文章编号：1000-8608(2013)01-0151-06

一种实用快速非负矩阵分解算法

程明松*， 刘勺连

（大连理工大学 数学科学学院，辽宁 大连 116024）

摘要：提出了一种基于快速非负矩阵分解算法的实用新算法.该实用快速非负矩阵分解算法扩展了快速非负矩阵分解算法的约束条件，并且保持了较高的收敛速度，更具一般性和实用性.然后对该新算法进行了一些稀疏非负矩阵分解的扩展应用.数值实验显示该实用快速非负矩阵分解算法和快速非负矩阵分解算法具有相近的收敛速度，与其他经典非负矩阵分解算法相比其收敛速度有明显的提高，同时对添加稀疏性约束条件的实验也有很好的效果.

关键词：非负矩阵分解；快速非负矩阵分解算法；实用快速非负矩阵分解算法；稀疏非负矩阵分解

中图分类号：O241.6 **文献标志码**：A

0 引 言

非负矩阵分解（non-negative matrix factorization，NMF）是最近提出的一种降维方法.给定非负矩阵 $\boldsymbol{V}\in\boldsymbol{R}_{+}^{m\times n}$（$\boldsymbol{V}$表示所有元素为非负实数的 $m\times n$矩阵）和一个正数 $r\ll\min\{m,n\}$，NMF 的目的是求两个非负矩阵 $\boldsymbol{W}\in\boldsymbol{R}_{+}^{m\times r}$ 和 $\boldsymbol{H}\in\boldsymbol{R}_{+}^{r\times n}$ 使得 $\boldsymbol{V}\approx\boldsymbol{WH}$，其中 $\boldsymbol{W}$ 被称为基矩阵.

求解 $\boldsymbol{W}\in\boldsymbol{R}_{+}^{m\times r}$ 和 $\boldsymbol{H}\in\boldsymbol{R}_{+}^{r\times n}$ 常用的方法是解下列优化问题：

$$\min_{\boldsymbol{W},\boldsymbol{H}} f(\boldsymbol{W},\boldsymbol{H})=\|\boldsymbol{V}-\boldsymbol{WH}\|_{\mathrm{F}}^{2}$$

$$\text{s.t. } \boldsymbol{W}\geqslant\boldsymbol{0},\ \boldsymbol{H}\geqslant\boldsymbol{0}$$

其中 $\|\cdot\|_{\mathrm{F}}$ 表示矩阵的 Frobenius 范数；$\boldsymbol{W}\geqslant\boldsymbol{0}$，$\boldsymbol{H}\geqslant\boldsymbol{0}$，表示它们为非负矩阵.

非负矩阵分解的思想最早由 Paatero 等[1]于 1994 年提出.之后很多学者提出了各种非负矩阵分解方法[2-8]，并做了很多应用.1999 年，Lee 等[7]在 Nature 上首次提出了非负矩阵分解的概念，并将其成功用于人脸识别.2001 年，他们又提出了更为简化的计算矩阵分解的两种算法[2]：一种算法以最小化欧氏距离平方为目标函数，如上面的模型；另一种算法以最小化 K-L 散度为目标函数.这两种算法均采用了交替迭代下降算法.随后基于最小二乘问题的分解方法[4-5]、基于投影梯度法的分解算法[3]以及加稀疏约束的分解算法[6]相继被提出.

Li 等提出了一种快速非负矩阵分解算法[9]，与其他方法相比具有非常快的收敛速度.他们把上面模型中的约束条件做了一点改动得到如下模型：

$$\min_{\boldsymbol{W},\boldsymbol{H}} f(\boldsymbol{W},\boldsymbol{H})=\|\boldsymbol{V}-\boldsymbol{WH}\|_{\mathrm{F}}^{2}$$

$$\text{s.t. } W_{ij}\geqslant\varepsilon,\ H_{ij}\geqslant\varepsilon$$

其中 ε 是一个小正数，$W_{ij}\geqslant\varepsilon$，$H_{ij}\geqslant\varepsilon$ 表示 $\boldsymbol{W}$、$\boldsymbol{H}$ 的每个元素都不小于 ε.通过固定 $\boldsymbol{H}$ 和$\{\boldsymbol{W}_{\cdot1},\boldsymbol{W}_{\cdot2},\cdots,\boldsymbol{W}_{\cdot r}\}/\boldsymbol{W}_{\cdot i}$ 使目标函数变为一个二次函数，然后利用抛物函数的性质依次更新 $\boldsymbol{W}_{\cdot i}(i=1,2,\cdots,r)$，然后再固定 $\boldsymbol{W}$ 和$\{\boldsymbol{H}_{1\cdot},\boldsymbol{H}_{2\cdot},\cdots,\boldsymbol{H}_{r\cdot}\}/\boldsymbol{H}_{j\cdot}$ 依次更新 $\boldsymbol{H}_{j\cdot}(j=1,2,\cdots,r)$.

