用于 k-means 聚类的降维方法研究

李纯羽　　樊金昊

2020 年 6 月 22 日

摘要

降维的方法包括特征选择和特征提取两种，本文使用 SVD 作为特征选择方法和随机投影作为特征提取方法，对 ORL 人脸数据集以及其他数据集（改）进行降维，并使用 k-means 对降维之后的结果进行聚类。我们比较了两种方法，并分别进行了一些优化。

目录

1 背景

聚类是大数据分析中的一种常用方法，从生物学的基因组分析到计算机科学中的社交网络，降维方法都有着广泛的应用 [13]。其中，最著名的就是 k-means 聚类算法，又称 Lloyd 算法。k-means 算法是一种迭代算法，其优化目标是使所有点到其对应的类中心的距离之和最小 [3]。由于其高度的有效性，k-means 算法获得了广泛的应用。

由于 k-means 算法需要大量进行距离计算，而后者的复杂度与特征的维度成正比，在实际的大数据分析中，高维的数据给 k-means 计算带来了巨大挑战。为了解决这些挑战，研究人员提出了**特征选择** (*feature selection*) 和**特征提取** (*feature extraction*) 等方法。其中，**特征选择**指的是选取所有特征的子集，在该子集上运行 k-means 聚类算法，而**特征提取**指的是根据原有特征构造出一些人造的特征，在这些特征上运行聚类算法。

在本文中，我们研究了一种基于随机投影的特征提取算法 (§ 2) 和一种基于特征值分解的特征选择算法 (§ 3)。我们通过理论推导和实验验证分别证明了以上两种方法的正确性。此外，针对这两种方法存在的问题，我们进行了分析并提出了我们的改进方案。对比实验证明我们的改进方案有明显的效果。

本文中算法的代码实现、实验结果和实验报告均在本项目的 GitHub 仓库中。

1.1 数学符号约定

2 基于随机投影的降维方法

2.1 背景和引入

随机投影属于特征提取方法的一种，所依据的原理是 JL 引理 [11]，即经过线性变换 f 之后原点集中两个点的距离仍然在一定误差范围内保持，这样的性质适合在 k-means 中使用。

K-means 的目标函数用矩阵表示为

$$X_{opt} = \arg\min_{X \in \chi} \|A - XX^\top A\|_F^2$$

其中 X 表示聚类的结果，当且仅当第 i 个点属于第 j 类的时候 X_{ij} 非零，并且 $X_{ij} = 1/\sqrt{z_j}$，其中 z_j 表示在对应类中的点的数量，这样就保证了后面一项表示各点所属的类类中心的位置。

在这个实验中我们使用 Lloyd 算法实现 k-means 聚类，虽然这个算法没有最差保证，但是通过初始点的选择保证聚类的效果不会很差。

2.2 算法描述和时间复杂度分析

2.2.1 算法描述

2.2.2 时间复杂度分析

Lloyd 算法一个循环的时间复杂度从 $O(knd)$ 降到了 $O(nk^2/\varepsilon^2)$. 使用 $\gamma = 1 + \varepsilon$ 近似的 k-means 算法, 对于 $\forall \varepsilon \in (0, 1/3)$, 时间复杂度为 $O(nd\lceil \varepsilon^{-2}k/log(d)\rceil + 2^{(k/\varepsilon)^{O(1)}}kn/\varepsilon^2)$.

Algorithm 1 k-means 聚类使用的随机投影算法

Input:

矩阵 $A \in R^{n\times d}$, 类的数目 k, 误差因子 $\varepsilon \in (0, 1/3)$, γ 近似的 k-means 算法.

Output:

矩阵 $X_{\tilde{\gamma}}$ 表示聚类结果

1: 设置目标维度 $t = \Omega(k/\varepsilon^2)$, 对足够大的常数 c, $t = t_0 \geq ck/\varepsilon^2$;

2: 产生随机矩阵 $R \in R^{d\times t}$, 对于所有 $i \in [d], j \in [t]$

$$R_{ij} = \begin{cases} +1/\sqrt{t}, & \text{w.p. } 1/2, \\ -1/\sqrt{t}, & \text{w.p. } 1/2. \end{cases}$$

3: 计算矩阵 $\tilde{A} = AR$.

