doi：10.3969/j.issn.1001-2400.2022.05.23

一步转移矩阵收敛速率与特征值的关系初探

XXX，XXX，XX

（西安电子科技大学 计算机科学与技术学院，陕西 西安 710071）

摘要：为了探究影响一步状态转移矩阵收敛速率的因素，本文将初始状态在矩阵上的迭代过程巧妙地替换为矩阵特征值与特征向量间的运算。从矩阵的特征值出发，着重分析了其绝对值大小与矩阵收敛速率之间的关系。在仿真验证过程中，我们发现了矩阵维度等因素可能影响一步状态转移矩阵收敛速率与次大特征值的线性关系，并构造一系列不同维度矩阵进行仿真，相互对比，完善了所得结论。最终确认一步状态转移矩阵的收敛所需的轮数与矩阵次大特征值的绝对值大小呈线性正相关，且矩阵维度升高时，这种线性关系逐渐减弱。

关键词：马尔可夫过程 收敛速度 特征值 矩阵维度

The Markov Chain: convergence speed and eigenvalues

*XXX，XXX，XX*

(School of Electronic Engineering, Xidian Univ., Xi’an 710071, China)

**Abstract:** A state transition matrix is a set of probabilities of each transition from different states in a stochastic process problem. It embodies the memorylessness and no aftereffect of Markov process. During learning, we found that different one-step state transition matrices will always get the same result after multiple self-multiplications, and finally reach a convergent state. In the state transition diagram, starting from any position, after several state transitions, the probability of finally reaching each position will reach a balanced state. This paper is mainly to verify the inevitability of the convergence of the state transition matrix and the relationship between the convergence speed and the matrix characteristics.

The article is divided into three parts. The first part introduces the background of the research and is derived from the knowledge of linear algebra, which proves the inevitability of matrix convergence. The second part further guesses that the convergence rate is related to the second largest eigenvalue sum based on the derived convergence expression, and simulates it through Matlab. The third part is the analysis of its correlation with the matrix dimension. Finally is the summary.

**Key Words:** Markov process Convergence rate Eigenvalue matrix dimension

1. **研究背景及收敛性证明**

**1.1 背景介绍**

马尔可夫链用于表示状态空间中从一个状态到另一个状态的转换的随机过程，在状态转换过程中具有无后效性，如在某个阶段的状态已知，则从此阶段以后过程的发展变化仅与此阶段的状态有关，而与过程在此阶段以前的阶段所经历过的状态无关。大部分的马尔科夫链的一步转移矩阵经过数次转移后将会达到收敛态，本文通过探究马尔科夫链的收敛性出发，找出一步转移矩阵及其相关参数的部分性质与矩阵收敛速度的对应关系，并通过模拟检验验证结论。

