

基于蒙特卡洛方法的非负矩阵分解初始化

陈红莉

(武汉大学数学与统计学院, 湖北 武汉 430072)

摘要: 在非负矩阵分解中, 初值的选择对于算法效果有很大的影响. 一些基于奇异值分解的初始化方法已有人提出^[7,8], 但当矩阵维数过大时, 直接对原矩阵进行奇异值分解是耗时的. 本文提出了一种更节时的初始化方法 (KFV-NMF), 而且通过数值实验, 此算法既在一定程度上保持了计算精度, 也节省了计算时间.

关键词: 非负矩阵分解; 初始化; 奇异值分解; FKV

MR(2010) 主题分类号: 97N30; 49M27

中图分类号: O241.6

文献标识码: A

文章编号: 0255-7797(2020)04-0498-07

1 引言

非负矩阵分解 (NMF) 是 1999 年由 Lee 和 Seung^[1] 首次提出的, 有着比较广泛的应用, 如图像处理^[1]、文本聚类^[2] 和社区发现^[3] 等. 相较于其他的矩阵分解, 由于元素的非负性, 所以可解释性很强. 比如: 在处理图像时, 负的像素点是没有意义的; 在文本聚类时, 正值代表文档以一定概率属于某个主题, 负值则没有任何实际意义.

非负矩阵分解数学描述如下: 给定非负矩阵 $A \in \mathbb{R}^{m \times n}$, 希望找到两个非负矩阵 $W \in \mathbb{R}^{m \times k}$ 和 $H \in \mathbb{R}^{k \times n}$, 使得 $A \approx WH$. 可以通过以下优化问题来找到 W 和 H ,

$$\min_{W, H \geq 0} f(W, H) = \frac{1}{2} \|A - WH\|_F^2,$$

其中 $\|\cdot\|_F$ 代表矩阵的 Frobenius 范数.

解决非负矩阵分解的方法有很多, 比较常见的是由 Lee 和 Seung 在 2001 年提出的基准算法 (Multiplicative Update Rules)^[4]、Lin 在 2007 年提出的投影梯度法^[5] 以及 Kim 和 Park 在 2008 年提出的交替非负最小二乘法 (ANLS)^[6].

不同于某些迭代算法, 非负矩阵分解初始值的选择对于算法效果有很大影响. 由 Qiao 提出的 SVD-NMF^[7] 和 Boutsidis 等人提出的 NNDSVD^[8] 算法都是基于矩阵 A 的奇异值分解, 但是当矩阵 A 的维数很大时, 对 A 直接进行奇异值分解是耗时的. 不同于这两种方法, 本文不直接对原矩阵 A 进行奇异值分解. 本文利用 Frieze 等人^[9] 的思想, 通过抽样构造一个更小的矩阵, 对这个小矩阵来做奇异值分解, 从而实现了对 W 和 H 的初始化, 这大大节省了算法的计算时间.

2 预备知识

*收稿日期: 2019-04-26

接收日期: 2019-07-08

作者简介: 陈红莉 (1993-), 女, 安徽淮北, 硕士, 主要研究方向: 数值代数.

符号说明: 用 $\|A\|_F$ 表示矩阵 A 的 Frobenius 范数; $\|x\|$ 表示向量的 2-范数; 矩阵 A 的第 i 行和第 j 列分别用列向量 $A(i, \cdot)$ 和 $A(\cdot, j)$ 来表示; A_{ij} 和 $x(i)$ 分别表示矩阵 A 的第 (i, j) 个元素和向量 x 的第 i 个元素; A^T 和 x^T 分别表示矩阵 A 和向量 x 的转置. 对于一个非零向量 $x \in \mathbb{R}^n$, 用 D_x 表示 x 在 $\{1, 2, \dots, n\}$ 上的分布, 其中概率密度函数为 $D_x(i) = |x(i)|^2 / \|x\|^2$. 通常把依据分布 D_x 产生的样本称为从 x 中产生的样本.

首先, 介绍一种最为经典的非负矩阵分解算法, 即由 Lee 和 Seung 在 2001 年提出的基准算法 (Multiplicative Update Rules) [4], 以下简称 MU 算法. 在本文的数值实验部分, 将会用到 MU 算法. MU 算法步骤如下

步骤 1 初始化 $W_{ia}^0 \geq 0, H_{bj}^0 \geq 0, \forall i, a, b, j$.