本文首先介绍这种快速非负矩阵分解算法，并说明其约束条件修改为 $W_{ij}\geqslant\varepsilon$，$H_{ij}\geqslant\varepsilon$ 后的不足之处.希望在条件为 $\boldsymbol{W}\geqslant\boldsymbol{0}$，$\boldsymbol{H}\geqslant\boldsymbol{0}$ 的情况下仍能获得快速非负矩阵分解算法.即通过分析快速非负矩阵分解算法的求解原理改进这种算法，得到一种新的算法，然后对新算法进行一些扩展应用.

1 实用快速非负矩阵分解算法

Li 等的算法中为了使固定 $\boldsymbol{H}$ 和$\{\boldsymbol{W}_{\cdot1},\boldsymbol{W}_{\cdot2},$

收稿日期：2011-12-16； **修回日期**：2012-11-27.
基金项目：中央高校基本科研业务费专项资金资助项目(DUT10LK04).
作者简介：程明松*(1976-)，男，讲师，硕士生导师，E-mail：mscheng@dlut.edu.cn.

$\cdots,\boldsymbol{W}_{\cdot r}\}/\boldsymbol{W}_{\cdot i}$ 后的目标函数为二次项系数大于零的抛物函数，把 $\boldsymbol{W}\geqslant\boldsymbol{0}$ 的条件改为 $W_{ij}\geqslant\varepsilon$，同理把 $\boldsymbol{H}\geqslant\boldsymbol{0}$ 改为 $H_{ij}\geqslant\varepsilon$. 这个小小的改动或许会影响分解结果的性质，尤其是稀疏性不能实现. 因此希望保持非负条件不变的情况下也能快速求解，从而得到一种既可保证分解结果具有某些性质（如稀疏性），又能实现分解速度快的算法.

在非负模型中固定 $\boldsymbol{H}$ 和 $\{\boldsymbol{W}_{\cdot 1},\boldsymbol{W}_{\cdot 2},\cdots,\boldsymbol{W}_{\cdot r}\}/\boldsymbol{W}_{\cdot i}$，则问题变为求解

$$\min_{\boldsymbol{W}_{\cdot i}\geqslant\boldsymbol{0}} f(\boldsymbol{W}_{\cdot i})=\left\|\boldsymbol{V}-\sum_{k=1,k\neq i}^{r}\boldsymbol{W}_{\cdot k}\boldsymbol{H}_{k\cdot}-\boldsymbol{W}_{\cdot i}\boldsymbol{H}_{i\cdot}\right\|_{\mathrm{F}}^{2} \tag{1}$$

如果固定 $\boldsymbol{W}$ 和 $\{\boldsymbol{H}_{1\cdot},\boldsymbol{H}_{2\cdot},\cdots,\boldsymbol{H}_{r\cdot}\}/\boldsymbol{H}_{j\cdot}$，则问题变为求解

$$\min_{\boldsymbol{H}_{j\cdot}\geqslant\boldsymbol{0}} f(\boldsymbol{H}_{j\cdot})=\left\|\boldsymbol{V}-\sum_{k=1,k\neq j}^{r}\boldsymbol{W}_{\cdot k}\boldsymbol{H}_{k\cdot}-\boldsymbol{W}_{\cdot j}\boldsymbol{H}_{j\cdot}\right\|_{\mathrm{F}}^{2} \tag{2}$$

式(1)中目标函数

$$\begin{aligned}f(\boldsymbol{W}_{\cdot i})&=\left\|\boldsymbol{V}-\sum_{k=1,k\neq i}^{r}\boldsymbol{W}_{\cdot k}\boldsymbol{H}_{k\cdot}-\boldsymbol{W}_{\cdot i}\boldsymbol{H}_{i\cdot}\right\|_{\mathrm{F}}^{2}=\\&\left\|\boldsymbol{V}-\sum_{k=1,k\neq i}^{r}\boldsymbol{W}_{\cdot k}\boldsymbol{H}_{k\cdot}-\right.\\&\left.(W_{1i}\quad\cdots\quad W_{ci}\quad\cdots\quad W_{mi})^{\mathrm{T}}\boldsymbol{H}_{i\cdot}\right\|_{\mathrm{F}}^{2}=\\&\left\|\boldsymbol{V}_{1\cdot}-\sum_{k=1,k\neq i}^{r}W_{1k}\boldsymbol{H}_{k\cdot}-W_{1i}\boldsymbol{H}_{i\cdot}\right\|_{2}^{2}+\cdots+\\&\left\|\boldsymbol{V}_{c\cdot}-\sum_{k=1,k\neq i}^{r}W_{ck}\boldsymbol{H}_{k\cdot}-W_{ci}\boldsymbol{H}_{i\cdot}\right\|_{2}^{2}+\cdots+\\&\left\|\boldsymbol{V}_{m\cdot}-\sum_{k=1,k\neq i}^{r}W_{mk}\boldsymbol{H}_{k\cdot}-W_{mi}\boldsymbol{H}_{i\cdot}\right\|_{2}^{2}\end{aligned}$$

