基于图正则化非负矩阵分解的图像分类算法

21921164，汪鑫

1. 引言

矩阵分解（Matrix Factorization）在数据表达领域是一种常用的手段，在信息提取、计算机视觉、模式识别等问题中，输入数据具有很高的维度，这使得直接从样本中学习变得不可行。一个合适的手段是将原始的输入数据矩阵近似为两个低维度矩阵的乘积，典型的矩阵分解技术有 LU 分解、QR 分解、向量量化和 SVD 等。

SVD 是一种非常流行的矩阵分解手段，其主要思想是将原始的$M \times N$维的矩阵$\boldsymbol{X}$分解为：

$$\boldsymbol{X} = \boldsymbol{U}\Sigma\boldsymbol{V}^T$$

其中$\boldsymbol{U}$和$\boldsymbol{V}$分别是$M \times M$和$N \times N$维的正交矩阵，Σ是$M \times N$维的对角矩阵。当去除足够小的奇异值以及与之对应的奇异向量，我们可以得到原始矩阵的低秩近似。就重构误差而言，SVD 算法是最优的，因此 SVD 在实际中得到了广泛应用。

非负矩阵分解（Nonnegative Matrix Factorization）是为了处理人脸识别、文本聚类等任务提出的，NMF 的主要思想是寻找两个非负的低秩矩阵，用他们的乘积近似原始矩阵。事实证明，在部分表示（Part-based Representation）任务中，NMF 能给出比 SVD 更好的结果。

近些年来，研究者关注了当数据是从某些分布中采样得到时的数据表示问题。在这种情况下，数据大多是处在一个高维空间中的子流形上，从局部上看，该流形是一个低维的欧式空间。为了考虑这种潜在的流形结构，许多流形学习算法如局部线性嵌入（LLE）、ISOMAP、tSNE 等被提出。这些算法都是基于同一思想：相近的数据点应该有着相似的嵌入表示。事实证明，在具有这种几何结构的数据中，流形学习能够给出更好的结果。

受流形学习的启发，论文[2]作者提出了图正则化非负矩阵分解算法（Graph Regularized Nonnegative Matrix Factorization, GNMF），与传统的 NMF 算法相比，作者还考虑了数据中存在的几何结构。这种几何结构在不同任务中可以从不同的渠道获取，如图像标签、用户社交信息等，很容易将其他信息融合到数据表示任务中。

2. 论文思想

2.1 NMF 算法回顾

传统的 NMF 是通过优化重构误差来获取原始数据的低维表示。给定数据矩阵$\boldsymbol{X} = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_N] \in \mathbb{R}^{M \times N}$，$\boldsymbol{X}$的每一列表示一个样本，矩阵$\boldsymbol{U} \in \mathbb{R}^{M \times K}$和$\boldsymbol{V} \in \mathbb{R}^{N \times K}$是原始矩

阵的两个低秩乘积因子。考虑重构误差，当采用平方误差和 KL 距离时，优化目标分别为：

$$O_1 = \|\boldsymbol{X} - \boldsymbol{U}\boldsymbol{V}^T\|_F^2 = \sum_{i,j}\left(x_{ij} - \sum_{k=1}^{K} u_{ik}v_{jk}\right)^2$$

$$O_2 = D(\boldsymbol{X}\|\boldsymbol{U}\boldsymbol{V}^T) = \sum_{ij}\left(x_{ij}\log\frac{x_{ij}}{y_{ij}} - x_{ij} + y_{ij}\right)$$

其中$\boldsymbol{Y} = [y_{ij}] = \boldsymbol{U}\boldsymbol{V}^T$。文献[1]给出了模型参数的计算公式：

$$u_{ik} \leftarrow u_{ik}\frac{(\boldsymbol{X}\boldsymbol{V})_{ik}}{(\boldsymbol{U}\boldsymbol{V}^T\boldsymbol{V})_{ik}},\quad v_{jk} \leftarrow v_{jk}\frac{(\boldsymbol{X}^T\boldsymbol{U})_{jk}}{(\boldsymbol{V}\boldsymbol{U}^T\boldsymbol{U})_{jk}}\ \text{(平方损失)}$$

$$u_{ik} \leftarrow u_{ik}\frac{\sum_j\left(x_{ij}v_{jk}/\sum_k u_{ik}v_{jk}\right)}{\sum_j v_{jk}}, v_{jk} = v_{jk}\frac{\sum_i\left(x_{ij}u_{ik}/\sum_k u_{ik}v_{ik}\right)}{\sum_i u_{ik}}\ \text{（KL 距离）}$$

