Rethinking Symmetric Matrix Factorization: A More General and Better Clustering Perspective

Mengyuan Zhang
Clemson University
Clemson, SC, USA
mengyuz@clemson.edu

Kai Liu
Clemson University
Clemson, SC, USA
kail@clemson.edu

***Abstract*—Nonnegative matrix factorization (NMF) is widely used for clustering with strong interpretability. Among general NMF problems, symmetric NMF is a special one which plays an important role for graph clustering where each element measures the similarity between data points. Most existing symmetric NMF algorithms require factor matrices to be nonnegative, and only focus on minimizing the gap between the original matrix and its approximation for clustering, without giving a consideration to other potential regularization terms which can yield better clustering. In this paper, we explore to factorize a symmetric matrix that does not have to be nonnegative, presenting an efficient factorization algorithm with a regularization term to boost the clustering performance. Moreover, a more generalized framework is proposed to solve symmetric matrix factorization problems with different constraints on the factor matrices.**

***Index Terms*—symmetric matrix factorization, generalized optimization framework, clustering**

I. Introduction

Nonnegative matrix factorization (NMF) [15], [32] problem is formulated as the following: given a data matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n] \in \mathbb{R}_+^{m\times n}$ containing n observations, each observation denoted as $\boldsymbol{x}_i$ is an m dimensional vector, where $\mathbb{R}_+^{m\times n}$ denotes the set of $m \times n$ element-wise nonnegative matrices. NMF aims to find a lower-rank matrix approximation represented by:

$$\boldsymbol{X} \approx \boldsymbol{U}\boldsymbol{V}^T, \tag{1}$$

where $\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_k] \in \mathbb{R}_+^{m\times k}$, and $\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_k] \in \mathbb{R}_+^{n\times k}$. Typically squared Frobenius norm is used to measure the distance between $\boldsymbol{X}$ and $\boldsymbol{U}\boldsymbol{V}^T$, so the objective in NMF is formulated as [10]:

$$\min_{\boldsymbol{U},\boldsymbol{V}\geq 0} \|\boldsymbol{X} - \boldsymbol{U}\boldsymbol{V}^T\|_F^2, \tag{2}$$

where $\|\cdot\|_F^2$ denotes squared Frobenius norm. Usually, k is assumed to be smaller than $\min\{m, n\}$, thus NMF can be regarded a lower-rank approximation problem.

Apparently, NMF paradigm described above conducts clustering based on input data directly and assume data is well linearly separable. However, for data lies in a specific manifold (say a certain sphere or two moons), it will yield poor result. Therefore, graph clustering is introduced to overcome the difficulty based on a matrix which measures the similarity between each data point [12]. The factorization of similarity matrix $\boldsymbol{A} \in \mathbb{R}_+^{n\times n}$ will yield a lower-rank matrix $\boldsymbol{H} \in \mathbb{R}_+^{n\times k}$ which plays a similar role as $\boldsymbol{V}$ for cluster assignment [4], [6], [9], [12], [20], [22]. Specifically, symmetric NMF formulates the objective as:

$$\min_{\boldsymbol{H}\geq 0} \|\boldsymbol{A} - \boldsymbol{H}\boldsymbol{H}^T\|_F^2, \tag{3}$$

where k is the number of clusters. Compared to classical NMF, symmetric NMF is more flexible in terms of admitting any reasonable measurement with mixed signs such as cosine similarity [21], [34], [36].

Previous work on symmetric NMF mostly requires that the matrix $\boldsymbol{H}$ is nonnegative. Therefore, even $\boldsymbol{A}$ is not explicitly constrained to be non-negative, in practice, it is equivalent to set the negative elements to be 0. When Symmetric NMF is applied to graph clustering, the result is directly obtained from $\boldsymbol{H}$ while ignoring some other techniques such as graph regularization to promote clustering performance.

In this paper, we comprehensively study symmetric matrix factorization with its application in graph clustering. Our contribution is threefold:

- We first extend vanilla Symmetric NMF and study a more general case where there is no non-negative constraint on $\boldsymbol{H}$ and interpret it in a meaningful manner. To obtain the solution, we propose a very efficient updating algorithm which can be extended to non-negative case.
- A regularization term is added to boost the clustering performance. In stead of merely focusing on minimizing the objective $\|\boldsymbol{A} - \boldsymbol{H}\boldsymbol{H}^T\|_F^2$, we impose graph regularization term to ensure that data points with higher similarity value share more similar cluster indicators, and vice versa, with $\boldsymbol{H}$ admitting mixed signs.
- We propose a very general framework which can deal with symmetric matrix factorization problems with various constraints by learning the stepsize adaptively.

This paper is organized as follows: Section II describes in details about the motivation behind the problem we proposed. Section III provides the problem formulation and several optimization methods, including one column-wise update method with fast speed, and one more general framework. Section IV shows the convergence rate of our method. Section V reports the experimental results on both large-scale image datasets and text datasets, followed by the Conclusion Section.

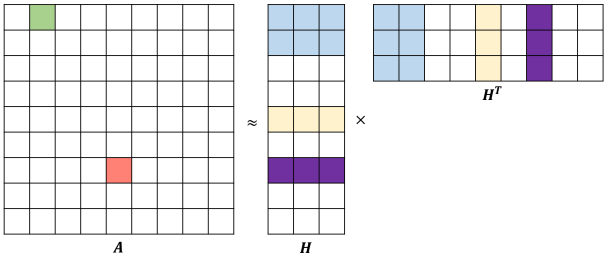

Fig. 1: An illustration of symmetric matrix factorization on $\boldsymbol{A} \in \mathbb{S}^n$. Green block indicates the similarity is high, the corresponding rows (first two) in $\boldsymbol{H}$ should also be similar, so they are filled with the same color. Red block indicates the similarity is low, where the corresponding rows (5th and 7th, respectively) in $\boldsymbol{H}$ should be dissimilar, filled with different colors.

II. Motivation

In symmetric NMF for clustering, the objective function (3) is to measure the gap between original similarity matrix $\boldsymbol{A}$ and $\boldsymbol{H}\boldsymbol{H}^T$, where $\boldsymbol{H}$ is clustering assignment matrix with nonnegative constraint. However, most algorithms only aim to minimize the gap $\|\boldsymbol{A} - \boldsymbol{H}\boldsymbol{H}^T\|_F^2$ while ignoring the potential over-fitting which may lower the clustering performance. Following the idea in graph regularization [3], [5], [31]: data points have high similarity (in $\boldsymbol{A}$) should have closer clustering indicators (rows of $\boldsymbol{H}$), and vice versa, which is demonstrated as Fig. 1. Accordingly, the regularization term is given by:

$$\min_{\boldsymbol{H} \geq 0} \sum_{i,j=1}^{n} \boldsymbol{A}_{ij} \|\boldsymbol{h}^i - \boldsymbol{h}^j\|_2^2, \tag{4}$$

where $\boldsymbol{h}^i$ denotes i-th row in $\boldsymbol{H}$.

Though theoretically $\boldsymbol{A}$ admits mixed signs, however, due to the nonnegative constraint on $\boldsymbol{H}$, negative elements are treated as 0 after projection and play no role in learning. Therefore, we remove nonnegative constraint on $\boldsymbol{H}$. Naturally, if $\boldsymbol{A}_{ij}$ is negative, the indicators should be significantly different and $\boldsymbol{A}_{ij}\|\boldsymbol{h}^i - \boldsymbol{h}^j\|_2^2$ remains negatively small, which is in accordance with the spirit of graph regularization. Negative element h_{ij} denotes the **unlikelihood** of i-th data belonging to j-th cluster while positive represents the very **likelihood**. Therefore we formulate the objective with regularization as:

$$\min \|\boldsymbol{A} - \boldsymbol{H}\boldsymbol{H}^T\|_F^2 + \lambda \sum_{i,j=1}^{n} \boldsymbol{A}_{ij} \|\boldsymbol{h}^i - \boldsymbol{h}^j\|_2^2, \tag{5}$$

where λ is the tuning regularization parameter.

