

图像表示的约束非负矩阵分解

Constrained Nonnegative Matrix Factorization for Image Representation
Haifeng Liu, Member, IEEE, Zhaohui Wu, Senior Member, IEEE, Xuelong Li, Senior Member, IEEE, Deng Cai, Member, IEEE, and Thomas S. Huang, Life Fellow, IEEE

摘要

非负矩阵分解(NMF)是一种流行的技术,用于寻找基于零件的非负数据的线性表示。它已成功地应用于模式识别、信息检索和计算机视觉等广泛的应用领域。然而,NMF本质上是一种非监督方法,不能利用标签信息。在本文中,我们提出了一种新的半监督矩阵分解方法,称为约束非负矩阵分解(CNMF),该方法将标签信息作为附加约束。具体来说,我们将展示如何显式地组合标签信息来提高所得到的矩阵分解的鉴别能力。对提出的CNMF方法进行了研究,并给出了优化问题的更新方案。通过一组基于真实世界应用的评估,经验实验证明了我们的新算法的有效性,与最先进的方法相比。

关键词: 非负矩阵分解, 半监督学习, 降维, 聚类。

1 介绍

在许多数据分析任务中,一个基本问题是找到合适的表示[1]、[2]、[3]、[4]、[5]、[6]、[7]、[8]。一个有用的表示通常使数据中的潜在结构显式,从而可以应用进一步的处理。矩阵分解技术作为此类数据表示的基本工具,已经受到越来越多的关注。通过使用不同的标准,已经发展出若干不同的方法来做这一点。最流行的技术包括主成分分析(PCA)[9]、奇异值分解(SVD)[10]和矢量量化[11]。矩阵分解的核心是找到两个或两个以上的矩阵因子,它们的乘积是原始矩阵的一个很好的近似。在实际应用中,分解后的矩阵因子的维数通常比原矩阵的维数小得多。这使得数据点的表示更加紧凑,这可以促进其他学习任务,如聚类和分类。

在矩阵分解方法中,非负矩阵分解(Nonnegative matrix factorization, NMF)[2], [3]的特点是它强制了因子矩阵必须是非负的约束,即所有元素必须等于或大于零。这种非负性约束将NMF引入到基于部件的对象表示中,因为它只允许对原始数据进行加法(而不是减法)组合。因此,对于图像处理、人脸识别[2]、[12]和文档聚类[13]、[14]来说,它是一种理想的降维算法。在这些降维算法中,很自然地会将物体看作是部分的组合而形成一个整体。

NMF是一种无监督学习算法。也就是说,NMF不适用于许多现实世界的问题,在这些问题上,来自领域专家的知识有限。然而,许多机器学习研究人员发现,当未标记数据与少量标记数据结合使用时,可以在学习准确度[15], [16], [17]上产生相当大的提高。与标记过程相关的成本可能使一个完全标记的训练集不可行,而获取一个小的标记数据集是相

对便宜的。在这种情况下，半监督学习可以有很大的实用价值。因此，将 NMF 的使用扩展到半监督的方式将会有很大的好处。

最近，Cai[1]提出了一种图形正则化 NMF(GNMF)方法来编码数据空间的几何信息。GNMF 构造一个最近邻图来建模局部流形结构。当标签信息可用时，可以很自然地将其合并到图结构中。具体来说，如果两个数据点共享相同的标签，则可以为连接它们的边分配一个较大的权重。如果两个数据点有不同的标签，则相应的权值设为 0。这就产生了半监督 GNMF。这种方法的主要缺点是，在理论上无法保证来自同一类的数据点会在新的表示空间中映射到一起，并且仍然不清楚如何以一种有原则的方式选择权重。

本文提出了一种新的矩阵分解方法——约束非负矩阵分解 (Constrained non-negative matrix Factorization, CNMF)，该方法将标签信息作为附加的硬约束。我们的方法的中心思想是，来自同一个类的数据点应该合并到新的表示空间中。这样，得到的基于零件的表示与原始数据具有一致的标签，从而具有更强的鉴别能力。我们的方法的另一个优点是它是无参数的，这避免了调优参数以获得最佳结果的成本。它使我们的算法适用于许多现实世界的应用容易和有效。讨论了如何有效地解决相应的优化问题。并给出了优化方案的收敛性证明。本文的贡献有：1、标准的 NMF 是一种不能包含标签信息的无监督学习算法。在本文中，我们将其推广到半监督学习算法中。此外，我们的方法将标签信息作为硬约束；因此，共享相同标签的数据点在新的表示空间中具有相同的坐标。通过这种方式，学习到的表征可以有更强的辨别能力。