**1.2 马尔科夫链收敛于稳态性质的证明**

已知一步转移矩阵有如下性质：

由此容易推出以下结论：

* + 1. 引论1：一步转移矩阵必然存在 的特征值

设向量

* + 1. 引论2: 一步转移矩阵的任一特征值满足

取马尔科夫链初始值 , 设初始值可表示为一步转移矩阵 A 线性无关特征向量 的线性组合, 不妨设

从而易得

根据特征值性质 ：

第 t 步状态：

若 ，设：

则

若存在 则 ，即，与已知矛盾

故而

* + 1. 引论3: 满足条件的马尔科夫链必然收敛于固定稳态

由引论1、2可知

满足 ，且存在 ，不妨设

易得

引论至此结束。

1. **关于第二大特征值对收敛速度影响的猜想及仿真研究**

**2..1 对于影响因素为第二大特征值的的猜想**

由1.2证明可知

则随状态转移次数增加，除 的项在转移过程中不变外，其余项均在转移过程中呈指数性缩减。

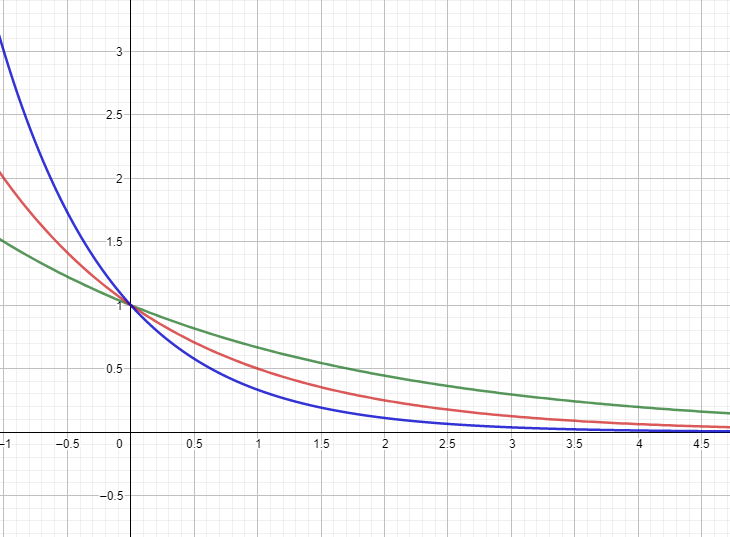


图 1 指数性缩减

可知除去的项外越大，缩减速度越慢，故而我们猜测一步状态转移矩阵的收敛快慢与矩阵特征值中绝对值第二大的特征值有关，越大，收敛速度越慢，反之越快。

* 1. **使用Matlab仿真验证**

为进一步探究对收敛速度的影响。我们使用Matlab随机生成60组随机的5维矩阵，计算其的值，以为横坐标，收敛轮数为纵坐标，画出散点图。结果如下：

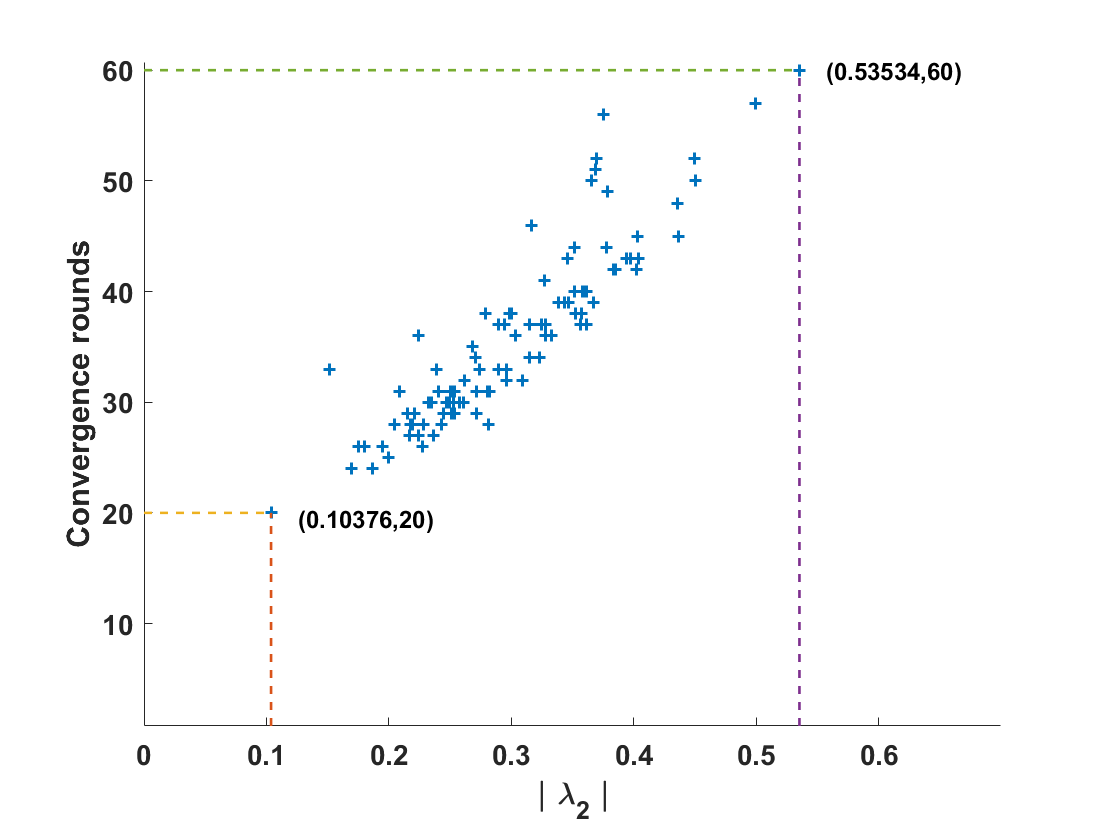


图 2 第二大特征值-收敛轮数散点图

