用于k-means聚类的降维方法研究

李纯羽　　樊金昊

2020 年 6 月 27 日

摘要

降维的方法包括特征选择和特征提取两种，本文使用基于特征值选择的特征选择方法和基于随机投影的特征提取方法，对ORL人脸数据集以及NIPS数据集进行降维，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 [15]。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。

方法	维度	时间复杂度	近似比
特征提取	$O(k/\epsilon^2)$	$nd[\epsilon^{-2}k/\log(n)]$	$2+\epsilon$
特征选择	$\Theta(k\log(k/\epsilon)/\epsilon^2)$	$min(nd^2, n^2d)$	$2+\epsilon$

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

由于下文涉及大量数学推导，因此在此给出其中使用的数学符号的约定：

- 对于$n \times d$数据矩阵A，约定$U_k \in R^{n\times k}$，$V_k \in R^{k\times d}$为前k个左（右）特征向量，$\Sigma_k \in R^{k\times k}$为 A的前k个特征向量组成的对角矩阵。
- 约定ρ为A的秩，则$A_{\rho-k}$等于$A - A_k$，其中$A_k = U_k\Sigma_k V_k$。
- 约定$\|A\|_F$和$\|A\|_2$为A的Frobenius范数和频谱范数。矩阵范数满足如下性质：$\|XY\|_F \leq \|X\|_F\|Y\|_2$且 $\|XY\|_F \leq \|X\|_2\|Y\|_F$。
- 称A^+为A的伪逆，$\|A^+\| = \sigma_{max}(A^+) = 1/\sigma_{min}(A)$，其中$\sigma_{max}$和$\sigma_{min}$分别代表矩阵最大和最小的奇异值。
- 对于一个方阵P，若其满足$P^2 = P$，则称其为投影矩阵。一个显而易见的结论是，若P为投影矩阵，则$I - P$也是投影矩阵。另外,对任意矩阵A，$\|PA\|_F \leq \|A\|_F$。

1.2 k-means目标函数的形式化

有了以上数学符号，我们可以从线性代数的角度给出k-means目标函数的定义：

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

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XX^TA表示各点所属的类中心的位置，$\|A - XX^TA\|_F^2$就是各点到类中心距离的平方和。

在下文中，我们都将从以上角度分析k-means聚类问题。事实上，不论是运用特征选择算法还是特征提取算法，降维后的矩阵中的列都是A中的列的线性组合，因此，降维后的矩阵都可以用$C = AD$来表示。 [8]的结果表明，对A的低秩估计可以被用来在k-means聚类时进行降维。我们可以通过$C = AD$得到A的低秩估计，因此下文的主要思路就是设法得到在k-means聚类时误差足够小的“好的估计”。

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 [12]，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。

在实验中，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

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$X_{\tilde{\gamma}}$

2.2.2 时间复杂度分析

使用随机投影降维之后，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$O(nd\lceil\varepsilon^{-2}k/log(d)\rceil+2^{(k/\varepsilon)^{O(1)}}kn/\varepsilon^2)$.

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

在本节简单证明上面关于近似比的结论

证明.

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

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，在此实验中我们使用分辨率112 *92 的数据集。由于数据按照类顺序排列，Lloyd算法初始化时我们取每一