2、早期的研究[18]表明，NMF 和 PLSA 都是多项 PCA 的实例。特别是，PLSA 解决了具有 KL 散度[19]，[20]的 NMF 问题。为了进一步探讨这方面的问题，我们将 CNMF 应用于 KL 发散公式，并提供了求解优化问题的更新规则。

3、与半监督 GNMF 不同，我们的方法的一个优点是它是无参数的。因此，为了获得最佳结果，不需要对参数进行调优。因此，CNMF 可以轻松高效地应用于许多实际应用。实验结果表明，该算法能显著提高聚类性能。

4、据我们所知，目前还没有直接得到 NMF 问题解的方法；最先进的算法是使用更新规则迭代地得到目标函数的最优值。因此，算法的效率对于实际应用非常重要。在本文中，我们定性地分析了算法的计算复杂性，并对算法的收敛速度进行了实验测试，以定量地证明算法的有效性。

本文的结构如下：在第二部分，我们简要回顾了 NMF 的背景和相关工作。第 3 节介绍了约束 NMF 的思想。第 4 节和第 5 节给出了两种公式中算法的详细算法和算法收敛性的理论证明。第 6 节讨论算法的计算复杂度。最后，第 7 节给出了实验结果，第 8 节总结了本文。

2 相关工作

矩阵的因式分解通常是非唯一的，通过合并不同的约束，已经发展出许多不同的方法来实现这一点。PCA[9]和 SVD[10]将矩阵分解为主成分的线性组合。Hoyer[21]和 Dueck 等人[22]计算稀疏矩阵分解。

NMF 与这些方法的不同之处在于，它强制约束因子矩阵的元素必须是非负的。假设有 n 个数据点 $\{\mathbf{x}_i\}_{i=1}^n$ 。每个数据点 $\mathbf{x}_i \in \mathbb{R}^m$ 是 m 维的，用一个

向量表示。向量放在列中，整个数据集由矩阵 $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n] \in \mathbb{R}^{m \times n}$ 表示。

NMF 的目标是找出两个非负矩阵因子 \mathbf{U} 和 \mathbf{V} ，其中这两个因子的乘积是原始矩阵的近似，表示为 $\mathbf{X} \approx \mathbf{UV}^T$ 。

该近似由一个成本函数来量化，该函数可以由一些距离测度来构造。一个简单的度量是两个矩阵[23]之间的欧几里德距离(也称为弗罗本尼乌斯范数)的平方。然后，NMF 的目标可以重述如下：将 \mathbf{X} 分解为 $m \times k$ 矩阵 \mathbf{U} 和 $k \times n$ 矩阵 \mathbf{V}^T ，从而最小化以下目标函数： $\mathcal{O}_F = \|\mathbf{X} - \mathbf{UV}^T\|^2$ 。[24]

中描述的另一个度量称为 \mathbf{X} 与 \mathbf{Y} 的“散度”：

$$\begin{aligned}\mathcal{O}_{KL} &= D(\mathbf{X} \parallel \mathbf{Y}) = D(\mathbf{X} \parallel \mathbf{UV}^T) \\ &= \sum_{i,j} \left(x_{ij} \log \frac{x_{ij}}{y_{ij}} - x_{ij} + y_{ij} \right)\end{aligned}$$

这里 $\mathbf{Y} = [y_{ij}] = \mathbf{UV}^T$ 。这个度量不是对称的。 \mathbf{X} 到 \mathbf{Y} 的距离不一定等于 \mathbf{Y} 到 \mathbf{X} 的距离。当 $\sum_{ij} X_{ij} = 1$ 和 $\sum_{ij} Y_{ij} = 1$ 时，它减小为 Kullback-Leibler 散度，即相对熵。

这两个目标函数在变量 \mathbf{U} 和 \mathbf{v} 上都不是凸的，因此，很难找到 \mathcal{O}_F 或 \mathcal{O}_{KL} 的全局极小值。Lee 和 Seung 提出了一种迭代更新算法[24]来寻找上述优化问题的局部最优解。

