

基于微创脑机接口的肢体运动意图辨识技术赛项说明

本赛题基于猪运动区 ECoG 信号, 开发肢体运动意图辨识算法, 旨在推进微创脑机接口技术的数据应用。

一、实验说明

本赛题同步采集猪的运动脑区 ECoG 数据和右侧前后肢关节运动角度数据, 初赛考察前肢运动意图辨识情况。猪在跑台以时速 1.2km/h、1.5km/h、1.8km/h 运动, 获取其在不同速度下的脑电和关节矢状面角度数据集, 其中, 脑电数据为 63 通道, 前肢关节角度数据为 4 通道, 依次为肩关节 (Sho)、肘关节 (Elb)、腕关节 (Wri)、掌指关节 (Mcp), 实验流程如图 1 所示。

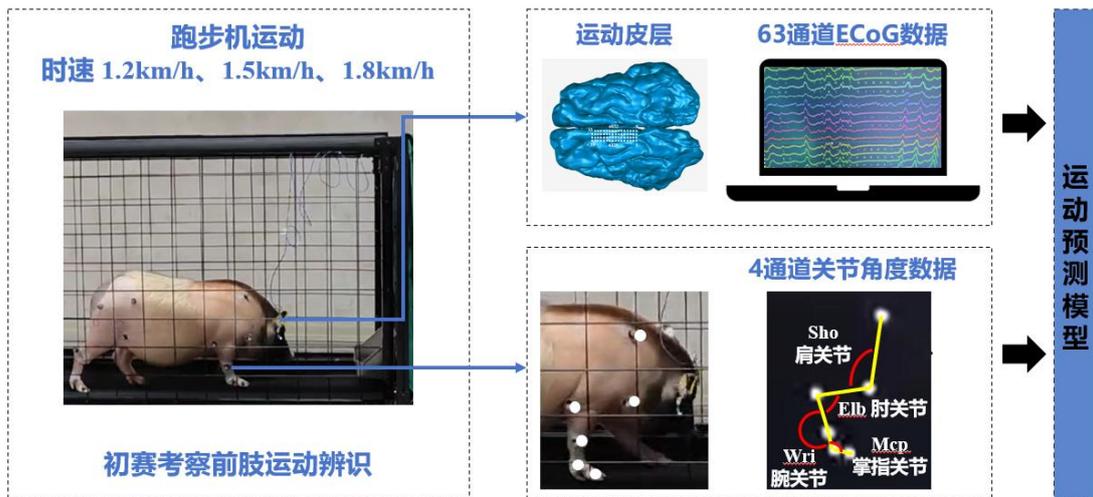


图 1 实验流程

二、数据说明

本赛题脑电数据采样率为 2kHz, 关节角度数据采样率为 200Hz, 对于提供给参赛方的训练数据, 其中脑电未做降采样或其他滤波处理。

脑电数据的导联排布规则如图 2 所示, 1-63 通道, 命名为 ch1-ch63。

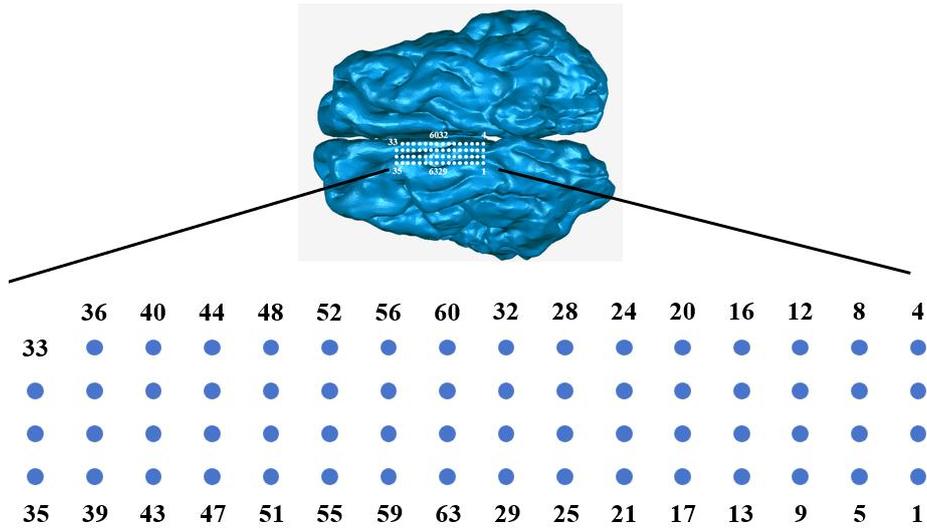


图 2 脑电导联排布规则

训练集数据包含 1.2km/h 数据 22min, 1.5km/h 数据 12min, 1.8km/h 数据 18min。