两者表现出很强的线性关系。我们进行一元线性回归拟合，结果如下：

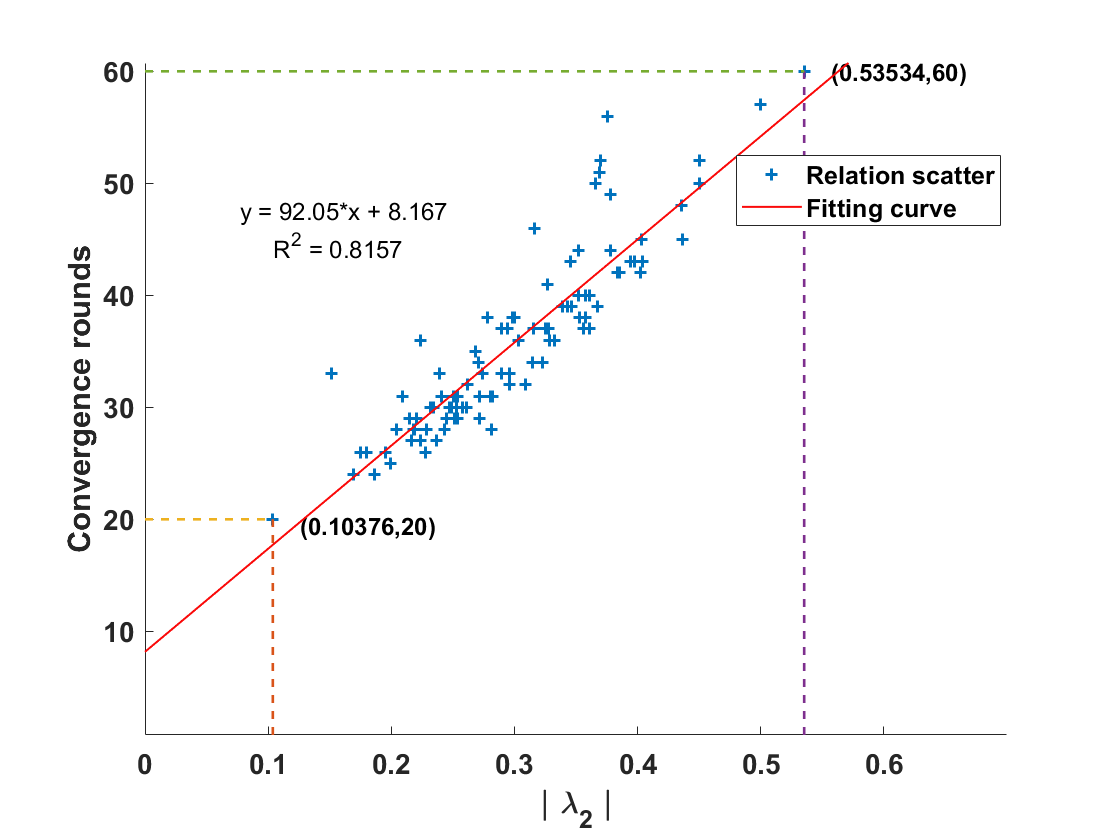


图 3 第二大特征值-收敛轮数拟合曲线

结果表明收敛轮数与成线性正相关。普遍而言越大，越接近于1，收敛速度越慢。越小，越接近于0，收敛速度越快。

有趣的是，当矩阵维度增加到8维时，收敛轮数与的线性相关性有减弱趋势，线性回归拟合图像如下：

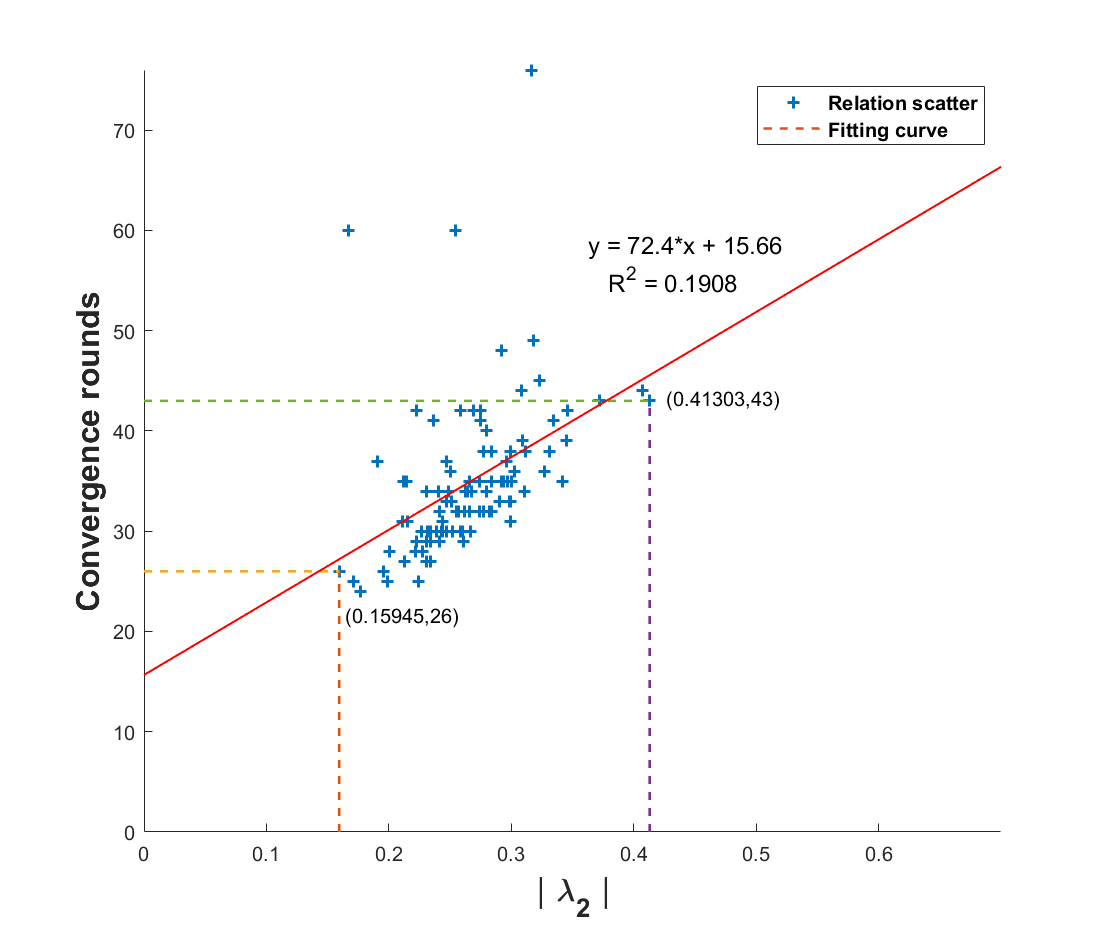


图 4 矩阵维度为8时的拟合曲线

由图4可知，当矩阵维度为8时，散点数据趋于分散， 下降，特征值与收敛轮数的线性关系减弱，不难得出猜想：该线性关系受一步转移矩阵维度的影响，矩阵维度越大，线性关系越弱。

