

# 基于多核学习的投影非负矩阵分解算法

李 谦<sup>1</sup> 景丽萍<sup>2</sup> 于 剑<sup>2</sup>

(北京交通大学计算机与信息技术学院数字出版技术国家重点实验室(筹) 北京 100044)<sup>1</sup>

(北京交通大学计算机与信息技术学院 北京 100044)<sup>2</sup>

**摘要** 非负矩阵分解(NMF)把给定的数据矩阵分解成低维的非负基矩阵和对应的系数矩阵,两者之间存在必然联系。为此,研究者将基矩阵转换为系数矩阵的投影,进一步提高分解效率。但是该方法无法处理非线性数据,核函数的引入部分解决了此问题,却同时导致核函数参数选择的问题。基于多核学习理论,提出了一种多核学习的投影非负矩阵分解(MKPNMF)算法,该算法有效地避免了核函数参数选择的问题,同时提高了学习性能。在实际人脸数据上的实验结果表明,MKPNMF 较已有的 NMF 类方法具备明显的性能优势。

**关键词** 投影非负矩阵分解,核函数,多核学习

**中图分类号** TP391 **文献标识码** A

## Multi-kernel Projective Nonnegative Matrix Factorization Algorithm

LI Qian<sup>1</sup> JING Li-ping<sup>2</sup> YU Jian<sup>2</sup>

(School of Computer Science, Beijing Jiaotong University, State Key Laboratory of Digital Publishing Technology, Beijing 100044, China)<sup>1</sup>

(School of Computer Science, Beijing Jiaotong University, Beijing 100044, China)<sup>2</sup>

**Abstract** Nonnegative Matrix Factorization (NMF) decomposes the data into non-negative base matrix and coefficient matrix, and both of them have relationship with each other. Thus, some researchers rebuild the base matrix based on the projection of coefficient matrix. However, these two NMF-type methods can not satisfy the requirement of non-linear data analysis. With the development kernel learning, kernel function is introduced into the traditional NMF model for non-linear data analysis, which results in another problem, i. e., kernel parameter selection. We presented a Multi-Kernel Projective Nonnegative Matrix Factorization(MKPNMF) method, which has ability to avoid the problem of kernel parameter selection and improves the final learning performance. A series of experiments on real-world face data were conducted. The results show that MKPNMF outperforms the existing NMF-type methods.

**Keywords** Projective nonnegative matrix factorization, Kernel function, Multi-kernel learning

## 1 引言

1999 年, Lee 等人提出的非负矩阵分解(NMF)<sup>[1]</sup>算法能够从原始数据中找到一组基,并将数据映射到这组基上,且数据在每个基的映射系数均为非负值。这种新的变换方法与感知原理(整体由部分组成)一致。同时,非负性约束在一定程度上增强了分解的稀疏性。自提出以来, NMF 算法受到广泛关注和研究<sup>[2-4]</sup>, 基于非负矩阵分解的变形算法<sup>[5-8]</sup>、半监督非负矩阵分解(Semi-NMF)算法<sup>[9-11]</sup>不断被提出并得到了很好的证明。非负矩阵分解算法的目标函数是:

$$\min_{W, H \geq 0} \|X - WH\|_F^2 \quad (1)$$

式中,  $X$  为数据矩阵,  $W$  为分解后的基矩阵,  $H$  为系数矩阵。

多篇文献已经提到, 该目标函数相对两个变量并不是联合凸优化的, 因此在已有的算法中大多采用交替迭代的方法。首先固定  $W$ (或者  $H$ ), 然后计算  $H$ (或者  $W$ )。然而这种策略

将影响算法的最终性能。2005 年, Yuan 等人<sup>[12]</sup>提出了一种新的非负矩阵分解方法——投影非负矩阵分解(PNMF)算法。PNMF 计算一个投影矩阵, 然后把数据映射到低维空间, 相应的系数矩阵可以用投影矩阵和原始数据来合理表示。因此, PNMF 算法只有一个待求解的变量。作者证明相对于 NMF, PNMF 在聚类性能上有一定程度的提高, 运行时间上有明显的改善。但是对于低维空间线性不可分的数据模式, NMF 和 PNMF 方法性能都不理想。Zhang 等<sup>[13]</sup>提出了使用核函数的思路来解决此类问题。

