

文章编号: 2095-2163(2019)04-0091-05

中图分类号: TP399

文献标志码: A

# 基于 PCA 的解大型超定线性方程组快速算法及应用

宋丛威<sup>1</sup>, 张晓明<sup>2</sup>

(1 浙江工业大学之江学院 理学院, 浙江 绍兴 312030; 2 绍兴数纺科技有限公司, 浙江 绍兴 312030)

**摘要:** 本文用主成分分析 (PCA) 降维技术解大型超定线性方程组  $AX = B$  的最小二乘解。工业上遇到的大型方程组的求解要消耗大量的时间和内存, 系数矩阵通常是病态的, 会带来不可忽视的误差。本文设计基于 PCA 的算法, 并从理论上说明其可行性。实验证明该方法有效, 不仅测试误差极小, 接近原方程的误差, 而且计算时间显著减少。

**关键词:** PCA; SVD; 线性方程组; 最小二乘解

## A fast algorithm of solving large-scale over-determined linear equations based on PCA and the application

SONG Congwei<sup>1</sup>, ZHANG Xiaoming<sup>2</sup>

(1 College of Science, Zhijiang College of Zhejiang University of Technology, Shaoxing Zhejiang 312030, China; 2 Shaoxing Digital Spinning Technology Co., Ltd., Shaoxing Zhejiang 312030, China)

**[Abstract]** The paper achieves Least-squares solution large-scale over-determined linear equations  $AX = B$  via the dimensional reduction technology of Principal Component Analysis (PCA). Solving the large equations in industry consumes massive time and memory, that is not economical. Moreover, the coefficient matrix is usually ill, which could cause innegligible error. An algorithm is designed based on PCA, and its feasibility in theory is validated. The experiments show that the method proposed in the paper is effective. On one hand, the test error is extremely small and close to the error of the original equation. On the other hand, the calculation time is significantly reduced.

**[Key words]** PCA; SVD; linear equations; least-squares solution

## 0 引言

很多工业问题最终会转化成解线性方程组问题。但是, 通常工业级的方程组规模巨大, 直接求解会消耗大量的时间和内存。本文提出一种基于主成分分析 (PCA) [1-5] 的快速降维算法求解大型线性方程组。

目前, 印染行业急需解决的难题是在染色过程中, 因为事先不清楚染料光谱的精确数值, 只能先从多个布匹的染色流程中, 获取染色光谱数据, 再估计使用染料的光谱数据, 最后用于新的布匹染色。设建立一个线性模型:

其中,  $A_{ij}$  为流程  $i$  染料  $j$  的用量 (百分比);  $X_{jk}$  为染料  $j$  色谱第  $k$  个分量;  $B_{ik}$  为流程  $i$  染色布料色谱第  $k$  个分量。3 个量具有线性关系:

$$B_{ik} = \sum_j A_{ij} X_{jk}, \quad (1)$$

其矩阵形式为:

$$AX = B。$$

其中,  $A \in \mathbb{R}^{N \times m}$ ;  $X \in \mathbb{R}^{m \times n}$ ;  $B \in \mathbb{R}^{N \times n}$ 。问题是: 使用了多种染料, 获取了大量光谱数据。方程规模非常巨大, 常规解法可能会消耗大量时间和内存。因此用机器学习的降维方法来缩小矩阵大小, 然后再解方程。

按照机器学习的说法, 这是一个针对模型  $AX = B$  的训练过程, 模型的输入 - 输出样本为  $A, B$ , 通过训练得到参数估计  $\hat{X}$ 。

本文采用了 5 000 多组数据, (而这只是工业数据中极小的一部分), 目的在于验证本文方法的有效性。

## 1 算法设计

解线性方程组  $AX = B$ , 其中矩阵  $A$  很大, 可能是病态的, 且一般是列满秩的 ( $A$  的行数远大于列

基金项目: 浙江省自然科学基金(LQ19F050004)。

作者简介: 宋丛威(1986-), 男, 博士, 讲师, 主要研究方向: 小波分析、调和分析、数据分析; 张晓明(1965-), 男, 博士, 企业高管, 主要研究方向: 大数据。

