On Constrained Sparse Matrix Factorization

Wei-Shi Zheng[1] Stan Z. Li[2] J. H. Lai[3] Shengcai Liao[2]
[1]Department of Mathematics, Sun Yat-sen University, Guangzhou, P. R. China, sunnyweishi@gmail.com
[2]CBSR & NLPR, Institute of Automation, CAS, Beijing, P. R. China, {szli,scliao}@nlpr.ia.ac.cn
[3]Department of Electronics Engineering, Sun Yat-sen University, P. R. China, stsljh@mail.sysu.edu.cn

Abstract

Various linear subspace methods can be formulated in the notion of matrix factorization in which a cost function is minimized subject to some constraints. Among them, constraints on sparseness have received much attention recently. Some popular constraints such as non-negativity, lasso penalty, and (plain) orthogonality etc have been so far applied to extract sparse features. However, little work has been done to give theoretical and experimental analyses on the differences of the impacts of different constraints within a framework. In this paper, we analyze the problem in a more general framework called Constrained Sparse Matrix Factorization (CSMF). In CSMF, a particular case called CSMF with non-negative components (CSMFnc) is further discussed. Unlike NMF, CSMFnc allows not only additive but also subtractive combinations of non-negative sparse components. It is useful to produce much sparser features than those produced by NMF and meanwhile have better reconstruction ability, achieving a trade-off between sparseness and low MSE value. Moreover, for optimization, an alternating algorithm is developed and a gentle update strategy is further proposed for handling the alternating process. Experimental analyses are performed on the Swimmer data set and CBCL face database. In particular, CSMF can successfully extract all the proper components without any ghost on Swimmer, gaining a significant improvement over the compared well-known algorithms.

1. Introduction

Learning object representation is an important topic in computer vision and pattern recognition. So far, linear subspace analysis is popular for object representation. It can be formulated in the notion of matrix factorization (MF) that training data matrix **X** is approximately factorized into a component matrix **W** and a coefficient matrix **H**.

As a well-known MF technique, principal component analysis (PCA) [15] gets the minimum reconstruction error conditioned that the extracted components are orthogonal. However, PCA can only extract holistic features. In view of this, inspired by psychological and physiological studies, many methods have been proposed to make **W** sparse. In computer vision, this would help extract local features. Local feature analysis (LFA) [4] is an early developed algorithm for extraction of sparse features, but it produces many redundant features. Independent component analysis (ICA) [10] finds statistically independent blind sources. Though it is argued that independent components have connection to localized edge filter, it is not guaranteed that they would be sparse. Recently, sparse principal component analysis (SPCA) [14] was proposed by incorporating lasso constraint in PCA, but the degree of the overlapping between components is not measured.

Unlike PCA, in non-negative matrix factorization (NMF) [1][3] components and coefficients are constrained to be non-negative, in accordance with biological evidences. However, it is experimentally found that non-negativity does not always yield sparsity and additional constraints may have to be used for pursuing sparseness. Local NMF (LNMF) [11], NMF with sparseness constraint (NMFsc) [8], and nonsmooth NMF (nsNMF) [13] etc are then developed.

Though it is argued that non-negativity is supported by biological evidences, mathematical interpretations are still not enough for why non-negativity could make sparseness and when it would fail. Even though some interpretation is given in [12], it is based on the generative model under which any positive sample is assumed to be constructed by a set of positive bases and this model may not be true. For example, any face image is hard to be constructed by a few bases. Moreover, we are curious about why imposition of non-negativity on both components and coefficients is preferred by NMF. Recently, some one-sided non-negativity [6][7] based methods are reported, but they fail to extract sparse components. Then, why do they fail? Is there any advantage of using one-sided non-negativity as constraint?

Though NMF may extract sparse components in some applications, however, not all real data are non-negative. It would be useful to seek a more general algorithm for potential applications in other domains. Moreover, as lasso penalty is popular for sparseness analysis in statistics, what if it is imposed on components or coefficients in vision problems? How does it make differences? If non-negativity is further imposed on components or on both components and coefficients, what will the results be? Recently, a technique called non-negative sparse PCA [2] is proposed, but no discussion on sparseness of coefficients is reported.

In this paper, we analyze the problem about extraction of sparse components in a more general framework called

978-1-4244-1631-8/07/$25.00 ©2007 IEEE

Constrained Sparse Matrix Factorization (CSMF). CSMF can provide a platform for discussion of the impacts of different constraints, such as absolute orthogonality, (plain) orthogonality, lasso penalty and non-negativity. We give analysis on why they can be applicable, how they perform and what their differences are.

We will further consider a special case of CSMF, namely *CSMF with non-negative components* (CSMFnc). Unlike NMF, CSMFnc allows both subtraction and addition in linear combination of components. Unlike some one-sided non-negativity based methods [6][7], CSMFnc can extract sparse features. Analysis will show the advantage of subtractive combination of non-negative components.

For optimization, we propose an alternating procedure as well as an update strategy called *gentle update strategy* in order to handle the alternating process, alleviating the problem of plunging into local optimum.

Experimental analysis as well as theoretical analysis on Swimmer and CBCL data sets is given. It is encouraged to show that CSMF can well extract all the proper components without any ghost on Swimmer, while ghost has been a known problem in existing algorithms.

In the reminder of the paper, we introduce CSMF and CSMFnc in Section 2. In Section 3, computational algorithms are developed for the optimization of CSMF. In Section 4, further theoretical and experimental analyses are given. Finally conclusion is provided in Section 5.

2. Constrained sparse matrix factorization

2.1. A general framework

Suppose given the data matrix $\mathbf{X}=(\mathbf{x}_1, \dots, \mathbf{x}_N) \in \Re^{p\times N}$, where $\mathbf{x}_i$ is the i^{th} sample. *Constrained Sparse Matrix Factorization* (CSMF) is formulated to factorize $\mathbf{X}$ into the product of $\mathbf{W}=(\mathbf{w}_1, \dots, \mathbf{w}_l) \in \Re^{p\times l}$ and $\mathbf{H}=(\mathbf{h}_1, \dots, \mathbf{h}_N) \in \Re^{l\times N}$, where $\mathbf{w}_i=(\mathbf{w}_i(1), \dots, \mathbf{w}_i(n))^T$ and $\mathbf{h}_i=(\mathbf{h}_i(1), \dots, \mathbf{h}_i(l))^T$, so that the reconstruction error $\|\mathbf{X}-\mathbf{WH}\|_F^2$ is minimized with penalty functions and some constraints as follows:

$$\begin{aligned} &\min_{\mathbf{W},\mathbf{H}} \; G(\mathbf{W},\mathbf{H}) = \left\|\mathbf{X}-\mathbf{WH}\right\|_F^2 + g_1(\mathbf{W}) + g_2(\mathbf{H}) \\ &s.t. \;\; \mathbf{W}\in D_1, \;\; \mathbf{H}\in D_2 \end{aligned} \tag{1}$$

where g_1 and g_2 are penalized functions of $\mathbf{W}$ and $\mathbf{H}$ respectively, and D_1 and D_2 are domains of $\mathbf{W}$ and $\mathbf{H}$ respectively. D_1 and D_2 should be specified for different problems. In this study, for constructing functions g_1 and g_2, we would like to consider the following possible penalties:

- Lasso penalty on components, denoted by entrywise l_1-norm of $\mathbf{W}$, i.e., $\|\mathbf{W}\|_1=\sum_{j=1}^{l}\sum_{c=1}^{n}|\mathbf{w}_j(c)|$.
- Lasso penalty on coefficients, denoted by entrywise l_1-norm of $\mathbf{H}$, i.e., $\|\mathbf{H}\|_1=\sum_{k=1}^{N}\sum_{j=1}^{l}|\mathbf{h}_k(j)|$.
- Penalty of absolute orthogonality between any two components $\mathbf{w}_{j_1}$ and $\mathbf{w}_{j_2}$, in the form of

$$I_{j_1,j_2} = \sum_{c=1}^{n} |\mathbf{w}_{j_1}(c)||\mathbf{w}_{j_2}(c)|. \tag{2}$$

The above penalties can be used for extraction of sparse and less overlapped components with sparse encodings. The lasso penalties are famous in statistics and they would yield the sparseness in components and coefficients. The absolute orthogonality penalty would yield low overlapping between components. When $I_{j_1,j_2}=0$, i.e., $\sum_{c=1}^{n}|\mathbf{w}_{j_1}(c)||\mathbf{w}_{j_2}(c)|=0$ for any $j_1\neq j_2$, we then called $\mathbf{w}_{j_1}$ and $\mathbf{w}_{j_2}$ are absolutely orthogonal, indicating no overlapping between them. The absolute operator is useful because even $\mathbf{w}_{j_1}^T\mathbf{w}_{j_2}=\sum_{c=1}^{n}\mathbf{w}_{j_1}(c)\mathbf{w}_{j_2}(c)=0$ but it may still exist that $\sum_{c=1}^{n}|\mathbf{w}_{j_1}(c)||\mathbf{w}_{j_2}(c)|>0$. In Section 4, detailed analysis between absolute orthogonality and lasso penalties are given. Using these penalties, g_1 and g_2 are then specified for discussion in this paper as follows:

$$\begin{aligned} g_1(\mathbf{W}) = {} & \alpha\sum_{j_1=1}^{l}\sum_{j_2=1,j_2\neq j_1}^{l}\sum_{c=1}^{n}|\mathbf{w}_{j_1}(c)||\mathbf{w}_{j_2}(c)| \\ & + \beta\sum_{j=1}^{l}\sum_{c=1}^{n}|\mathbf{w}_j(c)| \end{aligned} \tag{3}$$

$$g_2(\mathbf{H}) = \lambda\sum_{k=1}^{N}\sum_{j=1}^{l}|\mathbf{h}_k(j)| \tag{4}$$

where α, β and λ are non-negative importance weights.