To verify whether the regularization term can help to boost the clustering performance, a pilot experiment is conducted on COIL-20 data set [23]. Fig. 2 shows clustering accuracy comparison obtained from same initialization and constraint (admitting mixed signs on $\boldsymbol{H}$) with the only difference being the existence of regularization term. We can see that the clustering performance is significantly improved by incorporating the regularization term into the objective, especially when the number of clusters grows larger.

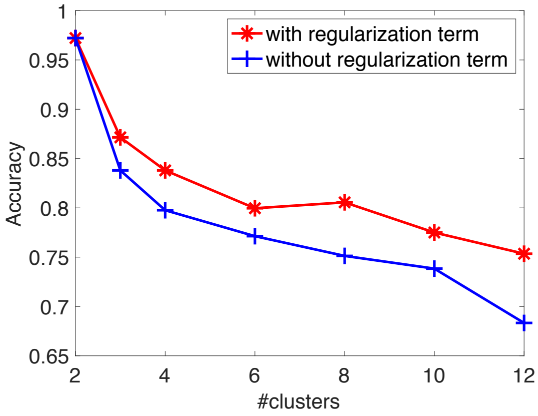

Fig. 2: Accuracy of clustering comparison on COIL20 data set with different cluster numbers.

Moreover, regarding $\boldsymbol{H}$, there can be different constraints on it. One can verify that if $\boldsymbol{H}^T\boldsymbol{H} = \boldsymbol{I}$, then it becomes spectral clustering [1], [24]; if all is zero except one element is 1 in each row of $\boldsymbol{H}$, then it is K-means; if each row is nonnegative and the sum is 1 [16], [26], then it is the probability distribution to each cluster. Instead of proposing ad-hoc algorithms with various constraints, we will systematically address the problem by providing a very general framework which can obtain optimal solution with updates efficiently.

III. Formulation and Algorithm

In this section, we propose our algorithm to solve (5) which enjoys faster convergence than existing counterparts.

A. Reformulation

By noting the regularization term is in row-wise form, we are to reformulate the objective in a more compact way as:

Theorem 1. *(5) is equivalent to:*

$$\min_{\boldsymbol{H}} \|\boldsymbol{A} - \lambda \boldsymbol{L} - \boldsymbol{H}\boldsymbol{H}^T\|_F^2, \tag{6}$$

where $\boldsymbol{L}$ is the Laplacian matrix given by $\boldsymbol{L} := \boldsymbol{D} - \boldsymbol{A} \in \mathbb{R}^{n \times n}$ and $\boldsymbol{D}$ is the degree matrix, which is diagonal defined as: $\boldsymbol{D}(i,i) = \sum_j (A(i,j))$.

Proof of Theorem 1. Following [3], we have:

$$\sum_{i,j=1}^{n} \boldsymbol{A}_{ij} \|\boldsymbol{h}^i - \boldsymbol{h}^j\|_2^2 = 2tr(\boldsymbol{H}^T \boldsymbol{L} \boldsymbol{H}). \tag{7}$$

Therefore, (5) is equivalent to

$$\min_{\boldsymbol{H}} \|\boldsymbol{A} - \boldsymbol{H}\boldsymbol{H}^T\|_F^2 + 2\lambda tr(\boldsymbol{H}^T \boldsymbol{L} \boldsymbol{H}). \tag{8}$$

By expanding (6), we obtain:

$$\begin{aligned} &\min_{\boldsymbol{H}} \|\boldsymbol{A} - \lambda \boldsymbol{L} - \boldsymbol{H}\boldsymbol{H}^T\|_F^2 \\ =&\min_{\boldsymbol{H}} tr(\boldsymbol{A} - \lambda \boldsymbol{L} - \boldsymbol{H}\boldsymbol{H}^T)^T(\boldsymbol{A} - \lambda \boldsymbol{L} - \boldsymbol{H}\boldsymbol{H}^T) \\ =&\min_{\boldsymbol{H}} \|\boldsymbol{A} - \boldsymbol{H}\boldsymbol{H}^T\|_F^2 + 2\lambda tr(\boldsymbol{H}^T \boldsymbol{L} \boldsymbol{H}) \\ &+ \textit{ terms irrelevant to } \boldsymbol{H}. \end{aligned} \tag{9}$$

Thus we conclude that solution to (5) is the same as to (6) in terms of optimizing $\boldsymbol{H}$. $\square$

B. Column-wise Fast Update

Given the new formulation in (6), we now turn to provide detailed updating rule for $\boldsymbol{H}$.

We first denote $\boldsymbol{A} - \lambda \boldsymbol{L}$ as $\boldsymbol{M}$, obviously $\boldsymbol{M}$ is still symmetric. And the optimization problem can be formulated as:

$$\min_{\boldsymbol{H}} f(\boldsymbol{H}) = \|\boldsymbol{M} - \boldsymbol{H}\boldsymbol{H}^T\|_F^2. \tag{10}$$

Before we begin our formal result, we start with a toy case: $f(h,p) = (t-hp)^2 + \lambda(h-p)^2$. By taking its partial derivative we have: $\partial_h f(h,p) = 2p(hp-t) + 2\lambda(h-p), \partial_p f(h,p) = 2h(hp-t) + 2\lambda(p-h)$, and therefore any critical point of f satisfies $(p-h)(hp-t-2\lambda) = 0$. One can see that as long as $\lambda > \frac{hp-t}{2}$, then $h^* = p^*$. The above example hints that instead of solving (10) directly, we may turn to solve:

$$\min_{\boldsymbol{H},\boldsymbol{P}} \|\boldsymbol{M} - \boldsymbol{H}\boldsymbol{P}^T\|_F^2 + \lambda\|\boldsymbol{H} - \boldsymbol{P}\|_F^2, \tag{11}$$

where as long as λ satisfies certain requirement, then $\boldsymbol{H}^* = \boldsymbol{P}^*$, which is easier to obtain the optimal solutions by utilizing any practical method such as alternating minimization.

Theorem 2. *Suppose $(\boldsymbol{H}^*, \boldsymbol{P}^*)$ be any critical point of (11) satisfying $\lambda > \frac{\|\boldsymbol{H}^*\boldsymbol{P}^{*T}\|_F - \sigma_n(\boldsymbol{M})}{2}$, where $\sigma_n(\cdot)$ denotes the n-th largest eigenvalue. Then $\boldsymbol{H}^* = \boldsymbol{P}^*$ and $\boldsymbol{H}^*$ is a critical point of (10).*

Proof of Theorem 2. We first introduce the following lemma which is very useful for later proof [17], [37].

Lemma 1. *For any symmetric $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ and PSD matrix $\boldsymbol{B} \in \mathbb{R}^{n\times n}$, we have:*

$$\sigma_n(\boldsymbol{A})\,\mathrm{trace}(\boldsymbol{B}) \le \mathrm{trace}\,(\boldsymbol{A}\boldsymbol{B}) \le \sigma_1(\boldsymbol{A})\,\mathrm{trace}(\boldsymbol{B}),$$

where $\sigma_i(\boldsymbol{A})$ is the i-th largest eigenvalue of $\boldsymbol{A}$.