核函数(Kernel Function)通过非线性映射把低维空间的数据映射到高维特征空间或者无限维的空间, 从而实现线性可分。常用的核函数<sup>[14]</sup>主要有: 高斯核函数(Gaussian kernel)、线性核函数(Polynomial kernel)、余弦核函数(Cosine kernel)。

Zhou 等<sup>[15]</sup>将核映射的思想应用于非负矩阵分解, 提出

到稿日期:2013-05-20 返修日期:2013-07-30 本文受中央高校基金科研业务费专项基金(2011JBM030, 2013JBZ005), 教育部博士点基金(20120009110006), 北方方正集团有限公司数字出版技术国家重点实验室开放课题资助。

李 谦(1986—), 男, 硕士生, 主要研究方向为非负矩阵分解; 景丽萍(1978—), 女, 副教授, 硕士生导师, 主要研究方向为文本挖掘、机器学习; 于 剑(1969—), 男, 教授, 博士生导师, 主要研究方向为机器学习、图像处理。

了基于核函数的非负矩阵分解(KNMF)算法。Yu 等人<sup>[16]</sup>又把核映射的思想应用于投影非负矩阵分解,提出了基于核函数的投影非负矩阵分解(KPNMF)算法。相对于 NMF 和 PNMF 算法,KNMF 和 KPNMF 能够提取到更多有价值的特征信息,并且能很好地处理非线性数据集。然而,这两种方法存在一个潜在的问题:核函数参数的选取。不同的核函数应用于 KNMF 或者 KPNMF 算法,得到的结果存在着较大的波动。多核学习(Multiple Kernel Learning, MKL)<sup>[17]</sup>能够有效地解决这个问题。该方法的思想是用多个核函数构造一个“合成”核。Yun 等<sup>[18]</sup>、Wang 和 Jabbar<sup>[17]</sup>将 MKL 的思想应用于传统的 NMF 模型。但是,上述方法沿袭了传统 NMF 的缺点。

本文基于多核学习理论,在投影非负矩阵分解算法的基础上,提出了一种基于多核学习的投影非负矩阵分解(MKPNMF)算法。本文第 2 节简要叙述相关工作;第 3 节提出本文的多核投影 NMF 方法并对其进行分析;第 4 节给出相应的实验数据、参数设定和实验结果及讨论;最后总结。

## 2 相关工作

投影非负矩阵分解(PNMF)算法是一种新颖的非负矩阵分解方法,PNMF 算法的基本思想是:给定数据矩阵  $X^{m \times n}$ ,试图找到一个  $m$  维的子空间  $\mathcal{R}$  以及一个  $m \times m$  的投影矩阵  $P$ ,然后把原始数据矩阵投影到低维空间,如此一来非负性就得以保持。文献<sup>[12]</sup>将  $P$  重写为  $WW^T$ 。为求达到分解后的低维矩阵同时满足非负,需要满足非负的约束。因此 PNMF 的目标函数为:

$$\min_{W \geq 0} \|X - WW^T X\|_F^2 \quad (2)$$

对比式(1)的 NMF 模型,式(2)中的  $W$  就是基矩阵,  $W^T X$  对应于系数矩阵。该方法只有一个变量  $W$ ,也正因为如此,该算法不存在非凸优化的问题。Yuan 等<sup>[12]</sup>证明了 PNMF 算法在时间性能上优于 NMF。