步骤 2 对于 $k = 1, 2, \dots$,

$$W_{ia}^k = W_{ia}^{k-1} \frac{(A(H^{k-1})^T)_{ia}}{(W^{k-1}H^{k-1}(H^{k-1})^T)_{ia}}, \quad \forall i, a, \quad (2.1)$$

$$H_{bj}^k = H_{bj}^{k-1} \frac{((W^k)^T A)_{bj}}{((W^k)^T W^k H^{k-1})_{bj}}, \quad \forall b, j. \quad (2.2)$$

从式子 (2.1) 和 (2.2) 中可以看出, 如果在第 k 次迭代中 W 和 H 是非负的, 那么在第 $k+1$ 次迭代中, MU 算法依然能够保证 W 和 H 是非负的. 并且, Lee 和 Seung [4] 已经证明在 MU 算法下, 目标函数 $f(W, H)$ 是非增的, 即 $f(W^k, H^{k-1}) \leq f(W^{k-1}, H^{k-1})$, $f(W^k, H^k) \leq f(W^k, H^{k-1})$.

由于本文提出的算法涉及到抽样, 下面给出对向量和矩阵进行抽样的假设.

假设 2.1 [10] 给定向量 $x \in \mathbb{R}^n$, 假设可以根据分布 D_x 来对 $i \in \{1, 2, \dots, n\}$ 进行抽样, 也就是说 i 被抽中的概率为 $D_x(i) = \frac{|x(i)|^2}{\|x\|^2}$.

假设 2.2 [10] 给定矩阵 $A \in \mathbb{R}^{m \times n}$, 假设满足以下假设

1. 可以对矩阵 A 的行标 $i \in \{1, 2, \dots, m\}$ 进行取样, 矩阵 A 的第 i 行被选中的概率为

$$P_i = \frac{\|A(i, \cdot)\|^2}{\|A\|_F^2}.$$

2. 对于任意的 $i \in \{1, 2, \dots, m\}$, 可以根据分布 $D_{A(i, \cdot)}$ 来对矩阵 A 的列标 $j \in \{1, 2, \dots, n\}$ 进行抽样, 也就是说 j 被抽中的概率为

$$D_{A(i, \cdot)}(j) = \frac{|A_{ij}|^2}{\|A(i, \cdot)\|^2}.$$

定理 2.3 [9] 给定矩阵 $A \in \mathbb{R}^{m \times n}$, 独立的按照概率分布

$$\{\|A(1, \cdot)\|^2 / \|A\|_F^2, \dots, \|A(m, \cdot)\|^2 / \|A\|_F^2\},$$

从行标中取 p 个样本: i_1, \dots, i_p . 记 $S \in \mathbb{R}^{p \times n}$ 为规范化后的子矩阵, 即对于 $t \in \{1, \dots, p\}$, $S(t, \cdot) = A(i_t, \cdot) / \sqrt{p \|A(i_t, \cdot)\|^2 / \|A\|_F^2}$. 则对于任意给定的 $\theta > 0$, 满足

$$\Pr(\|A^T A - S^T S\|_F \geq \theta \|A\|_F^2) \leq \frac{1}{\theta^2 p}.$$

在第 3 节, 会根据定理 2.3 说明本文提出的初始化方法 (KFV-NMF) 的可行性.

3 NMF 初始化算法

由定理 2.3, 可以看出, 若按照满足假设 2.2 的概率分布来抽样: 假设抽取了矩阵 A 的 p 行, 并按照定理 2.3 所说的构造一个 $p \times n$ 阶的矩阵 S , 则可以用 $S^T S$ 来近似 $A^T A$. A 的奇异值分解可以通过 $A^T A$ 的谱分解得到, 尽管这里可以从 $S^T S$ 的谱分解入手, 但并不打算这样做, 因为 $S^T S$ 是一个 $n \times n$ 阶的矩阵, 维数也是很大的. 这里可以对 S 的列进行同样的操作, 从 S 中抽取 p 列, 类似的得到一个 $p \times p$ 阶的矩阵 W , 则 $W W^T$ 近似 $S S^T$, 故最终可以直接对一个 $p \times p$ 阶的矩阵 W 做奇异值分解, 从而近似得到矩阵 A 的右奇异向量, 这大大减少了计算时间. 这种思想是 Frieze 等人^[9]提出的, 本文借鉴这种思想提出了一种更节时的非负矩阵分解初始化方法, 记此算法为 KVV-NMF.