NMF 分解，每列向量 \mathbf{U} ， \mathbf{u}_i ，可以被视为一个基础和每个数据点 \mathbf{x}_i 是近似的线性组合这些 k 基地，加权的组件 \mathbf{v} 。换句话说，NMF 习每个数据映射从 m 维空间 k 维空间。新的表示空间由 k 个基的 \mathbf{u}_i 生成。在真实的应用程序中，如图像处理[2]，人脸识别[12]，[25]，和文档聚类[13]，[14]，我们通常设置 $k \ll m$ 和 $k \ll n$ 。然后，高维数据可以表示为一组低维向量，希望基向量可以发现潜在语义结构的数据集。与 PCA、LDA、LPP[4]等降维算法不同的是， \mathbf{U} 和 \mathbf{V} 上的非负约束只允许基向量的可加性组合，这也是 NMF 被认为是基于零件的表示的原因。

Ding 等人[26]提出了一种半非负矩阵分解算法，该算法只限制一个矩阵因子包含非负项，同时松弛了对基向量的约束。

除了用乘法更新方法寻找矩阵因子外，还有人提出了梯度下降算法[28]来解决优化问题。该算法也被称为交替非负最小二乘或“投影梯度”。Lin[29]表明项目梯度法比乘法更新法收敛更快。Heiler 等人[30]推导了基于序列二次和二阶规划的 NMF 优化方案。这种方法的一个关键优点是，NMF 可以通过在相同的优化框架中以额外约束的形式合并先验知识来扩展。附加的约束是，对于每个类以及属于该类的每个标记点，将其系数限制为围绕类中心的圆锥。这一思想与我们的论文不同，我们的论文将标签信息作为附加的硬约束，要求来自同一类的数据点在新的表示空间中合并在一起。

据我们所知，大多数现有的 NMF 变体和扩展集中在矩阵分解上，而没有将标签信息作为硬约束考虑在内。本文提出了一种新的半监督矩阵分解方法，该方法将标签信息作为附加约束。因此，数据点的新表示形式具有更强的鉴别能力。

3 半监督 NMF

NMF 是一种无监督学习算法。它不能直接应用于标签信息可用的情况。在本节中，我们介绍了一种新的矩阵分解方法，称为约束非负矩阵分解，它将标签信息作为附加的约束。该方法可以保证共享相同标签的数据点在低维空间中映射到相同的类中。

考虑一个由 n 个数据点 $\{\mathbf{x}_i\}_{i=1}^n$ 组成的数据集，其中前 l 个数据点 \mathbf{x}_1 的标签信息可用和其余 $n-l$ 数据点没有标记。假设有 c 类。每个数据点从 $\mathbf{x}_1, \dots, \mathbf{x}_l$ 被标记为一个类。我们首先构建指标矩阵 \mathbf{c} ，其中 $c_{i,j}=1$ ；如果 \mathbf{x}_i 被标记为第 j 类，则 $c_{i,j}=0$ 。利用指标矩阵 \mathbf{C} ，定义一个标签约束矩阵 \mathbf{A} ，如下所示：

$$\mathbf{A} = \begin{pmatrix} \mathbf{C}_{l \times c} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n-l} \end{pmatrix}$$

其中 \mathbf{I}_{n-l} 是一个 $(n-l) \times (n-l)$ 单位矩阵。以 n 个数据点为例，其中 \mathbf{x}_1 、 \mathbf{x}_2 标记为 I 类， \mathbf{x}_3 、 \mathbf{x}_4 标记为 II 类， \mathbf{x}_5 标记为 III 类，另外 $n-5$ 个数据点未标记。基于本例的标签约束矩阵 \mathbf{A} 表示为：

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \mathbf{I}_{n-5} \end{pmatrix}$$

回想一下，NMF 将每个数据点 x_i 映射到 v_i ，从 m 维空间映射到 k 维空间。为了合并标签信息，我们可以通过引入一个辅助矩阵 Z 来施加标签约束： $V=AZ$ 。

从上面的方程，很容易检查，如果 x_i 和 x_j 有相同的标签，那么 $v_i=v_j$ 。在标签约束下，通过寻找两个非负矩阵因子 U 和 Z ，将原始的 NMF 扩展为半监督学习算法(CNMF)，其中因子 U 、 Z 和 a 的乘积是原始矩阵的近似表示为 $\mathbf{X} \approx \mathbf{U}(\mathbf{AZ})^T$ 。

