2018 软件工程优才夏令营测试题

数据驱动的测试行为分析系统

题目描述:

在企业的产品生产中需要采购大量的电子元件，这些电子元件的质量可以通过全部或者抽样的测试方式进行检验。测试人员会测量并记录电子元件产品的关键参数，例如电容、导电率等，作为这些电子元件等级和判断的依据。在以往人工测试记录过程中，也有测试人员出于节省时间和资源，可能会出现简化质检过程，编造假数据的行为，这对于最终产品的质量带来了危害。

请你开发一个简单的数据真实性检测程序，可以对输入的数据文件进行分析，判断是否有数据复制的造假行为存在。你的程序应该具有图形化界面，运行起来后像下面这样:

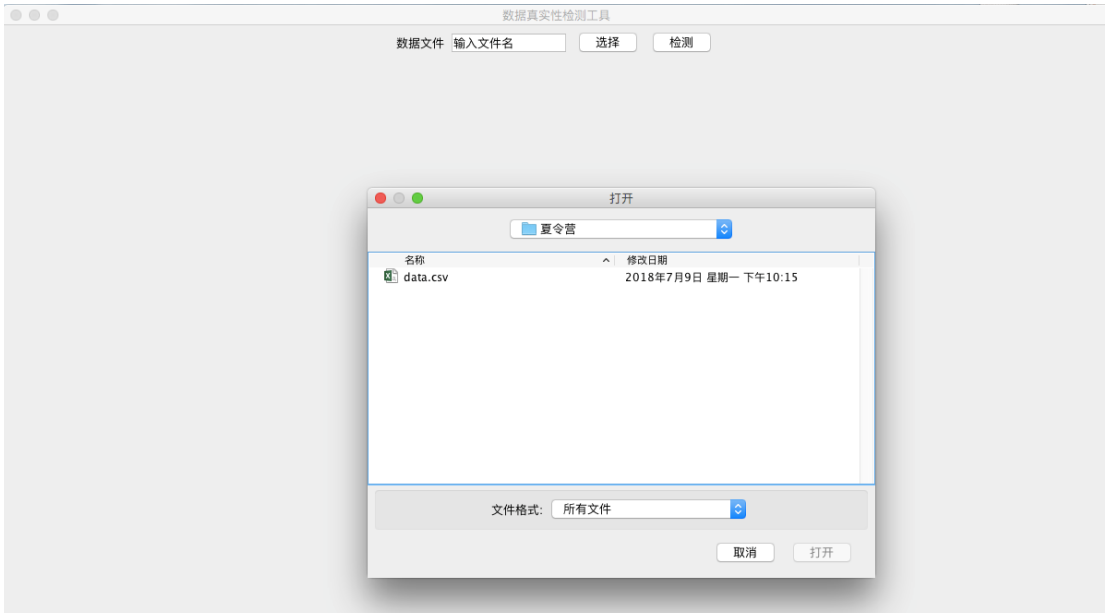

图 1 初始界面

在进行数据真实性检测时，用户需要选择一个目标文件。用户在选择时，可以直接在文本框中输入文件名，也可以点击 选择 按钮，通过弹出的文件选择对话框来选择文件。在选择好目标文件后，点击 检测 按钮，就执行数据真实性检测，并按照图 2 所示的方式显示图形化的分析结果:

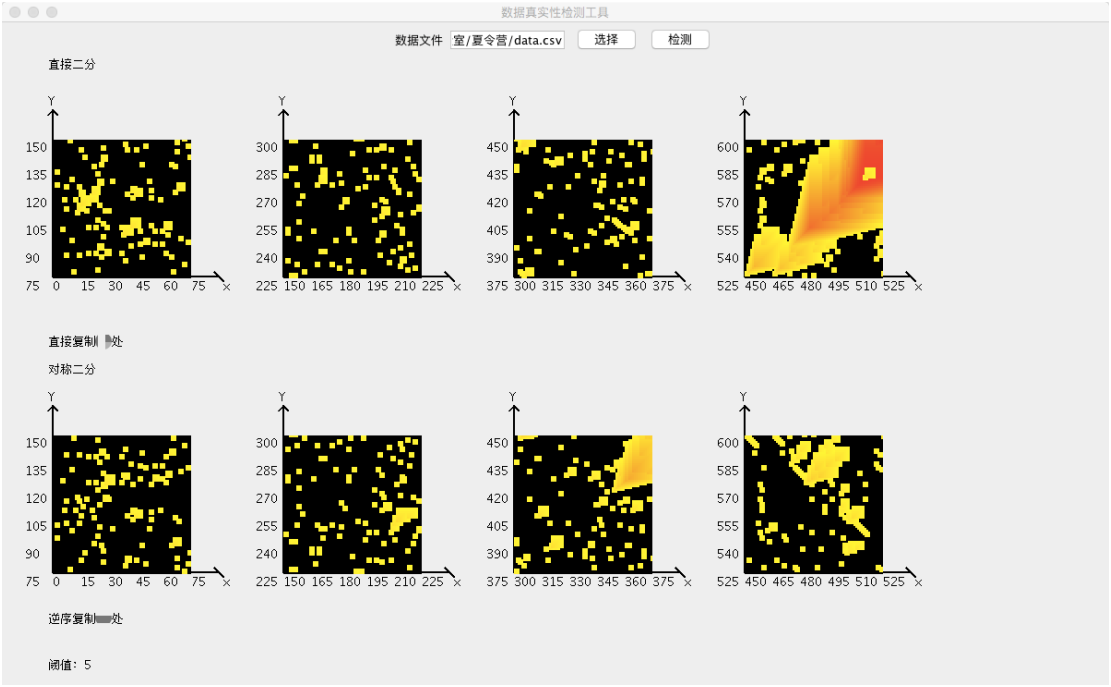

图 2 结果展示界面

分析结果显示的要求为:

1. 将数据切分为 4 个等长区间；将每个区间分为 2 个子序列，得到两个子序列的相似度矩阵，以图表（灰度图、热力图等均可）的方式可视化展示每个区间中序列的相似程度。例如，用一个子序列作为横坐标，另一个作为纵坐标，用灰度表示相似度。
2. 使用两种子序列拆分方式检验两种不同的数据复制行为:
   a) 直接二分，将区间从中间分为两个等长的子序列。这种拆分方式用于检测一种按顺序直接复制的行为。例如，在图 2 中的第一行第四个图表，可以看出第 510-525 行数据与 585-595 行数据的相似度程度极高，可能存在将 510-525 行数据直接复制到 585-595 行的数据造假行为。
   b) 对称二分，将区间从中间分为两个等长序列后，将其中一个序列反转，即令所有 $a_i=a_{n-1-i}$。这种拆分方式用于检测一种按逆序复制的行为。例如，在图 2 中的第二行第三个图表，可以看出第 360-375 行数据与第 435 行-450 行数据的相似程度极高，可能存在将将第 360-375 行数据逆序复制到第 375-390 行的数据造假行为。

   为了检测是否存在这两种典型的复制行为，请分别用这两种拆分方式计算并展示结果。
3. 当序列相似度超过某一阈值时，即可认定为复制造假行为。在图表结果显示区域下方，显示统计数据，包括直接复制和逆序复制各有多少处，和使用的阈值。注意，在进行统计时，每一对数据算作一处。根据经验，可使用 5 作为初始阈值。有余力的同学，可以利用统计学知识通过合理的方法生成动态阈值。
4. 最后给出你认为的存在造假行为的数据区间和阈值的优化方法（若有），在验收上机测试答案时需做说明。

考核要求:

---

1. (15 分)能够实现打开由用户指定的源文件，具体要求为:
   - (5 分)对文件类型进行检查:源文件被限定为 csv 文件，如果用户选择其他类型的文件，应该提示用户重新选择。
   - (5 分)输入文件名打开文件:支持用户在文本框中直接使用输入文件名打开文件。
   - (5 分)通过文件选择对话框打开文件:支持用户通过在文件选择对话框中选择并打开文件
2. (30 分)可视化显示分析结果，具体要求为:
   - (10 分)能够按照题目要求显示横纵坐标。
   - (10 分)能够按照题目要求以图的形式显示相似程度。
   - (10 分)能够展示对应数量的图表。
3. (55 分)对数据文件进行分析，具体要求为:
   - (15 分)能够实现直接二分的拆分方式，按数据原始顺序展示正确的相似度图表。
   - (10 分)能够正确统计原始阈值下直接复制的部分。
   - (15 分)能够实现对称二分的拆分方式，按数据原始顺序展示正确的相似度图表。
   - (10 分)能够正确统计原始阈值下逆序复制的部分。
   - (5 分)能够设计合理的阈值计算方法。

考试要求:

---

1. 编程语言:C、C++、Java、Python 或其他，你可以根据自己意愿选择。
2. 运行环境:使用自己的笔记本电脑，集成开发环境可以自选。
3. 提交方式:将你所编写的工程压缩后命名为“身份证号_姓名.rar”，例如“9999999999999900_张三.rar”，拷贝到发放题目的 U 盘上，考试结束时上交。请仔细检查是否拷贝成功。
4. 上机考试时间:17:30-21:30，考题结束后提交 U 盘后方能离开考场。
5. 批改与验收方式:考试结束后，按照名单顺序验收考试答案，其中答案以提交的 U 盘上的版本为准。验收时，将 U 盘上答案拷贝到你的电脑上，然后按照指令运行你的程序，并回答问题。

提示:

---

1. 你可以使用给你提供的 **data.csv** 来进行测试。
2. 为了避免在可视化调试上浪费过多时间，我们建议将窗口大小设置为 **1200*800** 像素，将单个图表大小设置为 **200*200** 像素。
3. 这个问题可以借助求最长公共子串**(LCS)**的算法来实现，计算 **LCS** 的算法很多，但是时间复杂度和空间复杂度基本上都是 **O(M*N)**，其中 **M** 和 **N** 分别是两个字符串的长度。本题中需要对局部序列进行比对，下面给出计算 相似度矩阵 **H** 的 **Smith-Waterman** 算法的描述:
   设:
   1) **A=$a_1a_2.....a_N$**，表示 **A** 是由 **$a_1a_2.....a_N$** 这 **N** 个字符组成，**Len(A)=N**
   2) **B=$b_1b_2.....b_M$**，表示 **B** 是由 **$b_1b_2.....b_M$** 这 **M** 个字符成，**Len(B)=M**

   定义置换矩阵**S(a,b)**，有公式:

   1) 若**a=b**，则**S(a,b)=3**

   2) 若**a≠b**，则**S(a,b)=-1**

   定义空位罚分**W=1;**

   定义相似度矩阵**H**大小为**N+1**行**M+1**列，**H(i, j)=H($a_1a_2.....a_i$, $b_1b_2.....b_j$)**,其中 **$0 \le i \le N, 0 \le j \le M$**

   对于 **$1 \le i \le N, 1 \le j \le M$**,有公式:

   $H(i, j)=Max(H(i-1, j-1) + s(a_i,b_j), H(i-1, j) - W, H(i, j-1) – W, 0)$

   例如, **A=TGTTACGG, B=GGTTGA**，计算 **H** 的步骤如下:
   第一步:初始化 **H** 矩阵

		T	G	T	T	A	C	G	G
	0	0	0	0	0	0	0	0	0
G	0								
G	0								
T	0								
T	0								
G	0								
A	0								

第二步:利用公式,计算矩阵的第一行

		T	G	T	T	A	C	G	G
	0	0	0	0	0	0	0	0	0
G	0	0	3	2	1	0	0	3	3
G	0								
T	0								
T	0								
G	0								
A	0								

第三步:利用公式，计算矩阵的其余各行

		T	G	T	T	A	C	G	G
	0	0	0	0	0	0	0	0	0
G	0	0	3	2	1	0	0	3	3
G	0	0	3	2	1	0	0	3	6
T	0	3	2	6	5	4	3	2	5
T	0	3	2	5	9	8	7	6	5
G	0	2	6	5	8	7	6	10	9
A	0	1	5	4	7	11	10	9	8

则得到相似度矩阵