从上面可以看出包含 W_{ci} 的部分与其他变量 $\{W_{1i},W_{2i},\cdots,W_{mi}\}/W_{ci}$ 是互不相关的，因此如果只看包含变量 W_{ci} 的目标函数，可以简化为

$$\begin{aligned}f(W_{ci})&=\left\|\boldsymbol{V}_{c\cdot}-\sum_{k=1,k\neq i}^{r}W_{ck}\boldsymbol{H}_{k\cdot}-W_{ci}\boldsymbol{H}_{i\cdot}\right\|_{2}^{2}=\\&\sum_{j=1}^{n}\Big(V_{cj}-\sum_{k=1,k\neq i}^{r}W_{ck}H_{kj}-W_{ci}H_{ij}\Big)^{2}=\\&W_{ci}^{2}\sum_{j=1}^{n}H_{ij}^{2}-2W_{ci}\sum_{j=1}^{n}\Big(V_{cj}-\\&\sum_{k=1,k\neq i}^{r}W_{ck}H_{kj}\Big)H_{ij}+\sum_{j=1}^{n}\Big(V_{cj}-\\&\sum_{k=1,k\neq i}^{r}W_{ck}H_{kj}\Big)^{2}=(\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}})_{ii}W_{ci}^{2}-\\&2W_{ci}\big[(\boldsymbol{V}\boldsymbol{H}^{\mathrm{T}})_{ci}-\sum_{k=1,k\neq i}^{r}W_{ck}(\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}})_{ki}\big]+\\&\sum_{j=1}^{n}\Big(V_{cj}-\sum_{k=1,k\neq i}^{r}W_{ck}H_{kj}\Big)^{2}=\\&(\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}})_{ii}W_{ci}^{2}-2\eta_{ci}W_{ci}+\gamma\end{aligned}$$

其中

$$\eta_{ci}=(\boldsymbol{V}\boldsymbol{H}^{\mathrm{T}})_{ci}-\sum_{k=1,k\neq i}^{r}W_{ck}(\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}})_{ki} \tag{3}$$

$\gamma=\sum_{j=1}^{n}\Big(V_{cj}-\sum_{k=1,k\neq i}^{r}W_{ck}H_{kj}\Big)^{2}$，都是与 W_{ci} 无关的常量.

从上式可以看出如果 $(\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}})_{ii}\neq 0$（即大于0），则目标函数是一个抛物方程，则函数 $f(W_{ci})$ 最小点在

$$W_{ci}=\eta_{ci}/(\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}})_{ii}$$

处取到，由于 $W_{ci}\geqslant 0$，则解应该为

$$W_{ci}=\max\{\eta_{ci}/(\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}})_{ii},0\}$$

如果 $(\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}})_{ii}=0$，即 $H_{i1}=H_{i2}=\cdots=H_{in}=0$，可知 $f(W_{ci})$ 中 W_{ci} 的一次项和二次项的系数都为0，因此 $f(W_{ci})$ 是关于 W_{ci} 的常数，此时求 $f(W_{ci})$ 最小，可令 W_{ci} 保持不变或任给一个值，此处选择随机生成一个值来更换它，之后给出解释. 即

$$W_{ci}=\begin{cases}\max\{\eta_{ci}/(\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}})_{ii},0\}; & \|\boldsymbol{H}_{i\cdot}\|>0\\ rand; & \|\boldsymbol{H}_{i\cdot}\|=0\end{cases}$$

其中 $rand$ 表示随机生成一个0到1之间的数.

由于包含 $W_{1i},\cdots,W_{ci},\cdots,W_{mi}$ 的部分与其他变量是互不相关的，即 $f(W_{c1}),f(W_{c2}),\cdots,f(W_{cr})$ 是互不相关的，那么在 $\boldsymbol{W}_{\cdot i}\geqslant\boldsymbol{0}$ 条件下求 $f(\boldsymbol{W}_{\cdot i})$ 最小，得解为

$$\boldsymbol{W}_{\cdot i}=\begin{cases}\max\{\boldsymbol{\eta}_{i}/(\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}})_{ii},\boldsymbol{0}\}; & \|\boldsymbol{H}_{i\cdot}\|>0\\ \boldsymbol{rand}(m,1); & \|\boldsymbol{H}_{i\cdot}\|=0\end{cases}$$

其中 $\boldsymbol{rand}(m,1)$ 表示随机生成一个元素值在0到1之间的 m 维列向量，

$$\begin{aligned}\boldsymbol{\eta}_{i}&=(\eta_{1i}\quad\eta_{2i}\quad\cdots\quad\eta_{mi})^{\mathrm{T}}=\\&(\boldsymbol{V}\boldsymbol{H}^{\mathrm{T}})_{\cdot i}-\sum_{k=1,k\neq i}^{r}\boldsymbol{W}_{\cdot k}(\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}})_{ki}\end{aligned} \tag{4}$$

