SSVEP_algorithms

Github的在线Markdown公式编译模块有bug，想看公式过程的建议下载**算法说明**文件本地查看。鉴于部分公式推导步骤属于本人硕士学位论文内容，在此提醒各位，ctrl+C/V 请慎重。

README.html、README.md、**README.pdf** 以及 **算法说明.pdf** 内容都是一样的，我只是在测试导出文本文件的不同方法……

目前发现在未安装 *Latex* 插件的平台上浏览 **html** 以及 **md** 文件会出现公式无法正常显示的问题，所以建议观看pdf文件。

更新进展：今天写了msTRCA部分的公式说明（2022/7/23）
近期计划：编写eCCA、msCCA的说明内容。

教资科目二怎么这么难背的，吐了

建议各位同僚读完硕士赶紧去就业吧，千万不要盲目读博、投身火海。

公式变量符号及说明

符号名称	物理含义
$\boldsymbol{\chi}$	EEG测试数据矩阵
$\boldsymbol{X}$	EEG训练数据矩阵
$\boldsymbol{x}$	EEG训练数据序列
$\boldsymbol{Y}$	人工构建正余弦模板
N_e	刺激类别数
N_t	训练样本数
N_c	导联数
N_p	单试次采样点数
N_h	正余弦信号谐波个数
N_k	保留子空间个数
$\boldsymbol{X}^i, \boldsymbol{x}^i$	第 i 试次或第 i 导联数据，详见各部分具体说明
$\boldsymbol{X}_k, \boldsymbol{x}_k$	第 k 类别数据
$\bar{\boldsymbol{X}}_k, \bar{\boldsymbol{x}}_k$	类别样本中心，由 $\boldsymbol{X}_k$ 或 $\boldsymbol{x}_k$ 按试次叠加平均获得

符号名称	物理含义
$\bar{\bar{\boldsymbol{X}}}, \bar{\bar{\boldsymbol{x}}}$	总体样本中心，由 $\bar{\boldsymbol{X}}_k$ 或 $\bar{\boldsymbol{x}}_k$ 按类别叠加平均获得
f_s	EEG 信号采样率
$\boldsymbol{\omega}, \boldsymbol{U}, \boldsymbol{V}$ ...	低维空间滤波器
$\boldsymbol{W}$	高维空间滤波器，由低维空间滤波器集成获得

（在无特殊说明的情况下，所有训练数据默认经过了零均值化处理）

常见SSVEP信号处理算法（空间滤波器）

1. 典型相关性分析

Canonical correlation analysis, CCA

1.1 标准CCA：CCA

论文链接 | **代码**：cca.cca()

对于第 k 类别、第 i 试次数据 $\boldsymbol{X}_k^i \in \mathbb{R}^{N_c \times N_p}$，其对应频率的人工构建正余弦模板 $\boldsymbol{Y}_k \in \mathbb{R}^{(2N_h) \times N_p}$ 可表示为：

$$\boldsymbol{Y}_k = \begin{pmatrix} \sin(2\pi fn) \\ \cos(2\pi fn) \\ \sin(4\pi fn) \\ \cos(4\pi fn) \\ ... \\ \sin(2N_h \pi fn) \\ \cos(2N_h \pi fn) \end{pmatrix}, n = [\frac{1}{f_s}, \frac{2}{f_s}, ..., \frac{N_p}{f_s}] \tag{1-1}$$

CCA的优化目标为 $\hat{\boldsymbol{U}}_k^i$ 和 $\hat{\boldsymbol{V}}_k^i$，使得一维信号 $\hat{\boldsymbol{U}}_k^i \boldsymbol{X}_k^i$ 与 $\hat{\boldsymbol{V}}_k^i \boldsymbol{Y}_k$ 之间相关性最大化，其目标函数为：