Many existing subspace algorithms could be involved in CSMF. For instance, CSMF is NMF when $D_1=\{\mathbf{W}\geq\mathbf{0}\}$, $D_2=\{\mathbf{H}\geq\mathbf{0}\}$ and $\alpha=\beta=\lambda=0$; CSMF is PCA when $D_1=\{\mathbf{W}\in \Re^{p\times l} \mid \mathbf{w}_{j_1}^T\mathbf{w}_{j_2}=0 \text{ for } \forall j_1\neq j_2\}$, $D_2=\{\mathbf{H}\in\Re^{l\times N}\}$ and $\alpha=\beta=\lambda=0$; CSMF is ICA when $D_1=\{\mathbf{W}\in\Re^{p\times l}\}$, $D_2=\{\mathbf{H}\in\Re^{l\times N} \mid \mathbf{h}_i(1),\dots,\mathbf{h}_i(l) \text{ are independent}, \forall i\}$ and $\alpha=\beta=\lambda=0$. It can also be seen that many variants of NMF, PCA and ICA are also involved in this framework.

The contribution of CSMF is to provide a platform for discussion of the impacts of different constraints on extraction of sparse and localized components. Moreover, the optimization algorithm proposed in Section 3 is general and can be used to generate some interesting special cases of CSMF, such as the CSMFnc discussed in next section.

2.2. Constrained sparse matrix factorization with non-negative components (CSMFnc)

In particular, we study the case when only the components are constrained to be non-negative, obtaining the following criterion, termed *Constrained Sparse Matrix Factorization with non-negative components* (CSMFnc):

$$\begin{aligned} &\min_{\mathbf{W},\mathbf{H}} G(\mathbf{W},\mathbf{H}) = \left\|\mathbf{X}-\mathbf{WH}\right\|_F^2 \\ &\quad + \alpha\sum_{j_1=1}^{l}\sum_{j_2=1,j_2\neq j_1}^{l}\sum_{c=1}^{n}\mathbf{w}_{j_1}(c)\mathbf{w}_{j_2}(c) \\ &\quad + \beta\sum_{j=1}^{l}\sum_{c=1}^{n}\mathbf{w}_j(c) + \lambda\sum_{k=1}^{N}\sum_{j=1}^{l}|\mathbf{h}_k(j)| \\ &s.t. \;\; \mathbf{W}\geq\mathbf{0} \;\&\; \mathbf{H}\in\Re^{l\times N} \end{aligned} \tag{5}$$

Unlike the NMF based methods in which only additive combination of non-negative components is allowed, CSMFnc allows both subtractive and additive combination of non-negative components. It is intuitively understood that representation of a complex object such as face using CSMFnc can be done by adding some sparse non-negative components and meanwhile removing some ones. If sometimes sparse encodings rather than sparse components are highly desirable, CSMFnc rather than NMF may be preferred. For example, there are six images illustrated in Fig. 1 (a), where black pixel indicates the zero gray value. Then, CSMFnc could ideally yield at least two smallest groups of components as shown in (d) and (e) respectively, but NMF only yields Fig. 1 (e) as its basis components. Imagining the pessimistic case that the first image where an

entire white rectangle exists in the left in Fig. 1 (a) appears frequently in an image series, NMF then will not yield sparse encodings of the image series. In contrast, sparse encodings could be gained by CSMFnc when Fig. 1 (d) is selected as the basis components.

Some one-sided non-negativity based algorithms such as NICA [6] which in contrast imposes non-negativity on the coefficients in ICA, so far could not exact sparse and non-negative components. Some reasons may be because no absolute orthogonality or sparseness constraint is used. Moreover the maximum number of the components learned by NICA would depend on PCA due to the whitening step and NICA may fail to extract proper components required.

3. Optimization algorithms

In Section 3.1, an alternating algorithm is first developed for optimization of the criterion of CSMF given by Eq. (1). In Section 3.2, a gentle update strategy is further provided to handle the alternating process. Convergence of the algorithm is finally discussed. Analysis could be generalized to CSMFnc with tiny modifications.

3.1. An alternating algorithm for optimization

As Eq. (1) is not convex, it may be hard to find a globally optimal solution. Thus, we herein develop an alternating algorithm to find a locally optimal solution. The algorithm first addresses the general case when $D_1=\{\mathbf{W}\in \Re^{m\times l}\}$ and $D_2=\{\mathbf{H}\in \Re^{l\times N}\}$ and can be easily generalized to other cases.

■ **Optimize $\mathbf{w}_i(r)$ for fixed $\{\mathbf{w}_j(c),(c,j)\neq(r,i)\}$ and $\{\mathbf{h}_k(j)\}$**

We first rewrite the criterion so that it is useful for derivation of the optimal solution. Denote $\mathbf{W}^T$ by $\widetilde{\mathbf{W}}$ and let $\widetilde{\mathbf{W}}=(\widetilde{\mathbf{w}}_1,\cdots,\widetilde{\mathbf{w}}_n)$, where $\widetilde{\mathbf{w}}_c=(\widetilde{\mathbf{w}}_c(1),...,\widetilde{\mathbf{w}}_c(l))^T$ and $\widetilde{\mathbf{w}}_c(j)=\mathbf{w}_j(c)$. Denote $\widetilde{\mathbf{X}}=\mathbf{X}^T=(\widetilde{\mathbf{x}}_1,\cdots,\widetilde{\mathbf{x}}_n)$, then the criterion could be rewritten as follows:

$$\begin{aligned} G(\mathbf{W},\mathbf{H}) &= \|\widetilde{\mathbf{X}}-\mathbf{H}^T\widetilde{\mathbf{W}}\|_F^2+\alpha\sum_{c=1}^{n}\sum_{j1=1}^{l}\sum_{j2=1,j2\neq j1}^{l}|\widetilde{\mathbf{w}}_c(j_1)||\widetilde{\mathbf{w}}_c(j_2)| \\ &\quad +\beta\sum_{c=1}^{n}\sum_{j=1}^{l}|\widetilde{\mathbf{w}}_c(j)|+\lambda\sum_{k=1}^{N}\sum_{j=1}^{l}|\mathbf{h}_k(j)| \\ &= \sum_{c=1}^{n}\Big[\|\widetilde{\mathbf{x}}_c-\mathbf{H}^T\widetilde{\mathbf{w}}_c\|_F^2+\alpha\sum_{j1=1}^{l}\sum_{j2=1,j2\neq j1}^{l}|\widetilde{\mathbf{w}}_c(j_1)||\widetilde{\mathbf{w}}_c(j_2)| \\ &\quad +\beta\sum_{j=1}^{l}|\widetilde{\mathbf{w}}_c(j)|\Big]+\lambda\sum_{k=1}^{N}\sum_{j=1}^{l}|\mathbf{h}_k(j)| \end{aligned} \tag{6}$$

Note that optimizing $\mathbf{w}_i(r)$ is equal to optimizing $\widetilde{\mathbf{w}}_r(i)$. Thus for fixed $\{\mathbf{w}_j(c),(c,j)\neq(r,i)\}$ and $\{\mathbf{h}_k(j)\}$, $\widetilde{\mathbf{w}}_r(i)$ can be equivalently optimized by minimizing the formula below:

$$\widetilde{G}(\widetilde{\mathbf{w}}_r(i))=\|\widetilde{\mathbf{x}}_r-\mathbf{H}^T\widetilde{\mathbf{w}}_r\|_F^2+\alpha\sum_{j1=1}^{l}\sum_{\substack{j2=1,\\ j2\neq j1}}^{l}|\widetilde{\mathbf{w}}_r(j_1)||\widetilde{\mathbf{w}}_r(j_2)|+\beta\sum_{j=1}^{l}|\widetilde{\mathbf{w}}_r(j)| \tag{7}$$