类的第一张图片作为初始类中心。

在这里对于K-means结果，我们选取三个评价指标，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

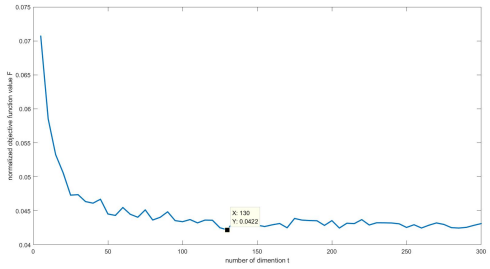


图 1: 归一化的目标函数

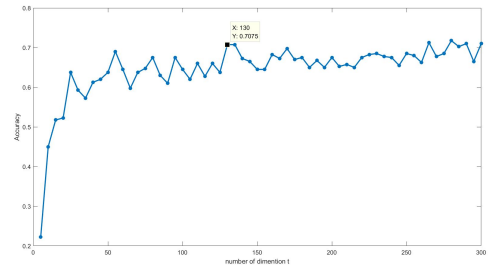


图 2: 分类的准确率

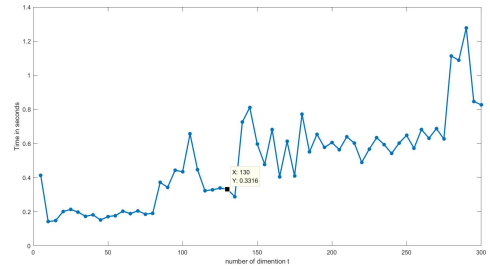


图 3: 算法运行的总时间

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。

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[2]

- $\forall i,j \quad R_{i,j} \sim N(0,1)$
- 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$P \in R^{k\times d}, H, D \in R^{d\times d}$
  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.
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

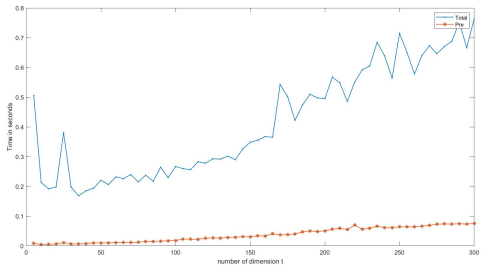


图 4: 算法总时间和预处理时间

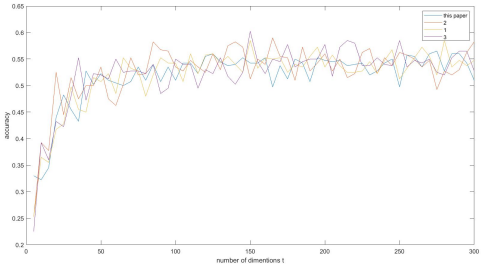


图 5: 四种随机投影矩阵的分类准确率比较

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 [13]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$A \in R^{m\times n}, \vec{x} \in R^{n\times 1}$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。

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

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的特征选择

尽管特征提取方法可以得到非常好的降维效果，但由于该方法得到的特征都是人为构造的，因而可解释性略差。相比之下，特征选择算法直接从原矩阵中抽取一些特征，我们可以清晰地看到哪些特征出现在了降维后的矩阵里（§ 3.3）, 可解释性非常好。

事实上，SVD和k-means聚类有着十分紧密的联系。我们知道，X_{opt}的秩至多为k，而A_k是A的最佳秩为k的估计，因此一个有趣的结论如下：

$$\|A_{\rho-k}\|_F^2 = \|A - A_k\|_F^2 \leq \|A - X_{opt}X_{opt}^T A\|_F^2$$

在 [8]中，作者证明了将A投影到其最佳k维子空间得到的$n \times k$矩阵$\tilde{A} = U_k\Sigma_k$后进行k-means聚类，可以得到一个最优聚类的2-approximation：

$$\|A - \tilde{X}_{opt}\tilde{X}_{opt}^T A\|_F^2 \leq 2 \min \|A - XX^T A\|_F^2$$

这实际上是本小节中介绍的算法的出发点。在本算法中，我们将以上k个构造的特征替换为$\Theta(k\log(k/\epsilon)/\epsilon^2)$个实际的特征，得到一个$(2+\epsilon)$-approximate的结果。

3.1 算法概述

可以看到，该算法是通过随机采样的方式进行特征选择的。特别地，以上方法称为**杠杆得分采样**（*leverage score sampling*）。杠杆得分是在该算法第3步计算出的，它代表了特征的“重要性”，在 [15]中有对其清晰的解释：

