SSVEP无训练数据集

本赛题基于稳态视觉诱发电位设计多目标键盘拼写实验，采集多人数据集，旨在测试不同算法在无训练条件下的识别性能。

实验范式：

实验范式为键盘拼写，刺激包含40个目标（包括数字0-9、字母A-Z、“，”、“.”、空格和退格），刺激频率分别为8~15.8Hz，频率间隔0.2Hz，相邻频率之间相位差0.5π，具体如图1及表1所示。实验中，固定频率、相位的刺激通过对屏幕上每个目标亮度的正弦调制实现（亮度范围：0-255）。每个试次包含提示1秒，刺激4秒，反馈1秒。一个block共40个试次，每次实验共6个block。被试视力或矫正视力正常，在实验过程中被要求集中注意力，刺激时注视提示的目标。实验中各个试次使用光trigger进行同步，trigger标记在每个刺激开始时刻。



图1 40目标刺激界面

表1 标签、刺激频率、相位对照表

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **标签** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| 频率 | 8 | 8.2 | 8.4 | 8.6 | 8.8 | 9 | 9.2 | 9.4 | 9.6 | 9.8 |
| 相位 | 0 | 0.5π | π | 1.5π | 0 | 0.5π | π | 1.5π | 0 | 0.5π |
| **标签** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** |
| 频率 | 10 | 10.2 | 10.4 | 10.6 | 10.8 | 11 | 11.2 | 11.4 | 11.6 | 11.8 |
| 相位 | π | 1.5π | 0 | 0.5π | π | 1.5π | 0 | 0.5π | π | 1.5π |
| **标签** | **21** | **22** | **23** | **24** | **25** | **26** | **27** | **28** | **29** | **30** |
| 频率 | 12 | 12.2 | 12.4 | 12.6 | 12.8 | 13 | 13.2 | 13.4 | 13.6 | 13.8 |
| 相位 | 0 | 0.5π | π | 1.5π | 0 | 0.5π | π | 1.5π | 0 | 0.5π |
| **标签** | **31** | **32** | **33** | **34** | **35** | **36** | **37** | **38** | **39** | **40** |
| 频率 | 14 | 14.2 | 14.4 | 14.6 | 14.8 | 15 | 15.2 | 15.4 | 15.6 | 15.8 |
| 相位 | π | 1.5π | 0 | 0.5π | π | 1.5π | 0 | 0.5π | π | 1.5π |

实验数据：

实验数据使用博睿康8通道脑电采集设备采集，第9导联为trigger信息，原始采样率为1000Hz，数据集中降采样到**250Hz**，未做其他滤波处理。**线上测试时赛题数据的试次开始trigger全部为1**，仅用作同步，不包含试次具体信息。除试次开始trigger之外数据还包含其他trigger用于系统控制，具体trigger定义如表3所示，参赛者可只关注trial开始trigger。

表2 导联序号-名称对照表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 导联序号 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 导联名称 | ‘POz' | ‘PO3' | ‘PO4' | ‘PO5' | ‘PO6' | ‘Oz' | ‘O1' | ‘O2' | ‘Trigger’ |

表3 系统trigger定义

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 定义 | Trial开始 | Trial结束 | Block开始 | Block结束 | 系统预留 |
| Trigger号 | 1-240 | 241 | 242 | 243 | 244-255 |

数据流采用模拟在线方式提供。每调用一次数据读取方法，可获得一个新数据包，数据包中包含40ms的实验EEG数据（最后一个数据包长度可能小于40ms），以及在该数据包记录过程中收到的trigger信息。在同一block中，数据包按照时间顺序依次发送。若测试数据中包含多组block数据，则一组block数据发送完毕后，数据读取方法被再次调用时，将会开始下一组block数据的EEG。而当所有实验数据发送完毕后，程序终止标记finishedFlag将被置为1。参赛算法检测到finishedFlag为1后，需要自行结束run()方法执行。需要指出，由于实验数据来自真实EEG信号，每个block中最后一个数据包的长度可能不是一个定值，在算法开发过程中请特别注意。