下面来看为什么当 $\|\boldsymbol{H}_{i\cdot}\|=0$ 时 $\boldsymbol{W}_{\cdot i}$ 的更新要随机生成，已知问题的求解目标是找 $\boldsymbol{W}$ 和 $\boldsymbol{H}$ 使 $\boldsymbol{V}\approx\boldsymbol{W}\boldsymbol{H}$，而 $\boldsymbol{W}\boldsymbol{H}=\boldsymbol{W}_{\cdot 1}\boldsymbol{H}_{1\cdot}+\boldsymbol{W}_{\cdot 2}\boldsymbol{H}_{2\cdot}+\cdots+\boldsymbol{W}_{\cdot r}\boldsymbol{H}_{r\cdot}$，如果 $\|\boldsymbol{H}_{i\cdot}\|=0$，说明 $\boldsymbol{W}_{\cdot i}$ 没有起作用，而 $\boldsymbol{W}$ 的列向量可看成是一组基，从而说明基中第 i 个分量选取得不好，因此要重新选择 $\boldsymbol{W}_{\cdot i}$. 对于 $\boldsymbol{H}$，在模型中也可以把目标函数写为 $\min_{\boldsymbol{W},\boldsymbol{H}} f(\boldsymbol{W},\boldsymbol{H})=$

$\|\boldsymbol{V}^{\mathrm{T}}-\boldsymbol{H}^{\mathrm{T}}\boldsymbol{W}^{\mathrm{T}}\|_{\mathrm{F}}^{2}$，则可以利用上面所用的方法求解. 但是当$\|\boldsymbol{W}_{\cdot j}\|=0$时，由于 $\boldsymbol{W}$ 是基，$\|\boldsymbol{W}_{\cdot j}\|=0$是不合理的，在下次更新中，要重新选择 $\boldsymbol{W}_{\cdot j}$，因此令 $\boldsymbol{H}_{j\cdot}=\boldsymbol{0}$，由上面所介绍更新 $\boldsymbol{W}$ 的方法可知，在下次更新中 $\boldsymbol{W}_{\cdot j}$ 会被重新选择.

根据以上分析可得到如下结论.

定理 1　对任意的 $i\in\{1,2,\cdots,r\}$，固定 $\boldsymbol{H}$ 和$\{\boldsymbol{W}_{\cdot 1},\boldsymbol{W}_{\cdot 2},\cdots,\boldsymbol{W}_{\cdot r}\}/\boldsymbol{W}_{\cdot i}$，则问题

$$\min_{\boldsymbol{W}_{\cdot i}\geqslant\boldsymbol{0}} f(\boldsymbol{W}_{\cdot i})=\Big\|\boldsymbol{V}-\sum_{k=1,k\neq i}^{r}\boldsymbol{W}_{\cdot k}\boldsymbol{H}_{k\cdot}-\boldsymbol{W}_{\cdot i}\boldsymbol{H}_{i\cdot}\Big\|_{\mathrm{F}}^{2}$$

在下面点处取得最小值：

$$\boldsymbol{W}_{\cdot i}=\begin{cases}\max\{\boldsymbol{\eta}_i/(\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}})_{ii},\boldsymbol{0}\}; & \|\boldsymbol{H}_{i\cdot}\|>0\\ \boldsymbol{rand}(m,1); & \|\boldsymbol{H}_{i\cdot}\|=0\end{cases}$$

其中 $\boldsymbol{\eta}_i$ 为式(4) 所定义.

定理 2　对任意的 $j\in\{1,2,\cdots,r\}$，固定 $\boldsymbol{W}$ 和$\{\boldsymbol{H}_{1\cdot},\boldsymbol{H}_{2\cdot},\cdots,\boldsymbol{H}_{r\cdot}\}/\boldsymbol{H}_{j\cdot}$，则问题

$$\min_{\boldsymbol{H}_{j\cdot}\geqslant\boldsymbol{0}} f(\boldsymbol{H}_{j\cdot})=\Big\|\boldsymbol{V}-\sum_{k=1,k\neq j}^{r}\boldsymbol{W}_{\cdot k}\boldsymbol{H}_{k\cdot}-\boldsymbol{W}_{\cdot j}\boldsymbol{H}_{j\cdot}\Big\|_{\mathrm{F}}^{2}$$

在下面点处取得最小值：

$$\boldsymbol{H}_{j\cdot}=\begin{cases}\max\{\boldsymbol{\xi}_j/(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W})_{jj},\boldsymbol{0}\}; & \|\boldsymbol{W}_{\cdot j}\|>0\\ \boldsymbol{0}_{1\times n}; & \|\boldsymbol{W}_{\cdot j}\|=0\end{cases}$$

其中 $\boldsymbol{\xi}_j=(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{V})_{j\cdot}-\sum_{k=1,k\neq j}^{r}(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W})_{jk}\boldsymbol{H}_{k\cdot}$.

利用定理 1 和定理 2 的结论，非负约束模型可以通过固定 $\boldsymbol{H}$ 和$\{\boldsymbol{W}_{\cdot 1},\boldsymbol{W}_{\cdot 2},\cdots,\boldsymbol{W}_{\cdot r}\}/\boldsymbol{W}_{\cdot i}$，依次更新 $\boldsymbol{W}_{\cdot i}(i=1,2,\cdots,r)$，再固定 $\boldsymbol{W}$ 和$\{\boldsymbol{H}_{1\cdot},\boldsymbol{H}_{2\cdot},\cdots,\boldsymbol{H}_{r\cdot}\}/\boldsymbol{H}_{j\cdot}$，依次更新 $\boldsymbol{H}_{j\cdot}$ 来使目标函数不断减小. 基于以上做法得到算法 1：