通讯作者: 宋丛威 Email: williamzju@yahoo.com

收稿日期: 2019-05-06

数), 因此可用  $A^T A$  的条件数衡量方程病态程度。  $B$  有时也不只有一列。

由于各种原因, 原方程可能没有解, 只能寻求最小二乘解, 即解优化问题。

$$\min_X \|AX - B\|_2, \quad (2)$$

其中, 向量的 2 范数是  $\|x\|_2 = \sqrt{\sum_i x_i^2}$ 。 矩阵看成列向量的组合, 其范数也应该是列向量范数的组合, 即 Frobenius 范数。

$$\|X\|_F = \sqrt{\sum_j \|X_j\|_2^2} = \sqrt{\sum_{ij} |x_{ij}|^2}, \quad (3)$$

如果数据采集没有误差, 原方程组应该有非负解。 即使解包含负元素, 其绝对值也不会太大, 可直接设置为 0, 不会对解有显著影响。 增加限制条件  $x_{jk} \geq 0$  并无意义, 因为误差也传递到解上, 可代之以正则化约束。

本文记号遵循如下规范:  $A_{ij}$  表示矩阵元素;  $(V_1, V_2)$  表示分块矩阵;  $A^T$  表示矩阵转置;  $A^+$  表示广义逆;  $\lambda$  表示矩阵特征值; 直接用 1 代表单位矩阵。

### 1.1 矩阵分析

为了避免多余的矩阵运算, 没有按照标准的 PCA 流程, 而是直接对  $A^T A$  进行奇异值分解 (SVD) 或特征值分解:

$$A = CV^T = UAV^T, \quad (4)$$

其中,  $V$  是  $A^T$  特征向量构成的矩阵;  $C$  是主成分矩阵;  $A$  主对角线包含  $A$  的奇异值, 也是  $A^T A$  的特征值, 且与  $A$  形状一致。 可将  $A$  处理成对角方阵, 此时  $U$  不再是方阵。

选择  $C$  中  $p$  个列向量  $C_1$  作为主成分, 对应有:

$$A_1 = C_1 V_1^T = U_1 A_1 V_1^T, \quad (5)$$

问题转化成求解  $A_1 X = B$ 。 理论上可以得到最小二乘解  $A_1^+ B$ ,  $A_1^+ = V_1 A_1^+ U_1^T$ , 其中,  $A_1^+$  只需对  $A_1$  的非零元素求倒数, 通常  $A_1$  可逆。 通过进一步的线性代数演算, 可以证明:

$$\begin{aligned} AX - B &= AA_1^+ B - B \\ &= U_1 E U_1^T B - B \\ &= U(\bar{E} - 1)U^T B, \end{aligned} \quad (6)$$

其中,  $p$  阶对角矩阵  $E$  满足:

$$E_{ii} = \begin{cases} 1, & i \in C_1, \lambda_i \neq 0, \\ 0, & \lambda_i = 0. \end{cases}$$

$\bar{E}$  则是  $E$  填充适当个数的 0 元素得到的矩阵。 由于通常  $A$  满秩, 因此  $E$  是单位矩阵。 这里主成分矩阵和这些主成分 ( $C$  列向量或其编号) 构成的集合被等同起来看,  $i \in C_1$  代表第  $i$  个主成分被选中。

由式 (6) 可知, 对误差来说, 选择哪些主成分似乎并不重要, 相对误差大致可以写成:

$$\text{error} \sim \sum_{i \in C_1} w_i, \quad \sum_{i=1}^N w_i = 1, \quad w_i \geq 0.$$

但是, 计算  $A_1^+$  (或  $A_1^+$ ) 时, 如果  $A_1$  的元素太小, 可能会带来巨大的溢出误差, 这是选择对应最大几个特征值主成分的真正原因。

此外, PCA 可以控制方程病态性。 可用  $A^T A$  的条件数作为方程病态性的指标, 大致为  $\frac{\lambda_{\max}}{\lambda_{\min}}$ , 即最大特征值与最小特征值之比。 PCA 只选取特征值较大的几个分量, 因此分母变大, 则条件数变小。

若对  $B$  降维, 有如下公式:

$$AX - B = U(\bar{E} - 1)U^T B_1 + (B_1 - B), \quad (7)$$

其中,  $B_1$  是对  $B$  的 PCA 重构。 观察式 (7), 显然应该选取对应特征值绝对值最大的主成分。

### 1.2 算法

根据上述分析设计如下算法 (见表 1), 将一个大型方程组分解成 2 个小型方程组。

表 1 PCA 解线性方程组算法

Tab. 1 Algorithm for solving linear equations via PCA

输入方程 $AX = B$ , 输出解 $\hat{X}$
(1) 用 PCA, 分解 $A = CV^T$ , 方程归约为解 $CY = B$ , $Y = V^T X$
(2) $C = (C_1, C_2)$ , $V = (V_1, V_2)$ (根据主成分分块)
① 解 $C_1 Y_1 = B$ , 得到 $Y_1 \sim C_1^+ B = A_1^+ C_1^T B$
② 解 $V_1^T X = Y_1$ , 得到 $X \sim V_1 Y_1$
(3) 令 $\hat{X} = V_1 Y_1$ 作为对 $X$ 的估计, 计算误差 $\ A\hat{X} - B\ _2$