With some efforts, we can have:

$$\begin{aligned} &\|\widetilde{\mathbf{x}}_r-\mathbf{H}^T\widetilde{\mathbf{w}}_r\|_F^2=\widetilde{\mathbf{x}}_r^T\widetilde{\mathbf{x}}_r-2\widetilde{\mathbf{x}}_r^T\mathbf{H}^T\widetilde{\mathbf{w}}_r+\widetilde{\mathbf{w}}_r^T\mathbf{H}\mathbf{H}^T\widetilde{\mathbf{w}}_r \\ &=\widetilde{\mathbf{x}}_r^T\widetilde{\mathbf{x}}_r+\sum_{k=1}^{N}(\sum_{j=1}^{l}\widetilde{\mathbf{w}}_r(j)\mathbf{h}_k(j))^2-2\sum_{j=1}^{l}(\sum_{k=1}^{N}\widetilde{\mathbf{x}}_r(k)\mathbf{h}_k(j))\widetilde{\mathbf{w}}_r(j) \\ &=[\sum_{k=1}^{N}\mathbf{h}_k^2(i)]\widetilde{\mathbf{w}}_r^2(i)+[2\sum_{k=1}^{N}\mathbf{h}_k(i)\sum_{j=1,j\neq i}^{l}\widetilde{\mathbf{w}}_r(j)\mathbf{h}_k(j) \\ &\quad -2\sum_{k=1}^{N}\widetilde{\mathbf{x}}_r(k)\mathbf{h}_k(i)]\widetilde{\mathbf{w}}_r(i)+C_1^{r,i} \\ &\alpha\sum_{j1=1}^{l}\sum_{j2=1,j2\neq j1}^{l}|\widetilde{\mathbf{w}}_r(j_1)\|\widetilde{\mathbf{w}}_r(j_2)|+\beta\sum_{j=1}^{l}|\widetilde{\mathbf{w}}_r(j)| \\ &=\begin{cases}(2\alpha\sum_{j=1,j\neq i}^{l}|\widetilde{\mathbf{w}}_r(j)|+\beta)\widetilde{\mathbf{w}}_r(i)+C_2^{r,i}, & \widetilde{\mathbf{w}}_r(i)\geq 0\\ -(2\alpha\sum_{j=1,j\neq i}^{l}|\widetilde{\mathbf{w}}_r(j)|+\beta)\widetilde{\mathbf{w}}_r(i)+C_2^{r,i}, & \widetilde{\mathbf{w}}_r(i)<0\end{cases} \end{aligned}$$

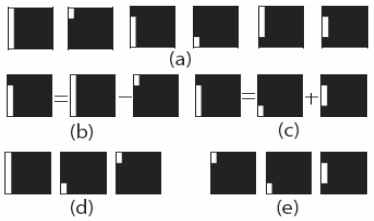


Figure 1. Illustration of the decomposition using CSMFnc. (a) Six image samples; (b) and (c) are two examples of construction; (d) and (e) are two alternative bases found by CSMFnc from (a).

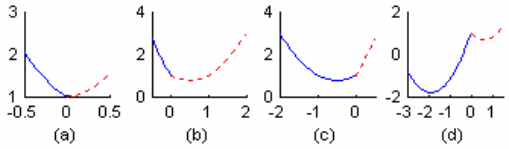


Figure 2. (a)~(c) are examples of $\widetilde{G}^{r,i}(\widetilde{\mathbf{w}}_r(i))$ exist in 3 possible ways, while the case (d) would not exist. Blue solid curve indicates $\widetilde{G}_-^{r,i}(\widetilde{\mathbf{w}}_r(i))$ and red dot curve indicates $\widetilde{G}_+^{r,i}(\widetilde{\mathbf{w}}_r(i))$.

where $C_1^{r,i}$ and $C_2^{r,i}$ are constant values independent of $\widetilde{\mathbf{w}}_r(i)$. It thus can be verified that for fixed $\{\widetilde{\mathbf{w}}_c(j),(c,j)\neq(r,i)\}$ and $\{\mathbf{h}_k(j)\}$, minimizing $G(\mathbf{W},\mathbf{H})$ with respect to $\widetilde{\mathbf{w}}_r(i)$ is equal to minimizing the following formula with respect to $\widetilde{\mathbf{w}}_r(i)$:

$$\widetilde{G}^{r,i}(\widetilde{\mathbf{w}}_r(i))=\begin{cases}\widetilde{G}_+^{r,i}(\widetilde{\mathbf{w}}_r(i))=a^{r,i}\widetilde{\mathbf{w}}_r^2(i)+b_+^{r,i}\widetilde{\mathbf{w}}_r(i)+C^{r,i}, & \widetilde{\mathbf{w}}_r(i)\geq 0\\ \widetilde{G}_-^{r,i}(\widetilde{\mathbf{w}}_r(i))=a^{r,i}\widetilde{\mathbf{w}}_r^2(i)+b_-^{r,i}\widetilde{\mathbf{w}}_r(i)+C^{r,i}, & \widetilde{\mathbf{w}}_r(i)<0\end{cases} \tag{8}$$

where

$$\begin{aligned} &a^{r,i}=[\sum_{s=1}^{N}\mathbf{h}_s^2(i)] \\ &b_+^{r,i}=b_0^{r,i}+2\alpha\sum_{j=1,j\neq i}^{l}|\widetilde{\mathbf{w}}_r(j)|+\beta \\ &b_-^{r,i}=b_0^{r,i}-2\alpha\sum_{j=1,j\neq i}^{l}|\widetilde{\mathbf{w}}_r(j)|-\beta \\ &b_0^{r,i}=2\sum_{k=1}^{N}\mathbf{h}_k(i)\sum_{j=1,j\neq i}^{l}\widetilde{\mathbf{w}}_r(j)\mathbf{h}_k(j)-2\sum_{k=1}^{N}\widetilde{\mathbf{x}}_r(k)\mathbf{h}_k(i) \\ &C^{r,i}=C_1^{r,i}+C_2^{r,i} \end{aligned}$$

It is interesting to show the possible existing styles of function $\widetilde{G}^{r,i}(\widetilde{\mathbf{w}}_r(i))$ in Fig. 2, where Fig. 2 (d) would not exist because $2\alpha\sum_{j=1,j\neq i}^{l}|\widetilde{\mathbf{w}}_r(j)|+\beta\geq 0$ for ever.

Finally, the optimal solution is given by the following lemma and theorem, and the proofs are omitted.

Lemma 1. $\widetilde{\mathbf{w}}_r^+(i)=\arg\min_{\widetilde{\mathbf{w}}_r(i)\geq 0}\widetilde{G}_+^{r,i}(\widetilde{\mathbf{w}}_r(i))=\max(-\frac{b_+^{r,i}}{2a^{r,i}},0)$ *and* $\widetilde{\mathbf{w}}_r^-(i)=\arg\min_{\widetilde{\mathbf{w}}_r(i)<0}\widetilde{G}_-^{r,i}(\widetilde{\mathbf{w}}_r(i))=\min(-\frac{b_-^{r,i}}{2a^{r,i}},0)$.

Theorem 1. *Suppose* $\widetilde{\mathbf{w}}_r^{opt}(i)$ *is the minimum solution of* $\widetilde{G}^{r,i}(\widetilde{\mathbf{w}}_r(i))$. *Then* $\widetilde{\mathbf{w}}_r^{opt}(i)=\widetilde{\mathbf{w}}_r^+(i)$ *if* $\widetilde{G}_+^{r,i}(\widetilde{\mathbf{w}}_r^+(i))<\widetilde{G}_-^{r,i}(\widetilde{\mathbf{w}}_r^-(i))$, *and* $\widetilde{\mathbf{w}}_r^{opt}(i)=\widetilde{\mathbf{w}}_r^-(i)$ *otherwise.*

■ **Optimize $\mathbf{h}_s(i)$ for fixed $\{\mathbf{w}_j(c)\}$ and $\{\mathbf{h}_k(j),(k,j)\neq(s,i)\}$**