根据SVD分解，我们可以将A的第j列表示为左奇异向量的线性组合：$A^j = \sum_{i=1}^{r}(\sigma_i u^i)v_j^i$，其中$v_j^i$表示第j个右奇异向量的第i个元素。我们也可以用前k个左奇异向量估计$A^j = \sum_{i=1}^{k}(\sigma_i u^i)v_j^i$。因此，特征的重要性可以根据其在前k个左奇异向量组成的线性空间中的长度决定，即： $\pi_j = \frac{1}{k}\sum_{i=1}^{k}(v_j^i)^2$，此处除以k是为了归一化。

Algorithm 2 用于k-means聚类的特征选择算法

Input:

矩阵 $A \in R^{n\times d}$, 类的数目k, 误差因子 $\varepsilon \in (0,1)$, 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 $r = \Theta(k\log(k/\epsilon)/\epsilon^2)$。

2: 计算前k个右奇异向量V_k。

3: 计算**杠杆得分**p_i，对于$i = 1,\ldots,d$，$p_i = \|(V_k)_{(i)}\|_2^2/k$。

4: 进行r次有放回的采样，每一次以p_i的概率保留第i个特征，并将其乘以系数$(rp_i)^{-1/2}$。

5: 在上一步产生的$n \times r$矩阵$\tilde{A}$上运行γ近似的k-means算法，返回矩阵$X_{\tilde{\gamma}}$。

可以证明，以上算法运行于近似比为γ，失败率为δ_γ的k-means聚类算法上时，可以以至少$0.5-\delta_\gamma$的概率得到$(1+(1+\epsilon)\gamma)$近似的聚类结果。降维算法中最复杂的运算为求前k个右奇异向量，因此时间复杂度为$O(\min(nd^2, n^2d))$。

3.2 理论推导

本小节对以上特征选择算法的PAC-界给出理论推导 [4,7]。

定理 1. *按照算法 2的得到的聚类结果$X_{\tilde{\gamma}}$，以至少 $0.5-\delta_\gamma$的概率满足：*

$$\|A - X_{\tilde{\gamma}}X_{\tilde{\gamma}^T}A\|_F^2 \le (1+(1+\epsilon)\gamma)\|A - X_{opt}X_{opt}^TA\|_F^2$$

为了证明以上定理，先给出以下两个概念：

采样矩阵S是一个$d\times r$矩阵，它是这样构造出的：在算法 2的有放回采样过程中，每次若第i个特征被选中，则将e_i（第i项为1，其它均为0的列向量）加入到矩阵S中都会出现。

缩放矩阵D是一个$r\times r$对角矩阵，在算法 2采样的过程中，若第t次采样，第i个特征被选中，则将第t列的对角项设为$1/\sqrt{rp_i}$。

有了以上两个定义，下面给出一个引理 [4]：

引理 1. *若r满足算法 2中的条件，则以下四个语句以至少0.5的概率同时成立：*

1. $\|V_k^TSD\|_2 = \sigma_{max}(V_k^TSD) \le \sqrt{1+\lambda}$。
2. $\|(V_k^TSD)^+\|_2 = 1/\sigma_{min}(V_k^TSD) \le \sqrt{1/(1-\lambda)}$。
3. V_k^TSD满秩。
4. $A_k = (ASD)(V_k^TSD)^+V_k^T + E$，其中$\|E\|_F \le \mu\|A-A_k\|_F$。

其中，λ和μ为给定的常数 [4]。

证明. 首先证明前两条：

定义随机向量$y \in R^k$如下：对于$i=1,\ldots,d$，$\mathrm{P}[y=y_i]=p_i$，其中$y_i = (1/\sqrt{p_i})\left(V_k^T\right)^{(i)}$。那么， S和D的定义表明：$V_k^TSDDS^TV_k = \frac{1}{r}\sum_{i=1}^d y_iy_i^T$， $p_i = \left\|\left(V_k^T\right)^{(i)}\right\|_2/k$使得$\|y\|_2 \le \sqrt{k}$.

