

第十二届“认证杯”数学中国

数学建模网络挑战赛

承 诺 书

我们仔细阅读了第十二届“认证杯”数学中国数学建模网络挑战赛的竞赛规则。

我们完全明白，在竞赛开始后参赛队员不能以任何方式（包括电话、电子邮件、网上咨询等）与队外的任何人（包括指导教师）研究、讨论与赛题有关的问题。

我们知道，抄袭别人的成果是违反竞赛规则的，如果引用别人的成果或其他公开的资料（包括网上查到的资料），必须按照规定的参考文献的表述方式在正文引用处和参考文献中明确列出。

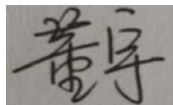
我们郑重承诺，严格遵守竞赛规则，以保证竞赛的公正、公平性。如有违反竞赛规则的行为，我们接受相应处理结果。

我们允许数学中国网站(www.madio.net)公布论文，以供网友之间学习交流，数学中国网站以非商业目的的论文交流不需要提前取得我们的同意。

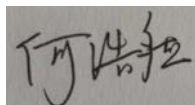
我们的参赛队号为：2196

参赛队员（签名）：

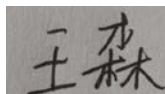
队员 1：



队员 2：



队员 3：



参赛队教练员（签名）：

参赛队伍组别（例如本科组）：研究生组

参赛队号 #2196

第十二届“认证杯”数学中国

数学建模网络挑战赛

编号专用页

参赛队伍的参赛队号：（请各个参赛队提前填写好）：
2196

竞赛统一编号（由竞赛组委会送至评委团前编号）：

竞赛评阅编号（由竞赛评委团评阅前进行编号）：

2019 年第十二届“认证杯”数学中国 数学建模网络挑战赛第一阶段论文

题 目 基于方差分布的方法对未知语言文本中重复片段
的自动搜索问题的研究

关 键 词 自动搜索算法 方差 非监督学习 聚类算法思想 MATLAB

摘 要

本文针对未知语言文本中重复片段自动搜索的问题，运用了模式识别、非监督学习中的聚类算法等思想理论，构建了含有重复字母序列片段的未知语言文本模型，综合运用了 Matlab, Excel 等软件编程以及数据分析，最终能够高效、准确的找到重复出现的字母序列片段。

本文的特色是借鉴模式识别中非监督学习的思想，利用方差这一数据统计特征把未知的样本数据中具有相似特点的数据归为一类，进而搜索出重复片段。由于重复出现字母序列片段的长度，所在文本段落中的出现位置都是随机的。针对这样的随机性和未知性，先通过方差这一数据统计特性，缩小搜索范围，相比于传统的穷举遍历法可减少搜索次数近 50 倍，大大提高了搜索速度。

针对问题一，要求解决文本长度在 5000-8000 个未知语言字母（未知语言的文字由 20 个字母构成）之间的 30 段文本中，搜索到长度为 15-21 个字母的重复出现的字母序列片段，并且此字母序列片段中会出现 0 至 4 个字母被篡改的替换错误的问题。首先，运用了随机取样的方法，构建了含有重复字母序列片段的未知语言文本模型，运用了 Matlab 软件编写基于方差分布的自动搜索算法，再通过该算法能够搜索到重复的字母序列片段。

针对问题二，要求解决评价所编写的算法的有效性及时效性问题，运用了 Matlab 软件编程求解。最终得出本文所编写的算法有较高的准确率和时效性的结论。

本文最后给出了基于方差分布的自动搜索算法的评价，客观地评价算法的优点和缺点。优点：1. 提高运算速度，简化搜索过程；2. 搜索到的重复字母序列片段准确性高；3. 此算法适应性强，对模型要求低。缺点：1. 搜索的结果中会有一定数量的字母片段丢失；2. 当样本增长时，搜索时间将急速增长，不适用于过大的样本数量情况下的搜索。

参赛队号： 2196

参赛密码 _____
(由组委会填写)

所选题目： B 题

Abstract

In this paper, for the problem of automatic searching in the repeat fragment of unknown language, it uses the pattern recognition, such as the clustering algorithm in unsupervised learning thought and theory, constructs the model of unknown language text with the repeat fragment, and utilizes the software Matlab and Excel. Finally the algorithm proposed in this paper is proved to show the efficiency and correctness in the experiment.

The feature of this paper is to use the idea of unsupervised learning in pattern recognition, and uses the statistical feature of variance to classify the data with similar characteristics in the unknown sample data into a category, and then searches out the repeated fragments. Because of the length of repeated letter sequence fragments, the occurrence position in the text paragraph is random. As such randomness and unpredictability, the data statistical characteristic of variance is used to narrow the search scope. Compared with the traditional exhaustive calendar, the searching times could be reduced by nearly 50 times, and the searching speed is greatly improved.

Problem 1: To solve the texts in the 5000-8000 of unknown language alphabet (unknown language text is composed of 20 letters) in 30 texts, it searches the length of 15 to 21 segments of repeated sequences of letters. And this letters will be 0 to 4 in sequence fragment are replaced by tampering with the alternative question. Firstly, random sampling method is used to build an unknown language text model containing repeated letter sequence fragments. Matlab software is used to write an automatic search algorithm based on variance distribution. Then, repeated letter sequence fragments can be searched by this algorithm.

Problem 2: The problem of validity and timeliness of the algorithm is required to be solved by using Matlab software. The conclusion is that the algorithm is more accurate and reduces the cost of time.

Finally, this paper gives the evaluation of the automatic searching algorithm based on variance distribution, and evaluates the advantages and disadvantages of the algorithm. Advantages: 1. Improve the operation speed, simplify the search process; 2. High accuracy of the searching repeated letter sequence fragments; 3. This algorithm is strong adaptability and low requirements for the model. Disadvantages: 1. A few of letter fragments will be lost in the searching results; 2. When the sample size increases, the searching time will increase rapidly, which is not suitable for the searching with a large sample size.

