Near-Separable Non-negative Matrix Factorization Using $\mathcal{L}_1$-Optimization

Aashish Nagpal
Indian Institute of Technology Delhi
India
aashunag@gmail.com

Chayan Sharma
Indian Institute of Technology Delhi
India
chayansharma576@gmail.com

Rahul Garg
Indian Institute of Technology Delhi
India
rahulgarg@cse.iitd.ac.in

Pawan Kumar
Indian Institute of Technology Delhi
India
pawan.kumar@cse.iitd.ac.in

ABSTRACT

In this paper we propose a new LP-based formulation to solve near separable non negative matrix factorization (NMF) problem using L_1 norm optimization. We also present a comprehensive experimental evaluation of the existing methods to solve separable NMF problem and compare them with the proposed formulation. The evaluation of this formulation on synthetic data shows that our new formulation gives significantly better quality of factorization as compared to the existing methods.

CCS CONCEPTS

• **Computing methodologies** **Dimensionality reduction and manifold learning**; **Non-negative matrix factorization**.

KEYWORDS

Non-negative Matrix Factorisation, NMF

ACM Reference Format:
Aashish Nagpal, Chayan Sharma, Rahul Garg, and Pawan Kumar. 2019. Near-Separable Non-negative Matrix Factorization Using $\mathcal{L}_1$-Optimization. In *6th ACM IKDD CoDS and 24th COMAD (CoDS-COMAD '19), January 3–5, 2019, Kolkata, India.* ACM, New York, NY, USA, Article 4, 7 pages. https://doi.org/10.1145/3297001.3297016

1 INTRODUCTION

Matrix factorization is an emerging feature selection and dimensionality reduction technique. The goal behind matrix factorization is to find a suitable representation of data which explains the hidden structure in the data explicitly that can further be used for high level computations and model building.

Given a matrix $X \in R^{m \times n}$, a matrix factorization algorithm decomposes it into two matrices $W \in R^{m \times r}$ and $H \in R^{r \times n}$ such that $r \ll min(m, n)$ and $X \approx WH$. There have been several variations of matrix factorization such as principal component analysis (PCA), independent component analysis (ICA), vector quantization etc. Essentially all of these variations decomposes the data matrix X into basis matrix W and coefficient matrix H under some constraints. In the case of non-negative matrix factorization, the constraint is non-negativity of basis and coefficient matrix coefficients i.e., $w_{ij} \geq 0$, $h_{ij} \geq 0$ assuming that $x_{ij} \geq 0$ where w_{ij}, h_{ij} and x_{ij} are coefficients of the matrices W, H and X, respectively. Constraint of non negativity can be used to learn a parts-based representation of the data [20]. Also it makes the representation purely additive whereas other variations like principal component analysis and independent component analysis allows subtractions only.

NMF was first introduced by Paatero and Tapper [24] as positive matrix factorization and since then it has been used in a variety of research areas such as *facial feature extraction* [8], *video tracking* [7], *document clustering* [27], *email analysis*, *topic detection* [4, 26], *acoustic signal processing*, *blind source separating* [9], *recommendation systems* [22] among many more. Exact NMF is NP-hard in general as shown by [25]. Several heuristic approaches have been proposed in the past such as the *multiplicative update rule* [21] which uses *alternating least squares minimization method* [17] to find a good solution. Many alternative approaches focus on some specific conditions on the input data solving NMF in polynomial time. The most important condition is the separability condition under which the matrix X is assumed to yield a non-negative matrix factorization such that columns of W are a subset of columns of X. The separable NMF has been shown to be useful for many applications such as *hyperspectral unmixing* [5], *document clustering* [10, 26, 27], *image analysis* [12].

One of the popular application of separable NMF is in blind hyperspectral unmixing in the presence of pure pixels [15]. In the context of topic modeling problems where X is a document-word matrix and W, H are document-topic and topic-word matrices, the separability implies that there exist anchor words in the vocabulary which uniquely describes a topic.

In the recent past, several new algorithms have been proposed to solve the problem of separable NMF. In this paper, we examine these algorithms critically and compare them with a new method that we have developed to solve the separable NMF problem. Evaluation on synthetic data shows that our new formulation performs significantly better than the existing methods.

CoDS-COMAD '19, January 3–5, 2019, Kolkata, India

ACM ISBN 978-1-4503-6207-8/19/01…$15.00
https://doi.org/10.1145/3297001.3297016

The rest of the paper is organised as follows. In Section 2, we outline related work on algorithms for the separable NMF problem. In section 3, we present our proposed formulation for the problem and give the main result for the case of noiseless recovery. In section 4, we present performance comparison using synthetic data. Section 5 concludes with some ideas for future work.

2 SEPARABLE NON-NEGATIVE MATRIX FACTORIZATION

Algebraically, the exact separable NMF can be represented as:

$$X = W[I_r, H^{'}]\pi, \tag{1}$$

where I_r is r by r identity matrix,W is a basis matrix and $H^{'}$ is the coefficient matrix s.t. $[I_r, H^{'}]\pi = H$ and π is a permutation matrix. If the input non-negative matrix X can be exactly decomposed as in (1), such that the columns of the matrix W are a subset of columns of the matrix X, then the matrix X is called r-separable.

Geometrically, r-separability means that in the cone generated by the columns of X, the r extreme rays of the cone correspond to the columns of W. Equivalently, if the columns of X are drawn in m-dimensional space, r-separability means that convex cone formed by the r columns of W encapsulate all the columns of X which are contained in positive orthant [11].