又有$E\left[yy^T\right]=\sum_{i=1}^{d}p_i\frac{1}{\sqrt{p_i}}\left(V_k^T\right)^{(i)}\frac{1}{\sqrt{p_i}}\left(V_k^T\right)^{(i)}\Big)^T=V_k^TV_k=I_k$。运用 [16]中的定理3.1，结合Markov不等式，得到对于足够大的c_0，以至少$1-1/6$的概率：

$$\left\|V_k^TSDDS^TV_k-I_k\right\|_2\leq 6c_0\sqrt{k\log(r)/r}$$

根据矩阵摄动理论 [4]，有：

$$\left\|V_k^TSDDS^TV_k-I_k\right\|_2=\left|\sigma_i^2\left(V_k^TSD\right)-1\right|\leq 6c_o\sqrt{k\log(r)/r}$$

前两条因此得证。为了证明第三条，我们只需证明V_k^TSD的第k个奇异值为正。事实上，ϵ的取值和引理的第二条已经保证了这一结果。

为了证明第四条，我们首先有：

$$\begin{aligned}\left\|A_k-ASD\left(V_k^TSD\right)^+V_k^T\right\|_F&=\left\|A_k-A_kSD\left(V_k^TSD\right)^+V_k^T-A_{\rho-k}SD\left(V_k^TSD\right)^+V_k^T\right\|_F\\&\leq\left\|A_k-A_kSD\left(V_k^TSD\right)^+V_k^T\right\|_F+\left\|A_{\rho-k}SD\left(V_k^TSD\right)^+V_k^T\right\|_F\end{aligned}$$

以上第一步，我们将A替换为$A_k+A_{\rho-k}$，第二步运用了矩阵范数的三角不等式。我们将第一项设为θ_1，第二项为θ_2，则有：

$$\begin{aligned}\theta_1&=\left\|A_k-U_k\Sigma_kV_k^TSD\left(V_k^TSD\right)^+V_k^T\right\|_F\\&=\left\|A_k-U_k\Sigma_kI_kV_k^T\right\|_F=0\end{aligned}$$

$$\begin{aligned}\theta_2&=\left\|U_{\rho-k}\Sigma_{\rho-k}V_{\rho-k}^TSD\left(V_k^TSD\right)^+V_k^T\right\|_F\\&\leq\left\|\Sigma_{\rho-k}V_{\rho-k}^TSD\left(V_k^TSD\right)^+\right\|_F\end{aligned}$$

对于后一项，如果前三项以至少$1-1/6$的概率成立，则以至少$1-1/3$的概率有：

$$\left\|\Sigma_{\rho-k}V_{\rho-k}^TSD\left(V_k^TSD\right)^+\right\|_F\leq\left(\epsilon\sqrt{6/\left(2c_1c_o^2\log\left(c_1c_o^2k/\epsilon^2\right)\right)}+\sqrt{6\lambda^2/(1-\lambda)}\right)\|A-A_k\|_F$$

因此，根据union bound，所有四条引理以至少$1-1/6-1/3=1-1/2$的概率成立。

证毕。 □

有了以上引理，我们可以对定理 1给出证明：

证明. 首先，我们将A分为$A=A_k+A_{\rho-k}$，再运用矩阵范数勾股定理 [7]，则有：

$$\left\|A-X_{\tilde{\gamma}}X_{\tilde{\gamma}}^TA\right\|_F^2=\left\|\left(I-X_{\tilde{\gamma}}X_{\tilde{\gamma}}^T\right)A_k\right\|_F^2+\left\|\left(I-X_{\tilde{\gamma}}X_{\tilde{\gamma}}^T\right)A_{\rho-k}\right\|_F^2$$