下面, 给出算法的具体流程.

1 KVV-NMF 初始化

输入: A, p, k, ϵ .

1. 对 A 的行进行取样: 独立的按照满足假设 2.2 的概率分布 $P = (P_1, P_2, \dots, P_m)$. 抽取 p 个行标, 记为 i_1, i_2, \dots, i_p , 即

$$P_i = \frac{\|A(i, \cdot)\|^2}{\|A\|_F^2}.$$

构造 $p \times n$ 阶矩阵 S , 其中 $S(t, \cdot) = A(i_t, \cdot) / \sqrt{p P_{i_t}}$, $t = 1, 2, \dots, p$.

2. 对 S 的列进行抽样: 独立的按照满足假设 2.1 的概率分布 $P' = (P'_1, P'_2, \dots, P'_n)$. 抽取 p 个列标, 记为 j_1, j_2, \dots, j_p , 即

$$P'_j = \sum_{t=1}^p D_{A(i_t, \cdot)}(j) / p.$$

构造 $p \times p$ 阶矩阵 W , 其中 $W(\cdot, t) = S(\cdot, j_t) / \sqrt{p P'_{j_t}}$, $t = 1, 2, \dots, p$.

3. 计算矩阵 W 前 k 个奇异值 $\hat{\sigma}_1, \hat{\sigma}_2, \dots, \hat{\sigma}_k$ 以及对应的左奇异向量 $\hat{u}_1, \hat{u}_2, \dots, \hat{u}_k$.

4. 构造 $n \times k$ 阶矩阵 \hat{V} : $\hat{V} = (\frac{S^T \hat{u}_1}{\hat{\sigma}_1}, \frac{S^T \hat{u}_2}{\hat{\sigma}_2}, \dots, \frac{S^T \hat{u}_k}{\hat{\sigma}_k})$.

5. 令 $W^0 = \max(\epsilon, A \hat{V})$, $H^0 = \max(\epsilon, \hat{V}^T)$.

输出: W^0 和 H^0 .

下面, 给出一个定理, 并以此来说明算法 KVV-NMF 中第 2 步的具体实施方法.

定理 3.1 假设矩阵 $A \in \mathbb{R}^{m \times n}$ 满足假设 2.2, S 和 W 如算法 KVV-NMF 中第 1 和 2 步所构造, 则 $\|A\|_F^2 = \|S\|_F^2 = \|W\|_F^2$.

证 由 $P_{i_t} = \|A(i_t, \cdot)\|^2 / \|A\|_F^2$, $S(t, \cdot) = A(i_t, \cdot) / \sqrt{p P_{i_t}}$, 有

$$\|S\|_F^2 = \sum_{t=1}^p \|S(t, \cdot)\|^2 = \sum_{t=1}^p \frac{\|A(i_t, \cdot)\|^2}{p P_{i_t}} = \sum_{t=1}^p \frac{\|A\|_F^2}{p} = \|A\|_F^2.$$

接下来证明 $\|S\|_F^2 = \|W\|_F^2$.

$$\begin{aligned} P'_j &= \sum_{t=1}^p \frac{D_{A(i_t, \cdot)}(j)}{p} = \sum_{t=1}^p \frac{|A(i_t, j)|^2}{p \|A(i_t, \cdot)\|^2} = \frac{1}{\|A\|_F^2} \sum_{t=1}^p \frac{\|A\|_F^2}{\|A(i_t, \cdot)\|^2} \frac{|A(i_t, j)|^2}{p} \\ &= \frac{1}{\|A\|_F^2} \sum_{t=1}^p |S(t, j)|^2 = \frac{\|S(\cdot, j)\|^2}{\|A\|_F^2} = \frac{\|S(\cdot, j)\|^2}{\|S\|_F^2}. \end{aligned}$$