算法规范：

参赛算法调用数据读取方法获取脑电数据。数据读取方法被调用一次，比赛系统会返回一个新数据包，参赛算法可以对新数据包进行缓存并处理。当算法认为接收到的数据足以满足判决条件时，需要调用反馈方法向比赛系统报告识别结果。比赛系统根据数据读取方法的调用次数计算出算法使用有效数据长度，并结合反馈正确率，综合计算出模拟信息传输速率。

参赛算法需要同时满足以下几个约束条件：

1. 试次起止约束：

在对单一试次数据的检测识别过程中，参赛算法需要在接收到该试次trigger之后开始检测，并且最迟在接收到下一trigger时进行反馈报告。否则，报告结果将被错判为后续试次的识别结果。

1. 单试次最大数据长度约束：

本项目对于单一试次最长检测时间需小于4秒。从试次的trigger信号起，参赛算法最多采集4秒的EEG数据（不含4秒），就必须立刻给出反馈结果，否则该试次识别结果将被视为无效，该试次结果为0。不同被试，不同试次的检测时间可以不同。

1. 算法终止约束:

当接收到数据包中Endflag = 1时，意味着所有实验数据均已发送完毕，参赛算法需要停止处理并自行退出。

赛题框架

参赛者用例



图2 参赛者用例

系统主体框架如图3所示。



图 3 系统主体框架

1. FrameworkInterface框架接口

该接口主要负责赛题程序与外部执行系统的连接。该框架接口的实现类为AlgorithmSystemManager，实现接口定义的所有函数。

1. ProblemInterface题目接口

该接口是**面向参赛者**的赛题接口，主要负责题目与参赛算法之间的数据传递及结果报告。参赛者可以通过该接口获取比赛数据，并通过该接口报告识别结果。比赛题目需要根据参赛算法获取比赛数据的次数，以及报告结果的正确率综合给出比赛评分。

1. TaskInterface赛题接口

该接口主要负责实现赛题的数据填充、获取得分、清除数据以及清除报告结果，框架通过该接口实现对赛题的调用。

1. AlgorithmInterface算法接口

通过该接口比赛题目可以对参赛算法进行验证计算。**参赛者需要实现该接口**。在执行过程中，参赛算法需要通过ProblemInterface接口获取数据，并且通过该接口报告结果。同时，参赛者需要控制算法的计算复杂度，否则当运行时间超过预定长度时，系统将自动终止该计算进程，所获成绩无效。

数据模型

1. DataModel参赛者数据模型
2. data：float类型矩阵，分段数据。例如包含有64导EEG数据+1导trigger信号，在250Hz采样率下，以40ms对数据分段，则单次获取的data为65\*10个点。
3. startPosition: int类型标量，当前分段数据起始时刻相对于该block数据起始时刻的索引位置。
4. personID：int类型标量，当前数据来源受试者序号。
5. finishedFlag：bool类型标量，测试结束标志。当参赛算法通过ProblemInterface.getData()函数获取数据包中该字段为True时，需要自行退出程序运行。

参赛者相关接口函数

(1) ProblemInterface

该接口由出题方负责实现，包括数据获取方法及结果反馈方法。在算法运行前，该接口的实现类会被注入参赛算法实现类中。算法执行过程中，可以调用该接口获取数据，并通过结果反馈方法报告识别结果。出题方根据数据获取方法的被调用次数，及结果反馈的正确性进行综合评分。

1) def getData(self):

输入参数：无

输出参数：DataModel

实现功能：获取下一分段实验数据。

2) def report(self, reportModel):

输入参数：ReportModel

输出参数：无

实现功能：反馈识别结果。

(2) AlgorithmInterface

参赛者需要将程序运行过程填入run函数中。在算法执行过程中，通过ProblemInterface接口getData方法获取DataModel类型数据，并通过report方法返回ReportModel类型结果。当通过getData获取的DataModel数据中finishedFlag为true时，意味着数据处理完毕，该函数需要自行退出运行。

1) def run(self):