每一分钟为 1 个 block, 形成一个.pkl 文件, 文件 1-4 行为关节角度数据, 依次对应肩关节(Sho)、肘关节(Elb)、腕关节(Mcp)、掌指关节(Wri), 5-67 行为脑电数据, 最后一行 68 行为 trigger 信息。由于脑电的采样率为 2kHz, 关节的采样率为 200Hz, 在 1min 中的数据中, 脑电为 120000 列, 关节为 12000 列。关节数据置于前 12000 列, 后面补 0。数据规则如图 3 所示。

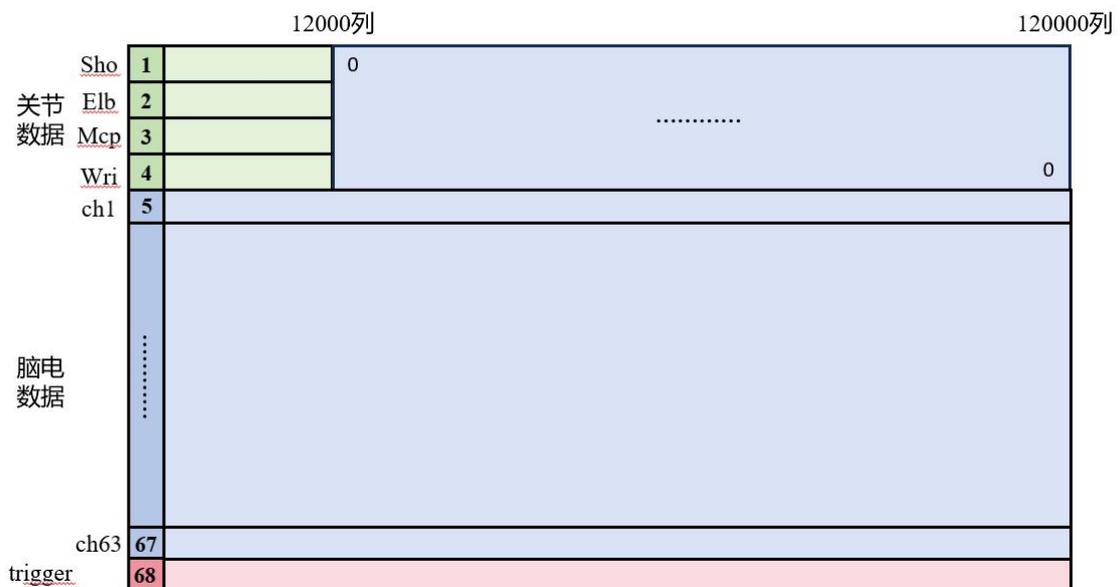


图 3 训练集数据排布规则

比赛中测试集数据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 ECoG 数据，以及在该数据包记录过程中收到的 trigger 信息，关节数据在选手端被屏蔽掉，全部设置为 0。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 数据的发送。而当所有实验数据发送完毕后，程序终止标记 finish_flag 将被置为 true。参赛算法检测到 finish_flag 为 true 后，需要自行结束 run() 方法执行。一组 block 数据为 1min，包含 20 个 trial (3s)。每个 trial 需提交一次结果，返回这一个 trial 时间内的 4 组关节角度数据 (以 200Hz 的时间分辨率反馈)。计算 20 个 trial 的平均结果作为一个 block 的成绩。比赛流程如图 4 所示。