与引理的证明相同，我们将第一项称为θ_3^2，第二项称为θ_4^2。对于第二项，由于$I-X_{\tilde{\gamma}}X_{\tilde{\gamma}}^T$是投影矩阵，去掉它不会使范数增大，再结合本章开头的结论，有：

$$\theta_4^2\leq F_{opt}$$

下面讨论第一项的界：

$$\begin{aligned}
\theta_3 &\le \left\| \left(I - X_{\tilde{\gamma}} X_{\tilde{\gamma}}^T\right) ASD \left(V_k SD\right)^+ V_k^T \right\|_F + \|E\|_F \\
&\le \left\| \left(I - X_{\tilde{\gamma}} X_{\tilde{\gamma}}^T\right) ASD \right\|_F \left\| \left(V_k SD\right)^+ \right\|_2 + \|E\|_F \\
&\le \sqrt{\gamma} \left\| \left(I - X_{opt} X_{opt}^T\right) ASD \right\|_F \left\| \left(V_k SD\right)^+ \right\|_2 + \|E\|_F \\
&\le \sqrt{\gamma} \left\| \left(I - X_{opt} X_{opt}^T\right) ASD \left(V_k SD\right)^+ \right\|_F \left\| \left(V_k SD\right) \right\|_2 \left\| \left(V_k SD\right)^+ \right\|_2 + \|E\|_F \\
&= \sqrt{\gamma} \left\| \left(I - X_{opt} X_{opt}^T\right) ASD \left(V_k SD\right)^+ V_k^T \right\|_F \left\| \left(V_k SD\right) \right\|_2 \left\| \left(V_k SD\right)^+ \right\|_2 + \|E\|_F
\end{aligned}$$

其中，第一步我们使用了引理1、三角不等式和投影矩阵的性质，第二步我们使用了数学符号约定（§ 1.1）里的结果，第三步是基于以下结论：

$$\left\| \left(I - X_{\tilde{\gamma}} X_{\tilde{\gamma}}^T\right) ASD \right\|_F^2 \le \gamma \min_{X \in \mathcal{X}} \left\| \left(I - X X^T\right) ASD \right\|_F^2 \le \gamma \left\| \left(I - X_{opt} X_{opt}^T\right) ASD \right\|_F^2$$

在第四步和第五步，我们分别加入了一些不影响矩阵范数的项。

我们设$\theta_5 = \left\| \left(I - X_{opt} X_{opt}^T\right) ASD \left(V_k SD\right)^+ V_k^T \right\|_F$，则可以有如下对其的讨论：

$$\begin{aligned}
\theta_5 &\le \left\| \left(I - X_{opt} X_{opt}^T\right) A_k \right\|_F + \left\| \left(I - X_{opt} X_{opt}^T\right) E \right\|_F \\
&\le \left\| \left(I - X_{opt} X_{opt}^T\right) A V_k V_k^T \right\|_F + \|E\|_F \\
&\le (1+\mu)\sqrt{F_{opt}}
\end{aligned}$$

在第一步，我们运用了引理1和三角不等式，第二步我们将前一项的A_k展开，后一项去掉了投影矩阵，第三步我们去掉了投影矩阵并使用了本章开头的结论。将上述结论带回θ_3，则：

$$\theta_3 \le \sqrt{\gamma}\left(\sqrt{\frac{1+\lambda}{1-\lambda}}(1+\mu)+\mu\right)\sqrt{F_{opt}}$$

易知，括号内的部分小于$\sqrt{1+\lambda}$，由此，我们得到：

$$\theta_3^2 \le \gamma(1+\epsilon)F_{opt}$$

由于定理 1失效当且仅当引理1或近似k-means算法失效，两者概率至多为$0.5+\delta_\gamma$。综上，定理得证。 □

3.3 实验验证

我们首先在原文所使用的NIPS数据集 [10]上进行了验证。数据集为一个183×6314的文本-词项矩阵，包含了2001-2003年发表在NIPS上的文章，分为神经科学、学习理论、控制和强化学习三类。 图 6和图 7表示了我们的实验结果。