Now we turn to check the sub-differential of f at any critical point $(\boldsymbol{H}^*, \boldsymbol{P}^*)$ which is given by:

$$\begin{aligned}\partial_{\boldsymbol{H}^*} f(\boldsymbol{H}^*, \boldsymbol{P}^*) &= 2[(\boldsymbol{H}^*\boldsymbol{P}^{*\mathrm{T}} - \boldsymbol{M})\boldsymbol{P}^* + \lambda(\boldsymbol{H}^* - \boldsymbol{P}^*)] = 0,\\ \partial_{\boldsymbol{P}^*} f(\boldsymbol{H}^*, \boldsymbol{P}^*) &= 2[(\boldsymbol{P}^*\boldsymbol{H}^{*\mathrm{T}} - \boldsymbol{M})\boldsymbol{H}^* - \lambda(\boldsymbol{H}^* - \boldsymbol{P}^*)] = 0.\end{aligned} \tag{12}$$

By subtracting the second line in the above equation from the first line, we have:

$$(2\lambda\boldsymbol{I} + \boldsymbol{M})(\boldsymbol{H}^* - \boldsymbol{P}^*) = \boldsymbol{P}^*\boldsymbol{H}^{*\mathrm{T}}\boldsymbol{H}^* - \boldsymbol{H}^*\boldsymbol{P}^{*\mathrm{T}}\boldsymbol{P}^*. \tag{13}$$

Taking the inner product $\boldsymbol{H}^\star - \boldsymbol{P}^\star$ on both sides:

$$\begin{aligned}&\langle 2\lambda\mathbf{I} + \boldsymbol{M}, (\boldsymbol{H}^* - \boldsymbol{P}^*)(\boldsymbol{H}^* - \boldsymbol{P}^*)^{\mathrm{T}}\rangle\\ =&\langle \boldsymbol{P}^*\boldsymbol{H}^{*\mathrm{T}}\boldsymbol{H}^* - \boldsymbol{H}^*\boldsymbol{P}^{*\mathrm{T}}\boldsymbol{P}^*, \boldsymbol{H}^* - \boldsymbol{P}^*\rangle.\end{aligned} \tag{14}$$

By applying Lemma 1 on the LHS, we have:

$$\begin{aligned}&\langle 2\lambda\mathbf{I} + \boldsymbol{M}, (\boldsymbol{H}^* - \boldsymbol{P}^*)(\boldsymbol{H}^* - \boldsymbol{P}^*)^{\mathrm{T}}\rangle\\ \ge&(2\lambda + \sigma_n(\boldsymbol{M}))\|\boldsymbol{H}^* - \boldsymbol{P}^*\|_F^2,\end{aligned} \tag{15}$$

while applying Lemma 1 on the other side we have:

$$\begin{aligned}&\langle \boldsymbol{P}^*\boldsymbol{H}^{*\mathrm{T}}\boldsymbol{H}^* - \boldsymbol{H}^*\boldsymbol{P}^{*\mathrm{T}}\boldsymbol{P}^*, \boldsymbol{H}^* - \boldsymbol{P}^*\rangle\\ =&\left\langle \frac{\boldsymbol{P}^\star\boldsymbol{H}^{\star\mathrm{T}} + \boldsymbol{H}^\star\boldsymbol{P}^{\star\mathrm{T}}}{2}, (\boldsymbol{H}^\star - \boldsymbol{P}^\star)(\boldsymbol{H}^\star - \boldsymbol{P}^\star)^{\mathrm{T}} \right\rangle\\ &- \frac{\left\|\boldsymbol{H}^\star\boldsymbol{P}^{\star\mathrm{T}} - \boldsymbol{P}^\star\boldsymbol{H}^{\star\mathrm{T}}\right\|_F^2}{2}\\ \le&\left\langle \frac{\boldsymbol{P}^\star\boldsymbol{H}^{\star\mathrm{T}} + \boldsymbol{H}^\star\boldsymbol{P}^{\star\mathrm{T}}}{2}, (\boldsymbol{H}^\star - \boldsymbol{P}^\star)(\boldsymbol{H}^\star - \boldsymbol{P}^\star)^{\mathrm{T}} \right\rangle\\ \le&\sigma_1\left(\frac{\boldsymbol{P}^\star\boldsymbol{H}^{\star\mathrm{T}} + \boldsymbol{H}^\star\boldsymbol{P}^{\star\mathrm{T}}}{2}\right)\|\boldsymbol{H}^\star - \boldsymbol{P}^\star\|_F^2\\ \le&\|\frac{\boldsymbol{P}^\star\boldsymbol{H}^{\star\mathrm{T}} + \boldsymbol{H}^\star\boldsymbol{P}^{\star\mathrm{T}}}{2}\|_F\|\boldsymbol{H}^\star - \boldsymbol{P}^\star\|_F^2\\ \le&\|\boldsymbol{H}^*\boldsymbol{P}^{*\mathrm{T}}\|_F\|\boldsymbol{H}^\star - \boldsymbol{P}^\star\|_F^2.\end{aligned} \tag{16}$$

Combining the above two equations we have:

$$(2\lambda + \sigma_n(\boldsymbol{M}))\|\boldsymbol{H}^* - \boldsymbol{P}^*\|_F^2 \le \|\boldsymbol{H}^*\boldsymbol{P}^{*\mathrm{T}}\|_F\|\boldsymbol{H}^\star - \boldsymbol{P}^\star\|_F^2. \tag{17}$$

Therefore, if $\lambda > \frac{\|\boldsymbol{H}^*\boldsymbol{P}^{*T}\|_F - \sigma_n(\boldsymbol{M})}{2}$, then $\boldsymbol{H}^* = \boldsymbol{P}^*$ and any critical points satisfying (12) meets the critical condition in (10). $\square$

The following lemma gives a bound for $\|\boldsymbol{H}^*\boldsymbol{P}^{*T}\|_F$.

Lemma 2. *For (11), suppose the objective decreases with initialization $\boldsymbol{P}_0 = \boldsymbol{H}_0$, then for any $k \ge 0$, the iterate $(\boldsymbol{H}_k, \boldsymbol{P}_k)$ generated by any algorithm satisfies:*

$$\|\boldsymbol{H}_k\boldsymbol{P}_k^{\mathrm{T}}\|_F \le \|\boldsymbol{M} - \boldsymbol{H}_0\boldsymbol{P}_0^{\mathrm{T}}\|_F + \|\boldsymbol{M}\|_F. \tag{18}$$

Proof of Lemma 2. By the assumption that the algorithm decreases the objective function, we have:

$$\begin{aligned}\|\boldsymbol{M} - \boldsymbol{H}_k\boldsymbol{P}_k^{\mathrm{T}}\|_F^2 + \lambda\|\boldsymbol{H}_k - \boldsymbol{P}_k\|_F^2 &\le \|\boldsymbol{M} - \boldsymbol{H}_0\boldsymbol{H}_0^{\mathrm{T}}\|_F^2\\ \implies \|\boldsymbol{M} - \boldsymbol{H}_k\boldsymbol{P}_k^{\mathrm{T}}\|_F^2 &\le \|\boldsymbol{M} - \boldsymbol{H}_0\boldsymbol{H}_0^{\mathrm{T}}\|_F^2\\ \implies \|\boldsymbol{H}_k\boldsymbol{P}_k^{\mathrm{T}}\|_F &\le \|\boldsymbol{M}\|_F + \|\boldsymbol{M} - \boldsymbol{H}_k\boldsymbol{P}_k^{\mathrm{T}}\|_F\\ &\le \|\boldsymbol{M}\|_F + \|\boldsymbol{M} - \boldsymbol{H}_0\boldsymbol{P}_0^{\mathrm{T}}\|_F.\end{aligned}$$

$\square$

Corollary 1. *If $\lambda > \frac{1}{2}(\|\boldsymbol{M}\|_F + \|\boldsymbol{M} - \boldsymbol{H}_0\boldsymbol{P}_0^T\|_F - \sigma_n(\boldsymbol{M}))$ and $\boldsymbol{P}_0 = \boldsymbol{H}_0$, then any algorithm decreases the objective in (11) will result in $\boldsymbol{P}^* = \boldsymbol{H}^*$.*

Proof. This is established by Lemma 2 and Theorem 2. $\square$

Discussion 1. *In our case, $\boldsymbol{H}$ is symmetric, not necessarily semi-positive definite (SPD). To compute $\sigma_n(\boldsymbol{M})$, which can be negative, it is very time-consuming if eigenvalue decomposition is utilized given its complexity level being $\mathcal{O}(n^3)$ when n is large. Therefore, we can divide into the following cases:*

- *when $\boldsymbol{H}$ is SPD, then $\lambda > \frac{1}{2}(\|\boldsymbol{M}\|_F + \|\boldsymbol{M} - \boldsymbol{H}_0\boldsymbol{P}_0^T\|_F)$ will naturally satisfy the requirement, admitting desired optimal solutions.*