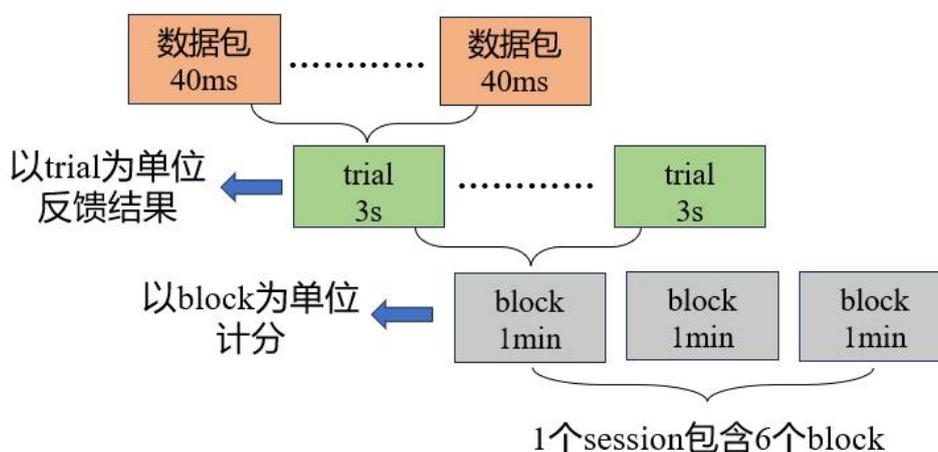


图 4 比赛流程

表 1 Trigger 定义

定义	Trial 开始	Trial 结束	Block 开始	Block 结束
Trigger 号	240	241	242	243

三、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。比赛系统根据返回结果计算出皮尔森相关系数。

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(240)之后开始检测，并且在接收到该试次结束 trigger(241)时进行反馈报告。否则，报告结果将被判定为无效的识别结果。

2、单试次最大数据长度约束：

从试次开始的 trigger 信号开始，参赛算法给出反馈结果使用的数据长度不应超过试次结束的 trigger 信号，否则该试次识别结果将被视为无效，该试次结果为 0。

3、算法终止约束：

当接收到数据包中 finish_flag = true 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。

四、赛题框架

1、参赛者用例

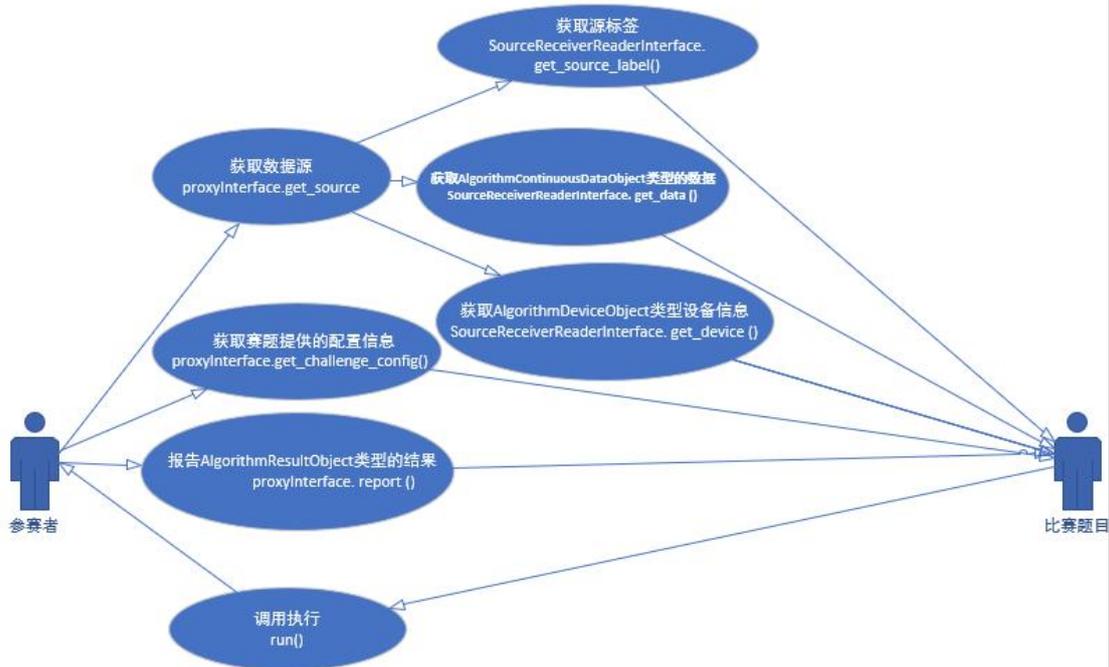


图 5 参赛者用例

2、系统框架

(1) ProxyInterface 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，还可以获取赛题提供的配置信息。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报告结果的正确率给出比赛评分。