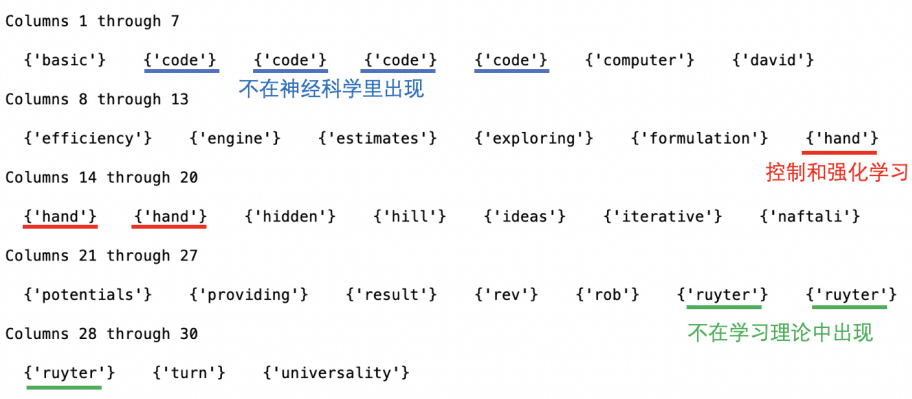


图 6: 算法 2选出的特征

从图 6可以看出，有一些特征被重复选了多次，根据领域知识可以知道，这些特征恰恰是不同领域中具有代表性的词语，这说明我们的选择算法是有效的。此外，我们可以直观地看到算法选出的特征，也说明了特征选择算法良好的可解释性[§ 3]。

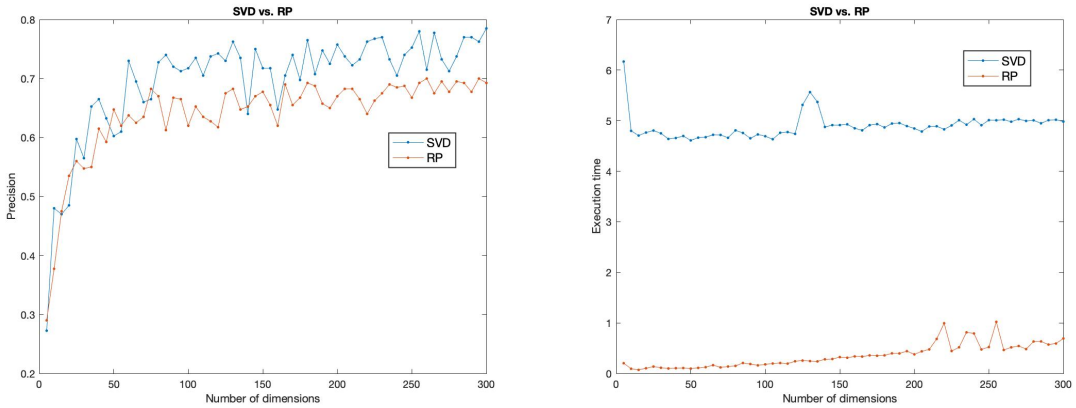
	$r = 5k$	$r = 10k$	$r = 20k$	不降维
P	0.7826	0.8098	0.8207	0.8750
F	0.7574	0.7539	0.7524	0.7472

图 7: 算法 2的分类正确率和目标函数值

从图 7可以看出，降维后聚类的正确率与不降维的差距不大。事实上，尽管我们的算法要求目标维度$r = \Theta(k \log(k/\epsilon)/\epsilon^2)$，但是在实验中选择远小于该值的$r$就可以良好地完成降维任务。

3.3.1 两种方法的比较

我们还在ORL数据集上对特征选择和特征提取算法的正确率和执行时间进行了比较：



(a) 正确率 (b) 执行时间

图 8: 两种降维算法的比较

比较的结果非常直观，相对来讲，基于SVD的特征选择算法比基于随机投影的特征提取算法正确率稍高，但执行时间则长了很多。这是由于SVD运算的时间复杂度是三次方量级。事实上，在实际的大数据计算中，我们难以承受对巨大的矩阵进行SVD分解的时间和空间开销[3, 11, 11]。

3.4 思考与改进

在本小节中，针对上一小节实验中发现的SVD执行速度慢的问题，我们利用一种近似SVD算法进行改进[§ 3.4.1]，我们还将就杠杆得分函数的有效性进行讨论[§ 3.4.2]，最后，我们将我们的降维方法与确定性的 PCA方法进行比较[§ 3.4.3]。