Algorithm 1 Efficient update to optimize (11)

```
**Initialization:** $\boldsymbol{H}_0 = \boldsymbol{P}_0 \in \mathbb{R}^{n\times k}$.
**while** not converge **do**
    **for** $i = 1 : k$ **do**
        $\overline{\boldsymbol{M}} = \boldsymbol{M} - \sum_{j\neq i} \boldsymbol{h}_j \boldsymbol{p}_j^T$.
        $\boldsymbol{h}_i^+ = \frac{(\overline{\boldsymbol{M}}+\lambda \boldsymbol{I})\boldsymbol{p}_i}{\|\boldsymbol{p}_i\|^2+\lambda}, \quad \boldsymbol{p}_i^+ = \frac{(\overline{\boldsymbol{M}}+\lambda \boldsymbol{I})\boldsymbol{h}_i^+}{\|\boldsymbol{h}_i^+\|^2+\lambda}$.
    **end for**
**end while**
**Output:** $\boldsymbol{H}^* = \boldsymbol{P}^*$.
```

- *when $\boldsymbol{H}$ is not SPD, we can first compute the leading eigenvalue (t) by power iteration or Lanczos method [13], which gives $\mathcal{O}(log(n)/k)$ and $\mathcal{O}((log(n)/k)^2)$ convergence rate, respectively [33]. By setting $\lambda > \frac{1}{2}(\|\boldsymbol{M}\|_F + \|\boldsymbol{M} - \boldsymbol{H}_0\boldsymbol{P}_0^T\|_F + |t|)$, one can verify that it will obtain desired solution with significantly reduced complexity than eigenvalue decomposition.*

To end this proof part, we finally propose an efficient algorithm which will **decrease the objective monotonically**.

By noticing $\boldsymbol{H}\boldsymbol{P}^T = \boldsymbol{h}_i\boldsymbol{p}_i^T + \sum_{j\neq i}\boldsymbol{h}_j\boldsymbol{p}_j^T$, we can optimize $\boldsymbol{H}$ and $\boldsymbol{P}$ column by column. Different from the column-wise update process proposed in other studies [27], [29], there is no assumption needed in our update algorithm, the update formula in our method is much more straightforward and a detailed proof of sufficient decrease in objective function with our method is provided.

Denote $\overline{\boldsymbol{M}} = \boldsymbol{M} - \sum_{j\neq i}\boldsymbol{h}_j\boldsymbol{p}_j^T$, we have:

$$\begin{aligned}\boldsymbol{h}_i^+ &= \arg\min_{\boldsymbol{h}_i} \|\overline{\boldsymbol{M}} - \boldsymbol{h}_i\boldsymbol{p}_i^{\mathrm{T}}\|^2 + \lambda\|\boldsymbol{h}_i - \boldsymbol{p}_i\|_2^2 \\ &= \arg\min_{\boldsymbol{h}_i} (\|\boldsymbol{p}_i\|^2 + \lambda)\boldsymbol{h}_i^2 - 2\langle \boldsymbol{h}_i, \lambda\boldsymbol{p}_i + \overline{\boldsymbol{M}}\boldsymbol{p}_i\rangle \\ &= \frac{(\overline{\boldsymbol{M}} + \lambda\boldsymbol{I})\boldsymbol{p}_i}{\|\boldsymbol{p}_i\|^2 + \lambda}.\end{aligned} \tag{19}$$

Moreover, by noticing the strongly convexity ($\alpha = 2(\|\boldsymbol{p}_i\|^2 + \lambda)$) of objective w.r.t. $\boldsymbol{h}$, one have:

$$\begin{aligned}&f(\boldsymbol{h}_i, \boldsymbol{p}_i) - f(\boldsymbol{h}_i^+, \boldsymbol{p}_i) \\ \geq&\langle \nabla_{\boldsymbol{h}_i} f(\boldsymbol{h}_i^+, \boldsymbol{p}_i), \boldsymbol{h}_i - \boldsymbol{h}_i^+\rangle + \frac{\alpha}{2}\|\boldsymbol{h}_i - \boldsymbol{h}_i^+\|^2 \\ \geq&\lambda\|\boldsymbol{h}_i - \boldsymbol{h}_i^+\|^2,\end{aligned} \tag{20}$$

which indicates sufficient decrease by updating $\boldsymbol{h}_i$. Similarly, one can have same conclusion while updating $\boldsymbol{p}_i$.

Fig. 3 shows the convergence curves of Algorithm1, SymANLS [37] and alternating direction method of multipliers (ADMM) [2] when solving symmetric matrix factorization problem. From the figure we see Algorithm 1 converges very fast.

C. A more general Framework for constrained Optimization

Though the above subsection describes a simple, yet very efficient algorithm to obtain optimal solution, still it can't deal with most constraint problems such as $\|\boldsymbol{h}^i\|_1 = 1, \|\boldsymbol{h}^i\|_0 \leq s$, *etc.* [18], where different constraint indicate various meanings such as sparsity, probability distribution *etc.* In this subsection we propose a more general framework to solve (10).

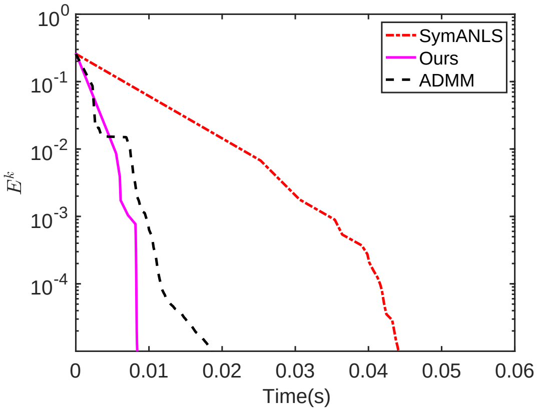

Fig. 3: Typical convergence curve which shows the superiority of our proposed method.

Recall the relationship between second-order Taylor expansion and gradient descent:

$$f(\boldsymbol{y}) \approx f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^T(\boldsymbol{y} - \boldsymbol{x}) + \frac{1}{2}\langle \boldsymbol{y} - \boldsymbol{x}, \nabla^2 f(\boldsymbol{x})(\boldsymbol{y} - \boldsymbol{x})\rangle. \tag{21}$$

If we replace Hessian $\nabla^2 f(\boldsymbol{x})$ with $\frac{1}{t}\boldsymbol{I}$, then $\min f(\boldsymbol{y})$ is to minimize $\|\boldsymbol{y} - \boldsymbol{x} + t\nabla f(\boldsymbol{x})\|_F^2$ where t is the step size in gradient descent method. It can be verified that:

$$\nabla f(\boldsymbol{H}) = 4(\boldsymbol{H}\boldsymbol{H}^T\boldsymbol{H} - \boldsymbol{M}\boldsymbol{H}). \tag{22}$$

By invoking (21) to update $\boldsymbol{H}^+$, we get:

$$\begin{aligned}\boldsymbol{H}^+ &= \arg\min_{\boldsymbol{H}'\in\mathbb{H}} f(\boldsymbol{H}') \\ &= \arg\min_{\boldsymbol{H}'\in\mathbb{H}} f(\boldsymbol{H}) + \nabla f(\boldsymbol{H})^T(\boldsymbol{H}' - \boldsymbol{H}) + \frac{1}{2t}\|\boldsymbol{H}' - \boldsymbol{H}\|_F^2 \\ &= \arg\min_{\boldsymbol{H}'\in\mathbb{H}} \frac{1}{2t}\|\boldsymbol{H}' - (\boldsymbol{H} - t\nabla f(\boldsymbol{H}))\|_F^2 \\ &= \mathcal{P}_{\mathbb{H}}(\boldsymbol{H} - t\nabla f(\boldsymbol{H})).\end{aligned} \tag{23}$$

$\mathbb{H}$ denotes the feasible set satisfying the constraint. Stepsize t in the above update rule should be relatively small to avoid gradient explosion. However, if it is too small, the convergence becomes slow, which should be avoided as well. As a contribution, we propose a method with **adaptive stepsize** which will make the objective decrease monotonically.