注: 算法没有按照 PCA 标准实现, 即没有对  $A$  中心化, 而是直接对  $A^T A$  进行奇异值分解:

$$A^T A = V \Lambda V^T, \quad C = AV.$$

并得到  $C_1^+ = A_1^+ C_1^T$ 。 算法并没有真的使用最小二乘法, ① 和 ② 基本不影响计算效率。

若使用 NMF 或其它分解技术, 分解  $A = WH$ , 则不能避免做最小二乘法。

理论上, 最小二乘解是方程误差最小的解, 但降维之后, 这个解就不再是原方程的最优解了。

### 1.3 解的处理

设最后得到的  $\hat{X} = V_1 Y_1$ , 是原方程  $AX = B$  的降维 (最小二乘) 解。 工业中有时默认所有量都是非负的, 但是降维解可能含有负数。 为了满足非负性, 可将其修正为:

$$\hat{X}_+ = \max(\hat{X}, 0), \quad (8)$$

实际上, 为了方便计算, 还可对  $B$  进行降维,

此时方程变为:

$$A(XW) = BW = D,$$

其中,  $BW$  代表  $B$  的主成分矩阵。只选用前面几个, 记为  $D_1 = BW_1$ , 方程的解为  $X_0 = XW_1$ , 不再是  $X$ 。

因为  $X_0W_1^T$  并不能完全重构  $X$ , 此时不需要做非负化处理。此外, 数值实验表明“非负处理”对误差的影响非常微小。

考虑上述对  $B$  降维处理, 最终得到解的估计为:

$$\hat{X} = V_1(C_1^+ D_1)W_1^T. \tag{9}$$

### 2 数值实验

本文算法用 Python 实现, 运行于 MacOS 上。所用数据、源代码和运行结果都托管在 Github 上, 网址为 <https://github.com/Freakwill/PCA-linear-equations>。

#### 2.1 主成分选择

对矩阵  $A, B$  降维, 通过观察累计百分比曲线, 选择适当的主成分数。如图 1 所示。

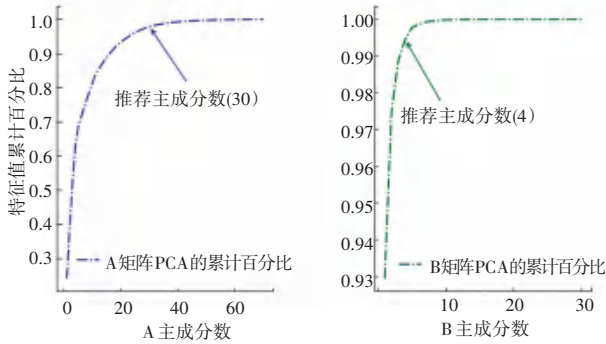


图 1 A、B 主成分数-累计百分比关系图

Fig. 1 Diagram of principal components-cumulative percentages of A, B

#### 2.2 误差分析

误差采用向量的 2 范数, 对矩阵来说就是 Frobenius 范数。本文用相对误差公式衡量算法逼近能力。

$$\frac{\|A\hat{X} - B\|_F}{\|B\|_F}. \tag{10}$$

如果对  $B$  降维, 那么  $\hat{X}$  包含重构  $X_0W_1^T$ , 即式 (9)。

20%的样本会被事先抽取出来, 测试误差通常比训练误差更有说服力。降维是对训练样本进行的, 而不是对测试样本进行。但对  $A$  的降维也可以应用于训练样本, 因为其是已知输入值, 而测试样本中的  $B$  作为目标值, 不参与降维。如图 2 所示。