3.4.1 用近似算法加速SVD计算

在§ 3.3中，我们发现SVD计算非常费时，影响了特征选择算法的执行速度。事实上，学术界对加速SVD计算的方法已经有了深入的研究 [11]，在本小节中，我们介绍一种随机算法来加速SVD计算 [19]，算法 3概述其计算步骤。

我们利用算法 3替换算法 2中计算前k个右奇异向量的过程，可以看到，它的执行时间与随机投影算法已经基本相同，主要受限于k-means的执行速度，而它的准确率却几乎没有下降。

Algorithm 3 近似SVD算法

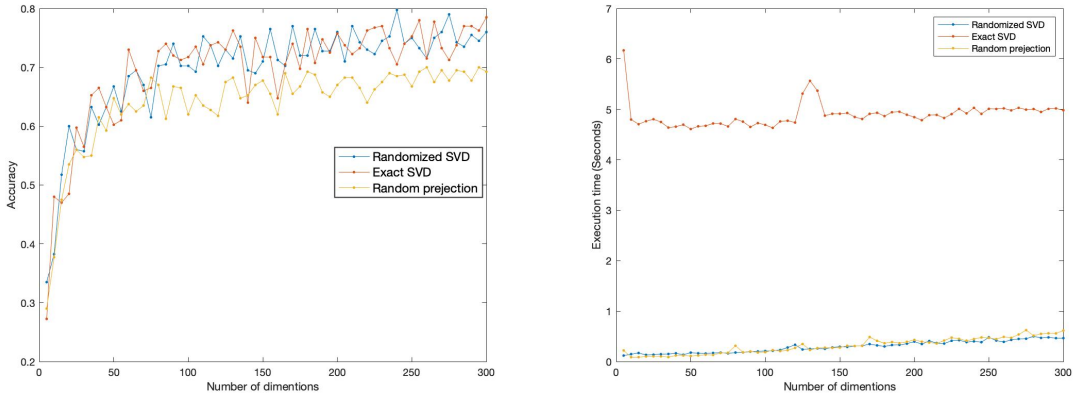
Input:

矩阵 $A \in R^{n \times d}$, 参数k。

Output:

近似的前k个奇异值和奇异向量。

1: 将A作用于k个独立同分布的随机向量得到B，可以证明，B以很大概率覆盖A的列向量空间 [11]。
2: 利用Gram-Schmidt等方法得到B的标准正交基Q。
3: 由于Q几乎覆盖了A的列向量空间，所以有$A \approx QQ^TA = Q(Q^TA)$。
4: 计算Q^TA的SVD：U、S、V，因为它只有$k \times n$维，所以计算很快。
5: $A \approx QQ^TA = (QU)SV^T$。