4: 在矩阵 $\tilde{A}$ 上运行 γ 近似的 k-means 算法得到 $X_{\tilde{\gamma}}$; 返回矩阵 $X_{\tilde{\gamma}}$

2.2.3 算法的近似比

假设上述算法使用 γ 近似的 k-means 算法, 则有

$$\|A - X_{\tilde{\gamma}} X_{\tilde{\gamma}}^\top A\|_F^2 \leq (1 + (1+\varepsilon)\gamma)\|A - X_{opt} X_{opt}^\top A\|_F^2$$

2.3 算法近似比的数学证明

下面简单的证明上面关于近似比的结论

证明. $A = A_k + A_{\rho-k}$

其中, ρ 代表 A 的秩, $A_k = U_k \sum_k V_k, U_k, V_k, \sum_k$ 分别是由前 k 个左右奇异向量和奇异值组成的矩阵. 由正交性

$$\begin{aligned} &\|A - X_{\tilde{\gamma}} X_{\tilde{\gamma}}^\top A\|_F^2 = \|(I - X_{\tilde{\gamma}} X_{\tilde{\gamma}}^\top)A_k\|_F^2 + \|(I - X_{\tilde{\gamma}} X_{\tilde{\gamma}}^\top)A_k\|_F^2 \\ &\because (I - X_{\tilde{\gamma}} X_{\tilde{\gamma}}^\top) \text{ 为投影矩阵} \\ &\therefore \|(I - X_{\tilde{\gamma}} X_{\tilde{\gamma}}^\top)A_{\rho-k}\|_F^2 \leq \|A_{\rho-k}\|_F^2 = \|A - A_k\|_F^2 \leq \|A - X_{opt} X_{opt}^\top A\|_F^2 \end{aligned}$$

R 为算法产生的随机投影矩阵, $E = A_k - (AR)(V_k^\top R)^\dagger V_k^\top$, 并且可以证明 $||E||_F \leq 4\varepsilon\, ||A - A_k||_F$,

则

$$
\begin{aligned}
&\|(I - X_{\tilde{\gamma}}X_{\tilde{\gamma}}^\top)A_k\|_F \\
&\leq \|(I - X_{\tilde{\gamma}}X_{\tilde{\gamma}}^\top)AR(V_k^\top R)^\dagger V_k^\top\|_F + \|E\|_F \\
&\leq \|(I - X_{\tilde{\gamma}}X_{\tilde{\gamma}}^\top)AR\|_F\|(V_k^\top R)^\dagger V_k^\top\|_2 + \|E\|_F \\
&\leq \|(I - X_{\tilde{\gamma}}X_{\tilde{\gamma}}^\top)AR\|_F\|(V_k^\top R)^\dagger\|_2 + \|E\|_F \\
&\because \|(I - X_{\tilde{\gamma}}X_{\tilde{\gamma}}^\top)AR\|_F^2 \leq \gamma \min_{X\in\chi}\|(I - XX^\top)AR\|_F^2 \leq \gamma\|(I - X_{opt}X_{opt}^\top)AR\|_F^2 \\
&\therefore \|(I - X_{\tilde{\gamma}}X_{\tilde{\gamma}}^\top)AR\|_F\|(V_k^\top R)^\dagger\|_2 + \|E\|_F \\
&\leq \sqrt{\gamma}\|(I - X_{opt}X_{opt}^\top)AR\|_F\|(V_k^\top R)^\dagger\|_2 + \|E\|_F \\
&\leq \sqrt{\gamma}\|(I - X_{opt}X_{opt}^\top)A\|_F\sqrt{1+\varepsilon}\|(V_k^\top R)^\dagger\|_2 + \|E\|_F \\
&\because \forall i \in [k]|1 - \sigma_i(V_k^\top R)| \leq \varepsilon\text{其中}\sigma_i\text{表示 A 的第 i 个非负奇异值} \\
&\therefore \|(V_k^\top R)^\dagger\|_2 \leq \frac{1}{1-\varepsilon} \\
&\therefore \sqrt{\gamma}\|(I - X_{opt}X_{opt}^\top)A\|_F\sqrt{1+\varepsilon}\|(V_k^\top R)^\dagger\|_2 + \|E\|_F \\
&\leq \sqrt{\gamma} \times \frac{\sqrt{1+\varepsilon}}{1-\varepsilon}\|(I - X_{opt}X_{opt}^\top)A\|_F + 4\varepsilon\|(I - X_{opt}X_{opt}^\top)A\|_F \\
&\leq \sqrt{\gamma}(1 + 6.5\varepsilon)\|(I - X_{opt}X_{opt}^\top)A\|_F
\end{aligned}
$$