The goal is to identify the r columns of X (matrix W) that will allow reconstruction of the whole matrix X. One another representation of separable NMF can be as follows: X = [W,WH$^{'}$]Π

Multiplying with Π and Π^{-1} on the right side gives

$$X = [[W, WH^{'}]\Pi\,\Pi^{-1}]\,\Pi$$

Since W has r columns and $WH^{'}$ has (n-r) columns, we can multiply RHS with matrix

$$X = \left[[W, WH^{'}]\Pi\,\Pi^{-1}\begin{bmatrix} I_r & H^{'} \\ 0_{(n-r)\times r} & 0_{(n-r)\times(n-r)} \end{bmatrix}\right]\Pi$$

$$X = \left[[W, WH^{'}]\Pi\right]\,\Pi^{-1}\begin{bmatrix} I_r & H^{'} \\ 0_{(n-r)\times r} & 0_{(n-r)\times(n-r)} \end{bmatrix}\Pi$$

$$X = XC$$

where C is called the *factorization localizing matrix* [6]

In practice non-negative matrices rarely admit separability. Therefore, in the case of approximate reconstruction of X, the problem is called near-separable NMF.

Near Separable NMF: Given a noisy matrix $\tilde{X} = X + N$, where N represents noise in the measurement of the matrix X. If X admits an r-separable NMF factorization, the goal is to find a set of r columns J such that the W approximately represents r columns of $\tilde{X}$.

$$\tilde{X}_{.,J} \approx W$$

Without loss of generality, we make the following two assumptions for the rest of the paper:

Assumption-1: Given a r-separable matrix X we assume that the columns of X are normalized *i.e.* $\sum_{i=1}^{m} x_{ij} = 1$. Note that this also implies that columns of W and $H^{'}$ sum to one as well. If the input columns do not sum to one, then they can be scaled suitably. The results after factoring can be scaled back to give a solution to the original problem.

Assumption-2: We assume that the matrix X (after normalization) has no duplicate columns. In case the original matrix had duplicate columns, those can be found easily and removed to get a matrix with no duplicate columns. The results thus obtained can be suitably modified to give a solution to the original problem.

2.1 Algorithms for Near Separable NMF

In this section we summarize some of the historically important and some recently discovered promising algorithms in more detail. Several algorithms have been proposed to solve the problem of near-separable NMF in the recent years. These algorithms can be categorized as linear programming (LP) based [2], [1], [6], [14], semidefinite programming based [16], [23] and geometric or combinatorial algorithms [15], [13],[19].There are also some specialized algorithms to handle the large scale data in streaming, multi-core architectures [3]. The LP based algorithms typically have strong guarantees of finding the right solution under a suitable noise model and are generally very slow for problems of large size. Geometric algorithms are generally faster but have weaker guarantees. They try to identify the vertices of the convex hull of the normalized columns of X or equivalently, the extreme rays of the convex cone generated by the columns of X.

AGKM: The first algorithm to solve this specific case of the NMF problem was [2], which shows that if the noise satisfies $||N||_{\infty,1}$[1] $\leq \epsilon \leq \frac{\alpha^2}{20+13\alpha}$, where columns of W are α-robust simplicial then their algorithm provably find a good separable NMF factorization. A linear program is designed to express a given column as a convex combination of other columns. This method, though provides provable guarantees, requires solving n feasibility linear programs each with *O(n)* variables and therefore is not suitable for solving large scale practical problems.

Hottopixx: The other LP based method to solve this problem is Hottopixx [6]. This formulation is also based on $(\infty, 1)$ norm of the matrix as outlined below.

$$\begin{aligned} &\min_{\forall C \in R_+^{n\times n}} p^T diag(C) \\ &\text{such that } ||\tilde{X} - \tilde{X}C||_{\infty,1} \leq 2\epsilon, \\ &tr(C) = r, \\ &c_{ii} \leq 1 \quad \forall i, \\ &c_{ij} \leq c_{ii} \quad \forall i, j \end{aligned} \tag{2}$$

where p is n-dimensional vector with distinct entries. This LP involves $O(n^2)$ variables. The authors also give a fast stochastic gradient descent algorithm to solve this optimization problem. Under the assumption $\tilde{X} = X + N$, where $X = WH$ is r-separable, $||N||_{\infty,1} \leq \epsilon$

[1]
$$||N||_{\infty,1} = \max_{1\leq i\leq m} \sum_{j=1}^{n} |N_{ij}|$$

and $\epsilon \leq \frac{\alpha^2}{8+4\alpha}$ and columns of W are α-robust simplicial, the solution to the above LP gives a rank-r factorization $\tilde{W}\tilde{H}$ such that $\|\tilde{X} - \tilde{W}\tilde{H}\| \leq 4\epsilon$. Although Hottopixx is a promising algorithm for separable NMF, but it doesn't do well in noisy data. This drawback is primarily due to the choice of $(\infty, 1)$ norm which is very sensitive to the outliers. If one column of $\tilde{X}$ is very noisy, the choice of the norm will ensure that accuracy w.r.t. other columns will also degrade. Also, Hottopixx requires the knowledge of ϵ, which may be hard to estimate for most practical applications.

LPSepNMF: This is another LP-base formulation [14] similar to Hottopixx, proposed to resolve its drawbacks. The LP model is as follows:

$$\begin{aligned} \min_{\forall C \in R_+^{n \times n}} \; & p^T diag(C) \\ \text{such that } & \|\tilde{X} - \tilde{X}C\|_{\infty,1} \leq \rho\epsilon, \\ & c_{ii} \leq 1 \quad \forall i, \\ & c_{ij} \leq c_{ii} \quad \forall i, j, \end{aligned} \tag{3}$$

where p is a positive vector and $\rho > 0$. The algorithm does not require the data matrix X to be normalised. It also detects the factorization rank (r) automatically. Hence, it is more flexible than Hottopixx. However, this algorithm also uses the $(\infty, 1)$ norm and ϵ and therefore has similar limitations as that of Hottopixx.