We know $\|\mathbf{X}\text{-}\mathbf{WH}\|_F^2=\sum_{k=1}^{N}\|\mathbf{x}_k\text{-}\mathbf{W}\mathbf{h}_k\|_F^2$. Hence for fixed $\{\mathbf{w}_j(c)\}$ and $\{\mathbf{h}_k(j),(k,j)\neq(s,i)\}$, minimizing $G(\mathbf{W},\mathbf{H})$ with respect to $\mathbf{h}_s(i)$ is equal to minimizing the formula below:

$$\begin{aligned} \widetilde{G}^{s,i}(\mathbf{h}_s(i)) &= \|\mathbf{x}_s-\mathbf{W}\mathbf{h}_s\|_F^2+\lambda\sum_{j=1}^{l}|\mathbf{h}_s(j)| \\ &= \mathbf{h}_s^T\mathbf{W}^T\mathbf{W}\mathbf{h}_s-2\mathbf{h}_s^T\mathbf{W}^T\mathbf{x}_s+\mathbf{x}_s^T\mathbf{x}_s+\lambda\sum_{j=1}^{l}|\mathbf{h}_s(j)| \\ &= \sum_{c=1}^{n}(\sum_{j=1}^{l}\mathbf{h}_s(j)\mathbf{w}_j(c))^2-2\sum_{j=1}^{l}\mathbf{h}_s(j)(\mathbf{w}_j^T\mathbf{x}_s) \\ &\quad +\lambda\sum_{j=1}^{l}|\mathbf{h}_s(j)|+\mathbf{x}_s^T\mathbf{x}_s \\ &= \sum_{j_1=1}^{l}\sum_{j_2=1}^{l}\sum_{c=1}^{n}\mathbf{h}_s(j_1)\mathbf{h}_s(j_2)\mathbf{w}_{j_1}(c)\mathbf{w}_{j_2}(c) \\ &\quad -2\sum_{j=1}^{l}\mathbf{h}_s(j)\mathbf{w}_j^T\mathbf{x}_s+\lambda\sum_{j=1}^{l}|\mathbf{h}_s(j)|+\mathbf{x}_s^T\mathbf{x}_s \end{aligned} \tag{9}$$

$$= \begin{cases} \widetilde{G}_{+}^{s,i}(\mathbf{h}_s(i)) = \widetilde{a}^{s,i}\mathbf{h}_s^2(i) + \widetilde{b}_{+}^{s,i}\mathbf{h}_s(i) + \widetilde{C}^{s,i}, \mathbf{h}_s(i) \geq 0 \\ \widetilde{G}_{-}^{s,i}(\mathbf{h}_s(i)) = \widetilde{a}^{s,i}\mathbf{h}_s^2(i) + \widetilde{b}_{-}^{s,i}\mathbf{h}_s(i) + \widetilde{C}^{s,i}, \mathbf{h}_s(i) < 0 \end{cases}$$

where

$\widetilde{a}^{s,i} = [\sum_{c=1}^{n} \mathbf{w}_i^2(c)]$

$\widetilde{b}_{+}^{s,i} = -2\mathbf{w}_i^T\mathbf{x}_s + 2\sum_{j=1,j\neq i}^{l}\sum_{c=1}^{n}\mathbf{h}_s(j)\mathbf{w}_j(c)\mathbf{w}_i(c) + \lambda$

$\widetilde{b}_{-}^{s,i} = -2\mathbf{w}_i^T\mathbf{x}_s + 2\sum_{j=1,j\neq i}^{l}\sum_{c=1}^{n}\mathbf{h}_s(j)\mathbf{w}_j(c)\mathbf{w}_i(c) - \lambda$

$\widetilde{C}^{s,i}$ is a constant independent of $\mathbf{h}_s(i)$.

Without proofs, we similarly get the following conclusions.

Lemma 2. $\mathbf{h}_s^{+}(i) = \arg\min_{\mathbf{h}_s(i)\geq 0}\widetilde{G}_{+}^{s,i}(\mathbf{h}_s(i)) = \max(-\frac{\widetilde{b}_{+}^{s,i}}{2\widetilde{a}^{s,i}}, 0)$ *and* $\mathbf{h}_s^{-}(i) = \arg\min_{\mathbf{h}_s(i)<0}\widetilde{G}_{-}^{s,i}(\mathbf{h}_s(i)) = \min(-\frac{\widetilde{b}_{-}^{s,i}}{2\widetilde{a}^{s,i}}, 0)$.

Theorem 2. *Suppose* $\mathbf{h}_s^{opt}(i)$ *is the minimum solution of* $\widetilde{G}^{s,i}(\mathbf{h}_s(i))$. *Then* $\mathbf{h}_s^{opt}(i) = \mathbf{h}_s^{+}(i)$ *if* $\widetilde{G}_{+}^{s,i}(\mathbf{h}_s^{+}(i)) < \widetilde{G}_{-}^{s,i}(\mathbf{h}_s^{-}(i))$, *and* $\mathbf{h}_s^{opt}(i) = \mathbf{h}_s^{-}(i)$ *otherwise*.

Based on the above optimization scheme of CSMF, optimization scheme of CSMFnc for minimizing Eq. (5) could be easily derived by only changing the update rule of the components. First, we find that when $\widetilde{\mathbf{w}}_r(i) \geq 0$, $\widetilde{G}^{r,i}(\widetilde{\mathbf{w}}_r(i))$ in Eq. (8) could be reduced to:

$$\widetilde{G}^{r,i}(\widetilde{\mathbf{w}}_r(i)) = a^{r,i}\widetilde{\mathbf{w}}_r^2(i) + b_{+}^{r,i}\widetilde{\mathbf{w}}_r(i) + C^{r,i}, \widetilde{\mathbf{w}}_r(i) \geq 0 \qquad (10)$$

Then the components in CSMFnc are updated as follows.

Theorem 3. *Suppose* $\widetilde{\mathbf{w}}_r^{opt}(i)$ *is the minimum solution of* $\widetilde{G}^{r,i}(\widetilde{\mathbf{w}}_r(i))$ *when* $\widetilde{\mathbf{w}}_r(i) \geq 0$. *Then*

$$\widetilde{\mathbf{w}}_r^{opt}(i) = \arg\min_{\widetilde{\mathbf{w}}_r(i)\geq 0}\widetilde{G}_{+}^{r,i}(\widetilde{\mathbf{w}}_r(i)) = \max(-\frac{b_{+}^{r,i}}{2a^{r,i}}, 0) \qquad (11)$$

3.2. Gentle update strategy

In the last section, a locally optimal solution is obtained. However, it would highly depend on the implementation of the alternating procedure and an unsatisfied locally optimal solution may be learned. In view of this, we propose an adaptive strategy to handle the alternating process for learning a better local optimum. The proposed strategy is to select a subset of parameters either from the component part or from the coefficient part for update at each step, rather than updating all of them. We call this strategy the *gentle update strategy*. Algorithm 1 is an overview of the strategy and details are given as follows.

■ **Gentle update of components**. As $\mathbf{W}$ can be updated by the update of $\widetilde{\mathbf{W}}$, at the t^{th} step a subset $\{\widetilde{\mathbf{w}}_{i'_j}\}_{j=1}^{N_t^{\mathbf{w}}}$ constituted by two parts is then selected for update. For the first part, $N_{t,1}^{\mathbf{w}}$ indexes $\{\widetilde{i}'_1, \cdots, \widetilde{i}'_{N_{t,1}^{\mathbf{w}}}\}$ are cyclicly selected from $\{1, \ldots, n\}$ in an orderly manner from the start of the alternating process. For the second part, $N_{t,2}^{\mathbf{w}}$ indexes $\{\hat{i}'_1, \cdots, \hat{i}'_{N_{t,2}^{\mathbf{w}}}\}$ are selected such that the subset $\{\widetilde{G}(\widetilde{\mathbf{w}}_r), r \in \{\hat{i}'_1, \cdots, \hat{i}'_{N_{t,2}^{\mathbf{w}}}\}\}$ consists of the largest $N_{t,2}^{\mathbf{w}}$ values in $\{\widetilde{G}(\widetilde{\mathbf{w}}_r), r = 1, \cdots, n\}$, where

$$\widetilde{G}(\widetilde{\mathbf{w}}_r) = \|\widetilde{\mathbf{x}}_r - \mathbf{H}^T\widetilde{\mathbf{w}}_r\|_F^2 + \alpha\sum_{j1=1}^{l}\sum_{j2=1,j2\neq j1}^{l}|\widetilde{\mathbf{w}}_r(j1)||\widetilde{\mathbf{w}}_r(j2)| + \beta\sum_{j=1}^{l}|\widetilde{\mathbf{w}}_r(j)| \qquad (12)$$

By Eq. (6)~(7), we find that $\widetilde{G}(\widetilde{\mathbf{w}}_r)$ can fully reflect the contribution of $\widetilde{\mathbf{w}}_r$ in minimization of the criterion. (Eq. (12) can also be used for CSMFnc). Denote the selected indexes by $\{i'_1, \cdots, i'_{N_t^{\mathbf{w}}}\} = \{\widetilde{i}'_1, \cdots, \widetilde{i}'_{N_{t,1}^{\mathbf{w}}}\} \bigcup \{\hat{i}'_1, \cdots, \hat{i}'_{N_{t,2}^{\mathbf{w}}}\}$. Then, the first part of this subset gives the opportunity to update all parameters and the second part is to appropriately accelerate the convergence of the algorithm.