□

2.4 实验复现

实验使用 ORL 人脸数据集，这个数据集包含 40 类，每一类包含 10 张人脸图片，因为刚开始没有找到分辨率 64*64 的数据，使用的是分辨率 112 *92 的数据集。由于数据按照类顺序排列，使用 Lloyd 算法时取每一类的第一张图片作为初始类中心。

选取三个评价指标，分别是

- 归一化的目标函数，表示 k-means 收敛的情况
- 分类的准确率, 使用正确分类的图片数除以总数
- 从算法第一步开始到结束总的执行时间

初始维度是 10304 维，将目标维度 t 从 5 增长到 300，计算三个评价指标的结果如1,2,3

可以看出随着 t 的增长除了执行时间一直在增长之外，目标函数和准确率的变化趋势逐渐减小，从这次实验的结果看大概在维度是 130 的时候，目标函数和准确率可以达到几乎最优的程度，再继续增加 t 结果变化程度比较小，同时在这个维度执行时间也比较短。

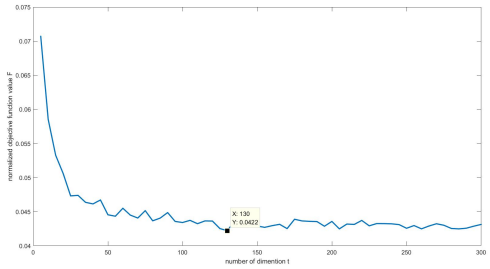

图 1: 归一化的目标函数

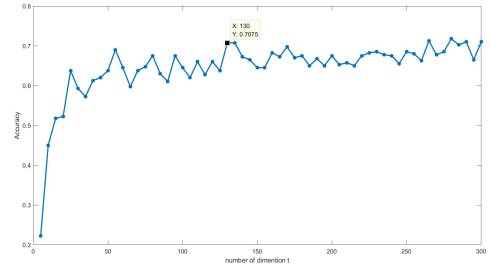

图 2: 分类的准确率

2.5 思考与改进

因为这种基于随机投影的降维方法可解释性不是特别强，我们暂时没有想到怎样从精度上做改进，下面的改进思路主要都是在速度上的
主要的三个改进思路分别是

1. 构造新的随机投影矩阵
2. 使用 mailman 算法加速矩阵向量乘法
3. 多次实验取最好的结果

2.5.1 构造新的随机投影矩阵

下面除了论文中随机投影矩阵的构造方法之外，另外使用三种随机投影矩阵的构造方法

- $\forall i,j \quad R_{i,j} \sim N(0,1)$
- 在本实验随机矩阵的基础上，使用稀疏矩阵，满足

$$R_{ij} = \begin{cases} +\sqrt{3}/\sqrt{t}, & \text{w.p. } 1/6, \\ 0, & \text{w.p. } 2/3, \\ -\sqrt{3}/\sqrt{t}, & \text{w.p. } 1/6. \end{cases}$$