XRAY: This is a geometric algorithm [19] [18] which iteratively extracts r columns of the original data matrix. At each step, it selects an exterior column of the data matrix and projects all the columns of X onto the cone generated by the columns of X extracted so far and repeats the process. This is a relatively faster algorithm than other LP based models but it tends to select near duplicate columns of X (similar columns to the columns of W). Also, since it extends the cone one extreme ray at each step, if some non-anchor columns lie outside the cone generated by the anchor columns (columns of W) due to added noise to the data matrix X, XRAY might select that non-anchor column. This is illustrated in Section 4.

Successive Projection Algorithm (SPA): This is also a geometric algorithm [15] similar to XRAY. It extracts one column of $\tilde{X}$ at each iteration which maximizes a convex function over the set **S**, set of currently extracted columns of matrix $\tilde{X}$. It then projects all vectors in S onto the orthogonal complement of the extracted column. In this way, it extracts one vertex of the convex hull at a time. SPA is very fast and effective in practice but it requires matrix W to be full rank. Also for ill-conditioned matrices, SPA fails even for small noise levels (see the discussion in [13]).

Successive Nonnegative Projection Algorithm (SNPA): This is a slightly modified version of Successive Projection Algorithm [13]. Similar to SPA, it extracts the the column of the data matrix which maximizes a strongly convex function f over the set S. But instead of projecting on the orthogonal space of the extracted columns it projects the vectors of S onto the convex hull of the columns extracted so far and the origin using the semi metric induced by f. SNPA is shown to be more robust than SPA and applies to a broader class of nonnegative matrices [13]. SNPA is relatively slower that SPA and XRAY, because the projection requires solving n linearly constrained least squares problems.

3 NEAR SEPARABLE NMF USING L$_1$ OPTIMIZATION

We propose the following linear optimization to identify columns of W among the columns of matrix $\tilde{X} = X + N$

$$\begin{aligned} \min_{\forall C \in R_+^{n \times n},\, a \in R_+^n} \; & \|\tilde{X} - \tilde{X}C\|_{1,1} \\ \text{subject to: } & 0 \leq c_{ij} \leq a_i \quad \forall i, j \\ & \sum_{i=1}^{n} a_i \leq r \end{aligned} \tag{4}$$

where $|M|_{1,1}$ represents the element-wise sum of absolute values of the entries in the matrix M. Algorithm 1 gives the complete details of the proposed method.

Algorithm 1: Separable Non-negative Matrix Factorization Algorithm using L$_1$ optimization

```
Input  : A noisy $r$-separable matrix $\tilde{X} = WH + N$ satisfying
         Assumption 1 and Assumption 2
Output : Matrix $C$ and index set $K$ satisfying $\tilde{X} = \tilde{X_K}C + \tilde{N}$,
         where $W \approx \tilde{X_K}$
Compute the matrix $C$ by solving LP (4).
Let $C_r$ be the column vector generated by summing matrix $C$ along the rows.
Compute the index set $K$ corresponding to the indices of largest $r$ entries in $C_r$.
Zero out all the rows of $C$ not in the index set $K$.
```

The LP based Hottopixx [6] or the LP separable NMF [14] formulations do not work well in practice due to three main reasons. Firstly, the $(\infty, 1)$ norm is used while evaluating the candidate solutions. The term $\|\tilde{X} - \tilde{X}C\|_{\infty,1}$ represents the maximum L$_1$ error over n columns of $\tilde{X}$. So, if there is one noisy column of $\tilde{X}$, it can potentially make ϵ very large. This makes the final solution, picked by the algorithm, insensitive to the quality of approximation of all the other columns of $\tilde{X}$. Secondly, the error ϵ needs to be estimated well. The algorithm considers all the potentially feasible solutions in the region $\|\tilde{X} - \tilde{X}C\|_{\infty,1} < \rho\epsilon$ and selects one of them. If the choice of ϵ is bad then many of these potential solutions will also tend to be bad. Finally, out of all the feasible solutions, the algorithm chooses the one minimizing $p^T diag(C)$ (where the choice of p is arbitrary), not the one minimizing the error of approximation $\|\tilde{X} - \tilde{X}C\|$ (under a suitable norm). Our proposed formulation directly minimizes the approximation error using the L$_1$ norm instead of using it in a constraint. As a result, the noise level ϵ is no longer required to be estimated. Moreover it minimizes the error for every column leading to much improved solutions.

Notice that the rounding method proposed to get the final set of the columns of W is also slightly different. While the earlier approaches select the indices corresponding to the largest diagonal entries in C, our algorithm selects the rows of C that have the largest sum. This rounding method ensures that the columns of $\tilde{X}$ that are most useful in explaining the other columns get picked. This leads to significantly improved approximation of $\tilde{X}$.

3.1 Correctness for a Noiseless Separable Matrix

The following theorem and its proof shows that the proposed formulation finds the optimal factorization in the noiseless case. The proof technique presented here is more general and uses an approach which can significantly shorten and simplify the correctness proofs for the noiseless case in earlier works [6].

THEOREM 3.1. *Given a r-separable matrix $X = WH$ (where $X \in R_+^{m\times n}$, $W \in R_+^{m\times r}$ and $H \in R_+^{r\times n}$), which is not $r-1$ separable, as an input, Algorithm 1 gives an index set K and a matrix C such that the columns of X corresponding to the index set K are identical to columns of W and rows of C corresponding to the index set K are identical to the rows of H.*

We first show that the columns of W define the extreme rays of the convex cone of columns of X using the fact that X is r-separable but not $r-1$ separable. We then show that the optimal solution of LP (4) has zero objective function value. Using the above two facts, we finally show that any optimal solution of LP (4) must have $c_{ii} = 1$ for column indices i of X corresponding to the columns of W.