1. **特征值与收敛轮数的线性关系与矩阵维度的关联性研究**

**3.1 问题引入**

由上述仿真结果可知，不同维度的一步状态转移矩阵对于特征值与收敛轮数的线性关系有较大影响，我

们设计选取不同维度的矩阵，在其他条件不变的基础上，分析矩阵维度与该线性关系的关联程度。

* 1. **编程仿真及结果分析**

综合展示2-8维一步转移矩阵收敛轮数与次大特征值的散点图，呈现至三维坐标系。

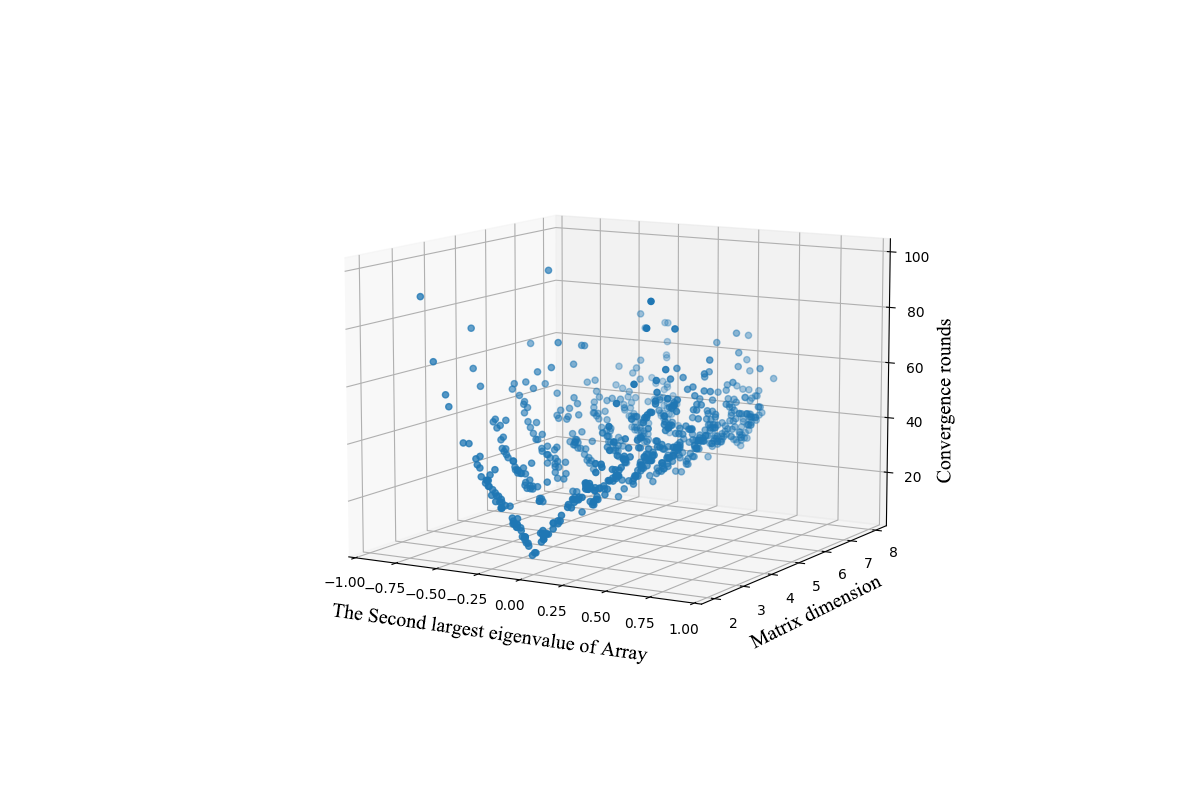


图 5 2-8维次大特征值-收敛轮数3d散点图

分别在2维、4维、6维和8维矩阵上重复上述实验，得到以下四个散点图：

**图表, 散点图

描述已自动生成图表, 散点图

描述已自动生成**

**图表, 散点图

描述已自动生成**

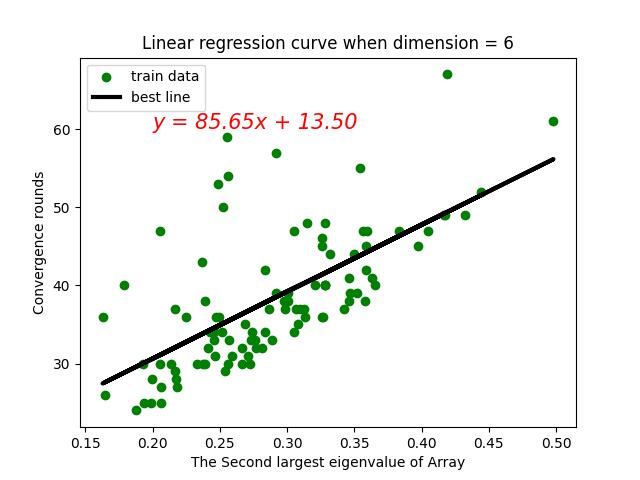


图 6 2-8维拟合曲线

**3.2.1** **结果分析：**

现象1: 在四个图中多个相同次大特征值处观察所对应迭代轮数，发现矩阵收敛所需的迭代轮数并未随矩阵维度变动而发生明显变化

现象2：当矩阵维度升高，观察到关系散点愈来愈离散，意味着线性回归效果越来越差。同时，拟合直线斜率逐步下降

现象3：随着矩阵维度的增加，所生成的次大特征值的绝对值趋向原点分布

对于观察到的现象1，经过多次作图核验及仿真验证，不同维度下矩阵平均收敛轮数并未发生较大变化，基本排除了矩阵维度对收敛快慢的影响。

而对于观察到的现象2，在仿真复现时，高维矩阵的收敛轮数与特征值的关系仍多次展现出较强的非线性性，这对我们在2.2中得出的结论提出了挑战。由此猜想矩阵维度的变化明显地影响了拟合曲线的线性程度。为了证明该猜想，我们对矩阵维度与拟合曲线线性的关系进行探究，从而纠正或补充所得结论。

在对拟合曲线进行回归分析的过程中，决定系数反映了拟合曲线与原始数据的线性相关性：

当增大时，原始数据与拟合曲线的线性相关程度较强，反之较弱。我们给出矩阵维度与的关系图：