2.2 GNMF 算法

NMF 算法仅仅是在欧式空间中进行学习，在融合数据内在的几何结构方面，NMF 算法并不奏效。GNMF 算法在传统的 NMF 算法中融入了数据的几何结构信息。数据的几何结构由一个权重矩阵$\boldsymbol{W}$表达，论文给出了三种定义方式：

1. 0-1 加权　如果样本$\mathrm{x_i}$和$\mathrm{x_j}$有联系，$\boldsymbol{W}_{ij} = \boldsymbol{W}_{ji} = 1$

2. 热核加权　如果样本$\mathrm{x_i}$和$\mathrm{x_j}$有联系，$\boldsymbol{W}_{ij} = \boldsymbol{W}_{ji} = \exp(-\frac{\|\mathrm{x_j}-\mathrm{x_i}\|^2}{\sigma})$

3. 内积加权　如果样本$\mathrm{x_i}$和$\mathrm{x_j}$有联系，$\boldsymbol{W}_{ij} = \boldsymbol{W}_{ji} = \mathrm{x_i^T x_j}$

融合几何结构信息是通过图正则项实现，其定义为：

$$\mathcal{R} = \sum_{ij} d(z_i, z_j)w_{ij}$$

其中$d(\cdot,\cdot)$表示距离度量，z_i为样本x_i低维表示。当采用二范数时，图正则项为：

$$\mathcal{R}_1 = \frac{1}{2}\sum_{i,j=1}^{N}\|\boldsymbol{z}_i - \boldsymbol{z}_j\|^2 w_{ij}$$

$$= \sum_{i=1}^{N}\boldsymbol{z}_i^T\boldsymbol{z}_i D_{ii} - \sum_{i,j=1}^{N}\boldsymbol{z}_i^T\boldsymbol{z}_j w_{ij}$$

$$= \mathrm{Tr}(\boldsymbol{V}^T\boldsymbol{D}\boldsymbol{V}) - \mathrm{Tr}(\boldsymbol{V}^T\boldsymbol{W}\boldsymbol{V}) = \mathrm{Tr}(\boldsymbol{V}^T\boldsymbol{L}\boldsymbol{V})$$

其中$D_{ii} = diag\{\sum_j w_{ij}\}$，$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$。

当采用 KL 距离时，图正则项为：

$$\mathcal{R}_2 = \frac{1}{2}\sum_{i,j=1}^{N}\sum_{k=1}^{K}\left(v_{ik}\log\frac{v_{ik}}{v_{jk}} + v_{jk}\log\frac{v_{jk}}{v_{ik}}\right)w_{ij}$$

GNMF 的优化目标为：

$$O_1 = \|\boldsymbol{X} - \boldsymbol{U}\boldsymbol{V}^T\|^2 + \lambda\mathrm{Tr}(\boldsymbol{V}^T\boldsymbol{L}\boldsymbol{V})$$

$$O_2 = \sum_{ij}\left(x_{ij}\log\frac{x_{ij}}{y_{ij}} - x_{ij} + y_{ij}\right) + \frac{\lambda}{2}\sum_{ij}\sum_{k=1}^{K}\left(v_{ik}\log\frac{v_{ik}}{v_{jk}} + v_{jk}\log\frac{v_{jk}}{v_{ik}}\right)w_{ij}$$

与 NMF 求解类似，得到更新公式为：

$$u_{ik} \leftarrow u_{ik}\frac{(\boldsymbol{XV})_{ik}}{(\boldsymbol{UV}^T\boldsymbol{V})_{ik}},\ v_{jk} \leftarrow v_{jk}\frac{(\boldsymbol{X}^T\boldsymbol{U} + \lambda\boldsymbol{WV})_{jk}}{(\boldsymbol{VU}^T\boldsymbol{U} + \lambda\boldsymbol{DV})_{jk}}$$ (平方损失)

$$u_{ik} \leftarrow u_{ik}\frac{\sum_j\left(x_{ij}v_{jk}/\sum_k u_{ik}v_{jk}\right)}{\sum_j v_{jk}}$$

$$\mathbf{v}_k = \left(\sum_i u_{ik}\mathbf{I} + \lambda\boldsymbol{L}\right)^{-1}\begin{bmatrix} v_{1k}\sum_i\left(x_{i1}u_{ik}/\sum_k u_{ik}v_{1k}\right) \\ v_{2k}\sum_i\left(x_{i2}u_{ik}/\sum_k u_{ik}v_{2k}\right) \\ \cdots \\ v_{Nk}\sum_i\left(x_{iN}u_{ik}/\sum_k u_{ik}v_{Nk}\right)\end{bmatrix}$$ （KL 距离）