与 NMF 一样,PNMF 存在一个问题:当处理低维不可分数据模式时性能会急剧下降。核函数方法能够通过非线性映射把低维空间映射到高维特征空间或者无限维的空间,从而实现线性可分。核函数形式化定义如下:

$$K(x, y) = \langle \phi(x), \phi(y) \rangle = (\phi(x)^T \phi(y))$$

通过核映射的方式,可以把数据从输入空间非线性地映射到更高维,甚至无限维的特征空间,从而实现线性可分。Yu 等人在此基础上提出 KPNMF<sup>[16]</sup>算法,把输入数据  $X$  做一次核映射:  $\phi: x \in X \rightarrow \phi(x) \in F$ ,通过如下优化模型的求解获得结果:

$$\min_{W \geq 0} \|\phi(X) - W_\phi W_\phi^T \phi(X)\|$$

为了使用梯度和迭代的方法计算优化模型中的  $W_\phi$ ,上述模型可改写为:

$$\begin{aligned} & \|\phi(X) - W_\phi W_\phi^T \phi(X)\|^2 \\ &= \|(I - W_\phi W_\phi^T) \phi(X)\|^2 \\ &= \text{Tr}((I - W_\phi W_\phi^T)(I - W_\phi W_\phi^T)^T K) \end{aligned} \quad (3)$$

式中,  $K = \phi(X)^T \phi(X)$  是核函数。

相对于 PNMF, KPNMF 能够很好地解决如下问题:1)能够提取到更有用的隐藏特征信息;2)能够很好地处理非线性的数据集;3)能够处理有非负值的数据集;4)能够处理没有确定值而只有相互关系(相似,不相似,或者有多相似)的数据

集。然而,与 KNMF 方法一样, KPNMF 需要选择一个合理的核函数来对输入数据做核映射。这个合理的核函数如何选择?是否某一个核函数能够对所有的数据都有很好的性能?Yun 等人<sup>[18]</sup>把 MKL 的思想引入非负矩阵分解算法提出了 MKNMF。作者证明该方法能够有效地提高 NMF 算法处理数据的性能。

## 3 基于多核学习的投影非负矩阵分解(MKPNMF)算法

KPNMF 算法存在核函数的选取问题;MKNMF 算法是基于 NMF 算法思想的一种改进算法,也就同样存在着 NMF 算法的问题:非联合凸优化、时间性能得不到保证。因此,为了在保证时间性能的前提下,避免类似于 KPNMF 算法的核函数选取问题,我们提出了一种基于多核学习的投影非负矩阵分解方法(MKPNMF)。根据 MKL 理论,我们的方法构建了一个合成的核函数:

$$\beta_1 K_1 + \beta_2 K_2 + \dots + \beta_m K_m = \sum_{i=1}^m \beta_i K_i = K \quad (4)$$

式中,  $\beta_i$  是对应核的权重值,并且需要满足  $\beta_i \geq 0$ ;  $\sum \beta_i = 1$ 。权重值非负,能够保证每一个核函数对核映射都有所贡献;权重值之和为 1,相当于进行了一次归一化处理。  $K_i$  是选取的核函数。

在实验中,高斯核函数相对于其他核函数而言,能够取得最佳的性能。所以,在我们的算法中,也选择了高斯核函数:

$$K_j^g = \exp\left\{-\frac{\|x_p - x_q\|^2}{\alpha_j}\right\}$$

式中,  $x_p, x_q$  分别表示两个数据点,  $\alpha_j$  为对应高斯核函数的参变量。为了就高斯核构造不同的核函数,我们令  $\alpha_i = \gamma \sigma^2$ ,  $\sigma$  是标准方差,  $\sigma = \sqrt{\text{var}(\sum_{j=1}^n \|x_j - \bar{x}\|^2 / n)}$ ,  $\bar{x}$  为均值,  $\gamma$  选取不同的非负值,从而构造出不同的核函数,具体参考实验部分。文献<sup>[14, 17, 18]</sup>也证明了高斯核的优越性。

基于式(3),为了计算核函数系数向量  $\beta$ ,我们作如下推导:

$$\begin{aligned} & \min_{W_\phi \geq 0, 0 \leq \beta_i \leq 1} \text{Tr}\{(I - W_\phi W_\phi^T)(I - W_\phi W_\phi^T)^T K\} \\ &= \min_{W_\phi \geq 0, 0 \leq \beta_i \leq 1} \text{Tr}\{(I - W_\phi W_\phi^T)(I - W_\phi W_\phi^T)^T \sum_{j=1}^m \beta_j K_j\} \end{aligned} \quad (5)$$