4 最小化 F 范数代价的算法

4.1 更新算法

使用 Frobenius 范数作为代价函数，我们的带有标签约束的 CNMF 算法减少，以最小化以下目标函数： $\mathcal{O}_F = \|\mathbf{X} - \mathbf{UZ}^T \mathbf{A}^T\|$ 。约束为 $u_{i,j}$ 和 $z_{i,j}$ 是非负的。

(1)中 CNMF 的目标函数在变量 U 和 Z 上都不是凸的，因此寻找 \mathcal{O}_F 的全局极小值是不现实的。在下文中，我们描述了一种迭代更新算法来获得 \mathcal{O}_F 的局部最优值。

使用矩阵属性 $Tr(\mathbf{AB}) = Tr(\mathbf{BA})$ ，可以将目标函数 \mathcal{O}_F 重写为：

$$\begin{aligned}\mathcal{O}_F &= Tr\left((\mathbf{X} - \mathbf{UZ}^T \mathbf{A}^T)(\mathbf{X} - \mathbf{UZ}^T \mathbf{A}^T)^T\right) \\ &= Tr(\mathbf{XX}^T) - 2Tr(\mathbf{XAZU}^T) + Tr(\mathbf{UZ}^T \mathbf{A}^T \mathbf{AZU}^T)\end{aligned}$$

让 α_{ij} 和 β_{ij} 是约束的拉格朗日乘子 $u_{ij} > 0$ 和 $z_{ij} > 0$ ，分别和 $\boldsymbol{\alpha} = [\alpha_{ij}]$, $\boldsymbol{\beta} = [\beta_{ij}]$ 。

拉格朗日函数 \mathcal{L} 是 $\mathcal{L} = \mathcal{O}_F + Tr(\boldsymbol{\alpha} \mathbf{U}^T) + Tr(\boldsymbol{\beta} \mathbf{Z}^T)$ 。

要求 \mathcal{L} 对 U 和 Z 的导数为零，我们得到：

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \mathbf{U}} &= -2\mathbf{XAZ} + 2\mathbf{UZ}^T \mathbf{A}^T \mathbf{AZ} + \boldsymbol{\alpha} = 0, \\ \frac{\partial \mathcal{L}}{\partial \mathbf{Z}} &= -2\mathbf{A}^T \mathbf{X}^T \mathbf{U} + 2\mathbf{A}^T \mathbf{AZU}^T \mathbf{U} + \boldsymbol{\beta} = 0.\end{aligned}$$

利用 Kuhn-Tucker 条件 $\alpha_{ij} u_{ij} = 0$ 和 $\beta_{ij} z_{ij} = 0$ ，我们得到以下关于 u_{ij} 和 z_{ij} 的方程：

$$\begin{aligned}(\mathbf{XAZ})_{ij} u_{ij} - (\mathbf{UZ}^T \mathbf{A}^T \mathbf{AZ})_{ij} u_{ij} &= 0 \\ (\mathbf{A}^T \mathbf{X}^T \mathbf{U})_{ij} z_{ij} - (\mathbf{A}^T \mathbf{AZU}^T \mathbf{U})_{ij} z_{ij} &= 0\end{aligned}$$

这些方程得到以下更新规则：

$$u_{ij} \leftarrow u_{ij} \frac{(\mathbf{XAZ})_{ij}}{(\mathbf{UZ}^T \mathbf{A}^T \mathbf{AZ})_{ij}},$$

$$z_{ij} \leftarrow z_{ij} \frac{(\mathbf{A}^T \mathbf{X}^T \mathbf{U})_{ij}}{(\mathbf{A}^T \mathbf{AZU}^T \mathbf{U})_{ij}}.$$

对于上述迭代更新规则，有如下定理：

定理 1：在(2)和(3)的更新规则下，(1)中的目标函数 \mathcal{O}_F 是不增长的。当

且仅当 \mathbf{U} 和 \mathbf{Z} 处于平稳点时，目标函数在这些更新下是不变的。

定理 1 保证(2)和(3)的迭代收敛，因此最终的解将是一个局部最优解。

下面，我们将给出定理 1 的证明。

4.2 收敛性的证明

为了证明定理 1，我们使用了一个辅助函数的如下性质，正如期望最大化算法[31]，[32]所使用的。

引理 2。如果 $F(x)$ 存在辅助函数 G ，满足条件 $G(x, x') \geq F(x)$ 和 $G(x, x) = F(x)$ ，

则 F 不增加在更新 $x^{t+1} = \arg \min_x G(x, x')$ 。

等式 $F(x^{t+1}) = F(x^t)$ 仅当 x^t 是 $G(x, x')$ 的局部最小值时成立。通过迭代(4)