Since X is r-separable, $X = WH$ where $X \in R_+^{m\times n}$, $W \in R_+^{m\times r}$ and $H \in R_+^{r\times n}$, such that all the columns of W are also the columns of X.

Claim 1: No column of W is contained in the convex cone of the remaining columns of W.

Proof: Assume for contradiction (without loss of generality) that the r^{th} column of W is contained in the convex cone of the first $r-1$ columns of W. Thus, there exist scalars $\alpha_1, \alpha_2, \ldots, \alpha_{r-1} \in R_+$, such that $\sum_{j=1}^{r-1} \alpha_j w_{ij} = w_{ir}, \forall i \in [1 \ldots m]$.

Define the matrices W' comprising the first $r-1$ columns of W and H' comprising the entries $h'_{jk} = h_{jk} + \alpha_j h_{rk}$. It may be verified that W' and H' are both non-negative and $X = W'H'$, implying that the matrix X is $r-1$ separable, leading to a contradiction. Thus, no column of W may be contained in the convex cone of its remaining columns.

In fact no column of W is contained in the convex cone of the remaining columns of X. To see this, again assume for contradiction that j^{th} column of W is contained in the convex cone of remaining columns of X. Thus $w_{.,j} = X\alpha$, for some $\alpha \in R_+^n$, which has at least two non-zero entries. So, $w_{.,j} = WH\alpha = W\beta$, where $\beta = H\alpha$. From Claim 1, is it only possible when β has only one non-zero entry equal to 1. For this, either H should have duplicate columns or α must have only one non-zero entry equal to 1. Since X does not have duplicate columns (Assumption 2), H cannot have duplicate columns and α has at least two non-zero entries. Thus, no column of W is contained in the convex cone of the remaining columns of X.

Claim 2: The optimal solution to the LP (4) has zero objective function value.

Proof: We construct a feasible solution that gives zero objective function value. Since the objective function is always non-negative, this solution must be one of the optimal solutions.

From r-separability of X, we have $X = WH$, where columns of W are subset of columns of X. Let $K(j)$ be the column of X which is identical to the j^{th} column of W. Let K represent the set of all column indices of X corresponding to the columns of W ($K = \cup_{j=1}^{r}\{K(j)\}$).

Construct the matrix C^* with entries

$$c^*_{jl} = \begin{cases} h_{j'l} & \text{if } j \in K \text{ where } j' \in [1 \ldots r] \text{ such that } K(j') = j \\ 0 & \text{if } j \notin K \end{cases}$$

Now, $X = XC^*$, by construction and by the fact $X = WH$. By Assumption 1, $\sum_{i=1}^{r} h_{ij} = 1$ and since $h_{ij} \geq 0$, this implies $0 \leq h_{ij} \leq 1$ for all i, j. Thus we have, $c^*_{il} \leq 1$ if $j \in K$ and $c^*_{il} = 0$ if $j \notin K$.

Set $a_i = 1$ for all $i \in K$ and $a_i = 0$ for all $i \notin K$. Thus, (a, C^*) gives a feasible solution to LP (4) and since $X = XC^*$, the objective function value is zero.

Let C' be an optimal solution of LP (4). Let $K = \{i : c'_{ii} = 1\}$.

Claim 3: There is a mapping κ from $[1 \ldots r]$ to K such that $w_{ij} = x_{i\kappa(j)}$ for all i.

Proof: Assume for contradiction that there is a column j of W such that there is no column of X in K is identical to it. Since X is r-separable, all columns of W must be in X as well. Let j' be the corresponding column in X such that $w_{ij} = x_{ij'} \forall i$. Since $j' \notin K$, $c'_{j'j'} < 1$. Also since the objective function value is zero, $x_{ij'} = \sum_{l=1}^{n} x_{il}c'_{lj'} = \sum_{l=1, l\neq j'}^{n} x_{il}c'_{lj'} + x_{ij'}c'_{j'j'}$ for all i. Thus, $x_{ij'}(1-c'_{j'j'}) = \sum_{l=1, l\neq j'}^{n} x_{il}c'_{lj'}$. Substituting w_{ij} in place of $x_{ij'}$ and using the fact that $1 - c_{j'j'} \neq 0$, we get $w_{ij} = \frac{1}{1-c'_{j'j'}} \sum_{l=1, l\neq j'}^{n} x_{il}c'_{lj'}$ for all i. Thus, the j^{th} column of W lies in the convex cone of other columns of X leading to a contradiction. Hence every column j of W, must have a corresponding column $\kappa(j) \in K$ such that $w_{ij} = x_{i\kappa(j)}$ for all i and $c'_{\kappa(j)\kappa(j)} = 1$.

From an optimal solution C' of the LP (4), the matrix W can be constructed by picking columns j of X that correspond to $c'_{jj} = 1$. The corresponding row-restricted matrix of C' gives the matrix H.

4 NUMERICAL EXPERIMENTS

In this section we present numerical experiments on synthetic data to compare our proposed approach with the following existing methods and their variations: (a) Hottopixx LP model [6] solved using CPLEX (b) Hottopixx LP model [6] solved using Augmented Lagrangian Coordinate Descent (ALCD) (c) Near-Separable NMF using Linear optimization (LPSepNMF) [14] (d) Successive Projection Algorithm (SPA) [15] (e) XRAY or Fast Conical Hull Algorithm [19] and (f) Successive Non-negative Projection Algorithm (SNPA) [13].