图表, 条形图, 直方图

描述已自动生成

图 7 R^2与矩阵维度关系图

经过多次作图，我们亦在其中发现了相同的规律，即随着矩阵维度的增加，次大特征值的绝对值与收敛轮数的线性相关性逐步减弱，这很大程度上证明了我们的猜想是正确的。

对于现象3，由公式(1)(2)，随矩阵维度升高，各元素值仍满足 ，则 随矩阵维度升高而下降，易得矩阵维度升高，次大特征值λ的统计特征更易趋向原点分布，结合现象2，我们总结出矩阵维度升高后，线性相关性下降的原因。

**3.2.2 原因思考：**

随着矩阵维数的增加，下降，次大特征值绝对值的统计均值 亦随之减小。由1.2的结论可知，之外的特征值的累积影响因素会逐渐增加，因此与的相关性会逐渐减弱。

由此得出结论，当矩阵维度较低时，一步转移矩阵收敛速度与 呈较强的线性相关性， 越大，收敛速度越慢。随着矩阵维度增加，该线性相关性减弱，但收敛速度仍受矩阵其他特征值的复合影响。

1. **结论**

本论文主要为探究马尔可夫链一步状态转移矩阵收敛速度与特征值的关系。论文严格证明了一步状态矩阵收敛的必然性。通过证明结论进行合理引申猜想，提出了收敛速度主要受次大特征值影响的假设。进一步用Matlab仿真结果证实了5维矩阵收敛轮数与 的线性相关性。

在此基础上，我们进一步探究了矩阵维度对收敛速率的影响。通过Matlab仿真结果的分析，我们发现了维度的增加会使一步状态转移矩阵收敛速率与次大特征值的线性关系减弱的现象，并通过理论推导，得出了随矩阵维度升高，收敛轮数更易受到多个特征值综合影响的结论。

**参考文献：**

[1] S. Jency, “720 degree performance appraisal – an emerging technique,” International journal of informative and futuristic research, vol. 3, issue 8, pp. 2956-2964, 2016.