1 问题重述

一、研究背景

随着经济的不断发展和人类文明的不断进步，人类从未停止过对宇宙空间的探究。从澳大利亚帕克斯望远镜宣布：确认了FAST望远镜在2017年8月22日发现的一颗脉冲星候选体，这是FAST望远镜确认发现的第一颗新脉冲星，被称为FAST脉冲星一号（FP1），自转周期1.83秒，距离粗估1.56万光年，被命名为南仁东星。这是人类对外太空探索中的一个里程碑。纵观探索的历史脚步：1972年3月2日，NASA发射了第一艘越过小行星带的飞行器，它携带了一块“名片”，是人类向可能存在的外星人问候，并表明我们在银河系位置的镀金铝板；1973年4月6日，NASA发射了先驱者11号，船上同样携有一封“电报”，一块载有人类讯息的镀金铝板；1977年8月20日和9月5日，NASA又分别发射了“旅行者2号”和“旅行者1号”两艘宇宙飞船，两艘宇宙飞船上各带有一张名片为“地球之音”的铜质镀金激光唱片，这张金唱片承载着人类与宇宙星系沟通的使命。人类不停地探索外太空，同时也对外太空有很多美好的构想。

从最初的《星际迷航》，到《飞向太空》，再到最经典的《E.T》，人类探索外太空的脚步不曾停歇，也一直想要在外太空找到一丝生命的迹象，希望与之交流沟通，互惠互利，可以读懂外星语言就显得尤为重要。因此，我们需要对未知语言——外星语言进行详细地分析。

二、研究意义及算法思想

想要了解和学习一门未知的语言，首先要找到此种语言的主要特征，然后再由主要特征拓展到整体语言体系，最后再分析细节词汇，从而达到研究问题由点及面，再由面及点的全面系统的研究方法。语言学家预测：“如果有的序列片段在每段文本中都会出现，这些片段就很可能具备某种固定的含义（类似词汇或词根），可以以此入手进行进一步的研究。在文本的获取过程中，由于我们记录技术的限制，可能有一些位置出现了记录错误。可能出现的错误分为三种：1. 删失错误：丢失某个字母；2. 插入错误：新增原本不存在的字母；3. 替换错误：某个字母被篡改成其他字母。”因此，找到外星语言文本中多次出现的高频片段并分析出含有错误信息的目标片段，成为研究此种语言的首要问题，也是破解未知语言—外星语言的关键步骤。

模式识别(Pattern Recognition)就是通过计算机用数学技术方法来研究模式的自动处理和判读。通常把环境与客体统称为“模式”。模式识别是指对表征事物或现象的各种形式的(数值的、文字的和逻辑关系的)信息进行处理和分析,以对事物或现象进行描述、辨认、分类和解释的过程,是信息科学和人工智能的重要组成部分。

模式识别又常称作模式分类，从处理问题的性质和解决问题的方法等角度，模式识别分为有监督的分类（Supervised Classification）和无监督的分类(Unsupervised Classification)两种。二者的主要差别 在于，各实验样本所属的类别是否预先已知。文字识别是利用计算机自动识别字符的技术，是模式识别应用的一个重要领域。文字识别一般包括文字信息的采集、信息的分析与处理、信息的分类判别等几个部分。文字识别方法基本上分为统计、逻辑判断和句法三大类。常用的方法有模板匹配法和几何特征抽取法。本文中就是借用了模式识别的思想。

三、具体问题

1. 假设我们已经获取了30段文本，每段文本的长度都在5000-8000个字母中间。我们希望找到的片段长度在15-21个字母之间。为简单起见，我们假设文本中出现的错误只是替换错误，而且对我们要找的片段而言，在文本中每次出现时，最多只会出现4个字母的替换错误。设计有效的数学模型，快速并尽可能多地找到符合要求的字母片段。

2. 自行编撰算例验证所设计的算法的有效性。

2 模型的假设

- (1) 为方便计算，假设每段文本的长度均相同；
- (2) 假设希望找到的片段长度在 15-21 个字母之间；
- (3) 为了简化问题，假设问文本中出现的错误只有替换错误，并且所找片段中最多只出现 4 个字母的替换错误；
- (4) 为了方便提取随机样本，假设随机抽取的 30 段样本均满足均匀分布；
- (5) 由于语言未知，目前已知此语言由 20 个字母构成，为了方便生成样本研究，故使用英文字母 A~T（共 20 个）代表未知语言的 20 个未知字母。

3 符号的说明

序号	符号	符号说明
1	n	文本中每 n 个数值为一组
2	m	方差区间划分为 m 个等份
3	w	每段文本中有 w 个字母
4	x_n	目标片段中第 n 个数值 x_n
5	c	每个目标片段中被替换掉字母的个数 c , c 为正整数且 $c \leq 4$
6	a	片段对比时数值不相同的次数

4 模型的建立和求解

一、问题一的分析与求解

1. 对问题的分析

该问题要求在文本长度在 5000-8000 个未知语言字母（未知语言的文字由 20 个字母构成）之间的 30 段文本中，搜索到长度为 15-21 个字母的重复出现的字母序列片段，并且此字母序列片段中会出现 0 至 4 个字母被篡改的替换错误。首先，所研究的语言文字—字母未知，所以需要先将用已知语言的字母标记未知语言的字母。其次，实验所需的 30 段文本样本未知，我们需要建立 30 段未知语言的文本库。再次，保证每段文本中会含有重复出现的字母序列片段，同时也需要建立随机的目标字母序列片段库，并将产生的目标字母序列片段随机插入 30 段文本中的随机位置。最后，根据非监督学习中的聚类算法的思想，编写程序算法，在文本中快速且多地搜索到含有替换错误的重复出现的目标字母序列片段。

2. 对问题的求解

模型 I ——基于随机采样含有目标字母序列片段的未知语言文本模型

(1) 模型的准备

由于未知语言的文字由 20 个字母组成，但字母未知，所以我们采取用大写字母 A—T 这 20 个字母来标记未知语言中的字母 1 至字母 20。并且为了后续方便编写搜索算法、数学建模及运算，将此 20 个字母进行赋值。赋值列表如下：

表 1 未知语言文字、英文字母及赋值对应表

未知语言文字	字母 1	字母 2	字母 3	字母 4	字母 5
英文字母	A	B	C	D	E
赋值	1	2	3	4	5
未知语言文字	字母 6	字母 7	字母 8	字母 9	字母 10
英文字母	F	G	H	I	J
赋值	6	7	8	9	10
未知语言文字	字母 11	字母 12	字母 13	字母 14	字母 15
英文字母	K	L	M	N	O
赋值	11	12	13	14	15
未知语言文字	字母 16	字母 17	字母 18	字母 19	字母 20
英文字母	P	Q	R	S	T
赋值	16	17	18	19	20

(2) 模型的建立

前提：经过多次实验，直接随机生成 30 段文本，文本长度在 5000~8000 字之间，重复出现片段长度为 15~21 个字母的序列片的概率很低，不满足问题中的要求，且不利于实验研究及计算。因此，人工随机生成长度为 15-21 个字母的目标片段，替换到随机生成的 30 段文本中，以便于算法研究及计算。（说明：所有用于生成目标片段的参数均不在后续提出的搜索算法中使用，仅用于算法有效性的验证。）模型建立的程序框图如图 1 所示。