- Fast JL 变换
  构造 $\Phi = PHD$, 其中 $P \in R^{k\times d}, H, D \in R^{d\times d}$

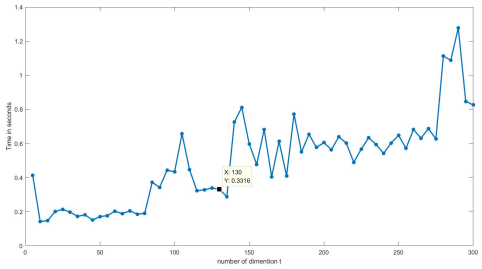

图 3: 算法运行的总时间

P 的构造如下:

$$P_{ij} = \begin{cases} N(0, 1/q), & \text{w.p. q,} \\ 0, & \text{w.p. 1-q} \end{cases} \quad q = min\Big\{\theta\Big(\frac{log^2 n}{d}\Big), 1\Big\}$$

在本实验中 q 的取值大约为 0.07 左右, 可以看出 P 非常稀疏,H 是归一化的 Hadamard 矩阵,D 是对角矩阵，对角线元素以 1/2 概率取-1 或 1.

理论上 FJLT 的方法可以达到 $\theta(nd/\varepsilon^2)$ 的时间复杂度，这里要求 d 是二的幂次，所以使用了 64*64 的 ORL 数据集

首先比较一下算法各个部分所用的时间，结果如4.

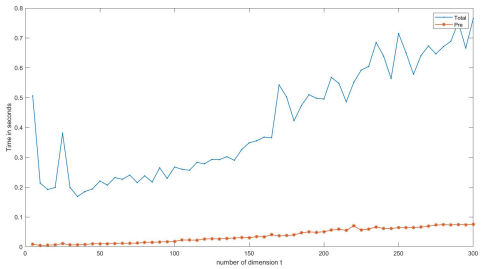

图 4: 算法总时间和预处理时间

这里把除了 k-means 算法之外的计算时间称为预处理时间，可以看到 k-means 算法的时间是整个算法主要的时间开销，所以不再单独比较四种 JL 变换的预处理时间

比较四种随机投影矩阵的分类准确率，结果如5.

考虑随机性等原因，我们认为使用这四种随机投影矩阵进行降维并用在 k-means 中对于分类准确率影响不大.

2.5.2 使用 mailman 算法加速矩阵向量乘法

当矩阵 A 的每个元素只取固定的值时，mailman 算法能够把矩阵向量乘法 $A\vec{x}$ 的时间复杂度从 $O(mn)$ 降到 $O(mn/log(maxm, n))$ Mailman 算法的核心思想是矩阵向量乘法 $A\vec{x}$, 其中

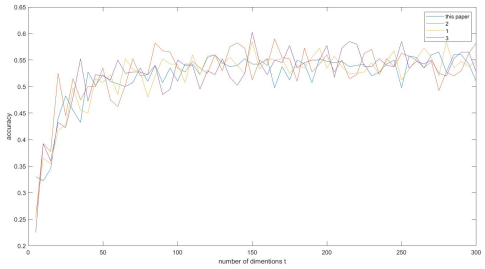

图 5: 四种随机投影矩阵的分类准确率比较

$A \in R^{m\times n}, \vec{x} \in R^{n\times 1}$ 可以分解成以下形式

$$A\vec{x} = \sum_{i=1}^{n} A^{(i)}x^{(i)}$$

其中，$A^{(i)}$ 表示 A 的第 i 列，$x^{(i)}$ 表示 x 的第 i 个元素，如果想象成是邮递员在送信，$A^{(i)}$ 就是地址，$x^{(i)}$ 是信，整个矩阵向量乘法就是邮递员遍历不同的地址送不同的信，但是因为地址是确定的，原始的乘法只是机械地走完全程，并没有考虑地址相同的情况。