当计算  $\beta$  时,  $(I - W_\phi W_\phi^T)(I - W_\phi W_\phi^T)^T$  被视为已知部分,我们用  $G$ (given 的首字母)来表示。式(5)可以表示为:

$$\min_{W_\phi \geq 0, 0 \leq \beta_j \leq 1} \text{Tr}\{G \sum_{j=1}^m \beta_j K_j\} = \min_{W_\phi \geq 0, 0 \leq \beta_j \leq 1} \sum_{j=1}^m \beta_j \text{Tr}\{GK_j\} \quad (6)$$

同理,我们把式(6)中的  $\text{Tr}\{GK_j\}$  表示成  $t_j$ ,故式(5)最终变形为:

$$\min_{W_\phi \geq 0, 0 \leq \beta_i \leq 1} t^T \beta \quad (7)$$

式中:

$$t_i = \text{Tr}((I - W_\phi W_\phi^T)(I - W_\phi W_\phi^T)^T K_i) \quad (8)$$

( $K_i$  为第  $i$  个核函数)

在固定  $W_\phi$  的情况下,根据  $\beta$  的性质,对式(7)引入式(4)的两个约束条件:

$$t^T \beta \quad \text{s. t. } \beta \geq 0; 1^T \beta = 1 \quad (9)$$

式(9)是一个有等式约束、不等式约束的标准线性规划表达式,可以用 matlab 中的 `linprog()` 函数计算得到上式中的  $\beta$  (请参阅 `linprog()`)。

得到一个合成的核函数之后,接下来就是在该合成核函数的基础上计算  $W_\phi$ :

$$\begin{aligned} F &= Tr((I - W_\phi W_\phi^T)(I - W_\phi W_\phi^T)^T K) \\ &= Tr(K - 2W_\phi W_\phi^T K + W_\phi W_\phi^T W_\phi W_\phi^T K) \\ &= Tr(K) - 2Tr(W_\phi W_\phi^T K) + Tr(W_\phi W_\phi^T W_\phi W_\phi^T K) \end{aligned} \quad (10)$$

令  $\lambda_{ij}$  是约束  $W_{\phi_{ij}} \geq 0$  的拉格朗日乘子,  $\lambda = [\lambda_{ij}]$ , 对式(10)做如下处理:

$$T = F + Tr(\lambda W_\phi^T) \quad (11)$$

为了利用梯度和迭代的方法计算  $W_\phi$ , 式(11)两端对  $W_\phi$  求导有:

$$\frac{\partial T}{\partial W_\phi} = 2KW_\phi - 2(W_\phi W_\phi^T KW_\phi + KW_\phi W_\phi^T W_\phi) + \lambda = 0 \quad (12)$$

式(12)两端同时乘以  $W_{\phi_{ij}}$ , 根据 Kuhn-Tucker 条件  $\lambda_{ij} W_{\phi_{ij}} = 0$  有:

$$(KW_\phi)_{ij} W_{\phi_{ij}} - (W_\phi W_\phi^T KW_\phi + KW_\phi W_\phi^T W_\phi)_{ij} W_{\phi_{ij}} = 0$$

根据上式导出计算  $W_\phi$  的迭代公式:

$$W_{\phi_{ij}} \leftarrow W_{\phi_{ij}} \frac{KW_\phi}{W_\phi W_\phi^T KW_\phi + KW_\phi W_\phi^T W_\phi}$$

综上, 我们的算法如下:

#### 算法 1 MKPNMF

输入:  $K_j (j=1, 2, \dots, m), r$ ;

输出: 核函数系数向量  $\beta$  和  $W_\phi$ 。

1. 初始化  $W_\phi$ ;

2. 重复如下步骤:

给定  $W_\phi$ , 应用线性规划的方法计算  $\beta$ ;