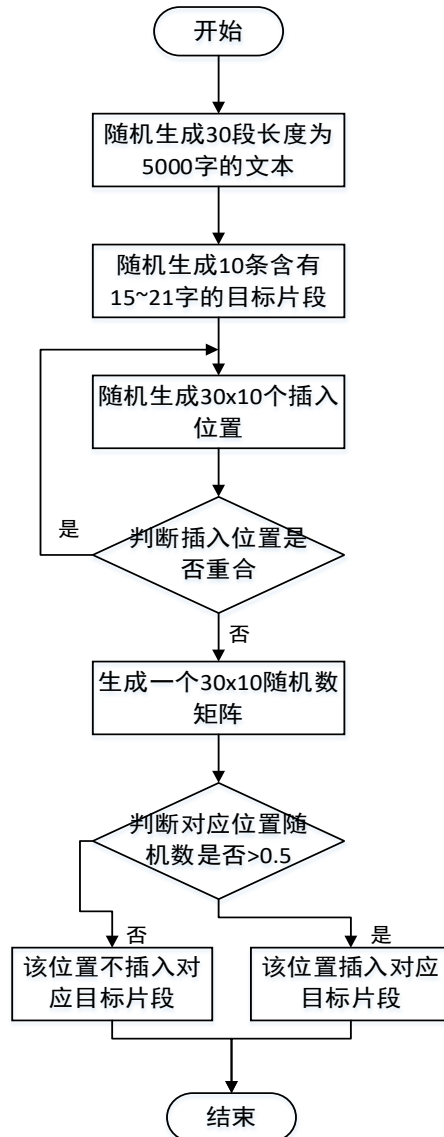


图 1 模型建立程序框图

模型建立步骤如下：

Step 1: 利用 Matlab 软件编写程序算法，创建出 30 段随机文本，并将此 30 段文本存入未知语言文本库，每段文本的字数为 5000 个；

Step 2: 随机生成 10 条含有 15-21 个字母的字母序列片段，将此字母序列片段作为目标搜索片段，将随机生成的目标搜索片段存入目标片段库；

Step 3: 生成文本中的随机插入位置，此步中要特别注意避免生成位置的覆盖，人为造成信息丢失；

Step 4: 在每次需要插入目标片段时，首先要将目标片段中的随机一个位置

的字母进行随机替换（此操作用于模拟替换错误），从而得到含有目标片段的文本。其中，文本之间插入目标片段的位置随机，每个文本中插入目标片段的个数随机，每个目标片段所含字母数随机（均在 15-21 个字母之间），替换掉的字母位置随机，替换的字母随机；

通过如上步骤，即可得到满足要求的数据模型，此模型已建立完成，可进行下一步求解运算。

（3）模型的求解

求解使用自动搜索算法，程序框图如图 2 所示。

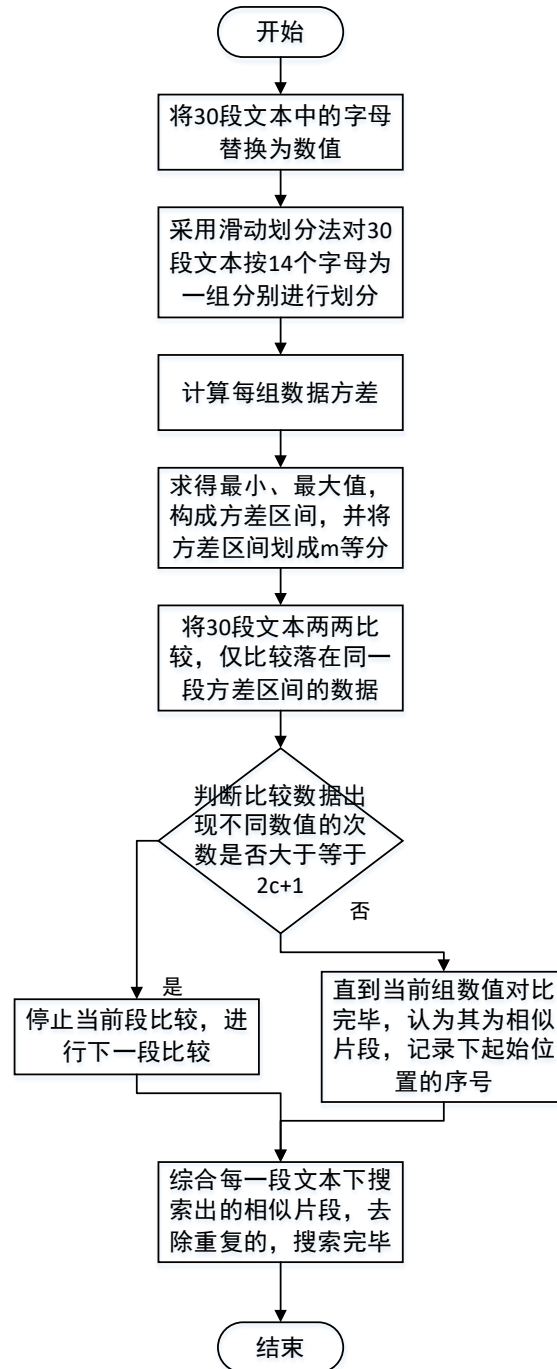


图 2 自动搜索算法程序框图

① 求解的步骤

Step 1: 在 Matlab 软件中编写搜索算法，提取未知语言文本库中的 30 段文本，并将文本中的字母按照表 1 中的赋值，将 20 个字母替换成数值，由于字母与数值是一一对应的，即每一个数值代表一个字母；

Step 2: 将所得到的每一段文本进行字段划分，每 n 个数值划分为一组，本算法将采取滑动法划分数据，将得到的数据共取得 $w-n+1$ 组数据。如：第 1 个数值至第 14 个数值为第一组，第 2 个数值至第 15 个数值为第二组，依次类推，第 $w-n+1$ 个数值至第 w 个数值为第 $w-n+1$ 组；

Step 3: 根据统计学中的数据特征，此算法中采取方差来分析数据特征。首先，计算每个文本中每组数据的方差，并将此方差记录出来，找到方差最大值和最小值，构成方差区间，再将此方差区间划分为 m 等份；

Step 4: 将 30 段文本分别进行每两段之间的数据对比，由于方差体现数据的离散分布情况，仅比较每两段文本中落在相同方差区间的数据，以此来缩减对比数据，加快辨识速度；

Step 5: 每两段之间数据的对比方法：首先，从每个目标片段中第一个数值 x_1 开始比对，依次对比 $x_2 \dots x_n$ ，并将每次对比出的不相同次数 a 记录下来，当对比时，若 $a \geq 2c+1$ ，则跳出当前次对比，认为当前两段片段中的数值排序不相似，则停止当前两段比较，再进行下两段比较；若 $a \leq 2c$ ，则认为当前两段片段中的数值排序相似，即为所要搜索的目标片段，并将此相同片段起始位置的序号记录并保存；