中的更新，估计序列将收敛到局部最小 $x_{\min} = \arg \min_x G(x, x')$ 。我们将通过

为(1)中的目标函数定义一个适当的辅助函数来说明这一点。

首先，我们证明(3)中更新规则的收敛性。对于 \mathbf{Z} 中的任意元素 z_{ab} ，设 $F_{Z_{ab}}$ 表示 \mathcal{O} 中与 z_{ab} 相关的部分。由于更新本质上是元素的，因此可以充分证明在(3)的更新步骤下，每个 $F_{Z_{ab}}$ 都是不递增的。我们通过定义关于 z_{ab} 的辅助函数来证明这一点：

引理 3。设 f_0 表示对应于 z 的一阶导数，函数

$$G(z, z_{ab}^t) = F_{z_{ab}}(z_{ab}^t) + F'_{z_{ab}}(z_{ab}^t)(z - z_{ab}^t) + \frac{(\mathbf{A}^T \mathbf{AZU}^T \mathbf{U})_{ab}}{z_{ab}^t} (z - z_{ab}^t)^2$$

是 $F_{Z_{ab}}$ 的辅助函数，是 \mathcal{O}_F 中仅与 z_{ab} 相关的部分。

证明。显然， $G(z, z) = F_{z_{ab}}(z)$ 。根据辅助函数的定义，我们只需证明

$G(z, z_{ab}^t) \geq F_{z_{ab}}(z)$ 。为了做到这一点，我们比较 $G(z, z_{ab}^t)$ 在(5)中的 $F_{Z_{ab}}$ 的

泰勒级数展开 $F_{z_{ab}}(z)$ ： $F_{z_{ab}}(z) = F_{z_{ab}}(z_{ab}^t) + F'_{z_{ab}}(z_{ab}^t)(z - z_{ab}^t) + \frac{1}{2} F''_{z_{ab}}(z_{ab}^t)(z - z_{ab}^t)^2$ 。

F'' 是对 z 的二阶导数，很容易检验

$$F'_{z_{ab}} = \left(\frac{\partial \mathcal{O}}{\partial \mathbf{Z}} \right)_{ab} = (-2\mathbf{A}^T \mathbf{X}^T \mathbf{U} + 2\mathbf{A}^T \mathbf{A} \mathbf{Z} \mathbf{U}^T \mathbf{U})_{ab}$$

$$F''_{z_{ab}} = 2(\mathbf{A}^T \mathbf{A})_{aa} (\mathbf{U}^T \mathbf{U})_{bb}$$

将(7)代入(6)并与(5)比较，我们可以看到，不是显示 $G(z, z_{ab}^t) \geq F_{z_{ab}}(z)$ ，就

等于证明 $\frac{(\mathbf{A}^T \mathbf{A} \mathbf{Z} \mathbf{U}^T \mathbf{U})_{ab}}{z_{ab}^t} \geq \frac{1}{2} F''_{z_{ab}} = (\mathbf{A}^T \mathbf{A})_{aa} (\mathbf{U}^T \mathbf{U})_{bb}$ 。

为了证明上述不等式，我们有

$$\begin{aligned} (\mathbf{A}^T \mathbf{A} \mathbf{Z} \mathbf{U}^T \mathbf{U})_{ab} &= \sum_{l=1}^k (\mathbf{A}^T \mathbf{A} \mathbf{Z})_{al} (\mathbf{U}^T \mathbf{U})_{lb} \\ &\geq (\mathbf{A}^T \mathbf{A} \mathbf{Z})_{ab} (\mathbf{U}^T \mathbf{U})_{bb} \\ &\geq \sum_{l=1}^k (\mathbf{A}^T \mathbf{A})_{al} z_{lb}^t (\mathbf{U}^T \mathbf{U})_{bb} \\ &\geq z_{ab}^t (\mathbf{A}^T \mathbf{A})_{aa} (\mathbf{U}^T \mathbf{U})_{bb}. \end{aligned}$$