Algorithm 1. *Gentle CSMF*

input :
Data matrix : $\mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_N)$
Parameters of gentle update strategy : $N_{0,1}^{\mathbf{w}}$, $N_{0,2}^{\mathbf{w}}$, $N_{0,1}^{\mathbf{h}}$, $N_{0,2}^{\mathbf{h}}$
Parameters of iteration : $loop$, $\varepsilon > 0$
Initilaize $\mathbf{W}$ *and* $\mathbf{H}$
$t \leftarrow 1$
While ($t \leq loop$)
  Let $\mathbf{W}_0 = \mathbf{W}$ *and* $\mathbf{H}_0 = \mathbf{H}$ // *save old values*
  // *update* $\mathbf{W}$ *using gentle update strategy*
  $\widetilde{\mathbf{W}} = \mathbf{W}^T$
  Set $N_{t,1}^{\mathbf{w}} = N_{0,1}^{\mathbf{w}}$, $N_{t,2}^{\mathbf{w}} = N_{0,2}^{\mathbf{w}}$ *and select the subset* $\{\widetilde{\mathbf{w}}_{i'_j}\}_{j=1}^{N_t^{\mathbf{w}}}$ *for update*
  For $j = 1 : N_t^{\mathbf{w}}$ *update* $\widetilde{\mathbf{w}}_{i'_j}$ *by* theorem 1; **End**
  $\mathbf{W} = \widetilde{\mathbf{W}}^T$
  // *update* $\mathbf{H}$ *using gentle update strategy*
  Set $N_{t,1}^{\mathbf{h}} = N_{0,1}^{\mathbf{h}}$, $N_{t,2}^{\mathbf{h}} = N_{0,2}^{\mathbf{h}}$ *and select the subset* $\{\mathbf{h}_{i''_j}\}_{j=1}^{N_t^{\mathbf{h}}}$ *for update*
  For $j = 1 : N_t^{\mathbf{h}}$ *update* $\mathbf{h}_{i''_j}$ *by* theorem 2; **End**
  // *see whether the iteration can be terminated*
  If $|G(\mathbf{W},\mathbf{H}) - G(\mathbf{W}_0,\mathbf{H}_0)| \geq \varepsilon$, $t \leftarrow t+1$. *Otherwise exist*.
End
output : $\mathbf{W}, \mathbf{H}$

■ **Gentle update of coefficients**. Similarly, at the t^{th} step, there are two parts constituting a subset $\{\mathbf{h}_{i''_j}\}_{j=1}^{N_t^{\mathbf{h}}}$ for update. First, $N_{t,1}^{\mathbf{h}}$ indexes $\{\widetilde{i}''_1, \cdots, \widetilde{i}''_{N_{t,1}^{\mathbf{h}}}\}$ are cyclicly selected from $\{1, \cdots, N\}$ in an orderly manner from the start of the alternating process. Second, the rest $N_{t,2}^{\mathbf{h}}$ indexes $\{\hat{i}''_1, \cdots, \hat{i}''_{N_{t,2}^{\mathbf{h}}}\}$ are selected such that the subset $\{\widetilde{G}(\mathbf{h}_s), s \in \{\hat{i}''_1, \cdots, \hat{i}''_{N_{t,2}^{\mathbf{h}}}\}\}$ consists of the largest $N_{t,2}^{\mathbf{h}}$ values in $\{\widetilde{G}(\mathbf{h}_s), s = 1, \cdots, N\}$, where

$$\widetilde{G}(\mathbf{h}_s) = \|\mathbf{x}_s - \mathbf{W}\mathbf{h}_s\|_F^2 + \lambda\sum_{j=1}^{l}|\mathbf{h}_s(j)| \qquad (13)$$

It is based on Eq. (9). So, for update of the coefficients, the selected index set is $\{i''_1, \cdots, i''_{N_t^{\mathbf{h}}}\} = \{\widetilde{i}''_1, \cdots, \widetilde{i}''_{N_{t,1}^{\mathbf{h}}}\} \bigcup \{\hat{i}''_1, \cdots, \hat{i}''_{N_{t,2}^{\mathbf{h}}}\}$.

For implementation of gentle update strategy, in the experiment we let $N_{t,1}^{\mathbf{w}}$, $N_{t,2}^{\mathbf{w}}$, $N_{t,1}^{\mathbf{h}}$ and $N_{t,2}^{\mathbf{h}}$ be constants $N_{0,1}^{\mathbf{w}}$, $N_{0,2}^{\mathbf{w}}$, $N_{0,1}^{\mathbf{h}}$ and $N_{0,2}^{\mathbf{h}}$ respectively, being independent of t as shown in Algorithm 1. Also, we will control the algorithm by the maximum iteration number while setting ε very small.

Convergence. The alternating algorithm would be converged, because $G(\mathbf{W},\mathbf{H}) \geq 0$ for any $\mathbf{W}$ and $\mathbf{H}$, and the function G will decrease after each update. So there must be a step at which $|G(\mathbf{W},\mathbf{H}) - G(\mathbf{W}_0,\mathbf{H}_0)| < \varepsilon$ in Algorithm 1.

Discussion. Though alternating technique is welcome for optimization of non-convex problem, however, little work is proposed for handling the alternating process. Alternating technique implemented in traditional way, for instance in our learning, will at each step first update all components and then update all coefficients. That is to set $N_{0,1}^{\mathbf{w}}=n$, $N_{0,2}^{\mathbf{w}}=0$, $N_{0,1}^{\mathbf{h}}=N$ and $N_{0,2}^{\mathbf{h}}=0$ in gentle update strategy. However, our study shows handling the process is important. Due to the limited room of the paper, we can only show an example in Fig. 4 (c) that the update using the traditional way would yield an unsatisfied result. Hence, handling the alternating process is useful.

4. Theory & experiment analyses

We first further interpret the sparseness in CSMF. Then experimental analysis and theoretical analysis are given on synthetic and real-world data sets in Section 4.2.

4.1. Sparseness in CSMF

As we state absolute orthogonality and lasso constraints can make components or coefficients sparse, the followings theorems tell how they work and the proofs are omitted.

Theorem 3. *If $\alpha\to+\infty$ or $\beta\to+\infty$, then by lemma 1, $\tilde{\mathbf{w}}_r^+(i)\to 0$ and $\tilde{\mathbf{w}}_r^-(i)\to 0$, and consequently $\tilde{\mathbf{w}}_r^{opt}(i)\to 0$ in theorem 1.*

Theorem 4. *If $\lambda\to+\infty$, then by lemma 2, $\mathbf{h}_s^+(i)\to 0$ and $\mathbf{h}_s^-(i)\to 0$, and consequently $\mathbf{h}_s^{opt}(i)\to 0$ in theorem 2.*

In applications, the sparseness could be gained without assigning large values to α, β and λ. Interestingly, both α and β could control the degree of component sparseness by theorem 3. Their differences would be analyzed later.

Next, we know that CSMF is NMF when $D_1=\{\mathbf{W}\geq\mathbf{0}\}$, $D_2=\{\mathbf{H}\geq\mathbf{0}\}$ and $\alpha=\beta=\lambda=0$. However, though non-negativity is popularly used for extracting sparse features, it still lacks of mathematical interpretations when it works, when it will fail and why imposing non-negativity on components and coefficients simultaneously are preferred. We now try to give some interpretations based on the optimization scheme in Section 3. First, we develop an alternative algorithm for solving NMF within the framework of CSMF. By modifying theorem 1 and theorem 2, the update rule is as follows.

Theorem 5. For $D_1=\{\mathbf{W}\geq\mathbf{0}\}$, $D_2=\{\mathbf{H}\geq\mathbf{0}\}$ and $\alpha=\beta=\lambda=0$, CSMF is updated by (1) $\tilde{\mathbf{w}}_r^{opt}(i)=\tilde{\mathbf{w}}_r^+(i)=\max(-\frac{b_+^{r,i}}{2a^{r,i}},0)$; (2) $\mathbf{h}_s^{opt}(i)=\mathbf{h}_s^+(i)=\max(-\frac{\tilde{b}_+^{s,i}}{2\tilde{a}^{s,i}},0)$.

It shows when $b_+^{r,i}$ or $\tilde{b}_+^{s,i}$ is positive, $\tilde{\mathbf{w}}_r^{opt}(i)$ or $\mathbf{h}_s^{opt}(i)$ will be zero, yielding sparseness in components or coefficients. This interprets why non-negativity constraint may yield sparseness. However, if $b_+^{r,i}$ or $\tilde{b}_+^{s,i}$ is negative, $\tilde{\mathbf{w}}_r^{opt}(i)$ or $\mathbf{h}_s^{opt}(i)$ will then be positive. Thus just imposing non-negativity may not be enough always. So additional constraints are used since larger positive values α, β and λ would bring $b_+^{r,i}$ and $\tilde{b}_+^{s,i}$ towards positive values, where in [9] sparseness is imposed on coefficients. With similar reason, when only one-sided non-negativity is imposed, the positivity of $b_+^{r,i}$ or $\tilde{b}_+^{s,i}$ seems more undetermined according to their formulas. Next section will give some more insight.