1. **附录**

|  |  |
| --- | --- |
|  | Figure\_scd\_times.m |
|  | clear;  clc;  jieshu=5;  % jieshu=8;  iniVector=zeros(1,jieshu); % 初始向量  iniVector(1,1)=10000;  trans = iniVector; % 当前迭代情况  preTrans=trans; % 前一轮迭代情况  generTimes=100; % 生成generTimes次一步转移矩阵  stage = jieshu; % 每个矩阵的阶数  lunshu=ones(1,generTimes);  secondMaxEigen=ones(1,generTimes);  i=1;  while i<=generTimes  temp=ones(stage);  for j=1:stage  x=rand(1,stage);  y=sum(x);  r=x./y;  temp(j,:)=r;  end  check=getSecondMaxEig(temp,stage);  if check==1  continue;  end    % 此处得到一个五阶的一步状态转移矩阵temp  preTrans=iniVector;  for j=1:100  trans=preTrans\*temp;  if trans == preTrans  lunshu(1,i)=j;  secondMaxEigen(1,i)=getSecondMaxEig(temp,stage);  break;  end  preTrans=trans;  if j == 100  i=i-1;  break;  end  end  i= i+1;  end  lunshu  secondMaxEigen  [sortedSecondMaxEigen,placeSME]=sort(secondMaxEigen,2);  sortedLunshuFollowingSME=lunshu(:,placeSME);  scatter(sortedSecondMaxEigen,sortedLunshuFollowingSME,'+','linewidth',1.5);  hold on  % grid()  axis([0,0.7,5,30])  pos = axis;  % title('一步转移矩阵次大特征值与收敛轮数关系','fontsize',14)  xlabel('\mid \lambda\_{2} \mid','fontsize',16,'fontweight','bold')  ylabel('Convergence rounds','fontsize',16,'fontweight','bold')  legend('Relation scatter')  plot([sortedSecondMaxEigen(1) sortedSecondMaxEigen(1)],[0 , sortedLunshuFollowingSME(1)],'linestyle','--','linewidth',1.3)  hold on  plot([0 , sortedSecondMaxEigen(1)],[sortedLunshuFollowingSME(1) sortedLunshuFollowingSME(1)],'linestyle','--','linewidth',1.3)  hold on  plot([sortedSecondMaxEigen(size(sortedSecondMaxEigen,2)) sortedSecondMaxEigen(size(sortedSecondMaxEigen,2))],[0 , sortedLunshuFollowingSME(size(sortedLunshuFollowingSME,2))],'linestyle','--','linewidth',1.3)  hold on  plot([0 , sortedSecondMaxEigen(size(sortedSecondMaxEigen,2))],[sortedLunshuFollowingSME(size(sortedLunshuFollowingSME,2)) sortedLunshuFollowingSME(size(sortedLunshuFollowingSME,2))],'linestyle','--','linewidth',1.3)  text(sortedSecondMaxEigen(1),sortedLunshuFollowingSME(1)-0.5,[' (',num2str(sortedSecondMaxEigen(1)),',',num2str(sortedLunshuFollowingSME(1)),')'])  text(sortedSecondMaxEigen(size(sortedSecondMaxEigen,2)),sortedLunshuFollowingSME(size(sortedLunshuFollowingSME,2)),[' (',num2str(sortedSecondMaxEigen(size(sortedSecondMaxEigen,2))),',',num2str(sortedLunshuFollowingSME(size(sortedLunshuFollowingSME,2))),')']) |

|  |  |
| --- | --- |
|  | getSecondMaxEig.m |
|  | function y = getSecondMaxEig(matrix,stage)  xushuchuxian = false;  xulie=ones(1,stage);  [~,eigValue]=eig(matrix);  for i=1:stage  if ~isreal(eigValue(i,i))  xushuchuxian = true;  break;  end  xulie(1,i)=eigValue(i,i);  end  xulie = abs(xulie);  [sorted,~]=sort(xulie,2,'descend');  y=sorted(1,2);  if xushuchuxian  y=1;  end  end  %% 如果碰到特征值为虚数的矩阵，则转向下一个 |