Algorithm 2 provides a generalized framework to solve any symmetric matrix factorization with different constraints. Below we provide some concrete examples:

- *Example I*: Nonnegative constraint $\boldsymbol{H} \geq 0$[1]:

$$\boldsymbol{H}^+ = \max\{\boldsymbol{H} - t\nabla f(\boldsymbol{H}), 0\}. \tag{24}$$

[1] One can verify that Algorithm 1 can also work if we set every column nonnegative, which can simply be obtained via $\boldsymbol{h}_i^+ = \max\{\frac{(\overline{\boldsymbol{M}}+\lambda\boldsymbol{I})\boldsymbol{p}_i}{\|\boldsymbol{p}_i\|^2+\lambda}, 0\}$, $\boldsymbol{p}_i^+ = \max\{\frac{(\overline{\boldsymbol{M}}+\lambda\boldsymbol{I})\boldsymbol{h}_i^+}{\|\boldsymbol{h}_i^+\|^2+\lambda}, 0\}$.

Algorithm 2 Optimize (10) where $\boldsymbol{H} \in \mathbb{H}$

```
**Input:** $\boldsymbol{M} = \boldsymbol{A} - \lambda \boldsymbol{L}$
**Initialization:** $\boldsymbol{H}_0 \in \mathbb{H}$, $i = 0$
**while** $i < K$ **do**
    $\nabla f(\boldsymbol{H}_i) = 4(\boldsymbol{H}_i \boldsymbol{H}_i^T \boldsymbol{H}_i - \boldsymbol{M}\boldsymbol{H}_i)$
    $L_i = 4\sigma_{max}(\boldsymbol{H}_i \boldsymbol{H}_i^T - \boldsymbol{M}) + 8\sigma_{max}(\boldsymbol{H}_i^T \boldsymbol{H}_i)$
    $\boldsymbol{H}_{i+1} = \mathcal{P}_{\mathbb{H}}(\boldsymbol{H}_i - t\nabla f(\boldsymbol{H}_i))$, where $t = 1/2L_i$
    $i = i + 1$
**end while**
**Output:** $\boldsymbol{H}_K$
```

- *Example II*: Unit constraint $\|\boldsymbol{h}\|_2 = 1$:

$$\boldsymbol{h}^+ = \frac{\boldsymbol{h} - t\nabla f(\boldsymbol{h})}{\|\boldsymbol{h} - t\nabla f(\boldsymbol{h})\|_2}. \tag{25}$$

- *Example III*: Sparsity constraint $\|\boldsymbol{h}^i\|_0 \leq s$: WLOG, assume the top s entry with maximum magnitude in $\boldsymbol{h}^i$ is indexed as $[1, s]$, then [19]

$$\boldsymbol{H}_{ij}^+ := \begin{cases} (\boldsymbol{H} - t\nabla f(\boldsymbol{H}))_{ij} & \text{if } j \in [1, s], \\ 0 & \text{otherwise.} \end{cases} \tag{26}$$

- *Example IV*: Orthogonality Constraint $\boldsymbol{H}^T\boldsymbol{H} = \boldsymbol{I}$:

$$\begin{gathered} \boldsymbol{H}^+ = \boldsymbol{U}^T\boldsymbol{V}, \\ where\ [\boldsymbol{U}, \boldsymbol{\Sigma}, \boldsymbol{V}] = svd(\boldsymbol{H} - t\nabla f(\boldsymbol{H})). \end{gathered} \tag{27}$$

IV. Convergence Analysis

In last section we mentioned the step size t should not be too large nor small, and in this section we will determine the best t in each update which guarantees the objective decreases monotonically by introducing the following lemma to begin:

Lemma 3. *For a function f with a Lipschitz continuous gradient L, if $\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\|_2 \leq L\|\boldsymbol{x} - \boldsymbol{y}\|_2$ then $f(\boldsymbol{y}) \leq f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^T(\boldsymbol{y} - \boldsymbol{x}) + \frac{L}{2}\|\boldsymbol{y} - \boldsymbol{x}\|_2^2$.*

Proposition 1. *For (10), in each i-th update, $L_i = 4\sigma_{max}(\boldsymbol{H}_i\boldsymbol{H}_i^T - \boldsymbol{M}) + 8\sigma_{max}(\boldsymbol{H}_i^T\boldsymbol{H}_i)$.*

Proof of Proposition 1. For sake of simplicity, we denote $\boldsymbol{H}$ as $\boldsymbol{H}_i$. With Lemma 3, it is equivalent to show $f(\boldsymbol{H}') - f(\boldsymbol{H}) - \nabla f(\boldsymbol{H})^T(\boldsymbol{H}' - \boldsymbol{H}) \leq \frac{L}{2}\|\boldsymbol{H}' - \boldsymbol{H}\|_F^2$. By denoting $\boldsymbol{H}'$ as $\boldsymbol{H} + \Delta\boldsymbol{H}$, we have:

$$\begin{aligned} & f(\boldsymbol{H} + \Delta\boldsymbol{H}) - f(\boldsymbol{H}) - \langle \Delta\boldsymbol{H}, \nabla f(\boldsymbol{H}) \rangle \\ &= 2\langle \boldsymbol{H}\boldsymbol{H}^T - \boldsymbol{M}, \Delta\boldsymbol{H}\Delta\boldsymbol{H}^T \rangle + \|\boldsymbol{H}\Delta\boldsymbol{H}^T + \Delta\boldsymbol{H}\boldsymbol{H}^T\|_F^2 \\ &\leq 2(\langle \boldsymbol{H}\boldsymbol{H}^T - \boldsymbol{M}, \Delta\boldsymbol{H}\Delta\boldsymbol{H}^T \rangle + \|\boldsymbol{H}\Delta\boldsymbol{H}^T\|_F^2 + \|\Delta\boldsymbol{H}\boldsymbol{H}^T\|_F^2) \\ &= 2tr(\Delta\boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{H}^T - \boldsymbol{M})\Delta\boldsymbol{H}) + 4tr(\Delta\boldsymbol{H}\boldsymbol{H}^T\boldsymbol{H}\Delta\boldsymbol{H}^T) \\ &\leq [2\sigma_{max}(\boldsymbol{H}\boldsymbol{H}^T - \boldsymbol{M}) + 4\sigma_{max}(\boldsymbol{H}^T\boldsymbol{H})]\|\Delta\boldsymbol{H}\|_F^2 = \frac{L}{2}\|\Delta\boldsymbol{H}\|_F^2, \end{aligned} \tag{28}$$

where $\sigma_{max}(\cdot)$ denotes the maximum singular value. □

Below we show that the objective function $f(\boldsymbol{H})$ in Algorithm 2 has sufficient decrease in each update with step size $t = \frac{1}{2L}$ and the generated sequence is convergent.

Theorem 3. *Let $g(\boldsymbol{H}_k) := f(\boldsymbol{H}_k) + \mathcal{C}(\boldsymbol{H}_k)$ be the objective function sequence generated by Algorithm 2 with constant step size $t_k = \frac{1}{2L_k}$. Then the sequence $g(\boldsymbol{H}_k)$ obeys sufficient decrease: $g(\boldsymbol{H}_{k-1}) - g(\boldsymbol{H}_k) \geq \frac{L_k}{2}\|\boldsymbol{H}_k - \boldsymbol{H}_{k-1}\|_F^2$.*

Proof of Theorem 3. From (23), noting that $\boldsymbol{H}_k$ minimizes $J(\boldsymbol{H}) = \nabla f(\boldsymbol{H}_{k-1})^T(\boldsymbol{H} - \boldsymbol{H}_{k-1}) + \frac{1}{2t}\|\boldsymbol{H} - \boldsymbol{H}_{k-1}\|_F^2 + \mathcal{C}(\boldsymbol{H})$, where $\mathcal{C}(\boldsymbol{H})$ represents any constraint $\mathcal{C}(\boldsymbol{H}) = \begin{cases} 0, \boldsymbol{H} \in \mathbb{H} \\ \infty, \text{ else} \end{cases}$, thus we naturally have $J(\boldsymbol{H}_k) \leq J(\boldsymbol{H}_{k-1})$, which implies:

$$\begin{aligned} & \nabla f(\boldsymbol{H}_{k-1})^T(\boldsymbol{H}_k - \boldsymbol{H}_{k-1}) + \frac{1}{2t}\|\boldsymbol{H}_k - \boldsymbol{H}_{k-1}\|_F^2 \\ & \leq \mathcal{C}(\boldsymbol{H}_{k-1}) - \mathcal{C}(\boldsymbol{H}_k). \end{aligned} \tag{29}$$