As a more general model of NMF, the *CSMF with non-negative components and coefficients* (CSMFncc) will be used for comparison in next section. Theorem 5 is still applicable for the optimization of CSMFncc and the condition "$\alpha=\beta=\lambda=0$" can be removed from the theorem.

4.2. Evaluation & analysis

We discuss the impacts of different constraints in CSMF in two cases: ground truth decomposition and approximate decomposition. In the experiment, the components of all iterative algorithms are initialized with the same positive values. The coefficients of CSMF and CSMFnc are initialized as the least square solutions of the reconstruction term since non-negativity is not needed, and for NMF based methods they are randomly initialized with the same positive values. The parameters of the compared methods are tuned by trying our best or suggested by the authors.

4.2.1 Case of ground truth decomposition

In ground truth decomposition, it is assumed that a group of images could be completely represented using local features without any overlapping. So experiment as well as theoretical analysis is performed on Swimmer data set [12]. It is constituted by 256 images of size 32×32. Each image is represented by five parts, a centered invariant part called "torso" of 12 pixels and four "limbs" of 6 pixels appear in one of 4 positions. Some images are shown in Fig. 3.

So far many algorithms have been tested on this data set. However, to the best of our knowledge, ghost is still a problem and the exact 17 factors are still not factorized out.

Figure 3. Some Images in Swimmer Data Set.

■ **Absolute orthogonality**. In ground truth decomposition, a group of images could be completely represented by a set of sparse components $\mathbf{w}_1,\ldots,\mathbf{w}_l$ with no overlapping between them. So the components should be absolutely orthogonal. Then, it is optimistic to only use absolute orthogonality for extraction of sparse features and get the same reconstruction performance as PCA. It is because when $\beta=\lambda=0$ for any $\alpha>0$, if absolute orthogonality is satisfied, we have:

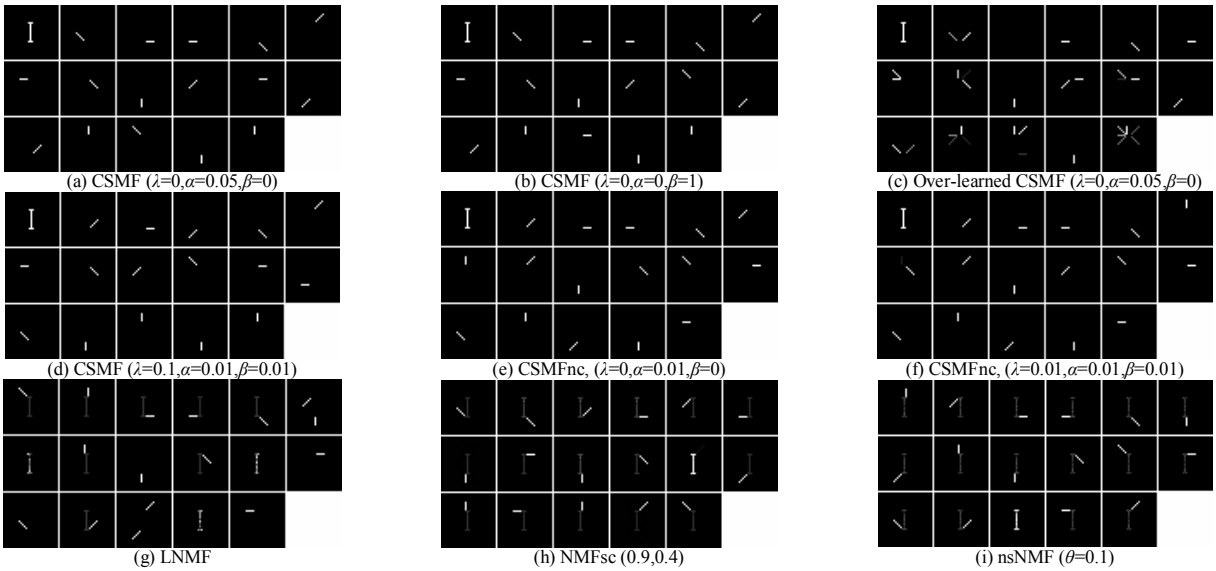
$$G(\mathbf{W},\mathbf{H})=\|\mathbf{X}-\mathbf{WH}\|_F^2+g_1(\mathbf{W})+g_2(\mathbf{H})=\|\mathbf{X}-\mathbf{WH}\|_F^2. \quad (14)$$

In the experiment, when $\alpha=0.05$, $\beta=\lambda=0$, $N_{0,1}^{\mathbf{w}}=600$, $N_{0,2}^{\mathbf{w}}=200$, $N_{0,1}^{\mathbf{h}}=200$ and $N_{0,2}^{\mathbf{h}}=100$, CSMF extracts all the proper components as shown in Fig. 4(a) and the difference of MSE (mean square error) results between CSMF and PCA is 1.4×10^{-29}. Moreover, in Fig. 4(e), we see absolute orthogonality is also useful for CSMFnc, where parameter setting for update is: $N_{0,1}^{\mathbf{w}}=200$, $N_{0,2}^{\mathbf{w}}=100$, $N_{0,1}^{\mathbf{h}}=100$ and $N_{0,2}^{\mathbf{h}}=50$.

■ **Lasso $\|\mathbf{W}\|_1$ Constraint**. As analyzed in theorem 3, $\|\mathbf{W}\|_1$ can also be used to control the sparseness of a component, and as shown in Fig. 4 (b) all the proper components are also extracted. In ground truth experiment, the differences between $\|\mathbf{W}\|_1$ and the absolute orthogonality in theory are:
(1). Absolute orthogonality measures the redundancy (overlapping) between components while $\|\mathbf{W}\|_1$ only measures the average cardinality of each component.
(2). In the optimistic case when global optimum is achieved, the learned components with absolute orthogonality penalty are also the best for reconstruction. The ones using constraint $\|\mathbf{W}\|_1$ is not sure, since $G(\mathbf{W},\mathbf{H})>\|\mathbf{X}-\mathbf{WH}\|_F^2$ for $\beta>0$.

■ **Lasso $\|\mathbf{H}\|_1$ Constraint**. To our surprise, in the ground truth decomposition as analyzed above, $\|\mathbf{H}\|_1$ is not required to find all the proper components. However $\|\mathbf{H}\|_1$ in CSMFnc is useful sometimes for learning sparse codings as exemplified in Section 2.2, and as seen later $\|\mathbf{H}\|_1$ is helpful for extraction of sparse components on real-world data set.

■ **Non-negativity**. The experiment of NMF on Swimmer was reported [13] and it is known NMF can not remove the ghost between components. Here we implement its three well-known variants, nsNMF, LNMF and NMFsc, where the parameters of nsNMF and NMFsc have been tuned by our best. The tuned parameter of nsNMF is not the same as



(a) CSMF (λ=0,α=0.05,β=0) (b) CSMF (λ=0,α=0,β=1) (c) Over-learned CSMF (λ=0,α=0.05,β=0)

(d) CSMF (λ=0.1,α=0.01,β=0.01) (e) CSMFnc, (λ=0,α=0.01,β=0) (f) CSMFnc, (λ=0.01,α=0.01,β=0.01)

(g) LNMF (h) NMFsc (0.9,0.4) (i) nsNMF (θ=0.1)

Figure 4. Experiment on Swimmer. (1) For CSMF, its absolute images are shown so that the more the darker pixels are the sparser the component is; (2) Dark pixels are for (almost) zero entries and white pixels are for positive ones; (3) The maximum iteration number in all methods is 2000; (4) For NMFsc, the parameters of controlling the component and coefficient sparseness are 0.9 and 0.4 respectively; for nsNMF, θ is defined in [13]; (5) See text for the parameter settings of CSMF and CSMFnc; (6) (c) is discussed at the end of Section 3.

the one reported in [13] since the initialization is different. The results in Fig. 4 show a ghost depicting the "torso" exists in each component learned by the variants of NMF.

Finally, Fig. 4(d) additionally shows the result of CSMF with λ=0.1, α=0.01, β=0.01 and $N^{\mathbf{w}}_{0,1}$=600, $N^{\mathbf{w}}_{0,2}$=200, $N^{\mathbf{h}}_{0,1}$=150 and $N^{\mathbf{h}}_{0,2}$=50, and Fig. 4(f) shows the results of CSMFnc with λ=0.01, α=0.01, β=0.01 and $N^{\mathbf{w}}_{0,1}$=600, $N^{\mathbf{w}}_{0,2}$=200, $N^{\mathbf{h}}_{0,1}$=150 and $N^{\mathbf{h}}_{0,2}$=50. So, success of CSMF is not restricted to one specific parameter setting.