算法 1(实用快速非负矩阵分解算法)

给定 V、r

随机初始化 $\boldsymbol{W}(0)\geqslant\boldsymbol{0}$，$\boldsymbol{H}(0)\geqslant\boldsymbol{0}$，令 $k=0$

重复下述步骤：

$\boldsymbol{D}(k)=\boldsymbol{H}(k)\boldsymbol{H}^{\mathrm{T}}(k)$，

$\boldsymbol{Q}(k)=\boldsymbol{V}\boldsymbol{H}^{\mathrm{T}}(k)$，$\boldsymbol{E}(k)=\boldsymbol{D}(k)$

对 $i=1:r$

$E_{ii}(k)=0$

如果 $D_{ii}(k)>0$

$$\hat{\boldsymbol{W}}=(\boldsymbol{W}_{\cdot 1}(k+1)\quad\cdots\quad\boldsymbol{W}_{\cdot i-1}(k+1)\quad\boldsymbol{W}_{\cdot i}(k)\quad\cdots\quad\boldsymbol{W}_{\cdot r}(k))$$

$$\boldsymbol{W}_{\cdot i}(k+1)=\max\left\{\frac{\boldsymbol{Q}_{\cdot i}(k)-\hat{\boldsymbol{W}}\boldsymbol{E}_{\cdot i}}{D_{ii}(k)},\boldsymbol{0}\right\}$$

否则

$\boldsymbol{W}_{\cdot i}(k+1)=\boldsymbol{rand}(m,1)$

$\boldsymbol{C}(k)=\boldsymbol{W}^{\mathrm{T}}(k+1)\boldsymbol{W}(k+1)$，

$\boldsymbol{R}(k)=\boldsymbol{W}^{\mathrm{T}}(k+1)\boldsymbol{V}$，$\boldsymbol{F}(k)=\boldsymbol{C}(k)$

对 $j=1,2,\cdots,r$

$F_{jj}(k)=0$

如果 $C_{jj}(k)>0$

$$\hat{\boldsymbol{H}}=(\boldsymbol{H}_{1\cdot}(k+1)\quad\cdots\quad\boldsymbol{H}_{j-1\cdot}(k+1)\quad\boldsymbol{H}_{j\cdot}(k)\quad\cdots\quad\boldsymbol{H}_{r\cdot}(k))$$

$$\boldsymbol{H}_{j\cdot}(k+1)=\max\left\{\frac{\boldsymbol{R}_{j\cdot}(k)-\boldsymbol{F}_{j\cdot}(k)\hat{\boldsymbol{H}}}{C_{jj}(k)},\boldsymbol{0}\right\}$$

否则

$\boldsymbol{H}_{j\cdot}(k+1)=\boldsymbol{0}_{1\times n}$

$k=k+1$

至此得到了一个新的算法，这个算法可以很容易地扩展到添加其他各种形式约束的应用，例如添加稀疏性约束的应用.

2　扩展应用

上章中的算法给出了求解一般非负矩阵分解的方法，下面再来研究添加其他约束的非负矩阵分解问题，以加稀疏性约束为例.

例 1

$$\min_{\boldsymbol{H},\boldsymbol{W}}\left\{\|\boldsymbol{V}-\boldsymbol{W}\boldsymbol{H}\|_{\mathrm{F}}^{2}+\alpha\|\boldsymbol{W}\|_{\mathrm{F}}^{2}+\beta\sum_{j}\|\boldsymbol{H}_{\cdot j}\|_{1}^{2}\right\}\tag{5}$$

s. t. $\boldsymbol{W}\geqslant\boldsymbol{0}$，$\boldsymbol{H}\geqslant\boldsymbol{0}$

求解式(5) 的一般方法是交替求解[10]

$$\min_{\boldsymbol{H}\geqslant\boldsymbol{0}}\left\|\begin{pmatrix}\boldsymbol{W}\\ \sqrt{\beta}\boldsymbol{e}_{1\times r}\end{pmatrix}\boldsymbol{H}-\begin{pmatrix}\boldsymbol{V}\\ \boldsymbol{0}_{1\times n}\end{pmatrix}\right\|_{\mathrm{F}}^{2}$$

和

$$\min_{\boldsymbol{W}\geqslant\boldsymbol{0}}\left\|\begin{pmatrix}\boldsymbol{H}^{\mathrm{T}}\\ \sqrt{\alpha}\boldsymbol{I}_k\end{pmatrix}\boldsymbol{W}^{\mathrm{T}}-\begin{pmatrix}\boldsymbol{V}^{\mathrm{T}}\\ \boldsymbol{0}_{k\times m}\end{pmatrix}\right\|_{\mathrm{F}}^{2}$$

其中 $\boldsymbol{I}_k$ 表示 k 阶单位阵.