According to the definition of Lipschitz continuous, Lemma. 3, and when $t = \frac{1}{2L_k}$ in (29), we have:

$$\begin{aligned} & f(\boldsymbol{H}_k) + \mathcal{C}(\boldsymbol{H}_k) - f(\boldsymbol{H}_{k-1}) - \mathcal{C}(\boldsymbol{H}_{k-1}) \\ \leq & \nabla f(\boldsymbol{H}_{k-1})^T(\boldsymbol{H}_k - \boldsymbol{H}_{k-1}) + \frac{L_k}{2}\|\boldsymbol{H}_k - \boldsymbol{H}_{k-1}\|_F^2 \\ & + \mathcal{C}(\boldsymbol{H}_k) - \mathcal{C}(\boldsymbol{H}_{k-1}) \\ \leq & - L_k\|\boldsymbol{H}_k - \boldsymbol{H}_{k-1}\|_F^2 + \frac{L_k}{2}\|\boldsymbol{H}_k - \boldsymbol{H}_{k-1}\|_F^2 \\ = & - \frac{L_k}{2}\|\boldsymbol{H}_k - \boldsymbol{H}_{k-1}\|_F^2. \end{aligned} \tag{30}$$

□

One can see that if $\boldsymbol{H}_k \in \mathbb{H}$, then (30) degenerates to $f(\boldsymbol{H}_{k-1}) - f(\boldsymbol{H}_k) \geq \frac{L_k}{2}\|\boldsymbol{H}_k - \boldsymbol{H}_{k-1}\|_F^2$. Therefore, (10) decreases with update. Now repeating for all k will give $\frac{\bar{L}}{2}\sum_{k=1}^{\infty}\|\boldsymbol{H}_k - \boldsymbol{H}_{k-1}\|_F^2 \leq g(\boldsymbol{H}_0)$, where $\bar{L} := \min\{L_1, \ldots, L_k\} > 0$ and establishes its convergence.

Theorem 4. *In Algorithm 2, to ensure $\min\|\nabla f(\boldsymbol{H}) + \partial\mathcal{C}(\boldsymbol{H})\|_F^2 \leq \epsilon$, we need at most $T = O(\frac{1}{\epsilon})$ iterations.*

This indicates that the algorithm reaches critical point at least sub-linear convergence rate. Below is the proof:

Proof. First, by definition:

$$\boldsymbol{H}^+ = \arg\min_{\boldsymbol{H}'} \nabla f(\boldsymbol{H})^T(\boldsymbol{H}' - \boldsymbol{H}) + \frac{1}{2t}\|\boldsymbol{H}' - \boldsymbol{H}\|_F^2 + \mathcal{C}(\boldsymbol{H}'), \tag{31}$$

which implies:

$$0 \in \nabla f(\boldsymbol{H}) + \frac{1}{t}(\boldsymbol{H}^+ - \boldsymbol{H}) + \partial\mathcal{C}(\boldsymbol{H}^+). \tag{32}$$

WLOG, now we define $\boldsymbol{A}_{\boldsymbol{H}^+} \in \nabla f(\boldsymbol{H}^+) + \partial\mathcal{C}(\boldsymbol{H}^+)$, apparently it indicates the gap to $\boldsymbol{H}^*$ which satisfies $0 \in \nabla f(\boldsymbol{H}^*) + \partial\mathcal{C}(\boldsymbol{H}^*)$. Accordingly:

$$\begin{aligned} \boldsymbol{A}_{\boldsymbol{H}^+} & \in \nabla f(\boldsymbol{H}^+) + \partial\mathcal{C}(\boldsymbol{H}^+) \\ & = \nabla f(\boldsymbol{H}^+) - \nabla f(\boldsymbol{H}) - \frac{1}{t}(\boldsymbol{H}^+ - \boldsymbol{H}). \end{aligned} \tag{33}$$

As f is L-Lipschitz gradient continuous, then:

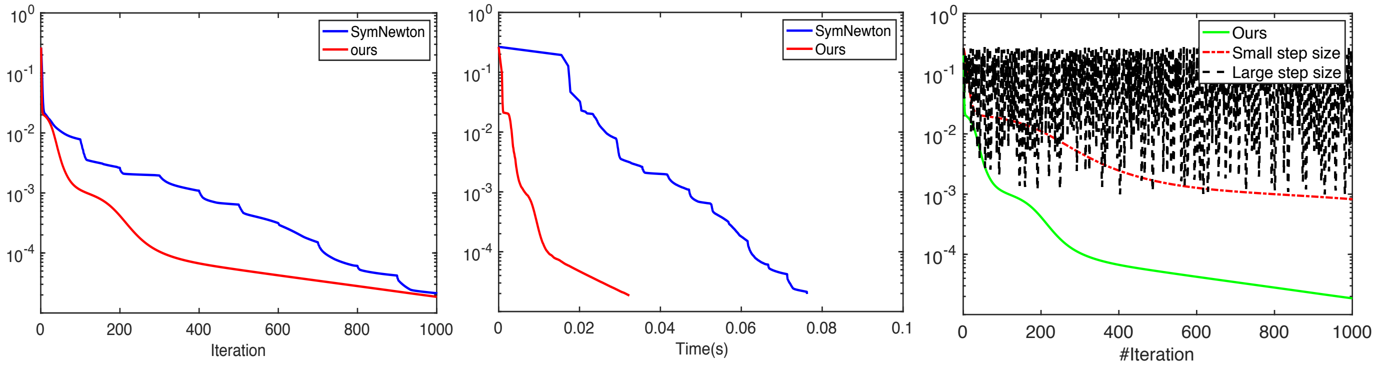

Fig. 4: Convergence comparison w.r.t. $E^k := \frac{\|\boldsymbol{M}-\boldsymbol{H}^k(\boldsymbol{H}^k)^T\|_F^2}{\|\boldsymbol{M}\|_F^2}$. **Left:** objective value versus iteration numbers. **Middle:** objective value versus time. **Right:** projected gradient descent with different step-size settings.

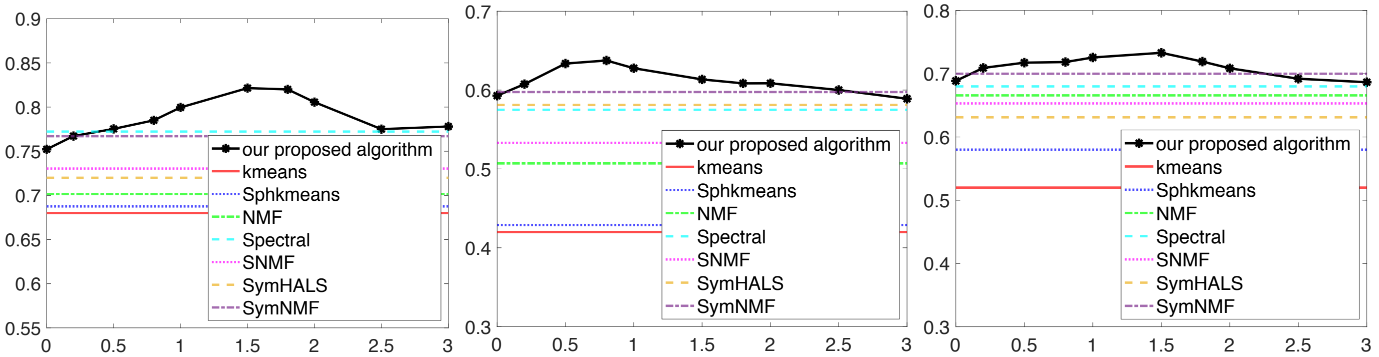

Fig. 5: Accuracy (Y-axis) comparison with λ changes (X-axis). **Left:** on a subset of 5 clusters from COIL-20. **Middle:** on a subset of 12 clusters from Reuters-21578. **Right:** on a subset of 10 clusters from TDT2.