Summary. (1) While lasso penalty has been widely used for constraint on generic data, in CSMF we introduce the absolute orthogonality penalty and justify its usefulness in ground truth experiment; (2) While non-negativity is popularly used as constraint for extraction of sparse components, CSMF and CSMFncc show constraints using absolute orthogonality and lasso penalties are also useful for matrix factorization to achieve this goal. Moreover, they help further remove the ghost and extract all proper components, while non-negative constraint may not do that.

4.2.2 Case of approximate decomposition

Unlike the ground truth experiment, there is no strong evidence that a group of face images could be completely represented using (a few) limited sparse components. However approximate decomposition with sparse components is still welcome. For evaluation, we first define the average overlapping degree (AOD) between components by the following formula:

$$AOD(\mathbf{W}) = [0.5 \cdot l \cdot (l-1)]^{-1} \sum_{r=1}^{l-1} \sum_{r'=r+1}^{l} \widehat{\mathbf{w}}_r^T \widehat{\mathbf{w}}_{r'} \qquad (15)$$

where $\widehat{\mathbf{w}}_r(i) = \left[\sum_{i=1}^{n} |\mathbf{w}_r(i)|\right]^{-1} \cdot |\mathbf{w}_r(i)|$. We see that $AOD(\mathbf{W})$ is a normalized absolute orthogonality measurement. If there is no overlapping between components, $AOD(\mathbf{W})$ is zero. Otherwise it shall be large, with maximum value being 1.

Experiment are performed on the training set of CBCL[1] [5] constituted by 2429 face images of size 19×19. The pixel values of images are ranging from 0 to 1. In the experiment, 49 components are found. For convenience of analysis, the parameters of gentle update strategy are fixed and indicated in Fig. 5.

First we investigate the effects of absolute orthogonality and lasso constraints on extraction of sparse components without the non-negativity constraint, i.e., $D_1=\{\mathbf{W} \in \Re^{n \times l}\}$ and $D_2=\{\mathbf{H} \in \Re^{l \times N}\}$ in CSMF. Table 1 and 2 show the impact of lasso penalty on coefficients in extraction of sparse components, where either absolute orthogonality or lasso penalty on components is used; table 3 and 4 show the impacts of absolute orthogonality and lasso penalty on components respectively when no penalty is imposed on coefficients. As shown, using lasso $\|\mathbf{H}\|_1$ as constraint is good for extracting sparse features on real-world data, though in ground truth experiment, it may not be needed. From table 1 to 3 we see that using absolute orthogonality penalty could accelerate the process of producing sparse features, while in table 4 using lasso constraint on components seems less effective. This can be further shown by table 2 as compared to table 1. However, we find that larger α would be easier to get $AOD(\mathbf{W})$=NaN. It indicates there are some over-learned components, which are (almost) empty, because $\left[\sum_{i=1}^{n} |\mathbf{w}_r(i)|\right]^{-1} \to +\infty$. In this case, one can set α properly large and then go on producing sparser components by further increasing β and λ as shown in table 5.

Next, we investigate the effect of non-negativity constraint and its differences from other constraints. First,

[1]No preprocessing is done on CBCL here, while some preprocessing, such as removing mean, clipping etc, are first applied by Lee [1] and Hoyer [8] in their codes (it is not clear in [13]). The preprocessing may perform some nonlinear transform on the data.

	λ=0	λ=0.01	λ=0.05	λ=0.08	λ=0.1	λ=0.3	λ=0.5
α=0.1	0.35187	0.27679	0.17524	0.13023	0.11722	0.05768	NaN

Table 1.Impact of Lasso ($\|\mathbf{H}\|_1$) Constraint, β=0, $AOD(\mathbf{W})$

	λ=0	λ=0.01	λ=0.05	λ=0.08	λ=0.1	λ=0.3	λ=0.5
β=0.1	0.52735	0.51248	0.46932	0.4466	0.43335	0.35167	0.31272

Table 2. Impact of Lasso ($\|\mathbf{H}\|_1$) Constraint, α=0, $AOD(\mathbf{W})$

	α=0	α=0.01	α=0.05	α=0.08	α=0.1	α=0.2	α=0.3
β=0.01	0.63806	0.44284	0.37664	0.37208	0.35124	0.32024	NaN
β=0.1	0.52735	0.42875	0.37435	0.36906	0.3478	0.31461	NaN

Table 3.Impact of Absolute Orthogonality, λ=0, $AOD(\mathbf{W})$

	β=0	β=0.01	β=0.05	β=0.08	β=0.1	β=0.2	β=0.3
α=0.01	0.44486	0.44284	0.43606	0.43101	0.42875	0.42081	0.41248
α=0.1	0.35187	0.35124	0.35053	0.34928	0.3478	0.34159	0.34049

Table 4. Impact of Lasso ($\|\mathbf{W}\|_1$) Constraint, λ=0, $AOD(\mathbf{W})$

	λ=0.05	λ=0.1	λ=0.2		λ=0.05	λ=0.1	λ=0.2		λ=0.05	λ=0.1	λ=0.2		λ=0.05	λ=0.1	λ=0.2
β=0.1	0.16034	0.11655	0.07082	β=0.5	0.14251	0.1077	0.06892	β=1	0.13026	0.08478	0.06538	β=1.5	0.1028	0.07391	NaN

Table 5. Impact of Lasso Constraints, α=0.1, $AOD(\mathbf{W})$

we give some visual results in Fig. 5. In the first four columns, non-negativity is gradually imposed in CSMF. It is imposed on components in the second row and on both components and coefficients in the third row. Note that in Fig. 5(k), CSMFncc is actually the NMF implemented by theorem 5. We see that non-negativity helps produce sparse components as supported by theorem 5. However, when non-negativity is imposed on components and coefficients simultaneously, it shows that the ghosts in some components are hard to be removed and also further incorporating other constraints is not effective to alleviate the overlapping between components, since there is no apparent difference among Fig. 5(k), Fig. 5(*l*) and Fig. 5(m). With further results shown in table 6 when non-negativity is imposed on both components and coefficients, $AOD(\mathbf{W})$ only changes a little when other constraints are used. In contrast, $AOD(\mathbf{W})$ changes obviously in table 3 and 4. Moreover, in the third row and fourth column CSMFncc is over-learned. This may be due to the improperly large weights of the constraints. As some more results of CSMFncc are shown in Fig. 6 with different parameter settings, it can still show that no significant change exhibits. On the other hand, when non-negativity is not simultaneously imposed on components and coefficients, the components become sparser when absolute orthogonality and lasso constraints are further imposed as shown from Fig. 5(a) to Fig. 5(d) and from Fig. 5(f) to Fig. 5(i). Very interestingly, using lasso constraint $\|\mathbf{H}\|_1$ could further remove the ghost in the components.

Finally, besides the example shown in Section 2, to see the further advantage of allowing subtraction of non-negative sparse components, we tabulate the MSE results corresponding to the visual results in Fig. 5 and Fig. 6 on CBCL in table 7. Compared to NMF [1], CSMFnc can produce much sparser (more localized) features while lower MSE values can be gained. Note that the MSE of CSMFnc with λ=0.05, α=0.1 and β=1 is 0.70183, while NMF is 0.8631 and CSMFncc is 0.76566. Though nsNMF is motivated for pursuing sparseness in both components and coefficients, however Fig. 5(i) is sparser than Fig. 5(o) with smaller MSE. Though LNMF and NMFsc can extract sparser features, their MSEs are high, being 5.051 and 1.3577 respectively. In fact there should be some trade-off between sparseness and reconstruction ability. Moreover, in Fig. 4 LNMF and NMFsc can not remove the ghost. So, allowing both additive and subtractive combinations of a set of non-negative sparse components may be more useful for representation of complex object in a more accurate way.