The source codes for the above mentioned algorithms were taken from their respective authors' websites. Since Hottopixx has been reported to perform better than AGKM algorithm [2], AGKM was not compared. While solving the LP model using ALCD [28], three different tolerance values $1e^{-3}$, $5e^{-3}$ and $1e^{-2}$ were used. Experiments were designed to find out how well these algorithms perform in finding the correct solution (i) under different noise levels (ii) with broader class of non-negative matrices such as ill-conditioned matrices, matrices with duplicate or near duplicate columns (iii) the size of the matrix X and its low dimensional representation k. All the tests were performed using Matlab on a Linux machine with Intel Core i7-7905 CPU @ 3.2GHz 8GB RAM.

4.1 Experimental Design

The experiments were divided into three stages. In the first stage, proposed algorithm is compared with other LP based algorithms (a,b,c). In the second stage, performance is compared with the geometric algorithms mentioned above (d,e,f). In the final stage, performance of the proposed algorithm is compared by varying tolerance value used for convergence in ALCD [28].

The recovery and running time of the above mentioned algorithms was compared by varying following parameters of the data matrix: (i) basis (number of columns r of W) (ii) number of rows m of the data matrix (iii) number of columns n of the data matrix (iv) sparsity of the coefficient matrix H (v) level of noise δ added to data matrix X.

4.1.1 *Synthetic Dataset Generation.*

The first step in generating a synthetic dataset is to generate the basis W. To analyze the robustness of algorithms in different cases of near-separable matrices, we follow the strategy used in [15] to generate the data matrix. Basis matrix $W \in R_{+}^{m\times r}$ is generated in two ways: (i) by choosing randomly from the uniform distribution $U(0,1)$ and (ii) by changing its condition number to 1000 as mentioned in [15]. Rest of the columns of the data matrix are generated : (i) from Dirichlet distribution and (ii) in a way such that each column is a middle point of the two vertices corresponding to the two columns of W. In total, we have 4 cases: (a) uniform Dirichlet (b) uniform middle points (c) ill-conditioned Dirichlet and (d) ill-conditioned middle points.

After generating W, H and N, matrices $\tilde{X}$ was computed as $\tilde{X} = WH + \delta N$ for different noise levels δ. These matrices were used as inputs for all algorithms and the output matrices $\hat{W}$ and $\hat{H}$ were compared with the ground-truth solutions.

4.1.2 *Performance parameters:*

The following metrics to evaluate the performance of the algorithms were used:

(1) **Fraction of Columns extracted** defined as:

$$\frac{\text{Number of correctly extracted columns of W}}{\text{Total number of columns of W (r)}}$$

(2) **Normalized Residual** defined as:

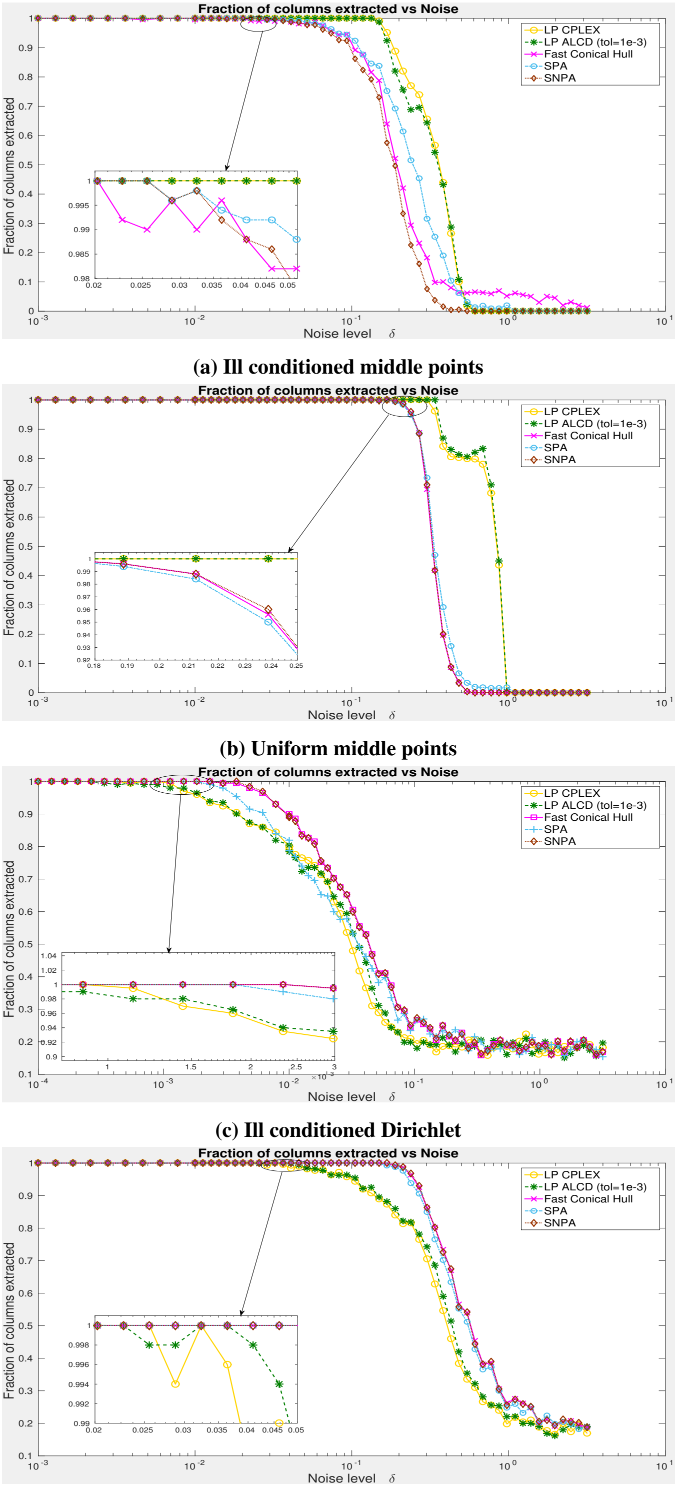
$$\frac{\|\tilde{X}-\tilde{X}C\|_{1,1}}{\|\tilde{X}\|_{1,1}} = \frac{\|\tilde{X}-\hat{W}\hat{H}\|_{1,1}}{\|\tilde{X}\|_{1,1}}$$