由 $P'_{j_t} = \|S(\cdot, j_t)\|^2 / \|S\|_F^2$, $W(\cdot, t) = S(\cdot, j_t) / \sqrt{p P'_{j_t}}$, 有

$$\|W\|_F^2 = \sum_{t=1}^p \|W(\cdot, t)\|^2 = \sum_{t=1}^p \frac{\|S(\cdot, j_t)\|^2}{p P'_{j_t}} = \sum_{t=1}^p \frac{\|S\|_F^2}{p} = \|S\|_F^2.$$

由定理 3.1 可知, 在算法 FKV-NMF 的第 2 步中定义的 P'_j , 经过计算, 即为 $\|S(\cdot, j)\|^2 / \|S\|_F^2$, 故 W 的取法和 S 的取法本质上是一样的, 这也为数值实验带来了便利. 但这并不说明算法第 2 步中 P'_j 的取法是无意义的, 因为当矩阵 A 很大时, 为了节省内存, 在知道抽样后得到的行标 i_1, i_2, \dots, i_p 后, 可以通过行标来访问原矩阵 A 中元素, 从而间接得到 S 的信息, 而不是把 S 显式计算出来.

由算法 FKV-NMF 的流程可以看出, 本文提出的初始化算法 FKV-NMF 的数值计算时间大大减少的原因在于: 本文不是对原矩阵 A 直接进行奇异值分解, 而是通过抽样构造一个更小的矩阵 W , 对小矩阵 W 做奇异值分解, 这使得算法 FKV-NMF 在奇异值分解上花费的时间大大减少, 而现有的基于奇异值分解的初始化算法都是直接对原矩阵 A 进行奇异值分解, 这是本文提出的算法数值计算时间较其他算法减少的原因. 另外, 通过简单计算可以发现: 算法 FKV-NMF 中矩阵 \hat{V} 中的每一列可以近似看作矩阵 A 的右奇异向量, Frieze 等人^[9] 中也说明了这一点, 这也是算法 FKV-NMF 的理论依据.

当然, 算法 FKV-NMF 也有弊端: 此算法虽然在奇异值分解上节省了很多时间, 但是它还要计算用于抽样的概率分布 P 和 P' , 这是有点耗时的. 一般来说, 基于奇异值分解的初始化算法使用的都是截断奇异值分解, 比如前文提到的初始化算法 SVD-NMF 和 NNDSVD. 随着 k 值的减小, 截断奇异值分解所使用的时间也在不断减少, 当 k 小到某一程度时, 算法 FKV-NMF 在进行奇异值分解上节省的时间便不再占优势. 但是, 这也是算法 FKV-NMF 可以进一步改进的地方, 比如可以考虑使用其他更加快速有效的抽样方法.

4 数值实验

在本次实验中, 测试了两组数据: 第一组是随机生成的服从均值为 0、方差为 1 的高斯分布的随机数, 这里取绝对值来保证矩阵元素的非负性, 即 $A = |N(0, 1)|$, 维数是 500×300 . 第二组是人脸数据库 ORL, ORL 数据库一共包含 400 张 92×112 像素的图片, 一共 40 个人, 每人 10 幅不同表情的脸. 在计算不同算法的时间和误差的实验中, 将这 400 张人脸图像组成了一个 10304×400 的矩阵, 作为算法的输入. 在进行人脸重构实验时, 选取其中 360 张图像作为训练集, 40 张图像作为测试集.

在实验中, 将本文提出的算法 FKV-NMF 与两种比较常见的基于 SVD 的初始化算法 SVD-NMF 和 NNDSVD 作了比较. 非负矩阵分解算法采用的是经典的 MU 算法. 误差采用公式 $\text{error} = \|A - WH\|_F / \|A\|_F$.

在表 1 和表 2 中, 分别计算了在 k 等于不同值时, SVD-NMF、NNDSVD 和 FKV-NMF 三种不同的初始化算法所用的时间和所产生的误差. 另外, 由于本文提出的算法涉及到抽样技术, 所以算法效果会有稍许波动, 为公平起见, 实验重复 20 次, 取平均值.