给定  $\beta$ , 构造“合成”核函数  $k = \sum_{j=1}^M \beta_j k_j$ , 更新  $W_\phi$ :

$$W_{\phi_{ij}} \leftarrow W_{\phi_{ij}} \frac{KW_\phi}{W_\phi W_\phi^T KW_\phi + KW_\phi W_\phi^T W_\phi}$$

直到收敛或者达到最大迭代次数退出循环。

收敛条件为:  $\|\phi(X) - W_\phi W_\phi^T \phi(X)\| \leq \xi$ ,

其中,  $\xi$  为一预设阈值。

在每一次迭代过程中, 我们的算法只需要更新一个变量, 所以, 相对于 MKNMF 算法, 我们的方法能够有效地减少子迭代的时间。虽然两个算法的时间复杂度的量级是一样的, 但是我们的方法时间复杂度的常数因子会更小, 从而能够获得更好的时间性能。

## 4 实验结果

### 4.1 实验数据

在实验中, 我们选择了 ORL、YaleDB、COIL20 这 3 个人脸数据集。其中, ORL 有 400 张人脸图片, YaleDB 数据集有 165 张人脸图片, COIL20 数据集有 1440 张人脸图片。上述 3 个数据集的人脸图片的维数均为  $32 \times 32$ 。

### 4.2 参数设定、比较算法及度量标准

我们的实验选取的是高斯核函数, 其形式为:

$$K_j^{pq} = \exp\left\{-\frac{\|x_p - x_q\|^2}{\alpha_j}\right\}$$

为了就高斯核构造不同的核函数, 我们令  $\alpha_i = \gamma\sigma^2$ ,  $\sigma$  是标准方差,  $\sigma = \sqrt{\frac{1}{n} \sum_{j=1}^n \|x_j - \bar{x}\|^2}$ ,  $\bar{x}$  为均值,  $\gamma$  选取不同的非负值, 从而构造出不同的核函数:

$$\gamma \in \left[\frac{1}{32}, \frac{1}{16}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}, 1, 2, 4, 8, 16, 32\right]$$

因为我们的方法是一个无监督的基于投影的非负矩阵分解算法, 所以选择做对比的算法为 PNMf, KPNMF。为了对比时间性能, 还选择了同样使用 MKL 理论的 MKNMF 算法作对比。同时, 上述算法在乘法迭代更新时的预设阈值和迭代次数都设定为 0.001 和 5000。最后统计的结果选取的是 10 次实验结果的平均值。

对于结果的统计, 我们选择了 accuracy 和 NMI 来度量聚类结果。因为 accuracy 比较容易理解, 所以我们仅详细介绍 NMI(normalized mutual information)。在聚类应用中, mutual information(MI)常用于度量两个簇集(cluster set)的相似度。给定两个簇  $c_1$  和  $c_2$ ,  $MI(c_1, c_2)$  定义为:

$$MI(c_1, c_2) = \sum_{c_i \in c_1, c_j \in c_2} p(c_i, c_j) \cdot \log \frac{p(c_i, c_j)}{p(c_i) \cdot p(c_j)}$$

其中,  $p(c_i)$  表示的是一幅图像选择自簇  $c_i$  的概率,  $p(c_i, c_j)$  定义了一幅图像选择自  $c_i$ , 同时选择自  $c_j$  的联合概率。

$$MI(c_1, c_2) \in [0, \max(H(c_1), H(c_2))]$$

其中,  $H(c_1)$  是  $c_1$  的熵。

当两个簇完全相同时,  $MI(c_1, c_2)$  取得最大值, 当两个簇是完全独立的时候, 得到最小值。对  $MI(c_1, c_2)$  做一次归一化处理, 以求将其取值范围统一到 0 到 1 之间:

$$NMI(c_1, c_2) = \frac{MI(c_1, c_2)}{\max(H(c_1), H(c_2))}$$

### 4.3 实验结果展示

在本节中, 将分别从对比算法的 accuracy 和 NMI 结果、KPNMF 算法在不同的核函数上的 accuracy 和 NMI、MKNMF 和我们的方法的时间性能 3 方面来验证 MKPNMF 算法的优势。