$$\begin{aligned}\|\boldsymbol{A}_{\boldsymbol{H}^+}\| &\le \|\nabla f(\boldsymbol{H}^+) - \nabla f(\boldsymbol{H})\| + \frac{1}{t}\|\boldsymbol{H}^+ - \boldsymbol{H}\| \\ &\le L\|\boldsymbol{H}^+ - \boldsymbol{H}\| + \frac{1}{t}\|\boldsymbol{H}^+ - \boldsymbol{H}\| \\ &= 3L\|\boldsymbol{H}^+ - \boldsymbol{H}\|,\end{aligned}\tag{34}$$

where the first line comes from subadditivity inequality, second line is by definition of Lipschitz gradient continuity while the last line is by definition of $t = \frac{1}{2L}$.

On the other hand, $f(\boldsymbol{H}) - f(\boldsymbol{H}^+) \ge \frac{1}{L}\|\boldsymbol{H}^+ - \boldsymbol{H}\|_F^2$, therefore:

$$\begin{aligned}\|\boldsymbol{A}_{\boldsymbol{H}^+}\|_F^2 &\le 9L^2\|\boldsymbol{H}^+ - \boldsymbol{H}\|_F^2 \\ &\le 9L^2 * \frac{2}{L}(f(\boldsymbol{H}) - f(\boldsymbol{H}^+)) \\ &= 18L(f(\boldsymbol{H}) - f(\boldsymbol{H}^+)).\end{aligned}\tag{35}$$

By repeating the above for all k:

$$\sum_i \|\boldsymbol{A}_i\|_F^2 \le 18L(f(\boldsymbol{H}_0) - f(\boldsymbol{H}_k)) \le 18Lf(\boldsymbol{H}_0).\tag{36}$$

Thus, $\min\|\boldsymbol{A}_i\|_F^2 \le \frac{18Lf(\boldsymbol{H}_0)}{T}$, that is as long as $T = \frac{18Lf(\boldsymbol{H}_0)}{\epsilon} = \mathcal{O}(\frac{1}{\epsilon})$, then $\min\|\boldsymbol{A}_i\|_F^2 = \epsilon$ which finishes the proof. Apparently, our algorithm has a at least sub-linear convergence rate. □

We compare our proposed method with nonnegative constraint on $\boldsymbol{H}$ using projected gradient descent and the Newton-like method in SymNMF [12] in terms of convergence speed on synthetic data in Fig. 4. With adaptive step-size, it converges way faster than others.

TABLE I: Datasets Information

Dataset	#Clusters	#Samples	Dimensionality
COIL-20	20	1440	1024
CIFAR-10	10	3000	1024
Reuters-21578	41	8213	18933
TDT2	30	9394	36771

V. Experiments

A. Datasets

Two image datasets and two text datasets are used in the experiment: COIL-20 [23], CIFAR-10 [11], Reuters-21578 [25], and TDT2 [28]. Detailed descriptions about the number of clusters, number of samples and dimensionality of these datasets can be found in Table I.

B. Experimental Settings

Clustering performances of the following 9 algorithms are compared:

TABLE II: Normalized mutual information (NMI) of different algorithms on four datasets with varying number of clusters

	COIL-20			CIFAR-10			Reuters-21578			TDT2		
Method	2	10	20	3	6	10	2	8	15	2	10	20
K-means	0.901	0.624	0.591	0.296	0.287	0.201	0.785	0.553	0.421	0.752	0.532	0.501
NMF	0.907	0.729	0.522	0.308	0.288	0.195	0.819	0.752	0.598	0.822	0.666	0.600
Spectral	0.877	0.701	0.677	0.309	0.298	0.201	0.828	0.611	0.499	0.829	0.607	0.592
SymHALS	0.911	0.688	0.652	0.308	0.255	0.198	0.855	0.631	0.552	0.822	0.611	0.588
SNMF	0.911	0.659	0.638	0.311	0.302	0.188	0.872	0.566	0.531	0.751	0.689	0.603
SymNMF	0.951	0.739	0.662	0.307	0.289	0.209	0.897	0.692	0.605	0.802	0.671	0.662
GNMF	0.951	0.701	0.652	0.319	0.301	0.217	0.852	0.772	0.618	0.798	0.699	0.652
DSC	0.949	0.752	0.701	0.318	0.302	0.297	0.876	0.779	0.682	0.851	0.691	0.664
Ours	**0.958**	**0.797**	**0.725**	**0.341**	**0.319**	**0.302**	**0.901**	**0.798**	**0.755**	**0.861**	**0.729**	**0.705**

TABLE III: Clustering accuracy (AC) of different algorithms on four datasets with varying number of clusters

	COIL-20			CIFAR-10			Reuters-21578			TDT2		
Method	2	10	20	3	6	10	2	8	15	2	10	20
K-means	0.921	0.674	0.631	0.316	0.297	0.221	0.815	0.563	0.503	0.800	0.581	0.533
NMF	0.923	0.765	0.586	0.330	0.306	0.198	0.900	0.777	0.616	0.839	0.693	0.629
Spectral	0.898	0.737	0.702	0.332	0.316	0.228	0.889	0.645	0.513	0.896	0.639	0.602
SymHALS	0.927	0.703	0.682	0.321	0.287	0.219	0.881	0.658	0.575	0.851	0.639	0.601
SNMF	0.932	0.686	0.653	0.328	0.312	0.202	0.890	0.586	0.552	0.781	0.703	0.630
SymNMF	0.972	0.772	0.695	0.321	0.305	0.231	0.911	0.721	0.627	0.823	0.703	0.689
GNMF	0.968	0.722	0.683	0.343	0.326	0.232	0.871	0.802	0.628	0.825	0.715	0.676
DSC	0.972	0.788	0.722	0.339	0.321	0.318	0.903	0.811	0.703	0.863	0.720	0.687
Ours	**0.975**	**0.815**	**0.787**	**0.369**	**0.346**	**0.328**	**0.922**	**0.823**	**0.782**	**0.878**	**0.756**	**0.736**

1) Standard K-means;
2) NMF using alternating nonnegative least squares algorithm [10]; The data matrix $\boldsymbol{X}$ is transformed into its normalized-cut weighted version;
3) Spectral clustering (Spectral) [24], [30];
4) Hierarchical Alternating Least Squares (HALS) for symmetric NMF (SymHALS) [37];
5) Symmetric NMF using Procrustes rotations (SNMF) [7];
6) Symmetric NMF (SymNMF) [12];
7) Graph regularized nonnegative matrix factorization (GNMF) [3];
8) Deep subspace clustering (DSC) [8];
9) Our method. Algorithm 2 is used to solve the objective function (10).

In order to randomize the experiments, we conduct the evaluation using subsets of the whole datasets with different cluster numbers. For each selected number of clusters K, 10 test runs are conducted on a randomly chosen subset with K clusters. When K is the total number of clusters in the complete data set, the test runs are repeated on the entire data set. The symmetric matrix $\boldsymbol{A}$ can be obtained by utilizing any similarity measures, for simplicity we use the inner product similarity. Throughout the experiments, we use Matlab R2019a on a laptop with 1.4 GHz QuadCore Intel Core i5 processor.

C. Results and Analysis

The clustering quality is measured by normalized mutual information (NMI) [14], a measurement of similarity from information theory, and clustering accuracy (AC), the percentage of items correctly clustered with the maximum bipartite matching [35]. AC is defined as follows:

$$AC = \frac{\sum_{i=1}^{n} \delta(r_i, map(l_i))}{n}, \tag{37}$$

where l_i is the obtained cluster label, r_i is the original provided label, n is the number of total samples, $\delta(x, y)$ equals 1 when $x = y$ and equals 0 otherwise, and $map(l_i)$ is the permutation function that maps each l_i to the equivalent cluster label provided via Hungarian algorithm.