$$\hat{\boldsymbol{U}}_k^i, \hat{\boldsymbol{V}}_k^i = \underset{\boldsymbol{U}_k^i, \boldsymbol{V}_k^i}{\arg\max} \frac{Cov(\boldsymbol{U}_k^i \boldsymbol{X}_k^i, \boldsymbol{V}_k^i \boldsymbol{Y}_k)}{\sqrt{Var(\boldsymbol{U}_k^i \boldsymbol{X}_k^i)} \sqrt{Var(\boldsymbol{V}_k^i \boldsymbol{Y}_k)}} = \underset{\boldsymbol{U}_k^i, \boldsymbol{V}_k^i}{\arg\max} \frac{\boldsymbol{U}_k^i \boldsymbol{C_{XY}} {\boldsymbol{V}_k^i}^T}{\sqrt{\boldsymbol{U}_k^i \boldsymbol{C_{XX}} {\boldsymbol{U}_k^i}^T} \sqrt{\boldsymbol{V}_k^i \boldsymbol{C_{YY}} {\boldsymbol{V}_k^i}^T}} \tag{1-2}$$

$$
\begin{cases}
\boldsymbol{C_{XX}} = \dfrac{1}{N_p - 1} \boldsymbol{X}_k^i {\boldsymbol{X}_k^i}^T \in \mathbb{R}^{N_c \times N_c} \\
\boldsymbol{C_{YY}} = \dfrac{1}{N_p - 1} \boldsymbol{Y}_k {\boldsymbol{Y}_k}^T \in \mathbb{R}^{(2N_h) \times (2N_h)} \\
\boldsymbol{C_{XY}} = \dfrac{1}{N_p - 1} \boldsymbol{X}_k^i {\boldsymbol{Y}_k}^T \in \mathbb{R}^{N_c \times (2N_h)} \\
\boldsymbol{C_{YX}} = \dfrac{1}{N_p - 1} \boldsymbol{Y}_k {\boldsymbol{X}_k}^T \in \mathbb{R}^{(2N_h) \times N_c}
\end{cases} \tag{1-3}
$$

根据最优化理论，函数 (1-2) 的等效形式为：

$$
\begin{cases}
\max\limits_{\boldsymbol{U}_k^i, \boldsymbol{V}_k^i} \ \boldsymbol{U}_k^i \boldsymbol{C_{XY}} {\boldsymbol{V}_k^i}^T \\
s.t. \ \boldsymbol{U}_k^i \boldsymbol{C_{XX}} {\boldsymbol{U}_k^i}^T = \boldsymbol{V}_k^i \boldsymbol{C_{YY}} {\boldsymbol{V}_k^i}^T = 1
\end{cases} \tag{1-4}
$$

利用*Lagrandian*乘子法构建多元函数 $J(\boldsymbol{U}_k^i, \boldsymbol{V}_k^i, \lambda, \theta)$：

$$
J = \boldsymbol{U}_k^i \boldsymbol{C_{XY}} {\boldsymbol{V}_k^i}^T - \frac{1}{2}\lambda(\boldsymbol{U}_k^i \boldsymbol{C_{XX}} {\boldsymbol{U}_k^i}^T - 1) - \frac{1}{2}\theta(\boldsymbol{V}_k^i \boldsymbol{C_{YY}} {\boldsymbol{V}_k^i}^T - 1) \tag{1-5}
$$

对函数 J 求偏导数并置零、化简：

$$
\begin{cases}
\dfrac{\partial J}{\partial \boldsymbol{U}_k^i} = \boldsymbol{C_{XY}} {\boldsymbol{V}_k^i}^T - \lambda \boldsymbol{C_{XX}} {\boldsymbol{U}_k^i}^T = 0 \\
\dfrac{\partial J}{\partial \boldsymbol{V}_k^i} = \boldsymbol{C_{YX}} {\boldsymbol{U}_k^i}^T - \theta \boldsymbol{C_{YY}} {\boldsymbol{V}_k^i}^T = 0
\end{cases} \tag{1-6}
$$

$$
\begin{cases}
{\boldsymbol{C_{XX}}}^{-1} \boldsymbol{C_{XY}} {\boldsymbol{C_{YY}}}^{-1} \boldsymbol{C_{YX}} \boldsymbol{U}_k^i = \lambda^2 \boldsymbol{U}_k^i \\
{\boldsymbol{C_{YY}}}^{-1} \boldsymbol{C_{YX}} {\boldsymbol{C_{XX}}}^{-1} \boldsymbol{C_{XY}} \boldsymbol{V}_k^i = \theta^2 \boldsymbol{V}_k^i
\end{cases} \tag{1-7}
$$