这样就需要把原问题的求解矩阵规模扩大，因此会增加计算量. 如果用本文所用的求解方法，模型中 W_{ci} 的目标函数

$$f_1(W_{ci})=[(\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}})_{ii}+\alpha]W_{ci}^{2}-2\eta_{ci}W_{ci}+\gamma_1$$

其中 η_{ci} 为式(3) 所定义，γ_1 是关于 W_{ci} 的一个常数. 因此求它的最小值得解为

$$W_{ci}=\max\{\eta_{ci}/[(\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}})_{ii}+\alpha],0\}$$

关于 H_{jf} 的目标函数为

$$f_1(H_{jf})=[(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W})_{jj}+\beta]H_{jf}^{2}-2\xi'_{jf}H_{jf}+\gamma_2$$

其中 $\xi'_{jf}=(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{V})_{jf}-\sum_{k=1,k\neq j}^{r}[(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W})_{jk}+\beta]H_{kf}$，$\gamma_2$ 是关于 H_{jf} 的一个常数. 求最小值得解

$$H_{jf}=\max\{\xi'_{jf}/[(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W})_{jj}+\beta],0\}$$

因此，模型(5) 的迭代规则为

$$\boldsymbol{W}_{\cdot i}=\max\{\boldsymbol{\eta}_i/[(\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}})_{ii}+\alpha],\boldsymbol{0}\}$$

$$\boldsymbol{H}_{j\cdot}=\max\{\boldsymbol{\xi}'_j/[(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W})_{jj}+\beta],\boldsymbol{0}\}$$

其中 $\boldsymbol{\eta}_i$ 为式(4) 所定义，$\boldsymbol{\xi}'_j=(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{V})_{j\cdot}-\sum_{k=1,k\neq j}^{r}[(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W})_{jk}+\beta]\boldsymbol{H}_{k\cdot}$.

再来看另外一种加稀疏性约束的模型.

例 2

$$\min_{\boldsymbol{W},\boldsymbol{H}}\left\{\|\boldsymbol{V}-\boldsymbol{W}\boldsymbol{H}\|_{\mathrm{F}}^2+\beta\sum_{k,l}H_{kl}\right\}\quad(6)$$

$$\text{s. t. } \boldsymbol{W}\geqslant\boldsymbol{0},\ \boldsymbol{H}\geqslant\boldsymbol{0}$$

关于 H_{jf} 的目标函数

$$f_2(H_{jf})=(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W})_{jj}H_{jf}^2-2\xi''_{jf}H_{jf}+\gamma_3$$

其中 $\xi''_{jf}=(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{V})_{jf}-\sum_{k=1,k\neq j}^{r}(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W})_{jk}H_{kf}-0.5\beta$，$\gamma_3$ 是关于 H_{jf} 的正常数. 因此解得最小点为，当 $\|\boldsymbol{W}_{\cdot j}\|>0$ 时，$H_{jf}=\max\{\xi''_{jf}/(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W})_{jj},0\}$，否则，$H_{jf}=0$. 从而可以用如下方式交替更新 $\boldsymbol{H}$ 和 $\boldsymbol{W}$：

$$\boldsymbol{H}_{j\cdot}=\begin{cases}\max\{\boldsymbol{\xi}''_j/(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W})_{jj},0\}; & \|\boldsymbol{W}_{\cdot j}\|>0\\ \boldsymbol{0}_{1\times n}; & \|\boldsymbol{W}_{\cdot j}\|=0\end{cases}$$

其中 $\boldsymbol{\xi}''_j=(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{V})_{j\cdot}-\sum_{k=1,k\neq j}^{r}(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W})_{jk}\boldsymbol{H}_{k\cdot}-0.5\beta\boldsymbol{e}_{1\times n}$，而 $\boldsymbol{W}$ 用定理 1 所示形式更新.

3 数值实验

在这一章中用本文提出的实用快速非负矩阵分解算法(Pfast)、快速非负矩阵分解算法(Fast)、乘性迭代算法(Mult)、块主轴旋转算法(Bp)来做实验.

首先用随机生成的矩阵进行实验验证. 用 MATLAB 随机生成 200×300 和 2 000×1 500 的矩阵进行实验，前一矩阵 r 分别取 10、15、20，后一矩阵 r 分别取 30、40、50，运行时间都为 20 s，比较矩阵分解后的误差 $\delta=\|\boldsymbol{V}-\boldsymbol{W}\boldsymbol{H}\|_{\mathrm{F}}$，其中用 $\Delta=\|\boldsymbol{V}\|_{\mathrm{F}}$ 表示原始矩阵 $\boldsymbol{V}$ 的 F-范数. 在实验中各种算法采用相同的初始化数据，初始化数据也由 MATLAB 随机生成，每个实验都运行多次，最后选用多次实验数据的平均结果. 实验结果如表 1 所示.