Step 6: 将已保存的目标片段中的重复片段筛选掉；

Step 7: 最后将得到的数据人工确认其准确性。

② 求解的结果

i. 生成 10 个目标片段，每个片段中所含字母个数随机（每个片段中所含的字母随机个数见表 2），个数均在 15-21 之间，见表 3。再将目标片段中的 1 个随机位置替换成一个随机字母（本文中仅以 1 个字母替换掉为例，此方法可以推广到 n 个字母被替换），这样为 30 段文本中的每段文本均生成 10 个不同的替换后的目标片段，由于此数据过大，故仅取其中一段（此文中取第 2 段文本）的替换后的目标片段结果在此呈现，替换位置已标黄，见表 4。算法程序请见附录一。

表 2 随机生成 10 个目标片段中的字母个数

目标片段序号	1	2	3	4	5	6	7	8	9	10
目标片段中所含字母数	20	21	15	21	19	15	16	18	21	21

其中，表 3 中数值为“0”的位置表示没有字母，如片段 1 的第 21 位数值为“0”，则此片段中仅含有 20 个字母，第 21 位没有字母。

表 3 随机生成的 10 个目标片段中的各个位置代表字母的数值

目标片段序号 位置序号	1	2	3	4	5	6	7	8	9	10
1	14	20	17	3	1	7	8	14	9	1
2	9	11	4	9	18	15	18	5	10	14
3	17	10	16	18	15	7	1	11	12	11
4	17	4	9	9	14	11	14	1	3	7
5	9	14	19	6	3	9	17	6	6	2
6	16	6	10	10	16	2	3	16	15	15
7	20	12	14	16	13	6	16	18	18	15
8	5	3	8	10	9	15	14	1	13	7
9	11	17	8	9	20	7	2	9	2	11
10	16	15	18	19	11	19	2	17	17	11
11	6	1	6	3	2	13	19	15	1	5
12	12	2	4	8	3	16	5	17	7	6
13	1	9	13	11	10	19	20	10	8	20
14	20	15	14	6	7	13	7	10	14	9
15	8	15	4	8	13	18	12	9	4	15
16	12	5	0	18	3	0	15	10	1	8
17	20	11	0	13	15	0	0	3	10	1
18	13	16	0	13	19	0	0	8	3	5
19	14	2	0	19	2	0	0	0	17	20
20	17	1	0	1	0	0	0	0	6	8
21	0	18	0	8	0	0	0	0	20	8

表 4 被替换后的 10 个目标片段中的各个位置代表字母的数值

目标片段序号 位置序号	1	2	3	4	5	6	7	8	9	10
1	14	20	17	3	1	7	8	14	9	1
2	9	11	4	9	18	15	18	5	10	14
3	17	10	16	18	15	7	8	11	2	11
4	17	4	9	9	8	11	14	1	3	7
5	9	14	19	9	3	9	17	6	6	2
6	16	6	10	10	16	2	3	16	15	15
7	20	12	14	16	13	6	16	18	18	15
8	5	3	8	10	9	15	14	1	13	7
9	12	17	13	9	20	7	2	9	2	11

10	16	15	18	19	11	19	2	17	17	11
11	6	1	6	3	2	13	19	15	1	5
12	12	2	4	8	3	16	5	17	7	6
13	1	9	13	11	10	19	20	19	8	20
14	20	15	14	6	7	8	7	10	14	9
15	8	15	4	8	13	18	12	9	4	15
16	12	5	0	18	3	0	15	10	1	8
17	20	11	0	13	15	0	0	3	10	1
18	13	16	0	13	19	0	0	8	3	5
19	14	2	0	19	2	0	0	0	17	20
20	17	1	0	1	0	0	0	0	6	9
21	0	14	0	8	0	0	0	0	20	8

ii. 需要将上述所得到的替换后的目标片段插入 30 段文本中的随机位置，见表 5，此表中数值为“0”的位置为未插对应的入目标片段。如，在第 3 段文本中旨在字母位置为 4729 处添加替换后的第 6 个目标片段，在此段文本中未添加第 5 个目标片段。

表 5 10 个片段在 30 篇文本中随机插入的字母位置

片段序号 文本序号	1	2	3	4	5	6	7	8	9	10
1	0	348	4025	2531	0	3066	2975	2226	4759	4246
2	3356	1032	0	4278	2096	4880	0	0	1000	4374
3	0	0	0	0	0	4729	1809	0	2723	1759
4	0	4451	0	0	657	1612	2936	418	4063	4920
5	0	2470	0	849	0	3318	782	0	0	1316
6	3606	0	0	0	0	1778	3479	0	0	651
7	0	0	3727	0	0	2384	0	1568	0	0
8	4162	666	0	976	0	4567	2064	170	1808	0
9	322	1634	35	4120	1030	0	0	1683	0	0
10	1157	0	785	4751	1764	2548	0	3819	1388	0
11	0	0	2250	747	193	3010	0	1173	0	3680
12	0	0	0	0	811	0	0	0	624	3763
13	0	86	469	0	0	0	0	2007	0	870
14	3289	4918	0	0	187	0	4399	0	1072	4487
15	0	2078	0	2030	4225	0	1893	0	0	0
16	4017	0	0	3010	3769	4368	0	0	588	0
17	3535	0	1151	0	4651	0	3833	1882	3346	0
18	0	3233	2099	4025	0	0	0	1199	0	549
19	3602	0	0	1659	0	0	0	4742	3238	0
20	0	1684	0	4354	0	98	0	0	3596	0
21	3809	0	3976	1797	0	2987	4247	2635	1363	0

22	233	3881	0	3088	0	1988	0	1850	0	0
23	0	0	1178	0	0	698	0	2410	2668	3807
24	0	0	0	0	0	0	4494	3212	3294	1746
25	673	126	0	2676	0	4535	0	0	4306	0
26	3071	0	0	0	0	0	0	1712	0	1291
27	237	1243	2379	1698	0	0	4479	0	0	0
28	0	0	0	0	3486	0	2898	1923	533	0
29	0	0	4655	0	0	19	0	0	1660	3411
30	0	0	4065	0	1792	0	1946	0	0	0

iii. 通过上部分找到目标片段的插入位置后，生成新的含有替换后的目标片段的 30 段文本。由于生成后的 30 段文本较大，此文章只举其中一个例子来证明 30 段文本已按题目要求生成。这里以第二段文本在第 4880 字母位置插入替换后的目标片段。具体数据见图 3。图 3 中横坐标为 30 段文本，纵坐标为文本中的数值位置（即所对应字母位置），图中的标记部分为随机插入位置后的目标片段，此文本中的目标片段与表 4 中给出的目标片段一致，说明生成随机插入目标片段的文本建立完成。