(2) ChallengeInterface 赛题接口

该接口主要负责实现赛题的清洗数据和计算得分，框架通过该接口实现对赛题的调用。

(3) 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。该接口由参赛者具体实现。在执行过程中，参赛算法需要通过 ProxyInterface 接口的 get_source 获取

数据源，继而通过获取的数据源 SourceReceiverReaderInterface 的 get_source_label() 方法获取源标签，SourceReceiverReaderInterface 的 get_data () 方法获取 AlgorithmContinuousDataObject 类型的数据，SourceReceiverReaderInterface 的 get_device () 方法获取 AlgorithmDeviceObject 类型设备信息，并且通过 proxyInterface.report () 报告 AlgorithmResultObject 类型结果。AlgorithmResultObject 类型结果中，需要根据赛题要求返回指定类型的数据。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。

3、数据模型

(1) AlgorithmContinuousDataObject 参赛者数据模型

1) start_position 变量

int 类型；数据包内数据的起始位置

2) data 变量

numpy.ndarray 类型；数据内容，每行表示一个导联，最后一行为 trigger 通道。例如包含 4 导关节 + 63 导脑电 + 1 导 trigger 信号，在 200Hz 关节采样，2kHz 脑电采样率下，以 1min 对数据分段，则单次获取的 data 为 68*120000 个点（关节数据为前 4 行*12000 列，后面列补 0）。

3) subject_id 变量

str 类型；当前数据包的被试编号

4) finish_flag 变量

bool 类型；处理结束标志位，当检测到该标志位为 true 时，算法需终止运

行。

4、参赛者基于 AlgorithmInterface 的开发说明

参赛者需要根据 “Algorithm\method\interface\AlgorithmInterface.py” 中的 AlgorithmInterface 类实现算法类，并将算法类名称和路径写入 “Algorithm\config\AlgorithmConfig.yml” 文件中。

参赛者需在算法类中重写以协程的形式重写 run 方法，并在获取设备信息、获取数据和汇报结果时使用 await 关键字异步执行。框架运行时，会主动调用 run 方法。因此参赛者需要在 run 方法中读取数据，进行计算，并汇报结果。

其中，参赛者可以通过调用 _proxy 属性的 get_source 方法，并传入 str 类型的数据源名称参数，获取数据源 (SourceInterface 实例)。获取到数据源之后，可以使用 await 关键字异步调用其 “get_device” 方法获取设备信息 (AlgorithmDeviceObject 实例)，调用其 “get_data” 方法获取一个数据包 (AlgorithmDataObject 实例)。

在需要汇报结果时，参赛者可以实例化一个 AlgorithmResultObject 类型的实例，并将结果赋值给该对象的 “result” 属性。随后将该实例作为参数传入并调用 “_proxy” 属性的 report 方法。

以下是关于上文中出现的类的说明。

4.1 ProxyInterface

算法类与框架交互的代理类，在算法类实例化时，框架会实例化一个此类的对象放入算法类的 “_proxy” 属性中。

(1) get_source 方法

无传入参数。返回一个 SourceInterface 对象。用于算法从框架获取数据源

对象。

(2) report 方法

必须传入且仅传入一个 AlgorithmResultObject 对象。无返回值。用于算法向框架报告结果。

4.2 SourceInterface

(1) get_source_label 方法

无传入参数。返回 str 类型的数据源名称。

(2) get_data 方法

无传入参数。返回一个 AlgorithmDataObject 对象。用于算法从数据源获取数据。调用时需使用 await 关键字异步调用。

(3) get_device 方法

无传入参数。返回一个 AlgorithmDeviceObject 对象。用于算法从数据源获取数据采集设备的设备信息。调用时需使用 await 关键字异步调用

4.3 AlgorithmDeviceObject

(1) data_type 变量

str 类型；数据源中的数据类型。

(2) channel_number 变量

int 类型；数据包的通道数。

(3) sample_rate 变量

float 类型；数据的采样率。

(4) channel_label 变量

str 类型构成的 list 类型；数据各个通道的标签

(5) other_config_map 变量

dict 类型; 其他设备配置信息

4.4 AlgorithmResultObject

(1) result 变量

将 list 转为 str 类型后发送; 算法汇报的结果。

```
str(plist.flatten(order='F').tolist())
```

返回结果包含 4 种关节角度 (Sho、Elb、Mcp、Wri), 以一个 trial 为例, 共计 4*600 个结果, 按照 F 模式展开为一维数组, 然后转为字符串。