	β=0.01	β=0.05	β=0.1	β=0.2	β=0.3
λ=0, α=0.1	0.18287	0.18283	0.18144	0.18015	0.17445
	α=0.01	α=0.05	α=0.1	α=0.2	α=0.3
λ=0, β=0.1	0.18433	0.18355	0.18144	0.17384	0.17012

Table 6. Impact of Constraints, with Non-negativity, $AOD(\mathbf{W})$

Method	MSE	Method	MSE
CSMF (λ=0,α=0,β=0)	0.61576	CSMFnc (λ=0,α=0,β=0)	0.6157
CSMFncc(λ=0.001,α=0.05,β=0.05)	0.75877	CSMFnc (λ=0.05,α=0.1,β=1)	0.70183
CSMFncc(λ=0.001,α=0.01,β=1)	0.76322	CSMFnc (λ=0.1,α=0.05,β=1)	0.74385
CSMFncc (λ=0.001,α=0.1,β=0.001)	0.7575	CSMFnc (λ=0.2,α=0.1,β=1)	0.85249
CSMFncc (λ=0,α=0,β=0)	0.76566	NMF [3] (Euclidean distance)	0.8631
NMFsc (0.8, 0) [8]	1.3577	LNMF [11]	5.051
nsNMF (θ=0.6) [14]	1.5578	PCA	0.60871

Table 7. Reconstruction Errors on CBCL

Summary. (1) Absolute orthogonality can accelerate the process of producing sparse components; (2) Lasso constraint $\|\mathbf{H}\|_1$ helps remove ghost in the components; (3) Non-negativity is useful, but it seems hard to deal with the ghost problem; (4) Subtractive combination of non-negative sparse components may be good for a trade-off between sparseness and lower MSE value.

5. Conclusions

In this paper, the impacts of different constraints for pursuing sparse components and the relationship among them are theoretically and experimentally analyzed in the framework called Constrained Sparse Matrix Factorization (CSMF). The conditions when non-negativity constraint is useful for extraction of sparse components are investigated. It is also found that subtractive combination of non-negative sparse components is effective. Moreover, a gentle update strategy, as a useful technique for pursuing a better local optimum, is suggested for the optimization in CSMF. The proposed model has been finally justified to be effective for elimination of the ghost between components. In future, we will further consider the impacts of different constraints for pursuing sparse components in the aspect of classification.

6. Acknowledgements

This work was supported by NSFC (60675016, 60633030), 973 Program (2006CB303104) and NSF of GuangDong (06023194).

References

[1]. D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. Nature, 401: 788–791, 1999.

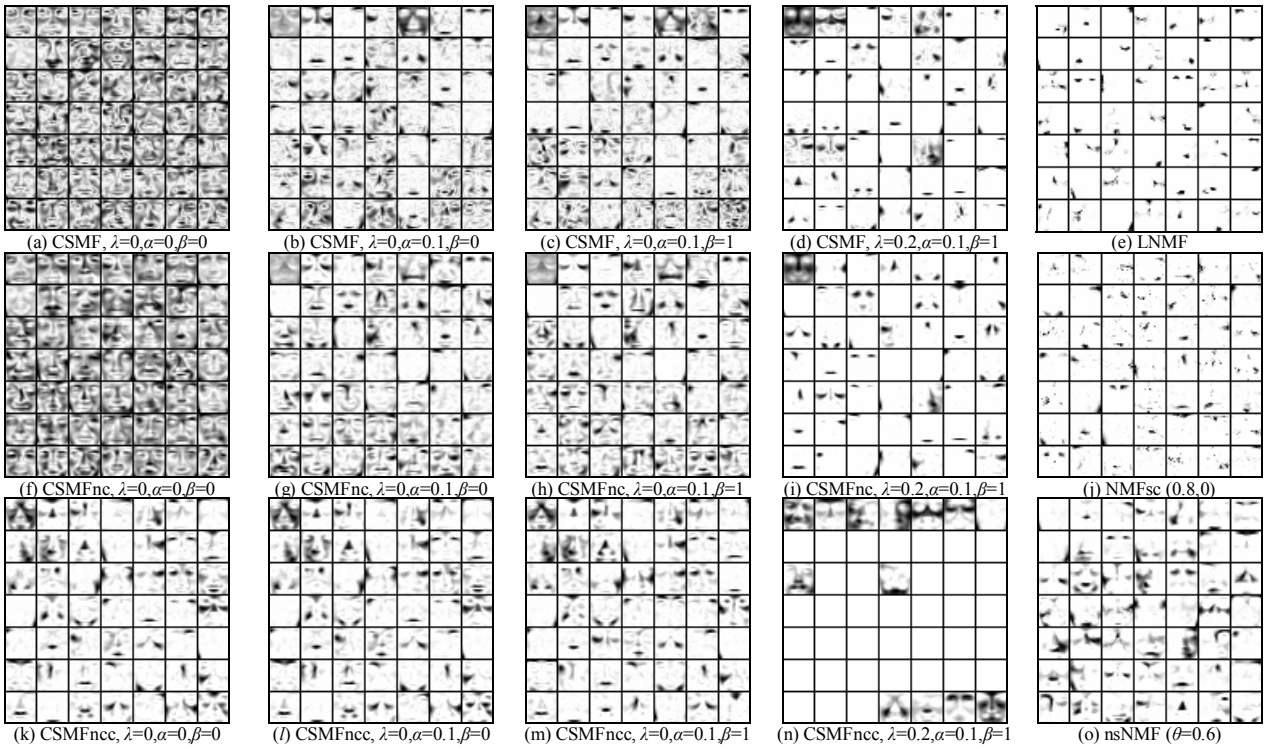


Figure 5. Experiment on CBCL. See text for detailed analysis. Setting: (1) For CSMFnc and CSMFncc, white pixels denote (almost) zero gray value and darker ones denote positive gray values; (2) The maximum iteration in all algorithms is 1000; (3) Figures of CSMF are shown as their absolute images, so the less the dark pixels are the sparser the component is. (The roles of the white and dark pixels here are different from those in Fig. 4 because of the traditional use for visualization); (4) Parameters of NMFsc are suggested in [8]; (5) The parameter setting of the gentle update strategy for CSMF, CSMFnc and CSMFncc is: $N^{w}_{0,1}$=100, $N^{w}_{0,2}$=50, $N^{h}_{0,1}$=300 and $N^{h}_{0,2}$=100.

[2]. R. Zass and A. Shashua. Nonnegative Sparse PCA. NIPS 2006.

[3]. D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. NIPS, 2000.

[4]. P. S. Penev and J. J. Atick. Local feature analysis: a general statistical theory for object representation. Network: Computation in Neural Systems, 7(3): 477-500, 1996.

[5]. MIT Center for Biological and Computation Learning, CBCL Face Database #1, http://www.ai.mit.edu/projects/cbcl.

[6]. M. D. Plumbley. Algorithms for non-negative independent component analysis. IEEE TNN, 14(3): 534-543, 2003.

[7]. H. Park and H. Kim. One-sided non-negative matrix factorization and non-negative centroid dimension reduction for text Classification. Proc. of the 4th Workshop on Text Mining, SDM 2006.

[8]. P.O. Hoyer. Non-negative matrix factorization with sparseness constraints. JMLR, 5: 1457-1469, 2004.

[9]. P.O. Hoyer. Nonnegative Sparse Coding. Proc. IEEE Workshop Neural Networks for Signal Processing, 2002.

[10]. Hyvärinen, J. Karhunen, and E. Oja. Independent Component Analysis. John Wiley and Sons, 2001.

[11]. S. Z. Li, X. Hou, H. Zhang, and Q. Cheng. Learning spatially localized parts-based representations. CVPR, 2001.

[12]. D. Donoho and V. Stodden. When does non-negative matrix factorization give a correct decomposition into parts? NIPS, 2003.

[13]. A. Pascual-Montano, J. M. Carazo, K. Kochi, D. Lehmann, and R. D. Pascual-Marqui. Nonsmooth non-negative matrix factorization (nsNMF). IEEE TPAMI, 28(3): 403-415, 2006.

[14]. H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," Technical Report, Statistics Department, Stanford University, 2004.

[15]. M. Kirby and L. Sirovich. Application of the karhunen-loeve procedure for the characterization of human faces. IEEE TPAMI, 12(1): 103-108, 1990.

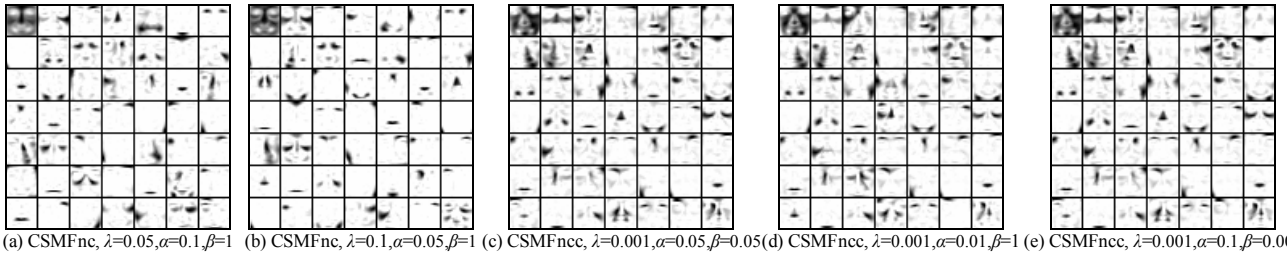


Figure 6. More results of CSMFnc and CSMFncc on CBCL (other parameter settings are the same as Fig. 5).