The experiments were repeated 50 times starting with different initial seeds for the random number generator. The average values as well as the standard error of the performance parameters were computed.

4.2 Results

Figures 1(a)-(d) show the fraction of columns extracted correctly by these algorithms as a function of noise levels for the four different ways of generating the input matrix X. In figure 1a and 1b, proposed LP CPLEX and ALCD outperforms all other algorithms. Note that the x-axis in these plot is in logscale. The proposed algorithms correctly identifies all the columns of basis matrix W for noise values (δ) which are nearly double than that of next best algorithms (SPA and SNPA).

In the first two cases, the data points lie outside the face of convex hull generated by the basis. So, geometric algorithms, which try to find the extreme points/rays of the convex hull/cone, performs relatively worse. From Figure 1(c) and 1(d), one can see that proposed algorithms perform rather poorly than geometric algorithms. The only case where SNPA outperforms the LP-ALCD is the case of near duplicate columns (cases (c) and (d)) showing that LP-ALCD

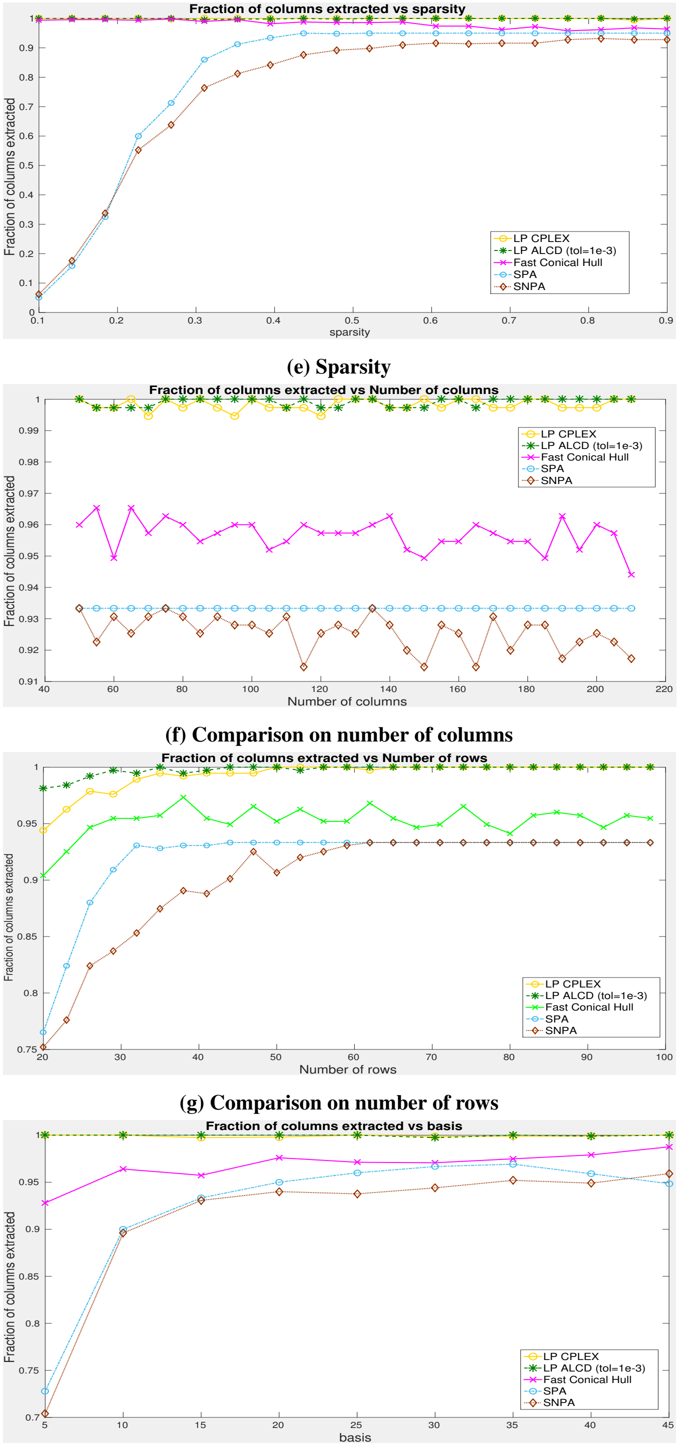


(a) Ill conditioned middle points

(b) Uniform middle points

(c) Ill conditioned Dirichlet

(d) Uniform Dirichlet



(e) Sparsity

(f) Comparison on number of columns

(g) Comparison on number of rows

(h) Comparison on number of basis

Figure 1: Comparison of near-separable NMF algorithms on different datasets (a)-(d) and for different conditions (e)-(h)

is not robust in this case. However, the near-duplicate columns can be identified and eliminated in a pre-processing step.