#### 4.3.1 对比算法的 accuracy、NMI 结果

表 1—表 3 是 PNMf、KPNMF、MKNMF 和我们的算法在 3 组数据集下的精度和 NMI 结果。

表 1 ORL 数据的聚类精度结果对比

(ORL)	PNMF	KPNMF	MKNMF	MKPNMF
accuracy	0.6200	0.2000~0.7475	0.7267	0.7490
NMI	0.8281	0.3945~0.8762	0.8661	0.8781

表 2 YaleDB 数据的聚类精度结果对比

(ORL)	PNMF	KPNMF	MKNMF	MKPNMF
accuracy	0.4000	0.1818~0.4788	0.4923	0.4801
NMI	0.4635	0.2381~0.5215	0.5319	0.5255

表 3 COIL20 数据的聚类精度结果对比

(ORL)	PNMF	KPNMF	MKNMF	MKPNMF
accuracy	0.6201	0.1368~0.6868	0.6612	0.6912
NMI	0.7444	0.4216~0.7833	0.7717	0.7867

注: 对于 KPNMF 的结果, 我们分别选取了每一个核函数, 得到了一个区间值。具体的可以参考下文统计结果。

从实验结果来看, 我们的方法一定程度上优于 PNMf 和 KPNMF 算法。相比于 MKNMF 算法, 我们的方法由于只需要更新一个变量, 不存在非联合凸优化的问题, 算法误差能够得到有效降低, 因此大部分情况下也能得到更佳的聚类性能。

#### 4.3.2 KPNMF 算法在不同核函数下的结果展示

图 1、图 2 展示的是在不同核函数情况下在给定数据集上的聚类结果, 从中可以看到, 不同的核函数, KPNMF 的结果有较大的差异; 并且实验结果先升后降, 大部分情况下, 最佳参数值出现在 1/2 到 1 之间。总体来讲, 我们能够得到如

下结论:不同的核函数,对单核 NMF 算法的聚类结果影响较大。

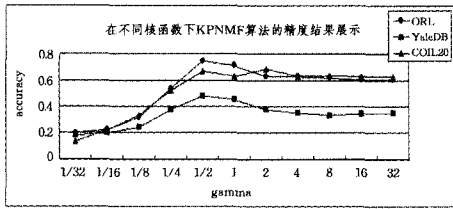


图1 不同数据集在不同核函数下 KPNMF 算法的精度对比

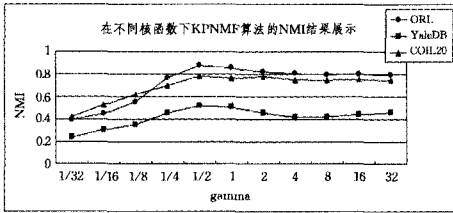


图2 不同数据集在不同核函数下 KPNMF 算法的 NMI 对比

#### 4.3.3 MKNMF 算法和 MKPNMF 算法的时间性能展示

为了对比我们的方法和 MKNMF 算法的时间性能,对 3 组数据做了时间性能的统计,运行环境如表 4 所列。

表4 算法的运行环境

Memory	31.4GB
Processor	Inter Xeon(R) CPU E5620 @ 2.40GHz ×16
OS Type	64-bit
Disk	2.1TB

与 MKNMF 算法相比,因为两种算法都需要必要的时间来计算核函数的系数向量  $\beta$ ,所以两者的时间差异主要体现在迭代算法的更新过程。从图 3 的实验结果可以看到,因为我们的方法只需要更新一个待求解向量,在每次迭代过程中,运行速度相对更快,我们的方法在运行时间性能上较 MKNMF 算法更优。