绘图

+

根据所选内容新建

打开

打印

变量

视图

行

列

4880:4895

2

插入

删除

转置

排序

编辑

5000x30 double

	1	2	3	4	5	6	7	8	9	10	11
4872	14	1	11	17	13	10	4	9	17	14	8
4873	14	11	12	5	3	19	7	14	20	5	16
4874	19	19	5	20	17	2	19	9	5	20	10
4875	11	13	9	19	20	8	6	19	15	4	10
4876	8	13	11	7	19	1	18	10	20	13	2
4877	14	13	17	16	4	1	10	15	20	1	7
4878	9	19	14	16	19	13	3	4	2	8	1
4879	3	5	10	12	3	11	13	6	2	11	9
4880	9	7	18	3	1	12	15	15	10	1	14
4881	11	15	10	1	14	6	12	15	14	1	5
4882	8	7	6	18	3	14	14	20	3	2	4
4883	8	11	15	12	5	13	18	2	1	7	14
4884	16	9	20	2	14	15	10	17	4	4	20
4885	11	2	8	5	7	13	6	13	17	7	13
4886	10	6	5	17	11	9	20	8	4	6	13
4887	2	15	5	2	11	15	10	17	7	16	17
4888	6	7	1	17	10	20	17	6	8	20	16
4889	9	19	20	18	11	11	15	2	14	8	7
4890	13	13	17	8	12	16	11	12	9	18	16
4891	15	16	12	17	8	20	20	16	12	5	8
4892	20	19	8	17	18	13	2	5	4	2	19
4893	19	8	14	4	8	6	3	8	8	13	18
4894	2	18	20	14	18	13	15	12	8	2	9
4895	20	8	15	14	9	13	2	20	11	14	9
4896	4	1	3	20	5	19	13	19	17	13	3
4897	20	5	9	8	15	13	10	16	2	13	2
4898	12	19	14	15	2	19	10	5	5	12	11
4899	5	18	7	2	4	6	9	11	16	5	2
4900	2	12	19	2	19	4	12	5	18	16	17
4901	7	12	13	11	12	9	15	14	18	2	4
4902	17	1	19	8	14	18	3	18	19	4	8

图 3 插入替换后目标片段的文本数据

iv. 计算划分数据组的方差，求得所有方差的最大最小值，生成方差区间，并将此方差区间等分为 12 份，方差区间分段点如图 4 所示，以此生成 30 份直方图。取其中第 4 段文本和第 17 段文本对应的方差直方图为例，如图 5 和图 6 所示。

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	5.7806	10.9014	16.0221	21.1429	26.2636	31.3844	36.5051	41.6259	46.7466	51.8673	56.9881	62.1088	67.2296