Test Case	**Proposed Cplex**	**Proposed ALCD ($tol = 1e^{-3}$)**	**Proposed ALCD ($tol = 1e^{-2}$)**	**Fast Conical Hull**	**SPA**	**SNPA**
a	$1.7e^{-1}$	$1.7e^{-1}$	$1.5e^{-1}$	$9.3e^{-2}$	$1.0e^{-1}$	$9.3e^{-2}$
b	$3.4e^{-1}$	$3.4e^{-1}$	$3.4e^{-1}$	$2.4e^{-1}$	$2.4e^{-1}$	$2.4e^{-1}$
c	$3.8e^{-3}$	$3.8e^{-3}$	$2.1e^{-4}$	$7.8e^{-3}$	$6.2e^{-3}$	$7.8e^{-3}$
d	$1.3e^{-1}$	$1.3e^{-1}$	$1.5e^{-1}$	$2.7e^{-1}$	$2.7e^{-1}$	$2.7e^{-1}$

Table 1: Maximum δ for more than 90% recovery: (a) Ill conditioned middle points (b) Uniform middle points (c) Ill conditioned Dirichlet (d) Uniform Dirichlet

Figures 1(e)-(h) show that the proposed algorithms LP-CPLEX and ALCD always outperform SPA, SNPA and Fast Conical Hull algorithms by large margin under different sparsity, number of columns, number of rows and basis size.

Table 1 contains the maximum noise level for which 90% recovery is possible for these algorithms. Proposed ALCD($1e^{-3}$) is more robust in case of 100% basis recovery but recovery performance less than 90% is similar for all the tolerance values ($5e^{-3}$ and $1e^{-2}$), refer table 2 and figure 2.

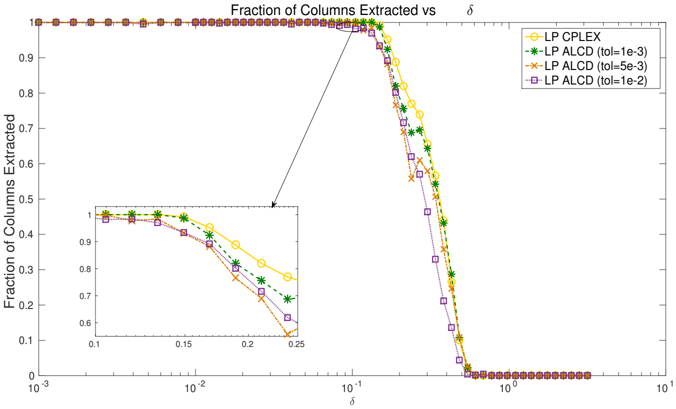


Figure 2: Robustness analysis: varying tolerance value of Proposed ALCD on Ill conditioned middle points dataset

Percentage	**Proposed Cplex**	**Proposed ALCD ($tol = 1e^{-3}$)**	**Proposed ALCD ($tol = 5e^{-3}$)**	**Proposed ALCD ($tol = 1e^{-2}$)**
100	$1.3e^{-1}$	$1.3e^{-1}$	$5.8e^{-2}$	$3.6e^{-3}$
90	$1.7e^{-1}$	$1.7e^{-1}$	$1.5e^{-1}$	$1.5e^{-1}$
80	$2.1e^{-1}$	$1.9e^{-1}$	$1.7e^{-1}$	$1.9e^{-1}$
70	$2.7e^{-1}$	$2.1e^{-1}$	$1.9e^{-1}$	$2.1e^{-1}$

Table 2: Robustness analysis: varying tolerance value of Proposed ALCD on Ill conditioned middle points dataset

Test Case	**Proposed Cplex**	**Proposed ALCD** ($tol = 1e^{-3}$)	**Proposed ALCD** ($tol = 1e^{-2}$)	**Fast Conical Hull**	**SPA**	**SNPA**
a	0.0311	0.0311	0.0321	0.0386	0.0394	0.0404
b	0.0274	0.0274	0.0274	0.0274	0.0274	0.0274
c	0.7696	0.7540	0.7549	0.7377	0.7426	0.7382
d	0.1470	0.1458	0.1449	0.1405	0.1406	0.1405

Table 3: Normalised Residual Analysis for δ=0.1: (a) Ill conditioned middle points (b) Uniform middle points (c) Ill conditioned Dirichlet (d) Uniform Dirichlet

Test Case	**Proposed Cplex**	**Proposed ALCD** ($tol = 1e^{-3}$)	**Proposed ALCD** ($tol = 1e^{-2}$)	**Fast Conical Hull**	**SPA**	**SNPA**
a	623.19	171.90	26.17	2.090	0.177	15.33
b	536.78	151.71	34.08	1.314	0.081	16.80
c	2261.3	346.49	73.45	0.937	0.073	8.492
d	1233.3	596.93	75.21	1.412	0.127	10.97

Table 4: Running Time (in seconds): (a) Ill conditioned middle points (b) Uniform middle points (c) Ill conditioned Dirichlet (d) Uniform Dirichlet

Table 3 shows that the normalized residuals for all the algorithms are comparable even though their recovery performances are different. Table 4 compares the running times of these algorithms. Though the proposed algorithm, when solved using CPLEX is very slow, its running time becomes comparable to that of SNPA when solved using ALCD using a tolerance of $1e^{-2}$, without significantly impacting the quality of its solutions (see Tables 1 and 3). However, in the case of Dirichlet, LP-ALCD is 7-10 times slower than SNPA.

5 CONCLUSIONS AND FUTURE WORK

Our evaluation on synthetic dataset showed that the proposed technique has higher column recovery rate than other algorithms. One issue with our technique is that it takes large amount of time as compared other geometric algorithms. This is mainly due to large number of constraints and variable in the proposed formulation. Also, the sparsity in the constraint matrix of the proposed LP formulation is O(m/n^2). Therefore, the current approach may be further improved by exploiting the sparsity.