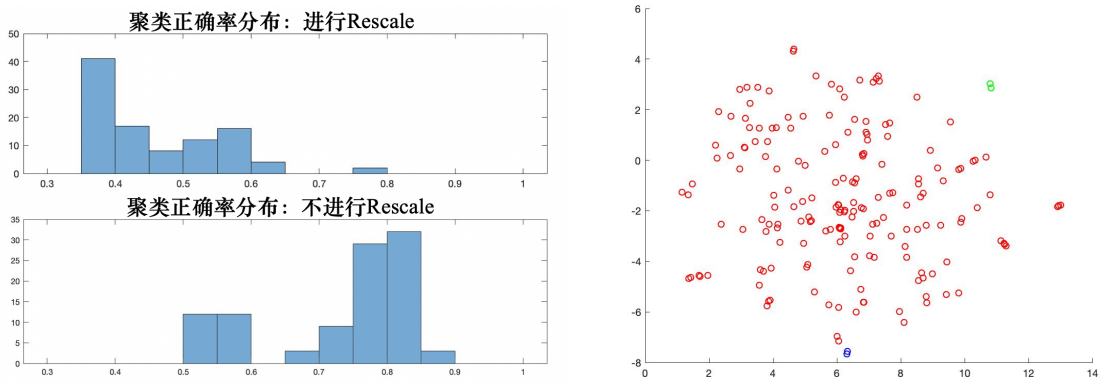


(a) 正确率 (b) 执行时间

图 9: 利用近似SVD加速后的效果

3.4.2 关于杠杆得分采样效果的讨论

在图 7中，我们得出的结论是特征选择算法的正确率很高，但这是我们进行了一百次重复实验后取目标函数最大的聚类结果得到的。实际上，这一百次实验的正确率分布如图 10，右侧的散点图表明了一个常见的聚类结果：出现离群点，几乎所有点都被分到了同一类。注意右侧的散点图是通过t-sne [14]降维后画出的。



(a) rescale前后的正确率分布

(b) 常见的聚类结果

图 10: 杠杆得分采样效果分析

经过对降维前后的矩阵进行对比分析，我们认为这是由于算法的rescale过程破坏了原始矩阵中不同特征之间的相对关系，导致某些特征被过度放大，拥有这些特征的点因此成为了离群点。为了证明这一猜测，我们对比了算法 2在ORL和NIPS数据集上选出特征的杠杆得分：

	ORL	NIPS
最小	4.03e-5	6.81e-6
最大	2.73e-4	0.14
方差	2.43e-9	3.4e-3

表 1: ORL和NIPS数据集的杠杆得分

在实验中我们发现，ORL数据集的聚类成功率非常稳定，而由表 1可以看出，它的特征杠杆得分也比较均匀。相比之下，NIPS数据集的特征重点比较突出，杠杆得分方差大，可能导致一些特征被放大很多倍，以至于特征的关系被破坏。

为了减小NIPS数据集杠杆得分之间的差异，我们将算法 2中的杠杆得分计算公式改为：

$$p_i = \left\| (V_k)_{(i)} \right\|_2 / k$$

再次实验后，我们可以将100次实验中聚类正确率达到50%以上的次数从约30次提高到约50次，我们认为这表明杠杆得分分布过于分散确实可能导致算法 2的实际效果不好。

3.4.3 与$\tilde{A} = AV_k$的比较

在算法 2的开头，我们介绍了 [8]中提到利用$\tilde{A} = U_k\Sigma_k$ 进行聚类可以得到得到最优聚类的一个近似比为2的估计。这个算法实际上就是我们常用的PCA算法，这个算法是一个确定性的算法，其提供的近似比似乎也是最好的，因此，我们以它为基准，比较了不同算法的正确率。在各算法需要进行SVD计算的时候，我们都用算法 3来替代，因此他们的执行时间都基本相同。

	ORL	NIPS
A	0.78	0.87
AV_k	0.77	0.87
ASD	0.76	0.82
AR	0.70	0.80

表 2: 不同算法的正确率对比

从表 2可以看到，利用PCA方法实际上可以得到和不降维几乎相同的聚类效果，而特征选择和特征提取算法尽管与前两者相比有差距，但也有不错的聚类效果。特征选择算法的可解释性、特征提取算法中随机投影计算的速度，都是他们各自的优势。

4 结论

在本文中，我们成功实现了基于特征值选择和随即投影的两种降维方法，并分别进行了优化工作。在基于随即投影的降维方法中，我们尝试使用不同的随机投影矩阵并学习了mailman算法加速矩阵向量乘法。在基于SVD的降维方法中，我们利用随机算法加速SVD计算，思考Rescale操作的实际效果和数据集的关系并进行优化，并尝试将A投影到V_k空间，得到了高效且聚类正确率高的降维方法。这些优化对原方法进行了有效的提升。

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] A. Globerson, G. Chechik, F. Pereira, and N. Tishby. Euclidean Embedding of Co-occurrence Data. *The Journal of Machine Learning Research*, 8:2265–2295, 2007.

[11] 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.

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

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

[14] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[15] 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.

[16] Mark Rudelson and Roman Vershynin. Sampling from large matrices: An approach through geometric functional analysis. *Journal of the ACM (JACM)*, 54(4):21–es, 2007.

[17] 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.

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

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