提交格式

本赛题程序使用 python 语言编写, 需提交基于 python 3.10.10 版本的扩展名为.py 的文件, 需进行源码复核。

5、评分方式

本系统以关节的真实轨迹与预测轨迹两条轨迹的 pearson 相关系数 r 作为评分标准。x=真实值, y=预测值, i=采样点。根据相关系数 r 的结果进行评分, r 越接近 1 的参赛团队为第一名, 依次排名。每个 trial 需提交一次结果, 分别计算 20 次提交的 trial 结果后取平均作为一个 block 的成绩。计算步骤如下:

(1) 计算一个试次的相关系数, 每个关节角度分别计算相关系数后对四个关节轨迹的相关系数求平均记为这一试次的相关系数。

$$r(\text{关节}) = \frac{\sum((x_i - \bar{x})(y_i - \bar{y}))}{\sqrt{\sum(x_i - \bar{x})^2 \cdot \sum(y_i - \bar{y})^2}} \quad r \in (-1, 1)$$
$$r = 1/4(r(\text{sho}) + r(\text{elb}) + r(\text{mcp}) + r(\text{wri}))$$

注: 1) 4 个关节轨迹的平均相关系数 $r < 0$ 记为 0;

2) 该试次未返回结果记为 0。

(2) 计算一个 block 的结果作为分数排名:

$$score = 1/20 (r1 + r2 + \dots + r20)$$

6、结果反馈异常处理

1)重复多次报告

在算法示例中，已经提供了读取脑电数据的代码，每获得一个完整的 trial 会调用一次_calculate 方法，初赛时参赛队在_calculate 中直接实现预测算法即可。在未再次调用 get_data 之前，当前 trial 可以重复报告，以最后一次报告结果为准计算评分。

2)结果未反馈

若一个 trial，参赛算法未反馈结果，该 trial 按照 0 分计算。

3)结果反馈超时

一个 trial 必须反馈一次结果，当未反馈当前 trial 结果，就继续获取新的数据时，当前 trial 反馈结果无效，按照 0 分计算。

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。

五、提交内容

入围决赛的队伍需提交如下内容:

1) 算法源代码及代码注释，交接代码时需本地能跑通和复现;

2) 算法报告 word 版和 pdf 版，包含**方法介绍和使用说明**两部分，**算法报告完整性影响最终排名。**

方法介绍（举例如下）：

(1) 传统机器学习方法

1. 方法介绍，包括：提取特征的类型（如：功率谱密度、均值方差等时域统计量、共空间模式等空域特征等）、计算步骤（含公式）、回归算法（如：多元线性回归、支持向量机回归算法、随机森林等集成回归算法）、算法的参数（如：SVM 的惩罚系数等）

2. 实验设置，包括：训练参数设置（如：XGBoost 的学习率等）、验证集的比例（如有）

(2) 深度学习方法

1. 方法介绍：网络的具体说明，包括网络的结构总览图、网络各个层的类型（如：卷积层或池化层等）参数设置（如：卷积核个数和尺寸等）；

2. 实验设置，包括：优化器类型、优化器参数（如：学习率、动量等）、损失函数类型（如：均方误差等）、批大小、验证集的比例（如有）、训练策略（如：早停机制、保存最优权重等）

使用说明：

1. 运行代码所需的库以及安装命令（如：需安装 mne，命令为 `pip install mne`）

2. 运行代码的命令（如：假设代码名字为 `test.py`，通过命令行输入命令 `python test.py` 运行）以及是否需要额外命令行参数（如有）

3. 项目结构

六、决赛名额与奖项设置

1) 本赛项获奖赛队数量为 22 支队，选取初赛前 10 支队伍在北京“2024 世界机器人大赛锦标赛（北京）”现场参加决赛。决赛中根据分数和算法报告完整

性进行排名，设特等奖 1 个、一等奖 3 个、二等奖 6 个，三等奖 12 个。

2) 本竞赛项目奖项设置如下：

特等奖 1 名：奖金 30000 元，获奖证书；

一等奖 3 名：奖金 10000 元，获奖证书；

二等奖 6 名：奖金 2500 元，获奖证书；

三等奖 12 名：获奖证书；

优秀组织奖若干，奖金总额 25000 元。

奖金为税前金额。