图 4 方差区间分段点

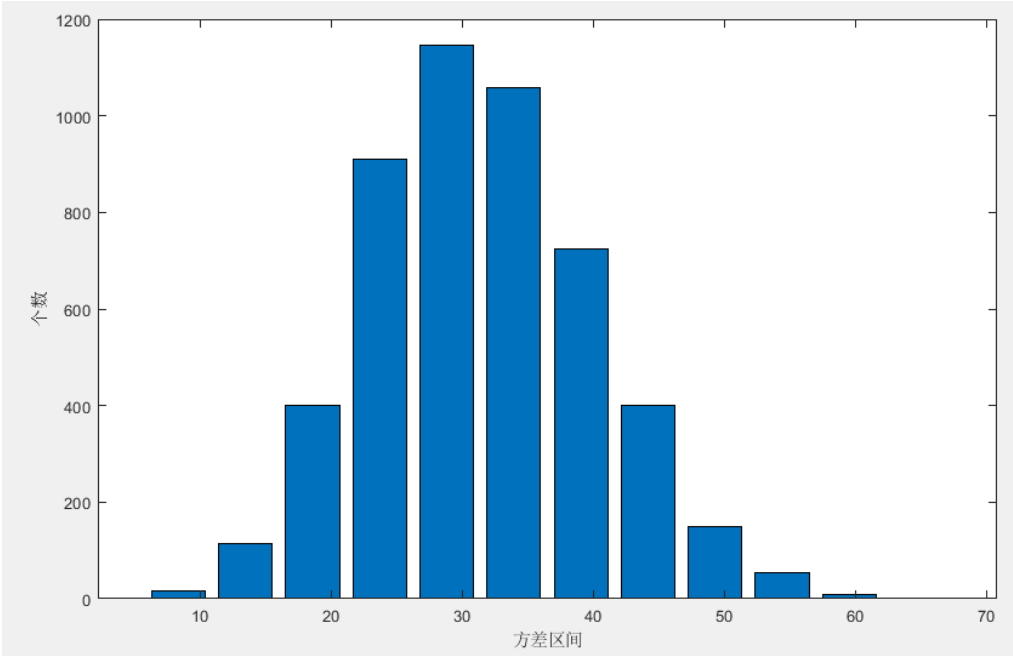


图 5 第 4 段文本各方差区间中的划分数据组个数

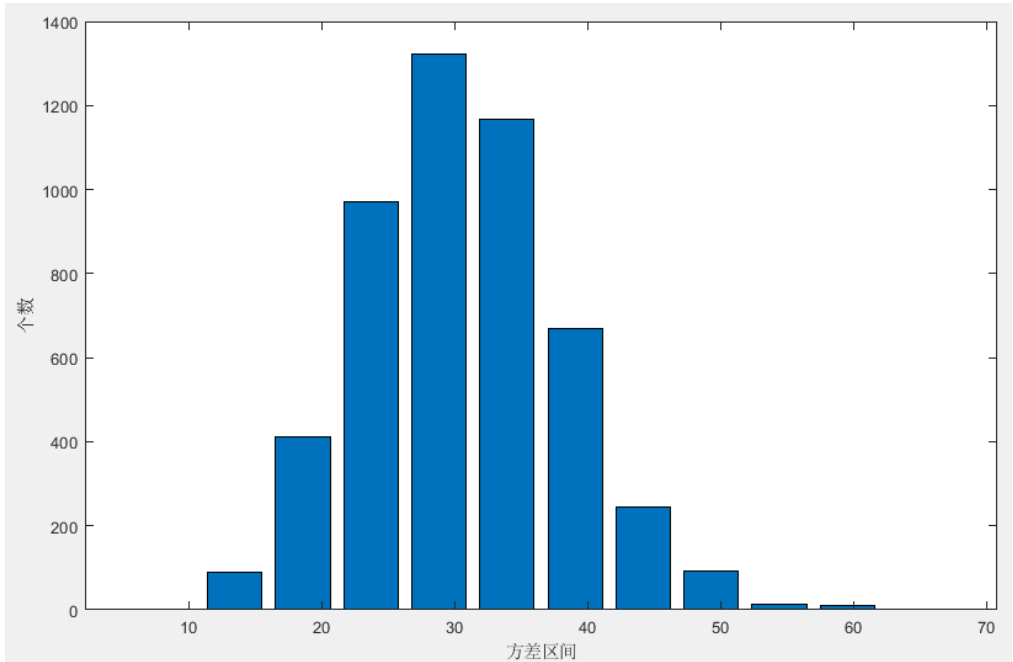


图 6 第 17 段文本各方差区间中的划分数据组个数

v. 找出落在同一方差区间的数据组，并计算程序运算时间。程序运算时间如图 7。

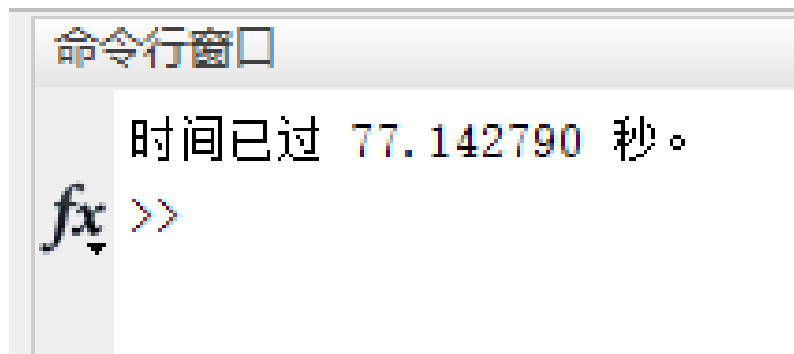


图 7 自动搜索算法运算时间

vi. 人工辅助校对片段，最终得到满足问题要求的重复出现片段。

二、问题二的分析与求解

1. 对问题的分析

由于模型一设计的算法已可以查找出问题中所要找的片段，但为了评价算法的查找能力，我们需要建立如下评价标准。我们通过实际算例验证所编写的算法的有效性及时效性。

2. 问题的求解

(1) 算法有效性

搜索算法结果如图 8 所示。表 6 代表最开始建立的 30 段文本含有目标片段的随机插入位置，此表用于自动搜索算法的结果对比，此表中的横坐标为目标片段序号，纵坐标为文本数。表 6 中的数值表示目标片段插入至文本中的位置，数值为“0”的位置，表示未插入目标片段。图 8 中每一行同一色块代表搜索到的一个片段，由于实验选得滑动切分数据组时，数据组的数据组成个数小于目标片段中所含数据的个数，所以在图中会出现连续位置的一些数据组，此数据组与目标片段为同一片段。表 6 与图 8 同一行同一颜色表示文本中的同一插入位置。通过算法自动搜索和人工辅助校对，并可从表 6 和图 8 中对比得出，能在 30 段文本中找到所插入的 10 各目标片段，从而体现算法的有效性。

表 6 最初 30 段文本 10 段目标片段生成的插入位置

	1	2	3	4	5	6	7	8	9	10
1	0	348	4025	2531	0	3066	2975	2226	4759	4246
2	3356	1032	0	4278	2096	4880	0	0	1000	4374
3	0	0	0	0	0	4729	1809	0	2723	1759
4	0	4451	0	0	657	1612	2936	418	4063	4920
5	0	2470	0	849	0	3318	782	0	0	1316
6	3606	0	0	0	0	1778	3479	0	0	651
7	0	0	3727	0	0	2384	0	1568	0	0
8	4162	666	0	976	0	4567	2064	170	1808	0
9	322	1634	35	4120	1030	0	0	1683	0	0

10	1157	0	785	4751	1764	2548	0	3819	1388	0
11	0	0	2250	747	193	3010	0	1173	0	3680
12	0	0	0	0	811	0	0	0	624	3763
13	0	86	469	0	0	0	0	2007	0	870
14	3289	4918	0	0	187	0	4399	0	1072	4487
15	0	2078	0	2030	4225	0	1893	0	0	0
16	4017	0	0	3010	3769	4368	0	0	588	0
17	3535	0	1151	0	4651	0	3833	1882	3346	0
18	0	3233	2099	4025	0	0	0	1199	0	549
19	3602	0	0	1659	0	0	0	4742	3238	0
20	0	1684	0	4354	0	98	0	0	3596	0
21	3809	0	3976	1797	0	2987	4247	2635	1363	0
22	233	3881	0	3088	0	1988	0	1850	0	0
23	0	0	1178	0	0	698	0	2410	2668	3807
24	0	0	0	0	0	0	4494	3212	3294	1746
25	673	126	0	2676	0	4535	0	0	4306	0
26	3071	0	0	0	0	0	0	1712	0	1291
27	237	1243	2379	1698	0	0	4479	0	0	0
28	0	0	0	0	3486	0	2898	1923	533	0
29	0	0	4655	0	0	19	0	0	1660	3411
30	0	0	4065	0	1792	0	1946	0	0	0