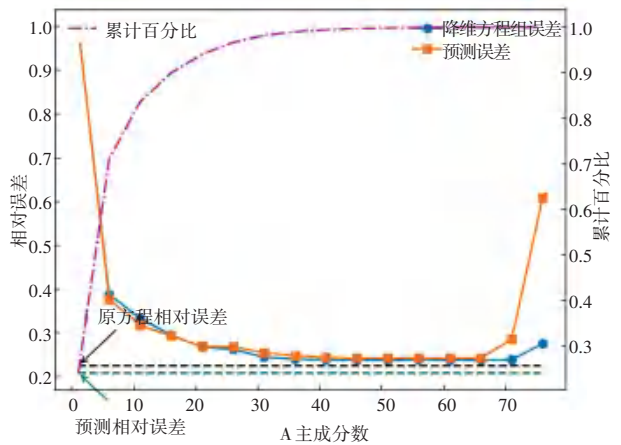


图 2 A 主成分数-误差关系图(参考累计百分比)

Fig. 2 Diagram of principal components of A and error (Refer to cumulative percentages)

由图 2 结果可知:

(1)原方程组没有解。图中原方程误差是理论上最小(在没有约束的情况下), 不妨看做一个相对 0 点。如果降维方法的误差低于这个点, 可认为是系统误差造成的, 不是数值分析意义上的误差。

(2)降维方程组的误差衰减达到预期。“负数值置 0”的处理, 对误差没有太大影响。当  $A$  的主成分数超过 70 时, 计算变得不稳定, 预测误差表现很夸张。

(3)从误差曲线不难看出, 解方程组时并不需要用到所有样本, 样本数量对本算法没有太大影响。实际应用中, 随机挑选充分数量的样本即可。

图 3 是对  $B$  取前 4 个主成分得到的误差图, 和之前的实验相比, 并没有显著影响误差。时间和主成分数基本上呈线性关系, 符合预期。为了获得较为准确的运行时间, 本次实验对每一个主成分数重复 50 次, 无论时间还是误差都取均值。

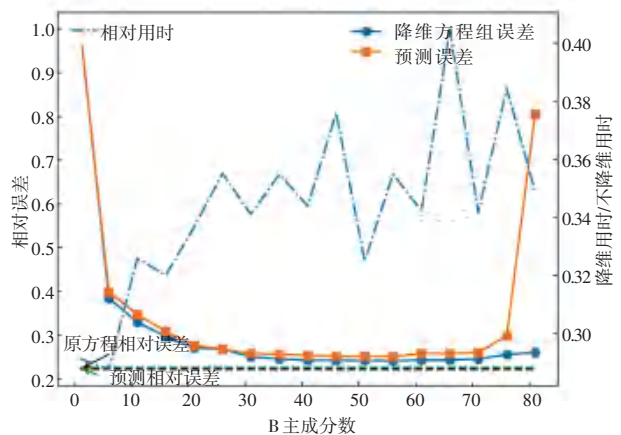


图 3 B 主成分数-误差关系图(包含对应的相对用时)

Fig. 3 Diagram of principal components of B and error (containing the corresponding relative use time)

由图4结果可知:

(1)  $B$  主成分数对误差影响没有  $A$  大 (这是优点), 由于  $B$  维数较低, 且主成分比较集中, 在第一个主成分处, 误差已降到 0.3 左右。

(2) 用时与主成分数近似呈弱线性相关,  $B$  的降维也相应提高了效率。

总之, 对  $B$  降维是完全合理的。

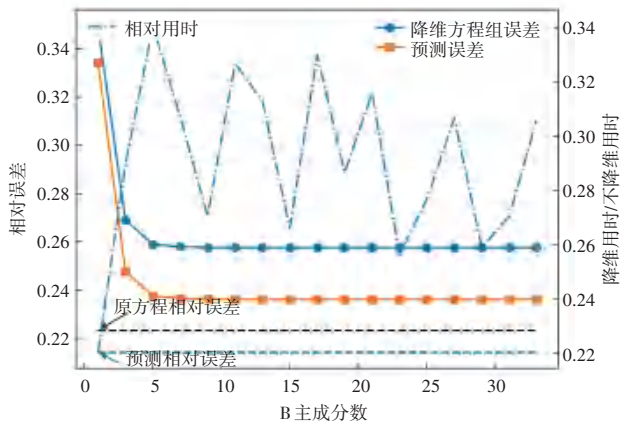


图4  $B$  主成分数-误差关系图 (包含对应的相对用时)

Fig. 4 Diagram of principal components of  $B$  and error (containing the corresponding relative use time)