输入参数：无

输出参数：无

实现功能：算法分析过程。

提交格式

本赛题程序使用python语言编写，需提交基于python 3.8版本的扩展名为.pyc的文件。

提交样例

参考配套代码。

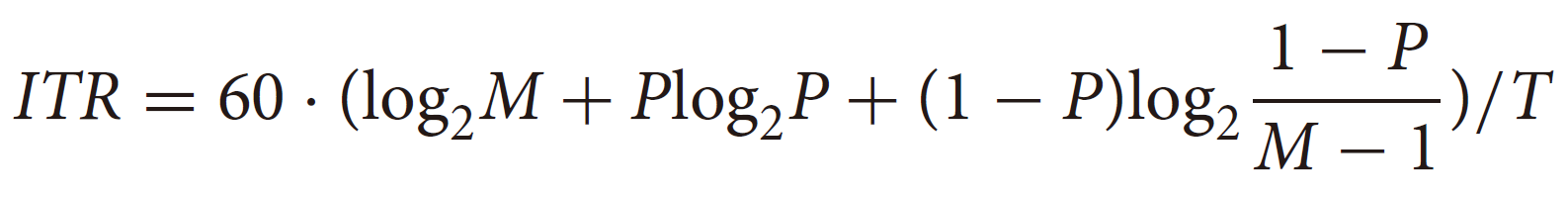
参赛者可通过修改Algorithm文件夹中的代码完成算法，为了避免未知错误，请勿在主目录内添加文件夹。完成后重新打包程序（包含AlgorithmImplement文件夹和config.toml）--> 分组 --> 具体分组 --> 计算单元 --> 定义计算单元 --> 上传程序包 --> 提交到比赛 --> 选择比赛 --> 部署 -->完成比赛。

部署完成后在具体到比赛中的排行榜中查看 比赛成绩；

需要注意的是，为防止参赛者修改代码框架作弊，保护评分程序会完全覆盖参赛者的代码（除了AlgorithmImplement 目录和config.toml） 在提交到比赛 --> 部署时 ，启动的实际为 评分程序 + 参赛者的AlgorithmImplement 目录，其余运行配套代码均为服务器内置程序（包括main.py等文件，服务器内置评分程序与范例中程序框架基本相同，但包含评分功能和读取服务器比赛数据功能）。

评分方式

本系统以模拟信息传输速率作为评分标准:



其中，T表示平均试次时长，M表示目标个数，P表示识别正确率。ITR的单位是bits/min。特别需要指出，本系统中ITR是按照理想ITR进行计算，即平均试次时间不包含模拟休息时长。

性能评估方法

参赛算法通过数据读取方法获取新数据包。当所得数据包内含有trigger信号时，评分系统将自动开始记录算法识别过程中所使用EEG信号的长度，直至反馈方法被调用。从trigger开始到反馈方法被调用时所获取的EEG数据长度将作为该试次的模拟试次时长。而平均准确率将根据算法反馈结果与真实刺激的一致性进行计算。

需要特别指出，在本比赛项目中每一个包含trigger的数据包，其依然被视为是前一试次的数据。 而新试次数据是从包含trigger数据包的下一个数据包开始计算。因此参赛算法不可在获取到包含trigger信号的数据包时立刻反馈，而最早需获取到下一数据包后才可反馈。

结果反馈异常处理

1)重复多次报告

在一个试次时间内，参赛算法多次反馈结果，则仅按照第一次反馈的时间及结果进行记录。

2)结果未反馈

若在一个试次时间内，参赛算法未反馈结果，则该试次时长将被记录为4秒，同时判决结果将被记录为误判。

3)结果反馈超时

结果反馈时，从当试次trigger开始计算参赛算法已经获取了超过4秒的EEG数据，则判决结果将被记录为误判，同时该试次时长将按照从trigger开始时刻到结果反馈时刻的数据长度进行计算。

4)算法执行超时

为满足脑-机接口系统实时处理需求，本项目同时对参赛算法的计算复杂度有一定要求。本比赛项目将会根据比赛数据量大小确定一个计算时间。若算法复杂度过高导致系统运行超时，则该算法比赛成绩将被视为无效。