表 1 三种不同的初始化算法所用的时间

time	Random (500×300)				ORL (10304×400)			
	$k = 15$	$k = 20$	$k = 25$	$k = 30$	$k = 25$	$k = 30$	$k = 35$	$k = 40$
SVD-NMF	0.41	0.45	0.54	0.57	6.31	7.27	8.24	8.93
NNDSVD	0.42	0.47	0.56	0.58	6.35	7.50	8.25	8.99
FKV-NMF	0.10	0.11	0.11	0.15	0.27	0.33	0.33	0.37

表 2 三种不同的初始化算法所产生的误差

error	Random (500×300)				ORL (10304×400)			
	$k = 15$	$k = 20$	$k = 25$	$k = 30$	$k = 25$	$k = 30$	$k = 35$	$k = 40$
SVD-NMF	0.81	0.94	1.08	1.22	0.77	0.84	0.89	0.95
NNDSVD	0.60	0.61	0.62	0.63	0.32	0.33	0.33	0.34
FKV-NMF	0.75	0.75	0.72	0.69	0.62	0.55	0.55	0.59

从表 1 和表 2 中, 可以看出: 就时间来方面说, 本文提出的 FKV-NMF 初始化算法所用的时间要比 SVD-NMF 和 NNDSVD 小的多. 就误差方面来说, 初始化误差比 NNDSVD 要大一些, 但是比 SVD-NMF 要小. 所以本文提出的初始化算法在大大节约了时间的同时, 误差也在可接受的范围内.

在图 1 中, 展现了 SVD-NMF、NNDSVD 和 FKV-NMF 三种不同初始化算法对于非负矩阵分解算法误差的影响. 图中曲线表示随着迭代次数增加, 误差的变化情况. 从图 1 中可以看

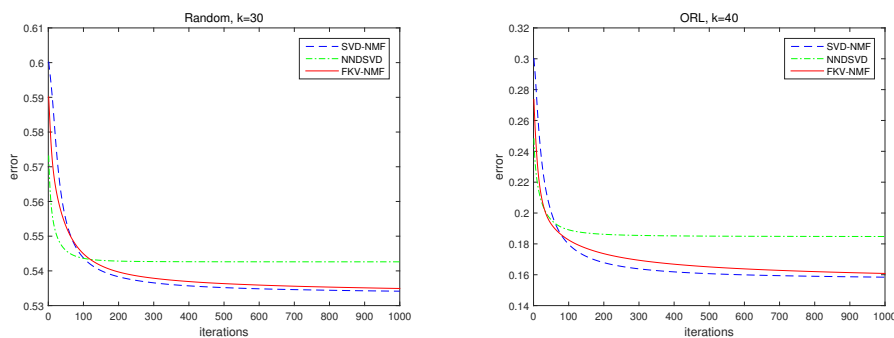


图 1 使用不同初始化算法后, 随着迭代次数增加, MU 算法误差变化情况

出, 在使用 SVD-NMF、NNDSVD 和 FKV-NMF 三种不同的初始化算法后, 使用 MU 算法进行非负矩阵分解时, 随着迭代次数的增加, 误差均在减少, 直到达到一个平稳值. NNDSVD 的初始化误差虽然最小, 但误差很快便不再下降. 在随机生成的数据 (Random) 上, 大约 100 次迭代后, 误差几乎不再下降, 在 ORL 数据集上, 大约 200 次迭代后, 误差几乎也不再下降, 而且最终误差比其他两种方法都要高的多. 本文提出的 FKV-NMF 虽然最终误差比 SVD-NMF

稍高,但是相差并不多.而且在迭代的最初始阶段,使用本文提出的FKV-NMF初始化方法的误差要比SVD-NMF低一些.

下面,展示了使用不同初始化算法后,MU算法迭代次数为1000时,ORL数据库部分人脸图像的重构情况.图2是人脸的重构图像,第1到第3行使用的初始化算法依次是SVD-NMF、NNDSVD和FKV-NMF.在表3中,计算了重构图像的信噪比(SNR),第1到第7列分别表示7个重构图像.信噪比是用来衡量图像质量的一种方式,信噪比越大,表示重构的人脸图像质量越好.这里信噪比的计算公式为: $SNR = 10 * \log_{10}(\sum_{i,j} g(i,j)^2 / \sum_{i,j} (f(i,j) - g(i,j))^2)$,其中 $g(i,j)$ 表示原始图像的灰度值, $f(i,j)$ 表示重构图像的灰度值.