注: 训练样本和测试样本是随机选取的, 各自的最小误差略有不同, 而且每次实验都会变动。可通过设定随机种子, 产生固定的训练-测试样本 (实验重复 50 次)。

选用  $A$  前 30 个主成分,  $B$  前 4 个主成分。获得实验结果见表 2。

表 2 数值实验报告

Tab. 2 Report on numerical experiments

方程	$AX = B$	$A: 4\ 177 \times 87$	$B: 4\ 177 \times 35$
设置	$A$ 主成分数 / 总数	30/87	
	$B$ 主成分数 / 总数	4/35	
	测试样本比例	20%	
实验结果	训练误差	0.252 2	
	用时	0.008 8 s	
	测试误差	0.250 2	
	原方程误差	0.224 2	

### 2.3 其它降维策略与拟合方法

NMF 降维的效果和 PCA 相近, 但矩阵分解后需解较为复杂的方程。PCA 的优势在于能分解出正交矩阵, 可以设计出更快捷的算法, 计算用时短。中心化 PCA 在主成分数较少时表现较好, 这是因为中心化 PCA 采用的是仿射变换, 比单纯的线性变换多了常数项, 但随着主成分数增加, 并没有表现出明显优势。目前中心化 PCA 只是简单重构  $A$ , 即:

$$A \sim C_1 V_1^T + M。$$

其中,  $C_1 V_1^T$  是中心化后的分解, 即通常意义上的 PCA,  $M$  是均值矩阵, 其无法简化方程, 计算时间维持在某个常数。若能设计出简化方程的算法, 那么中心化 PCA 或许是个不错的选择。

当主成分增加时, 负数解都是无法避免的。数值实验表明主成分过多还有可能出现异常。几种求解方法的测试比较结果如图 5 所示。

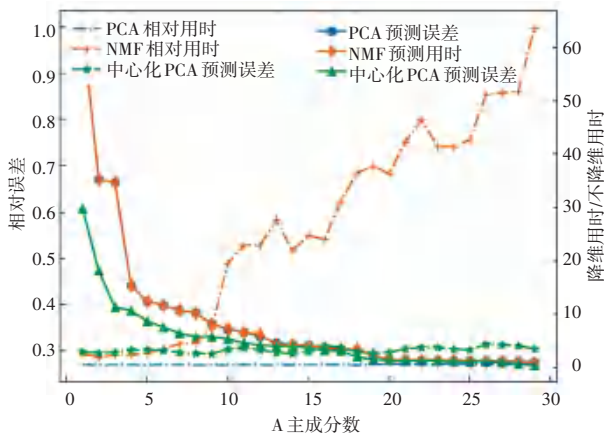


图5 基于 PCA、NMF、中心化 PCA 解方程性能比较

Fig. 5 Performance comparison of solving linear equations based on PCA, NMF and centralized PCA

显然, 完全可以用其它方法求解降维后的方程组, 尤其是当人们只关心预测, 而不在乎  $X$  时。如果只为了建立  $A$  与  $B$  的联系, 完全可以采用一些非线性方法。表 3 中给出一些线性模型测试结果, 用到了一些较复杂的计算策略, 含非线性成分, 误差无显著降低, 运行时间则较长。这些模型均由 Python 第三方库 scikit-learn 实现<sup>[7-8]</sup>。

表 3 其它线性回归方法 (以及和神经网络的比较)

Tab. 3 Other linear regression methods (and comparison with neural networks)

类型	方法	测试误差	用时/s	
线性回归	BayesianRidge	0.241 1	0.031 3	
	ElasticNet	0.241 5	0.062 3	
	ElasticNetCV	0.243 5	0.359 9	
	HuberRegressor	0.254 0	0.426 3	
	Lars	0.241 3	0.040 7	
	LarsCV	0.241 3	0.221 7	
	Lasso	0.241 5	0.065 8	
	LassoCV	0.241 9	0.394 0	
	LassoLars	0.565 2	0.011 0	
	LassoLarsCV	0.241 1	0.229 4	
	非线性	双层神经网络	0.210 9	15.520 5



最后,给出一般的计算框架,可为解线性方程组开发新的算法:

(1) 降维,  $AX = B$  转化成  $A_1X_1 = B_1$

(2) 拟合,解  $A_1X_1 = B_1$ , 得到  $\hat{X}_1$

(3) 重构,根据(1)从  $\hat{X}_1$  重构  $\hat{X}$

(4) 误差计算,采用(2)提供的范数

如果忽略不在乎计算  $X$ , 只关心预测,那么算法还可以这样设计:

(1) 降维,  $AX = B$  转化成  $A_1X_1 = B_1$

(2) 拟合,解  $A_1X_1 = B_1$ , 得到  $\hat{X}_1$

(3) 预测,  $B_1 \sim A_{11}\hat{X}_1$ , 其中  $A_{11}$  是测试数据  $A_1$  的降维, 重构  $B$

(4) 误差计算,采用(2)提供的范数

### 3 结束语

PCA 降维在解大型线性方程组表现的非常出色,随着主成分增加,误差快速递减。这种简单的代数学技巧,不仅计算快,算法设计简单,由于矩阵分解为对角矩阵和正交矩阵的乘积,之后也无需解任何线性方程组。可以根据需要任意压缩原方程。

本文提供的方法可以应用于工业生产中。但在实验中发现过多的主成分可能使得算法不稳定。今后的工作重点:

(上接第90页)

建后台数据库,利用服务器实现 Android 端与 Web 端数据的云同步。服务器方面,采用 Windows 操作系统,符合大众的常规使用习惯,便于对服务器进行后续的维护和改进。后端主要分为: Bean 层、Controller 层、Dao 层、DaoImple 层、Service 层、Test 层等,由此来构建系统框架,实现系统的功能模块。

### 5 结束语

本文对大学生时间利用率低、对生活没有规划、现有时间规划类平台受众广泛、不适合大学生使用等问题进行了研究与探索。文中构建了一个专门面向高校学生的时间管理类软件,采用了 Android 端技术来研发各类界面,Web 端搭建服务器用于数据的存储与传输,实现了业务逻辑的分离,构建了一个实用完备的大学生时间管理系统。希望本平台能够帮助在校大学生更加高效地利用时间,统筹规划自

(1)当变量有非负性或其它约束条件时,本文还没有给出更合理的处理办法。

(2)如何实现有效的基于中心化 PCA 的算法。本文提到的中心化 PCA 没有真正起到压缩方程组的作用。

(3)应利用系数矩阵的一些特点,如稀疏性,设计更有针对性的算法。

### 参考文献

- [1] HASTIE T, TIBSHIRANI R, FRIEDMAN J H. The elements of statistical learning: Data mining, inference, and prediction [M]. New York :Springer-Verlag, 2001.
- [2] HOTELLING H. Analysis of a complex of statistical variables into principal components [J]. Journal of Educational Psychology, 1933,24(6) : 417-441.
- [3] TURK M A, PENTLAND A P. Face recognition using eigenfaces [C]//Proceedings 1991 IEEE Computer Society Conference on ComputerVision and Pattern Recognition. Maui, HI, USA: IEEE, 1991:586-591.
- [4] SHALEV - SHWARTZ S, BEN - DAVID S. Understanding machine learning: from theory to algorithm[M]. New York, NY, USA :Cambridge University Press, 2014.
- [5] 焦李成,等. 稀疏学习、分类与识别[M]. 北京:科学出版社, 2017.
- [6] 解锋昌, 韦博成, 林金官. 零过多数据的统计分析及其应用 [M]. 北京:科学出版社, 2013.
- [7] HACKELING G. Mastering machine learning with scikit - learn [M]. 2<sup>nd</sup> ed. Birmingham :Packt Publishing, 2017.
- [8] COURNAPEAU D. Scikit - learn [EB/OL]. [2019]. <https://scikit-learn.org/stable/>.

己的生活,使学生生活更加充实,对于提升高等院校人才培养质量在一定意义上也有着积极的推动作用。

### 参考文献

- [1] 黄俊霞,梁雅丽,陈佰锋,等. 大学生手机使用及依赖现状研究 [J]. 中国卫生事业管理,2018(7) :534-538.
- [2] 吴晓珊,曹旭东,王森,等. 基于 B/S 架构的管理系统软件开发 [J]. 计算机测量与控制,2019,27(2) :123-128.
- [3] 陈瑞. 基于 Springboot 高并发 Java Web 开发模式[J]. 电脑编程技巧与维护,2019(4) :27-30.
- [4] 陈浩鑫,刘艳华,张桂娟. 强大的 Collection 集合框架[J]. 艺术科技,2016,29(12) :73.
- [5] 朱明元,孙媛媛. 番茄高效率学习一体化辅助平台[J]. 电脑知识与技术,2019,15(3) :104-105.
- [6] 刘姣. 基于数据挖掘技术的统计工作创新相关研究[J]. 中国管理信息化,2018,21(6) :148-149.
- [7] 张雪敏. 浅议 MVC 设计模式在 JavaWeb 中的作用[J]. 科技风, 2018(22) :58-59.