REFERENCES

[1] S. Arora, R. Ge, Y. Halpern, D. Mimno, A. Moitra, D. Sontag, Y. Wu, and M Zhu. 2013. A practical algorithm for topic modeling with provable guarantees. In *International Conference on Machine Learning (ICML)*, Vol. 28. 280–288.

[2] S. Arora, R. Ge, R. Kannan, and A. Moitra. 2012. Computing a nonnegative matrix factorization âĂŞ provably.. In *Proceedings of the 44th Symposium on Theory of Computing,*. STOC, 145–162.

[3] Austin R Benson, Jason D Lee, Bartek Rajwa, and David F Gleich. 2014. Scalable Methods for Nonnegative Matrix Factorizations of Near-separable Tall-and-skinny Matrices. In *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 945–953.

[4] M.W. Berry and M. Browne. 2005. Email Surveillance using nonnegative matrix factorization. In *Workshop on Link Analysis, Counterterroism and Security, SIAM conference on Data Mining.* 452–456.

[5] J. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot. 2012. Hyperspectral unmixing overview. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 5, 2 (2012), 354–379.

[6] V. Bittorf, B. Recht, E. R e, and J Tropp. 2012. Factoring nonnegative matrices with linear programs.. In *Advances in Neural Information Processing Systems (NIPS âĂŹ12).* 1223–1231.

[7] S. S. Bucak and B. Gunsel. 2006. Incremental subspace learning via nonnegative matrix factorization. *Pattern Recognit.* 42, 5 (2006), 788–797.

[8] Wen-Sheng Chen, Binbin Pan, Bin Fang, Ming Li, and Jianliang Tang1. 2008. Incremental Nonnegative Matrix Factorization for Face Recognition. *Wavelet Analysis and Pattern Recognition* (2008).

[9] A. Cichocki, R. Zdunek, A.H. Phan, and S.I. Amari. 2009. *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. Wiley.

[10] W. Ding, M. Rohban, P. Ishwar, and V. Saligrama. 2013. Topic discovery through data dependent and random projections. In *International Conference on Machine Learning (ICML âĂŹ13)*, Vol. 28. 471–479.

[11] David L. Donoho and Victoria C. Stodden. 2004. When Does Non-Negative Matrix Factorization Give a Correct Decomposition into Parts?. In *Advances in neural information processing systems 16: proceedings of the 2003 conference*. 283–290.

[12] E. Elhamifar, G. Sapiro, and R. Vidal. 2012. Sparse modeling for finding representative objects. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR âĂŹ12).* 1600–1607.

[13] N. Gillis. 2014. Successive Nonnegative Projection Algorithm for Robust Nonnegative Blind Source Separation. *SIAM J. on Imaging Sciences* 7, 2 (2014), 1420–1450.

[14] N. Gillis and R Luce. 2014. Robust Near-Separable Nonnegative Matrix Factorization Using Linear Optimization. *Journal of Machine Learning Research* (April 2014), 1249–1280.

[15] N. Gillis and S. Vavasis. 2013. Fast and robust recursive algorithms for separable nonnegative matrix factorization. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 36, 4 (2013), 698–714.

[16] N. Gillis and S. Vavasis. 2015. Semidefinite programming based preconditioning for more robust near-separable nonnegative matrix factorization. *SIAM Journal on Optimization* 25, 1 (2015), 677–698.

[17] H. Kim and H. Park. 2008. Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method. *SIAM J. Matrix Anal. Appl.* 30, 2 (2008), 713–730.

[18] Abhishek Kumar and Vikas Sindhwani. 2015. Near-separable Non-negative Matrix Factorization with âĎŞ1- and Bregman Loss Functions. In *SDM*.

[19] A. Kumar, V. Sindhwani, and P. Kambadur. 2013. Fast conical hull algorithms for near-separable non-negative matrix factorization. In *International Conference on Machine Learning (ICML âĂŹ13)*, Vol. 28. 231–239.

[20] DD Lee and HS Seung. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature* 401 (1999), 788–791.

[21] D.D. Lee and H.S. Seung. 2001. Algorithms for non-negative matrix factorization. In *Advancesin Neural Information Processing Systems 13*. 556–562.

[22] Tao Li, Jiandong Wang, Huiping Chen, Xinyu Feng, and Feiyue Ye. 2006. A NMF-based Collaborative Filtering Recommendation Algorithm. In *Intelligent Control and Automation.*, Vol. 6. WCICA, 267–273.

[23] T. Mizutani. 2014. Ellipsoidal rounding for nonnegative matrix factorization under noisy separability. *Journal of Machine Learning Research,* 15 (2014), 1011–1039.

[24] P. Paatero and U. Tapper. 1994. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics* 5 (1994), 111–126.

[25] S.A. Vavasis. 2009. On the complexity of nonnegative matrix factorization. *SIAM Journalon Optimization* 20, 3 (2009), 1364–1377.

[26] Zhi-Li Wu and Chun hung Li. 2007. Topic Detection in Online Discussion using Non-Negative Matrix Factorization. In *International Conferences on Web Intelligence and Intelligent Agent Technology*. ACM, 267–273.

[27] W. Xu, X. Liu, and Y. Gong. 2003. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*. ACM, 267–273.

[28] Ian E.H. Yen, Kai Zhong, Cho-Jui Hsieh, Pradeep Ravikumar, and Inderjit S. Dhillon. 2015. Sparse Linear Programming via Primal and Dual Augmented Coordinate Descent. In *Neural Information Processing Systems (NIPS)*.