表 1 随机矩阵在各算法下分解误差的对比

Tab. 1 Residual error comparison of different algorithms of random matrix

m	n	Δ	r	δ			
				Mult	Bp	Fast	Pfast
200	300	141.10	10	65.65	65.61	65.61	65.61
			15	63.49	63.46	63.44	63.42
			20	61.57	61.49	61.48	61.47
2 000	1 500	999.76	30	487.70	485.01	484.99	484.93
			40	485.00	480.80	480.74	480.67
			50	482.87	477.14	476.78	476.72

通过上表可看出 4 种算法的分解效果基本相同，后面 3 种算法比乘性迭代算法的分解效果略好一些，但是分解误差都比较大. 这是由于矩阵是随机生成的，矩阵的秩一般比较高，本身不能被很好地压缩，所以分解后误差仍然较大，压缩效果不是很好，为此，考虑对秩比较低的矩阵进行实验. 用 2 000×200 与 200×1 500 的随机矩阵的乘积和 3 000×500 与 500×8 000 的随机矩阵的乘积作为实验矩阵，前一矩阵 r 分别取 30、40、50，后一矩阵 r 分别取 50、75、100，运行时间同样选为 20 s，比较矩阵分解后的误差 $\delta=\|\boldsymbol{V}-\boldsymbol{W}\boldsymbol{H}\|_{\mathrm{F}}$，结果如表 2 所示.

表 2 低秩随机矩阵在各算法下的分解误差的对比

Tab. 2 Residual error comparison of different algorithms of low rank random matrix

m	n	Δ	r	δ			
				Mult	Bp	Fast	Pfast
2 000	1 500	1 331.90	30	35.21	29.69	27.88	27.80
			40	35.20	28.25	26.81	26.61
			50	34.82	27.85	25.65	25.46
3 000	8 000	3 220.70	50	122.23	49.99	46.60	46.30
			75	116.90	48.81	46.33	45.76
			100	108.14	47.83	45.78	44.91

从表 2 可以看出对低秩矩阵，各算法能很好地实现矩阵压缩，结果显示在相同的运行时间下新算法分解误差较小，与快速非负矩阵分解算法和块主轴旋转算法相比压缩效果略有提高，明显比乘性迭代算法压缩效果好.

然后用 ORL 人脸数据库和 CBCL 人脸数据库的数据来做实验. ORL 人脸库包含 400 幅人脸图像，每幅图像是 112 像素×92 像素，每个像素为 256 灰度级，即每幅图像是一个 112×92 的矩

阵，其中元素在 0 到 255 之间. CBCL 库包含 2 429 幅 19 像素×19 像素的人脸图像，每幅图像是一个 19×19 的矩阵. 把 ORL 库中每幅图像转化为一个 10 304 维的向量，则整个图像库可以表示为一个 10 304×400 的矩阵. 同样方法 CBCL 库可以表示为一个 361×2 429 的矩阵. 为了减小计算规模，把矩阵的每个元素都除以 255 使之介于 0～1.

首先用 ORL 人脸库作为实验数据，r 取为 50，在运行小于 35 s 的情况下，对各算法的误差进行对比，其中 $\|\boldsymbol{V}\|_F = 980.85$，结果如图 1 所示，由于起始误差较大，为了更好地比较各算法的误差变化选取了 5～35 s 的误差曲线图. 然后用 CBCL 人脸库作为实验数据，r 取为 49，在运行 20 s 的情况下，对各算法的误差进行对比，此时 $\|\boldsymbol{V}\|_F = 512.45$，结果如图 2 所示，同样为了对比明显图中选取 2～20 s 的曲线.

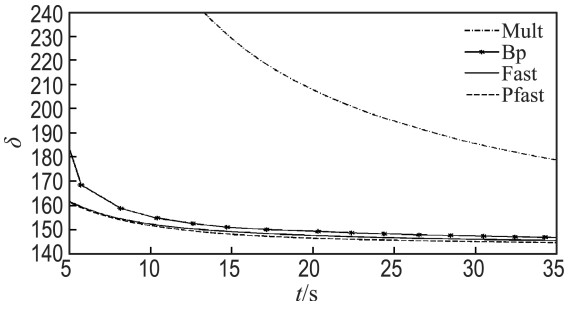

图 1　各算法对 ORL 人脸库分解误差的对比

Fig. 1　Residual error comparison of different algorithms of ORL face database

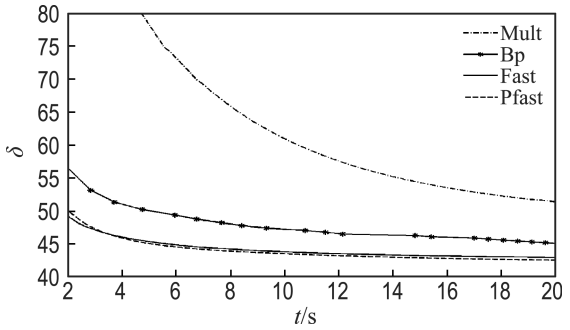

图 2　各算法对 CBCL 人脸库分解误差的对比

Fig. 2　Residual error comparison of different algorithms of CBCL face database

从图中可以看出新算法具有快速收敛效果，它和快速非负矩阵分解算法的收敛速度几乎一样快，甚至稍有提高，明显地比乘性迭代算法和块主轴旋转算法快.

对于稀疏性非负矩阵分解问题，分解的算法主要还是以保证分解结果与原矩阵近似为目的，因此选用的 α 和 β 都不宜过大. 用 ORL 人脸库作为实验数据，r 取为 49，运行时间取为 5 s 时，对于模型(5)新算法和块主轴旋转算法在 α 和 β 取各种值的情况下分解误差及稀疏性情况的对比结果如表 3 所示，然后对只有 β 约束的情况，对模型(5)应用块主轴旋转算法，对模型(6)应用新的算法进行了比较，结果如表 4 所示，其中 $\delta = \|\boldsymbol{V} - \boldsymbol{WH}\|_F$，$\Delta$ 表示 $\boldsymbol{H}$ 的零元个数占 $\boldsymbol{H}$ 总的元素个数的比例.