接下来，我们为(2)中的更新规则定义一个辅助函数。同样，让 $\mathbf{F} \mathbf{u}_{ab}$ 表示 \mathcal{O}_F 中与 \mathbf{u}_{ab} 相关的部分。那么，关于 \mathbf{u}_{ab} 的辅助函数定义如下：

$$G(u, u_{ab}^t) = F_{u_{ab}}(u_{ab}^t) + F'_{u_{ab}}(u_{ab}^t)(u - u_{ab}^t)$$

引理 4。函数 $\frac{(\mathbf{U} \mathbf{Z}^T \mathbf{A}^T \mathbf{A} \mathbf{Z})_{ab}}{u_{ab}^t} (u - u_{ab}^t)^2$ 是 $\mathbf{F} \mathbf{u}_{ab}$ 的辅助函数，是

\mathcal{O}_F 中仅与 \mathbf{u}_{ab} 相关的部分。

引理 4 的证明与引理 3 的证明在本质上是相似的，由于篇幅的限制，此处略去。利用上面的引理，现在我们给出定理 1 的证明。

定理 1 的证明。将 $G(z, z_{ab}^t)$ (5) 变成(4)，我们得到

$$z_{ab}^{t+1} = \arg \min_z G(z, z_{ab}^t) = z_{ab}^t \frac{(\mathbf{A}^T \mathbf{X}^T \mathbf{U})_{ab}}{(\mathbf{A}^T \mathbf{A} \mathbf{Z} \mathbf{U}^T \mathbf{U})_{ab}}.$$

根据引理 3，由于(5)是一个辅助函数，因此在此更新规则下 $\mathbf{F} \mathbf{Z}_{ab}$ 是非递增的。

虽然对于每一个更新步骤，KL 发散公式中 CNMF 的代价比 NMF 略高一些，但由于收敛的迭代次数不同，CNMFKL 的整体算法复杂度可能并不会更慢。我们将在下一节实验研究这些算法的收敛速度。假设 NMF、CNMF 和 CNMFKL 的相乘更新分别在 t_1 、 t_2 和 t_3 迭代后停止，则这些算法的总体计算复杂度为 $O(t_1 m n k)$ ， $O(t_2 m n k)$ 和 $O(t_3 (m+n) k)$ 。

7 实验结果

在本节中，我们将研究如何使用我们提出的 CNMF 算法进行数据聚类。通过实验验证了该算法在图像聚类中的有效性。

7.1 评价指标

我们使用两个指标来评估聚类性能[1], [13]。通过比较每个样本的聚类标签与数据集提供的标签来评估结果。一个度量是准确度(AC)，它用于测量获得的正确标签的百分比。给定一个包含 n 幅图像的数据集，对于每幅样本图像，设 l_i 为我们通过不同算法得到的聚类标签， r_i 为数据集

提供的标签。精度定义为 $AC = \frac{\sum_{i=1}^n \delta(r_i, \text{map}(l_i))}{n}$ 。

在 $\delta(x, y)$ 是如果 $x=y$ 等于 1，否则等于 0 的 delta 函数， $\text{map}(l_i)$ 是映射函数，将每个集群标签 l_i 映射到数据集中的等价标签。利用 Kuhn-Munkres 算法[34]可以找到最佳映射。

第二个度量是标准化互信息(MIc)。在聚类应用程序中，互信息用于度量两组聚类的相似程度。给定两组图像聚类 C 和 C' ，它们的互信息度量

$$MI(C, C') \text{ 定义为 } MI(C, C') = \sum_{c_i \in C, c'_j \in C'} p(c_i, c'_j) \cdot \log \frac{p(c_i, c'_j)}{p(c_i) \cdot p(c'_j)}。$$