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	
1	344	349	352	2227	2230	2231	2530	2531	2532	2535	2536	2972	2973	2977	2978	2979	3063	3065	3068	3070	4023	4027	4028	4029	4030	4242	4243	4244	4248	4253	4758	4759	4760	4765	4767	
2	999	1000	1001	1009	1033	1034	2095	2098	3354	3355	3356	3358	3361	4278	4279	4282	4370	4376	4381	4879	4880	4881														
3	1757	1761	1766	1805	1810	1811	1812	1813	2721	2722	2723	2729	4726	4727	4729	4730																				
4	420	422	423	424	659	664	1610	1612	1613	2933	2934	2935	2938	2939	2940	4063	4064	4069	4071	4452	4918	4922	4928													
5	780	781	784	785	786	787	849	851	852	853	854	855	1320	1323	1324	2471	2472	3316	3318	3321	3322	3323														
6	648	649	653	658	1777	1778	1779	3477	3478	3480	3482	3483	3603	3604	3605	3606	3608	3611	3613	3614																
7	1570	1572	2381	2382	2383	2384	2385	2386	3728	3729	3730																									
8	172	174	175	176	177	667	668	976	977	979	980	981	1806	1807	1808	1809	1814	2063	2066	2067	2068	2069	4167	4565	4566	4567	4569									
9	36	37	38	320	321	322	327	329	330	331	1028	1032	1630	1636	1638	1685	1687	1688	1689	1691	4119	4120	4121	4123	4124	4125										
10	785	788	1155	1156	1157	1158	1162	1164	1165	1386	1387	1388	1394	1396	1760	1766	2546	2548	2549	2550	3823	3824	3825	3826	4752	4753	4754	4755	4759							
11	195	199	748	750	751	752	758	1172	1175	1177	1180	2251	2252	2253	2255	3009	3010	3011	3013	3015	3677	3678	3682	3688												
12	624	625	627	630	612	613	617	3761	3765	3770																										
13	87	88	468	471	472	473	868	872	877	2006	2008	2011	2012	2013	2014	2015																				
14	187	188	189	1070	1071	1072	1078	3287	3291	3294	3297	3298	4397	4398	4401	4402	4403	4489	4494	4919	4920															
15	1891	1892	1894	1895	1896	2029	2031	2034	2036	2079	2080	4224	4226	4227	4231																					
16	586	587	588	598	3011	3013	3014	3015	3770	3771	3775	4016	4017	4024	4025	4026	4366	4368	4369	4371																
17	1152	1153	1154	1155	1886	1887	1888	1889	1890	3344	3345	3346	3352	3356	3533	3534	3535	3536	3543	3544	3835	3836	4652	4653												
18	547	551	556	1181	1203	1204	1205	2097	2098	2100	2101	2102	3234	3235	3346	4025	4026	4028	4029	4030																
19	1659	1660	1662	1663	1664	3234	3238	3239	3243	3244	3248	3292	3600	3602	3603	3607	4741	4742	4743	4746	4747	4748	4749													
20	95	98	99	101	102	1662	1665	1686	3595	3596	3597	3600	3602	3607	4312	4353	4354	4357	4358	4359	4365															
21	1361	1362	1363	1368	1369	1370	1796	1797	1798	1800	1801	1802	2637	2639	2640	2984	2985	2987	2988	3807	3809	3816	3817	3974	3975	3976	3977	3978	4245	4246	4248	4249	4250			
22	231	233	235	245	1850	1852	1854	1855	1985	1986	1988	1989	1991	1992	3084	3088	3089	3091	3092	3877	3882															
23	696	698	699	700	1177	1179	1180	1181	1182	2414	2415	2416	2417	2667	2668	2669	2678	3805	3809	3815																
24	1748	1753	3214	3216	3218	3219	3292	3293	3294	3297	3303	3304	4491	4492	4493	4496	4497																			
25	127	128	671	673	674	675	678	680	682	2672	2675	2677	2679	2680	2687	4304	4306	4307	4497																	
26	1292	1293	1295	1712	1714	1716	1718	3069	3070	3071	3073	3078	3079	3080																						
27	236	237	244	245	246	1239	1244	1245	1697	1698	1699	1701	1702	1703	1713	1718	2006	2378	2379	2381	2382	4475	4476	4477	4481	4482										
28	532	533	539	542	543	1922	1925	1927	1928	1929	2895	2896	2897	2900	2901	3484	3485	3486	3492	3493																
29	17	19	20	21	22	1659	1660	1666	3412	3413	3418	4654	4655	4656	4657	4658																				
30	1790	1793	1794	1944	1945	1948	1949	4065	4066	4067	4369	4658																								

图 8 最终搜索出的插入位置

(2) 算法时效性

本文提出的算法与穷举遍历算法相比较，由于遍历法需要的运算数据过大，经过数次实验 Matlab 运算时间均超过 0.5 小时，且未搜索出文本中的重复片段。本文中所提出的自动搜索算法，经过 30 次测试自动搜索算法的运算时间，测得算法运行时间如图 9 所示。从图 9 中的数据可得出，经过 30 次测试，自动搜索算法的平均运行时间为 76.42369 秒，此时间远小于穷举遍历法的时间，从而证

明自动搜索算法的较好的时效性。自动搜索算法的运行时间方差为 9.1315，证明此算法相对稳定。

```

命令行窗口
>> for i=1:30
test1_4
end
自动搜索算法运行时间: 74.0093 s
自动搜索算法运行时间: 78.2111 s
自动搜索算法运行时间: 74.0367 s
自动搜索算法运行时间: 77.7739 s
自动搜索算法运行时间: 76.2542 s
自动搜索算法运行时间: 74.0882 s
自动搜索算法运行时间: 78.3727 s
自动搜索算法运行时间: 74.2269 s
自动搜索算法运行时间: 82.2042 s
自动搜索算法运行时间: 77.8638 s
自动搜索算法运行时间: 81.9432 s
自动搜索算法运行时间: 77.5304 s
自动搜索算法运行时间: 74.5114 s
自动搜索算法运行时间: 73.902 s
自动搜索算法运行时间: 82.6764 s
自动搜索算法运行时间: 80.908 s
自动搜索算法运行时间: 78.8469 s
自动搜索算法运行时间: 76.2357 s
自动搜索算法运行时间: 75.7953 s
自动搜索算法运行时间: 78.6285 s
自动搜索算法运行时间: 75.4133 s
自动搜索算法运行时间: 75.6577 s
自动搜索算法运行时间: 75.2667 s
自动搜索算法运行时间: 72.7541 s
自动搜索算法运行时间: 74.2935 s
自动搜索算法运行时间: 72.9423 s
自动搜索算法运行时间: 71.4308 s
自动搜索算法运行时间: 71.975 s
自动搜索算法运行时间: 79.2933 s
自动搜索算法运行时间: 75.6652 s
    
```

图 9 30 次自动搜索算法运算时间

5 自动搜索算法的评价

一、 自动搜索算法的优点

对于我们所建立的数学模型本算法相对于一一对比的传统思维来讲即高效又准确。本算法打破的传统的思维方式，本算法不去直接选择一一对比的方式，也不直接采用已有的 20 个字母去对比。本算法的思路方式是分别先对已有的 20 个字母进行人为的数字排列，把直接的字母对比转换成数字模型的对比，这样做即加快了寻找速度，而且能够准确的寻找出对应字母。

1. 打破传统思维方式，另辟蹊径，寻找新的思维突破口。本算法提前先为已知的 20 个字母进行数字排序，把直接的字母对比转换成数学模型的对比，利用数学模型中的方差这一数字特征量来提前筛选出符合条件的字母片段，至此一步就可以把数据对比量缩减近 50 倍。

2. 提高运算速度，加快寻找结果。利用方差这一数学特征值可在寻找得到目标片段的前提下可直接缩减数据量近 50 倍，不仅提高了运算速度，而且还提高了寻找目标的准确性，从传统思维漫无目的的寻找，到很快锁定目标片段，这不仅是速度的提升还是质的飞跃。

3. 寻找到的目标片段准确性高。通过本算法寻找的字母片段都能保证是所寻找的目标片段，无一误差，在考虑替换错误的情况下，一样能保证做到正确性 100%。

4. 可满足多种目标片段的需求。本算法不仅能完成了题目所需要的功能，而且还能做功能扩展。本算法不仅可以完成搜索字母在 15-21 个的区间片段，还可以扩展到大于 21 个字母片段的文字搜索中，可以做到不仅能搜索文本片段常出现的词汇词根，如果出现常用的语言片段也可以搜索到。

二、 自动搜索算法的缺点

本算法是针对多数量的未知语言进行搜索文本，得出常用字母片段的数学模型，其运行数据之多，运算量之大是我们共同面对的问题，简化模型，优化算法是我们始终追求的目标。

1. 对于我们现有的本算法我们也做到了高效准确地寻找出目标字母片段，但我们还有进一步优化模型，简化算法的空间。

2. 在本算法中，如果搜寻的目标字母片段中的字母个数比较少，再加上中间设定了替换错误的判断条件，就有可能使 30 段文本中的每一段都有不同目标字母片段的丢失。

6 参考文献

- [1] 丁兴烁, 谢忠秋. 权数确定的新方法——概率法[J]. 统计与决策, 2003(7):13-14.
- [2] 博雅. 破解外星符号[J]. 大科技(科学之谜), 2008(9):14-15.
- [3] 熊承义, 李玉海. 统计模式识别及其发展现状综述[J]. 科技进步与对策, 2003, 20(8):173-175.
- [4] 宋佳. 模式识别综述及汉字识别的原理[J]. 科技广场, 2007(9):133-135.