表 3　Bp 和 Pfast 算法在不同参数值下的分解误差和 $\boldsymbol{H}$ 的零元比例的对比表

Tab. 3　Residual error and percentages comparison of zero elements in $\boldsymbol{H}$ of Bp and Pfast algorithms with various parameter values

α	β	δ		Δ	
		Bp	Pfast	Bp	Pfast
0.01	0.05	149.90	147.45	0.370	0.414
0.01	0.20	150.45	147.02	0.361	0.418
0.05	0.05	152.85	149.33	0.547	0.454
0.05	0.20	152.86	150.00	0.547	0.474

表 4　Bp 和 Pfast 算法在不同 β 约束下的分解误差和 $\boldsymbol{H}$ 的零元比例的对比表

Tab. 4　Residual error and percentages comparison of zero elements in $\boldsymbol{H}$ of Bp and Pfast algorithms with various β values

β	δ		Δ	
	Bp	Pfast	Bp	Pfast
0.05	149.27	146.35	0.218	0.373
0.20	149.27	146.39	0.218	0.374
0.50	149.78	146.47	0.219	0.378

从表 3 和 4 中可看出新算法对非负矩阵在相同约束和运行时间下的分解误差比较小，在表 3 中当 α 取值很小时，稀疏效果好于块主轴旋转算法，但是在 α 取值较大时则不如块主轴旋转算法，如果 α 取值继续增大，新算法中 $\boldsymbol{H}$ 会出现大量零行，说明算法失效，因此对于模型(5) 新算法有一定的缺陷. 从表 4 中可以看出在只对 $\boldsymbol{H}$ 进行约束的情况下，新算法具有较好的稀疏效果.

4 结 语

本文根据快速非负矩阵分解算法提出了一种实用快速非负矩阵分解算法,新算法放宽了快速非负矩阵分解算法的条件,使之既保证了分解的快速收敛性,又使之更具一般性.然后将新算法扩展应用到稀疏非负矩阵分解问题中.实验表明新算法具有快速收敛的性质并且在添加稀疏约束的情况下有很好的分解结果.

参考文献:

[1] Paatero P, Tapper U. Positive matrix factorization: a non-negative factor model with optimal utilization of error estimates of data values [J]. **Environmetrics**, 1994, **5**:111-126.

[2] Lee D D, Seung H S. Algorithms for non-negative matrix factorization [J]. **Advances in Neural Information Processing Systems**, 2001, **13**:556-562.

[3] Lin C J. Projected gradient methods for nonnegative matrix factorization [J]. **Neural Computation**, 2007, **19**(10):2756-2779.

[4] Kim H, Park H. Non-negative matrix factorization based on alternating non-negativity constrained least squares and active set method [J]. **SIAM Journal on Matrix Analysis and Applications**, 2008, **30**(2):713-730.

[5] Kim H, Park H. Toward faster nonnegative matrix factorization: a new algorithm and comparisons [C] // **Proceedings-8th IEEE International Conference on Data Mining**. Piscataway: IEEE, 2008:353-362.

[6] Hoyer P O. Non-negative matrix factorization with sparseness constraints [J]. **Journal of Machine Learning Research**, 2004, **5**:1457-1469.

[7] Lee D D, Seung H S. Learning the parts of objects by non-negative matrix factorization [J]. **Nature**, 1999, **401**(6755):788-791.

[8] XU Wei, LIU Xin, GONG Yi-hong. Document clustering based on non-negative matrix factorization [C] // **Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval**. New York: Association for Computing Machinery, 2003:267-273.

[9] LI Le, ZHANG Yu-jin. FastNMF: highly efficient monotonic fixed-point nonnegative matrix factorization algorithm with good applicability [J]. **Journal of Electronic Imaging**, 2009, **18**(3):033004.

[10] Kim H, Park H. Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis [J]. **Bioinformatics**, 2007, **23**(12):1495-1502.

A practical fast NMF algorithm

CHENG Ming-song*, LIU Shao-lian

(School of Mathematical Sciences, Dalian University of Technology, Dalian 116024, China)

Abstract: Based on the fast non-negative matrix factorization (NMF) algorithm, a new practical fast NMF algorithm is developed. The new algorithm extends the constrained conditions of the fast NMF algorithm, and it keeps having the fast convergence speed, so it is more general and practical. Then, some extended applications for sparse NMF are proposed based on the new algorithm. Numerical experiments show that the practical fast NMF algorithm and the fast NMF algorithm have similar convergence speed, which is improved significantly compared with other classical NMF algorithms. Moreover, the practical fast NMF algorithm also has good experimental results for the cases with sparse constrained conditions being added.

Key words: non-negative matrix factorization (NMF); fast NMF algorithm; practical fast NMF algorithm; sparse NMF