式中， $p(c_i), p(c'_j)$ 表示任意从数据集中选取的图像分别属于 c_j 和 c'_j 簇的概率， $p(c_i, c'_j)$ 表示任意选取的图像同时属于 c_j 和 c'_j 簇的概率。 $MI(C, C')$ 取值范围为 $0 \sim \max(H(C), H(C'))$ ， $H(C), H(C')$ 为 C 的熵值， $H(C_0)$ 为 C_0 的熵值。当两组图像簇相同时，达到最大 $H(C), H(C')$ ，当两组图像簇完全独立时，则为零。 $MI(C, C')$ 的一个重要特征： C' 是该值对于所有类型的排列保持相同。在我们的实验中，我们使用归一化度量 $MI(C, C')$ ，取值范围在 0 到 1 之间：

$$\widehat{MI}(C, C') = \frac{MI(C, C')}{\max(H(C), H(C'))}$$

7.2 评估与比较

为了展示数据聚类的性能，我们在四个数据集上比较了我们的算法和其他相关方法。我们评估的算法如下：

我们提出的约束非负矩阵分解算法最小化 f 范数代价。我们提出的约束非负矩阵分解算法最小化 KL 发散代价(CNMFKL)。基于非负矩阵分解的聚类。我们实现了一个标准化的 NMF 削减加权版本, 如[13]所建议的。非负张量分解(NTF)NTF 是 NMF 对张量数据的扩展。在 NTF 中, 每个人脸图像被表示为一个二阶张量, 而不是一个矢量。图正则化非负矩阵分解[1], 将数据空间的几何信息编码为矩阵分解。流形上的半监督图正则化非负矩阵分解(SemiGNMF)[1]。该方法通过修改权值矩阵, 将标签信息整合到图结构中。

分解聚类(CF)概念[36]。我们评估了在四个图像数据集上的聚类性能。这些数据集包含许多类别的图像。表 3 总结了这些数据集的重要统计数据。稍后我们将分别描述数据集的细节。对于每个数据集, 用 2 到 10 个不同数量的聚类进行评价。对于固定的聚类数 k , 我们进行如下实验:

1. 我们从数据集中随机选取 k 个类别, 将这 k 个类别的图像混合作为集合 X 进行聚类。对于半监督算法(CNMF、CNMFKL 和 SemiGNMF), 我们随机从 X 中的每个类别中选取 10% 的图像, 并使用它们的类别号作为可用的标签信息。例外情况是 ORL 数据库。ORL 中每个类别只有 10 张图片, 10% 的图片只有一张。对于 CNMF 来说, 一个标签是没有意义的, 因为该算法将具有相同标签的图像映射到同一点上。因此, 对于 ORL, 我们从每个类别中随机选择两幅图像来提供标签信息。

2. 在聚类集合上, 我们应用上述不同的矩阵分解算法来获得新的数据表示 v 。我们将新空间的维数设置为与聚类数量相同(光谱聚类中也使用了同样的技术)。因此, 这一步将数据从原始空间映射到低维(k 维)空间。

3. 然后将 K-means 应用到新的数据表示 V 中进行图像聚类。用不同的初始点重复 K-means 20 次, 得到 K-means 代价函数的最佳结果。

4. 我们将得到的聚类与原始图像分类进行比较, 计算精度和归一化互信息。

以上过程重复 10 次, 记录平均的聚类性能作为最终结果。正如我们前面提到的, 在我们的方法中没有参数。对于其他算法, 参数设置为每个算法能够达到其最佳结果的值。

7.2.1 ORL 数据库

AT&TORL 数据库 1 包含 40 个不同的受试者的 10 张不同的图像, 因此总共有 400 张图像。对于一些受试者, 图像是在不同的时间拍摄的, 光线变化, 面部表情(睁眼/闭眼, 微笑/不微笑), 面部细节(戴眼镜/不戴眼镜)。所有的图像都是在深色的均匀背景下拍摄的, 受试者处于直立的正面位置。

在所有的实验中, 图像都进行了预处理, 以确定人脸的位置。原始图像首先在尺度和方向上进行归一化, 使两只眼睛在同一位置对齐。然后, 将面部区域裁剪成最终的图像进行聚类。每个图像是 32x32 像素, 每个像素 256 个灰度级别。

图 1 显示了精度和归一化互信息与簇数的关系。我们提出的 CNMF 和 CNMFKL 算法始终优于所有其他算法。CF、NMF、GNMF 和 SemiGNMF 的性能相当。在该数据集中, CNMF 的性能最好, CNMFKL 的性能次之。

我们还从 ORL 数据库中随机选择了 25 名受试者，每个受试者有 10 张图片，并对总共 250 张图片运行算法。图 2 显示了效果。第一幅图像包含了 25 个主题，另外三幅是由 NMF、CNMF 和 CNMFKL 获得的基向量。