图2 ORL数据集人脸的重构图像

表3 重构图像的信噪比(SNR)

No	1	2	3	4	5	6	7
SVD-NMF	14.14	16.73	16.28	13.14	16.18	16.05	18.07
NNDSVD	13.45	16.03	15.98	12.76	15.49	15.17	17.30
FKV-NMF	13.88	16.65	16.53	13.09	16.13	16.03	18.04

从图2和表3中,可以看出:从重构图像的清晰度来看,使用SVD-NMF和本文提出的FKV-NMF初始化算法图像清晰度相差无几,而使用NNDSVD初始化算法的重构图像效果则稍差.从SNR的值来看,使用SVD-NMF和本文提出的FKV-NMF初始化算法图像的SNR值相差并不多,而使用NNDSVD初始化算法的重构图像SNR值要明显小于其他两者.

通过以上数值实验,可以看到:本文提出的初始化算法FKV-NMF的运行时间较SVD-NMF和NNDSVD大大减少,也在一定程度上保持了精度.

参 考 文 献

- [1] Lee D D, Seung H S. Learning the parts of objects by non-negative matrix factorization [J]. Nature, 1999, 401(6755): 788-791.
- [2] Ding C, He X F, Simon H D. On the equivalence of nonnegative matrix factorization and spectral clustering [A]. Kargupta H, Srivastava J, Kamath C, Goodman A. Proceedings of the 2005 SIAM International Conference on Data Mining [C]. Philadelphia: SIAM, 2005: 606-610.

- [3] Wang Fei, Li Tao, Wang Xin, Zhu Shenghuo, Ding C. Community discovery using nonnegative matrix factorization [J]. *Data Mining and Knowledge Discovery*, 2011, 22(3): 493–521.
- [4] Lee D D, Seung H S. Algorithms for non-negative matrix factorization [A]. Leen T K, Dietterich T G, Tresp V. *Advances in Neural Information Processing Systems 13* [C]. Cambridge: MIT Press, 2001: 556–562.
- [5] Lin C J. Projected gradient methods for nonnegative matrix factorization [J]. *Neural Computation*, 2007, 19(10): 2756–2779.
- [6] Kim H, Park H. Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method [J]. *SIAM Journal on Matrix Analysis and Applications*, 2008, 30(2): 713–730.
- [7] Qiao H. New SVD based initialization strategy for non-negative matrix factorization [J]. *Pattern Recognition Letters*, 2015, 63: 71–77.
- [8] Boutsidis C, Gallopoulos E. SVD based initialization: a head start for nonnegative matrix factorization [J]. *Pattern Recognition*, 2008, 41(4): 1350–1362.
- [9] Frieze A, Kannan R, Vempala S. Fast Monte-Carlo algorithms for finding low-rank approximations [J]. *Journal of the ACM*, 2004, 51(6): 1025–1041.
- [10] Chia N H, Lin H H, Wang C H. Quantum-inspired sublinear classical algorithms for solving low-rank linear systems [J/OL]. <https://arxiv.org/abs/1811.04852>, 2018.

INITIALIZATION OF NONNEGATIVE MATRIX FACTORIZATION BASED ON MONTE CARLO METHOD

CHEN Hong-li

(School of Mathematics and Statistics, Wuhan University, Wuhan, 430072, China)

Abstract: The selection of initial values is crucial for nonnegative matrix factorization (NMF), because it significantly influences the effectiveness of NMF algorithms. Some initialization methods based on singular value decomposition (SVD) have been proposed^[7,8]. However, when the dimension of the matrix is very large, it is time-consuming to compute the SVD of original matrix directly. In this paper, we propose a more time-saving initialization method (KFV-NMF). Numerical experiments show that our initialization algorithm needs less time and the accuracy is also maintained to some extent.

Keywords: nonnegative matrix factorization; initialization; SVD; FKV

2010 MR Subject Classification: 97N30 ; 49M27