三、实验结果

3.1 数据集

实验中采用了三个数据集：MNIST 手写数字、COIL20 物体数据集和 CelebA 人脸数据集，数据集的详细信息如下：

数据集	样本数	输入维度	类别数
COIL20	1440	16384	20
Caltech-101	9145	60000	101

3.2 比较算法

为了验证 GNMF 的效果，实验选用了四个算法进行比较，分别是：

1. **K-means**
2. **PCA 降维，在低维空间使用 K-means 聚类**
3. **NMF，根据模型参数的更新公式可知，当 GNMF 中的正则化系数$\lambda = 0$,GNMF 退化成了 NMF**
4. **GNMF**

3.3 模型度量

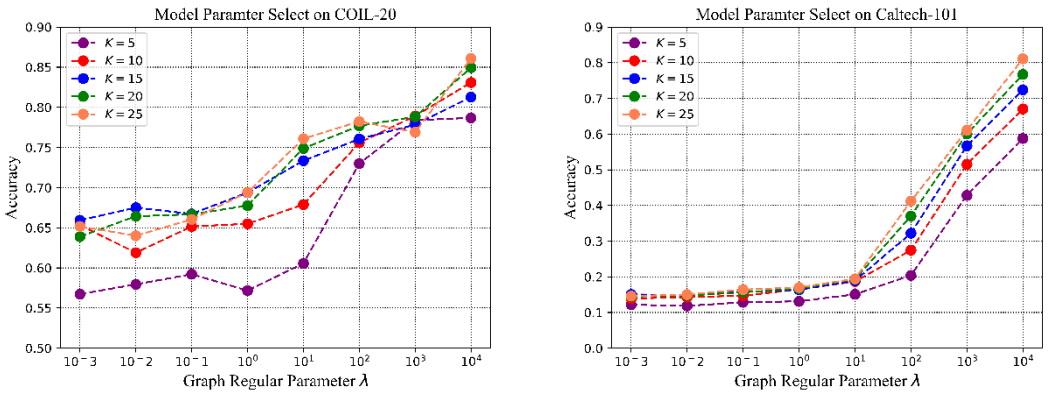
模型度量采用了聚类精度作为衡量指标，其定义为：

$$AC = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\{r_i = map(l_i)\}$$

其中n为聚类样本的个数，r_i为样本的真实类别，l_i为样本根据聚类得到的类别，函数$map(\cdot)$是聚类类别与真实类别之间的最优映射，该映射可以由 KuhnMunkres 算法得到。在实验中调用了 sklearn 库中的 linear_assignment()函数。

3.4 模型选择

在 GNMF 模型中，存在着两个重要的超参数：图正则化系数λ和子空间维度K。前者决定了对数据中几何结构的重视程度，后者反应了降维之后数据信息的丢失量。这两个参数对模型效果影响很大，因此需要对这两个参数的影响进行研究，从而获得一个较好的模型。在$\lambda \in [10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}, 10^{4}]$和$K \in [5, 10, 15, 20, 25]$的范围进行搜索，采用 5 折交叉得到如下结果：



从实验结果可以看出，聚类的准确性随着子空间维度和图正则化系数的增加而增加。

3.5 模型比较

表格 1 COIL-20

K	K-means	PCA	NMF	GNMF
10	0.664	0.683 ± 0.015	0.632 ± 0.034	0.831 ± 0.013
15		0.686 ± 0.025	0.653 ± 0.022	0.813 ± 0.018
20		0.706 ± 0.026	0.657 ± 0.026	0.849 ± 0.039
25		0.700 ± 0.043	0.651 ± 0.023	0.861 ± 0.036

表格 2 Caltech-101

K	K-means	PCA	NMF	GNMF
10	0.134	0.131 ± 0.006	0.139 ± 0.006	0.670 ± 0.014
15		0.137 ± 0.003	0.146 ± 0.006	0.724 ± 0.013
20		0.139 ± 0.004	0.141 ± 0.004	0.767 ± 0.016
25		0.134 ± 0.003	0.142 ± 0.004	0.811 ± 0.027

根据 3.4 节中搜索出的参数，将 GNMF 模型与另外三种模型进行比较，采用 5 折交叉，运行结果的格式为：*均值* ± *标准差*，结果列在表格 1 和 2 中。从实验结果可以看出，GNMF 的聚类精度远高于另外三种模型，尤其是在数据集 Caltech-101 上，目标的类别非常多时，GNMF 的效果远远好于基准模型。

四、参考文献

[1] Lee D D, Seung H S, “Learning the parts of objects by non-negative matrix factorization”, Nature, 401,6755,788-791, 1999.

[2] Cai D, He X, Han J, et al, “Graph regularized nonnegative matrix factorization for data representation”,IEEE transactions on pattern analysis and machine intelligence, 33,8,1548-1560, 2010.