[5] 刘迪, 李耀峰. 模式识别综述[J]. 科学技术创新, 2012(28):120-120.

7 附录

附录一

```
clc;
clear;

%假设有original_num段原始数据, 有target_num段目标数据, 有替换错误
出现(每段目标数据有一个被随机替换)

%假设原始数据长度一致,目标数据长度随机,每段原始数据随机插入随机
段不同目标数据

%生成原始数据及目标数据
original_num = 30;
original_length = 5000; %原始数据长度

target_num = 10; %目标数据段数

target_length = ceil(rand(1,target_num)*7)+14; %目标数据长度矩阵,每一
列为对应段目标数据长度,取值为15~21
Origin_Data = ceil(rand(original_length,original_num)*20);

for i=1:target_num
    temp = ceil(rand(target_length(1,i),1)*20); %按长度生成每段数据

    temp = [temp;zeros(21-length(temp),1)]; %将数据补零至21位(最大),
以便于合成矩阵

    Target(:,i) = temp; %生成的目标数据, 每一列为一段
end
```

```

temp = []; %清空temp

%初始化替换后的目标数据
for i=1:original_num
    Target_after(:,i) = Target; %每一列为目标数据,第三维为原始数据
    个数
end

%生成替换后的目标数据
for i=1:original_num
    for j=1:target_num
        replace_index(i,j) = ceil(rand*target_length(1,j));
        replace_value(i,j) = ceil(rand*20);
        Target_after(replace_index(i,j),j,i) = replace_value(i,j);
    end
end

%生成插入下标,0表示不插入

Insert_index = zeros(original_num,target_num); %初始化插入位置下标
for i=1:original_num
    index = 0; %当前插入下标

    last_index = 0; %记录上一次插入下标
    for j=1:target_num
        temp = rand;
        if temp>=0.5
            overlap_flag = 0; %下标重叠标志位
            while(overlap_flag==0)
                index = ceil(rand*(original_length-target_length(1,j))); %
                随机生成下标

                %如果当前生成下标与上一次生成下标差大于目标数据长度,则生成有效,
                防止覆盖上一次插入值
                if(abs(index-last_index)>target_length(1,j))

```

```

        Insert_index(i,j) = index;
        overlap_flag=1;
        last_index = index;
    end
end
end
end
end
temp = []; %清空temp

%将目标数据随机插入到原始数据中
for i = 1:original_num
    for j=1:target_num
        if Insert_index(i,j) ~= 0 %只插入下标不为0的

Origin_Data(Insert_index(i,j):Insert_index(i,j)+target_length(1,j)-1,i) =
Target_after(1:target_length(1,j),j,i);
        end
    end
end

%%以上为随机生成的模型代码

%%以下为自动搜索算法的代码

%开始计时
tic

%采样参数

sample_length = 14; %单次采样长度

sample_count = original_length-sample_length+1; %采样次数

%存放采样后的矩阵
Origin_Data_sample = zeros(sample_length,sample_count,original_num);

%存放方差的矩阵

```

```

Origin_Data_var = zeros(original_num,sample_count);

%采样并计算方差
for i=1:original_num
    for j=1:sample_length
        for k=1:sample_count
            Origin_Data_sample(:,k,i) = Origin_Data(k:k+sample_length-1,i);
            Origin_Data_var(i,k) = var(Origin_Data_sample(:,k,i),1);
        end
    end
end

%计算方差分布

var_divide_num = 13; %划分端点数，划分段数=段点数-1

max_var = max(max(Origin_Data_var)); %最大方差值

min_var = min(min(Origin_Data_var)); %最小方差值

var_divide_point = linspace(min_var,max_var,var_divide_num); %计算分割
点
var_divide_center = zeros(1,var_divide_num-1);
for i=1:var_divide_num-1
    var_divide_center(i) = mean(var_divide_point(i:i+1)); %计算分割中心
end
var_distribution_num = zeros(original_num,var_divide_num-1);
for i=1:original_num
    var_distribution_num(i,:) =
hist(Origin_Data_var(i,:),var_divide_center); %进行分割
end

%方差分布直方图
% figure(1)
% bar(var_divide_center,A_var_distribution_num);
% figure(2)
% bar(var_divide_center,B_var_distribution_num);
% figure(3)

```

```

% bar(var_divide_center,C_var_distribution_num);
%
%对方差及其下标进行排序

var_order = zeros(original_num,sample_count);
var_index_order = zeros(original_num,sample_count);
for i=1:original_num
    [var_order(i,:),var_index_order(i,:)] = sort(Origin_Data_var(i,:));
end

%生成下标重排序矩阵，按照方差分布排序，便于后续寻找下标

for i=1:original_num
    for j=1:var_divide_num-1
        for k=1:var_distribution_num(i,j)
            var_index_reorder(k,j,i) =
var_index_order(i,sum(var_distribution_num(i,1:j-1))+k);
        end
    end
end

%两两按方差分布比较,位于同一方差分布段内的才进行比较，减少比较次数

compare_count = 1;
similar_count = 1;
similar_num = 0;
for x=1:original_num-1
    for y=x+1:original_num
        for n=1:var_divide_num-1
            for i=1:var_distribution_num(x,n)
                for j=1:var_distribution_num(y,n)

                    error_count = 0; %初始化错误计数位

                    for k=0:sample_length-1
                        if Origin_Data(var_index_reorder(i,n,x)+k,x) ~=
Origin_Data(var_index_reorder(j,n,y)+k,y) %如果不相等则error_count+1

                            error_count = error_count+1;
                        end

                        if error_count>=4 %error_count>=4时，跳出
循环

```

```

        break;
    end
    if k == sample_length-1    %k到达最大值，判断
        AB相似，则记录下标

        similar_index(similar_count,compare_count) = var_index_reorder(i,n,x);

        similar_index(similar_count,compare_count+1) = var_index_reorder(j,n,y);

        similar_index(similar_count,compare_count+2) = var_index_reorder(i,n,x)-
        var_index_reorder(j,n,y);
        similar_count = similar_count+1;
        similar_num = similar_num+1;
    end
end
end
end
end
    compare_count = compare_count+3;
    similar_count = 1;
end
end

%计时结束
toc

```