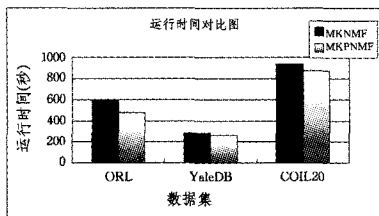


图3 MKNMF 算法和我们的算法的运行时间对比

**结束语** 非负矩阵分解(NMF)算法因为其合理的可解释性和稀疏性得到广泛研究。数十年时间研究者们不断地提出新的 NMF 算法旨在进一步提高分解效率。本文提出的多核学习的投影非负矩阵分解(MKPNMF)算法有效地避免了核函数参数选择的问题,同时提高了学习性能。在实际人脸数据上的实验结果表明,MKPNMF 算法的结果与最佳情况下的 KPNMF 算法性能相当,表明我们的方法在避免了核函数的选取的情况下,可以得到更好的性能。较已有的无监督非负矩阵分解方法,我们的方法具备明显的性能优势。

#### 参考文献

[1] Lee D D, Seung H S. Learning the parts of objects by nonnegative matrix factorization [J]. Letters to Nature, 1999, 401: 788-791

[2] Lee D D, Seung H S. Algorithm for nonnegative matrix factorization [C]// Neural Information Processing Systems Foundation. 2001:556-562

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

[4] Wang Yu-Xiong, Zhang Yu-Jin. Non-negative Matrix Factorization; a Comprehensive Review [J]. Knowledge and Data Engineering, IEEE Transaction, 2013, 25(6): 1336-1353

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

[6] Guillamet D, Bressan M, Vitria J. A weighted non-negative matrix factorization for local representation [C]// Computer Vision and Pattern Recognition, Proceedings of the 2001 IEEE Computer Society Conference. 2001, 1: 942-947

[7] Guillamet D, Vitria J, Schiele B. Introducing a weighted non-negative matrix factorization for image classification [J]. Pattern Recognition Letter, 2003, 24: 2447-2454

[8] Cichocki A, Amari S, Zdunek R. Extended smart algorithms for non-negative matrix factorization [C]// Proceedings of the 8th international conference on Artificial Intelligence and Soft Computing. 2006: 548-562

[9] Chen Yan-hua, Rege M, Ming Dong, et al. Nonnegative matrix factorization for semisupervised data clustering [J]. Knowledge and Information Systems, 2008, 17(3): 355-379

[10] Chen Yan-hua, Wang Li-jun, Dong Ming. Semisupervised document clustering with simultaneous text representation and categorization [J]. Machine Learning and Knowledge Discovery in Databases Lecture Notes in Computer Science, 2009, 5781: 211-226

[11] Chen Yan-hua, Wang Li-jun, Dong Ming. Non-negative matrix factorization for semi-supervised heterogeneous data co-clustering [J]. IEEE Trans. on Knowledge and Data Engineering, 2010, 22: 1459-1474

[12] Yuan Zhi-jian, Oja E. Projective nonnegative matrix factorization for image compression and feature extraction [C]// Image Analysis Lecture Notes in Computer Science. 2005, 3540: 333-342

[13] Zhang Dao-qiang, Chen Song-can. Clustering Incomplete Data using Kernel-based Fuzzy C-means Algorithm [J]. Neural Processing Letters, 2003, 18(3): 155-162

[14] Wang J, Jabbar M A. Multiple Kernel Learning for adaptive graph regularized nonnegative matrix factorization [C]// International Association of Science and Technology for Development. 2012: 115-122

[15] Zhang Dao-qiang, Zhou Zhi-hua, et al. Non-negative Matrix Factorization on Kernels [C]// Pacific Rim International Conferences on Artificial Intelligence. 2006: 401-412

[16] Yu Jun, Meng Jin-tao, Lu Xiao-xu. A kernel based non-negative matrix factorization [C] // Geoscience and Remote Sensing (IITA-GRS). 2010, 1: 376-379

[17] Gonen M, Alpaydin E. Multiple Kernel Learning Algorithms [J]. Journal Of Machine Learning Research, 2011, 12: 2211-2268

[18] An S, Yun J-M, Choi S. Multiple Kernel Nonnegative Matrix Factorization [C] // Acoustics, Speech and Signal Processing, IEEE International Conference. 2011: 1976-1979