下面简单描述一下如何使用 mailman 算法计算矩阵向量乘法，这里不妨假设 $m = log_2^n$，并且 $A(i,j) \in 0,1$, 如果不是可以把矩阵分成子矩阵或再进行扩展. 构造两个新矩阵 U_n 和 $n \times n$ 的矩阵 P，U_n 的每一列存储可能出现的由 m 个 0,1 组成的向量，那么 A 的每一列在 U_n 中都会出现。$P(i,j) = \delta(U^{(i)}, A^{(j)}) \quad \delta = 1 \quad \text{if} \quad U^{(i)} = A^{(j)}$, 所以 P 只有 n 个非零元。

$$\begin{aligned}
\because (U_nP)(i,j) &= \sum_{k=1}^{n} U_n(i,k)P(k,j) \\
&= \sum_{k=1}^{n} U_n^{(k)}(i)\delta(U_n^{(k)}, A^{(j)}) \\
&= A(i,j) \\
\therefore A\vec{x} &= (UP)\vec{x} = U(P\vec{x})
\end{aligned}$$

因为 P 只有 n 个非零元，计算 $P\vec{x}$ 的时间复杂度为 $O(n)$, 利用二的幂次的性质，递归计算 $U(P\vec{x})$ 的时间复杂度也为 $O(n)$, 更一般的情况下，在使用随机投影降维的算法中，理论上时间复杂度可以降低 $max\{n,d\}$，但是不论是在 mailman 算法的原始文献中还是在随机投影降维的文章中，作者都表示实际无法达到理论效果，所以并没有对该算法进一步尝试。

3 基于 SVD 的降维方法

4 结论

参考文献

[1] Nir Ailon and Bernard Chazelle. Faster dimension reduction. *Commun. ACM*, 53(2):97–104, February 2010.

[2] Salem Alelyani, Jiliang Tang, and Huan Liu. Feature selection for clustering: A review. In *Data Clustering*, pages 29–60. Chapman and Hall/CRC, 2018.

[3] Avrim Blum, John Hopcroft, and Ravindran Kannan. *Foundations of data science*. Cambridge University Press, 2020.

[4] Christos Boutsidis, Petros Drineas, and Michael W Mahoney. Unsupervised feature selection for the k-means clustering problem. In *Advances in Neural Information Processing Systems*, pages 153–161, 2009.

[5] Christos Boutsidis and Malik Magdon-Ismail. Deterministic feature selection for k-means clustering. *IEEE Transactions on Information Theory*, 59(9):6099–6110, 2013.

[6] Christos Boutsidis, Anastasios Zouzias, and Petros Drineas. Random projections for k-means clustering. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 1*, NIPS' 10, page 298–306, Red Hook, NY, USA, 2010. Curran Associates Inc.

[7] Christos Boutsidis, Anastasios Zouzias, Michael W Mahoney, and Petros Drineas. Randomized dimensionality reduction for k-means clustering. *IEEE Transactions on Information Theory*, 61(2):1045–1062, 2014.

[8] Petros Drineas, Alan M Frieze, Ravi Kannan, Santosh Vempala, and V Vinay. Clustering in large graphs and matrices. In *SODA*, volume 99, pages 291–299. Citeseer, 1999.

[9] Petros Drineas, Malik Magdon-Ismail, Michael W Mahoney, and David P Woodruff. Fast approximation of matrix coherence and statistical leverage. *Journal of Machine Learning Research*, 13(Dec):3475–3506, 2012.

[10] Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

[11] William B Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26(189-206):1, 1984.

[12] Edo Liberty and Steven W. Zucker. The mailman algorithm: A note on matrix–vector multiplication. *Inf. Process. Lett.*, 109(3):179–182, January 2009.

[13] Michael W. Mahoney and Petros Drineas. Cur matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences*, 106(3):697–702, 2009.

[14] Alessandro Rudi, Daniele Calandriello, Luigi Carratino, and Lorenzo Rosasco. On fast leverage score sampling and optimal learning. In *Advances in Neural Information Processing Systems*, pages 5672–5682, 2018.

[15] Gilbert Strang. *Linear algebra and learning from data.* Wellesley-Cambridge Press, 2019.

[16] Arthur Szlam, Yuval Kluger, and Mark Tygert. An implementation of a randomized algorithm for principal component analysis. *arXiv preprint arXiv:1412.3510*, 2014.