|  |  |
| --- | --- |
|  | Dimension.py |
|  | import numpy as np  import matplotlib.pyplot as plt  from sklearn.model\_selection import train\_test\_split  from sklearn.linear\_model import LinearRegression  from sklearn.metrics import mean\_squared\_error, r2\_score  from mpl\_toolkits.mplot3d.axes3d import Axes3D  def get\_matrix(n):  #生成一步转移矩阵  list = []  for i in range(n):  sum = 0  r = np.random.random(n)  for j in range(n):  sum += r[j]  for j in range(n):  r[j] /= sum  list.append(r)  return list  def get\_secondVaule(array, n):  #求次大(绝对值)特征值  value, evector = np.linalg.eig(array)  value.sort()  r = []  for i in range(n):  if type(value[i]) != np.float64:  return 0  if abs(value[i] - 1) > 1e-10:  r.append(value[i])  if abs(r[len(r)-1]) < abs(r[0]):  return r[0]  return r[len(r)-1]  def get\_round(array):  #求收敛轮数  n = 0  r = array  while(1):  n += 1  r = np.dot(r, array)  t = np.dot(r, array) == r  if False not in t:  return n  if n > 100:  return 0  if \_\_name\_\_ == '\_\_main\_\_':  x = []  y = []  z = []  for i in range(2, 9):  print("测试第{}维矩阵".format(i))  for j in range(100):  print("计算矩阵{}".format(j + 1))  while(1):  array = get\_matrix(i)  value = get\_secondVaule(array, i)  if not value:  continue  round = get\_round(array)  if not round:  continue  x.append(value)  y.append(i)  z.append(round)  break  '''  k = []  d = []  for i in range(7):  t = i\*100  xt = x[t:t + 100]  yt = y[t:t + 100]  d.append(i + 2)  model = LinearRegression()  xt = np.array(xt)  yt = np.array(yt)  model.fit(xt.reshape(-1, 1), yt.reshape(-1, 1))  k.append(r2\_score(yt, model.predict(xt.reshape(-1, 1)).reshape(1, -1)[0]))  print(d)  print(k)  plt.bar(d, k, width=0.6)  plt.xlabel("Dimension of Array")  plt.ylabel("R^2 of Linear regression curve")  #plt.ylim(0, 140)  plt.title("Relationship between Dimension and R^2")  plt.show()  '''  '''  # 绘制最佳拟合线  model = LinearRegression()  xt = np.array(x[400:500])  yt = np.array(y[400:500])  model.fit(xt.reshape(-1, 1), yt.reshape(-1, 1))  line\_xticks = xt  line\_yticks = model.predict(xt.reshape(-1, 1))  plt.scatter(xt, yt, color='g', label="train data")  plt.plot(xt, line\_yticks, color='black', linewidth=3, label="best line")  # 添加图标标签  plt.text(0.2, 60, "y = {:.2f}x + {:.2f}".format(float(model.coef\_), float(model.intercept\_)), size=15, color="r", style="italic")  plt.legend(loc=2)  plt.xlabel("The Second largest eigenvalue of Array")  plt.ylabel("Convergence rounds")  plt.title("Linear regression curve when dimension = 6")  # 显示图像  plt.show()  '''  fig = plt.figure(figsize=(12, 8))  ax3 = plt.axes(projection='3d')  ax3.view\_init(elev=10., azim=300)  ax3.set\_xlim(-1, 1)  ax3.scatter(x, y, z)  #ax3.set\_xlabel('The Second largest eigenvalue of Array', fontdict={'size': 25})  #ax3.set\_ylabel('Matrix dimension', fontdict={'size': 25})  #ax3.set\_zlabel('Convergence rounds', fontdict={'size': 25})  #ax3.set\_title('Array Analysis', fontdict={'size': 15})  plt.show() |