7.2.2 Yale 数据库

耶鲁数据库 2 包含 15 个个体的 165 张灰度图像。每个受试者有 11 张图片，每种不同的面部表情或配置都有一张：中心光、带眼镜、快乐、左光、不带眼镜、正常、右光、悲伤、困倦、惊讶和眨眼。我们对这个数据集进行与 ORL 数据集相同的预处理。因此，每个图像也由图像空间中的 1, 024 维向量表示。

聚类结果如表 6 所示。图示如图 4 所示。我们可以看到，NMF、GNMF 和 SemiGNMF 有相似的结果。虽然 SemiGNMF 利用了标签信息，但与 NMF 和 GNMF 相比，SemiGNMF 并没有表现出多大的优势。相比之下，CNMF 和 CNMFKL 在所有情况下都大大优于上述三种方法，其中 CNMFKL 的性能最好。从表 6 可以看出，与我们提出的 CNMF 算法之外的最佳算法 SemiGNMF 相比，CNMFKL 提高了 7.46%，CNMF 提高了 4.41% 的准确率。对于规范化的互信息，CNMFKL 和 CNMF 分别提高了 8.38% 和 4.81%。

7.2.3 Caltech-101 数据库

Caltech-101 是由加州理工大学创建的数字图像数据集，包含 101 个对象类别。每个类别包含大约 40 到 800 张图片。每个图像的大小大约是 300x200 像素。在我们的实验中，我们选择了 10 个最大的类别，除了 BACKGROUND_GOOGLE 类别。总的来说，我们测试的子集包含 3, 044 张图片。我们对每幅图像进行与 Corel 数据集相同的预处理。我们首先提取 SIFT 描述符，然后生成码字作为每幅图像的特征。对于 Caltech-101 数据集，SIFT 描述符的数量为 555, 292，我们还生成了 500 个码字。最后，Caltech-101 中的每幅图像都由一个 500 维频率直方图表示。

聚类结果如表 7 所示，图 5 所示。在这个数据集上，CNMFKL 仍然显示出最好的性能。CNMF 在归一化互信息测量和大多数精度测量情况下都是第二好的。总的来说，我们的方法在聚类中显示了更好的有效性。如表 7 所示，与 SemiGNMF 相比，CNMFKL 的准确度提高了 7.2%。对于标准化互信息，CNMFKL 提高了 9.01%，CNMF 提高了 2.87%。

7.3 收敛性研究

我们使用迭代更新规则来获得 CNMF 目标函数的局部最优值，无论成本度量是在 Frobenius 范数还是 KL 发散。在前几节中，我们已经证明了更新规则的收敛性，并分析了计算复杂性。本文通过实验证明了算法的收敛速度。

我们比较了原始 NMF 算法、我们的 CNMF 最小化 f 范数代价(CNMF)和 CNMF 最小化 KL 发散代价(CNMFKL)的收敛速度。图 6 为三种算法在四个图像数据库上的收敛速度。对于每个图，x 轴是迭代次数，y 轴是

目标函数的值。我们可以看到，CNMF 和 CNMFKL 算法收敛速度都非常快。对于 ORL、Corel 和 Caltech-101 数据库，所有三种算法在 20 次迭代中收敛。对于耶鲁的数据库，它们在 100 次迭代中收敛。特别是，我们注意到，对于 Yale 数据库和 Corel 数据库，CNMFKL 在不到 10 次迭代的时间内收敛，展示了非凡的性能。这也验证了我们在第 6 节中所做的陈述，即尽管 CNMFKL 在每个更新步骤中需要更多的操作，但整个过程的总体成本并不高，因为它可能比 NMF 收敛得更快。

8 结论

在本文中，我们提出了一种新的矩阵分解方法，称为约束非负矩阵分解，它利用有标记和无标记的数据点。CNMF 将标签信息作为硬约束强加于目标函数。这样，数据点的新表示形式可以具有更强的鉴别能力。我们在两个公式中展示了 CNMF 方法，并提出了两个优化问题的更新算法。在四个标准图像数据库上的实验结果证明了该方法的有效性。CNMF 和 CNMFKL 算法都优于其他现有算法。特别是在所有情况下，k1-散度形式的 CNMF 算法都比其他算法具有明显的优势。此外，我们的算法是无参数的。因此，我们提出的 CNMF 可以很容易地应用于广泛的实际问题。