Fig. 5 shows how the clustering accuracy of our method varies with different values of λ. The performance is not changing dramatically with respect to the parameter λ, and our method has consistently ideal performance if λ is within a reasonable range. It's reasonable to observe that the optimal value of λ is slightly dependent of the data since the dimensionality and magnitude of the data can all have some effect on it. Experiment results of normalized mutual information and clustering accuracy on the four datasets are shown in Table II and Table III. We report the mean of NMI and AC for each given cluster number K over 10 test runs, highest accuracy for each K is highlighted. We can see that for both image data and text data, our proposed method can always achieve the best clustering performance among all the methods, and the improvement is significant, both in NMI and AC. One potential reason that the performance on CIFAR-10 is not that good as on COIL-20 may be the images from CIFAR-10 have more complex and varying background than the images from COIL-20. Although as the number of clusters increases, all methods' clustering performance is getting worse, our method is relatively stable with increasing number of clusters compared to other methods.

VI. Conclusion

In this paper we study the symmetric matrix factorization problem with a regularization term. We propose an efficient column-wise update rule and provide a general framework which can be extended to solve symmetric matrix factorization problems with various constraints. We prove the convergence rate with theoretical analysis. The results of extensive experiments on real-world data sets validate the effectiveness of our algorithm and its superiority on data clustering tasks.

References

[1] Francis Bach and Michael Jordan. Learning spectral clustering. *Advances in neural information processing systems*, 16(2):305–312, 2004.

[2] Stephen Boyd, Neal Parikh, and Eric Chu. *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.

[3] Deng Cai, Xiaofei He, Jiawei Han, and Thomas S Huang. Graph regularized nonnegative matrix factorization for data representation. *IEEE transactions on pattern analysis and machine intelligence*, 33(8):1548–1560, 2010.

[4] M Catral, Lixing Han, Michael Neumann, and Robert J Plemmons. On reduced rank nonnegative matrix factorization for symmetric nonnegative matrices. *Linear Algebra and its Applications*, 393:107–126, 2004.

[5] Quanquan Gu and Jie Zhou. Co-clustering on manifolds. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 359–368, 2009.

[6] Zhaoshui He, Shengli Xie, Rafal Zdunek, Guoxu Zhou, and Andrzej Cichocki. Symmetric nonnegative matrix factorization: Algorithms and applications to probabilistic clustering. *IEEE Transactions on Neural Networks*, 22(12):2117–2131, 2011.

[7] Kejun Huang, Nicholas D Sidiropoulos, and Ananthram Swami. Nonnegative matrix factorization revisited: Uniqueness and algorithm for symmetric decomposition. *IEEE Transactions on Signal Processing*, 62(1):211–224, 2013.

[8] Pan Ji, Tong Zhang, Hongdong Li, Mathieu Salzmann, and Ian Reid. Deep subspace clustering networks. *arXiv preprint arXiv:1709.02508*, 2017.

[9] Yuheng Jia, Hui Liu, Junhui Hou, and Sam Kwong. Semisupervised adaptive symmetric non-negative matrix factorization. *IEEE transactions on cybernetics*, 51(5):2550–2562, 2020.

[10] Hyunsoo Kim and Haesun Park. Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method. *SIAM journal on matrix analysis and applications*, 30(2):713–730, 2008.

[11] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[12] Da Kuang, Chris Ding, and Haesun Park. Symmetric nonnegative matrix factorization for graph clustering. In *Proceedings of the 2012 SIAM international conference on data mining*, pages 106–117. SIAM, 2012.

[13] Jacek Kuczyński and Henryk Woźniakowski. Estimating the largest eigenvalue by the power and lanczos algorithms with a random start. *SIAM journal on matrix analysis and applications*, 13(4):1094–1122, 1992.

[14] Andrea Lancichinetti, Santo Fortunato, and János Kertész. Detecting the overlapping and hierarchical community structure in complex networks. *New journal of physics*, 11(3):033015, 2009.

[15] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[16] Tao Li and Chris Ding. The relationships among various nonnegative matrix factorization methods for clustering. In *Sixth International Conference on Data Mining (ICDM'06)*, pages 362–371. IEEE, 2006.

[17] Xiao Li, Zhihui Zhu, Qiuwei Li, and Kai Liu. A provable splitting approach for symmetric nonnegative matrix factorization. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[18] Kai Liu. Spherical matrix factorization. *arXiv preprint arXiv:2111.14536*, 2021.

[19] Kai Liu, Qiuwei Li, Hua Wang, and Gongguo Tang. Spherical principal component analysis. *arXiv preprint arXiv:1903.06877*, 2019.

[20] Kai Liu, Xiangyu Li, Zhihui Zhu, Lodewijk Brand, and Hua Wang. Factor-bounded nonnegative matrix factorization. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 15(6):1–18, 2021.

[21] Kai Liu and Hua Wang. High-order co-clustering via strictly orthogonal and symmetric l1-norm nonnegative matrix tri-factorization. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, 2018.

[22] Kai Liu, Hua Wang, Shannon Risacher, Andrew Saykin, and Li Shen. Multiple incomplete views clustering via non-negative matrix factorization with its application in alzheimer's disease analysis. In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pages 1402–1405. IEEE, 2018.

[23] S. A. Nene, S. K. Nayar, and H. Murase. Columbia object image library (coil-20). *Technical Report CUCS-005-96*, 1996.

[24] Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pages 849–856, 2002.

[25] Reuters-21578. http://www.research.att.com/ lewis. 1987.

[26] Mikkel N Schmidt, Ole Winther, and Lars Kai Hansen. Bayesian non-negative matrix factorization. In *International Conference on Independent Component Analysis and Signal Separation*, pages 540–547. Springer, 2009.

[27] Qingjiang Shi, Haoran Sun, Songtao Lu, Mingyi Hong, and Meisam Razaviyayn. Inexact block coordinate descent methods for symmetric nonnegative matrix factorization. *IEEE Transactions on Signal Processing*, 65(22):5995–6008, 2017.

[28] TDT2. http://projects.ldc.upenn.edu/tdt2/. 1998.

[29] Arnaud Vandaele, Nicolas Gillis, Qi Lei, Kai Zhong, and Inderjit Dhillon. Efficient and non-convex coordinate descent for symmetric nonnegative matrix factorization. *IEEE Transactions on Signal Processing*, 64(21):5571–5584, 2016.

[30] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[31] Jim Jing-Yan Wang, Halima Bensmail, and Xin Gao. Multiple graph regularized nonnegative matrix factorization. *Pattern Recognition*, 46(10):2840–2847, 2013.

[32] Yu-Xiong Wang and Yu-Jin Zhang. Nonnegative matrix factorization: A comprehensive review. *IEEE Transactions on knowledge and data engineering*, 25(6):1336–1353, 2012.

[33] John Wright and Yi Ma. *High-dimensional data analysis with low-dimensional models: Principles, computation, and applications*. Cambridge University Press, 2022.

[34] Wenhui Wu, Yuheng Jia, Sam Kwong, and Junhui Hou. Pairwise constraint propagation-induced symmetric nonnegative matrix factorization. *IEEE transactions on neural networks and learning systems*, 29(12):6348–6361, 2018.

[35] Wei Xu, Xin Liu, and Yihong Gong. 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*, SIGIR '03, page 267–273, New York, NY, USA, 2003. Association for Computing Machinery.

[36] Yan Zhao, Xing Chen, Jun Yin, and Jia Qu. Snmfsmma: using symmetric nonnegative matrix factorization and kronecker regularized least squares to predict potential small molecule-microrna association. *RNA biology*, 17(2):281–291, 2020.

[37] Zhihui Zhu, Xiao Li, Kai Liu, and Qiuwei Li. Dropping symmetry for fast symmetric nonnegative matrix factorization. *arXiv preprint arXiv:1811.05642*, 2018.