对式 (1-7) 中的两个*Hermitte*矩阵分别进行特征值分解，取最大特征值对应的特征向量作为投影向量，即为所求。

1.2 扩展CCA：eCCA

(Extended CCA)

论文链接 | 代码：cca.ecca()

1.3 多重刺激CCA：msCCA

(Multi-stimulus CCA)

论文链接 | 代码：cca.mscca()

1.x 跨个体空间滤波器迁移：CSSFT

(Cross-subject spatial filter transfer method)

论文链接 | 代码：cca.cssft()

---

2. 多变量同步化系数

Multivariate synchronization index, MSI

2.1 标准MSI：MSI

论文链接 | 代码：msi.msi()

2.2 时域局部MSI：tMSI

(Temporally MSI)

论文链接 | 代码：msi.tmsi()

2.3 扩展MSI：eMSI

(Extended MSI)

论文链接 | 代码：msi.emsi()

---

3. 任务相关成分分析

Task-related component analysis, TRCA

3.1 普通/集成TRCA：(e)TRCA

((Ensemble) TRCA, (e)TRCA)

论文链接 | 代码： trca.etrca()

与此前基于CCA改进的SSVEP算法相比，TRCA在构建思路上存在较大差别，具体表现在其关注对象（即信号模板）不再限定为具有正余弦波动性质的传统模型，而是充分包含了个体信息的 “任务相关成分” (Task-related components, TRCs)。关于TRC可以简单理解为：当受试者在多次接受相同任务时，其EEG信号中应当包含具有相同性质的诱发成分。由此可见，TRCA在理论上适用于任何诱发信号具有稳定波形特征的BCI范式特征信号解码。

Nakanishi 等人首次将TRCA应用至SSVEP信号解码上时，在公式推导部分使用了一个非常讨巧的办法：**跨试次信号相关性最大化**。之所以称其“讨巧”，是因为原版TRCA公式分子中强调的**跨试次协方差计算**操作，在实际编程过程中产生了大量冗余计算步骤；其分母的**矩阵拼接**操作也缺乏明确的物理意义对应说明。而上述“瑕疵”在后续算法改进工作中被不断研究透彻。因此本文不再按照原文思路推导算法，仅给出相对成熟的阐释：

对于第 k 类别、第 i、j 试次数据 $\boldsymbol{X}_k^i, \boldsymbol{X}_k^j \in \mathbb{R}^{N_c \times N_p}$ (假定 $i \neq j$)，其跨试次样本协方差以及单试次样本方差（自协方差）分别为：

$$\begin{cases} Cov(\boldsymbol{\omega}_k \boldsymbol{X}_k^i, \boldsymbol{\omega}_k \boldsymbol{X}_k^j) = \dfrac{1}{N_p - 1} \boldsymbol{\omega}_k \boldsymbol{X}_k^i {\boldsymbol{X}_k^j}^T {\boldsymbol{\omega}_k}^T, i \neq j \\ \\ Var(\boldsymbol{\omega}_k \boldsymbol{X}_k^i) = Cov(\boldsymbol{\omega}_k \boldsymbol{X}_k^i, \boldsymbol{\omega}_k \boldsymbol{X}_k^j) = \dfrac{1}{N_p - 1} \boldsymbol{\omega}_k \boldsymbol{X}_k^i {\boldsymbol{X}_k^i}^T {\boldsymbol{\omega}_k}^T \end{cases} \tag{3-1}$$

因此，TRCA的目标函数可写为：

$$\hat{\boldsymbol{\omega}}_k = \underset{\boldsymbol{\omega}_k}{\arg\max} \frac{\sum_{j=1, j \neq i}^{N_t} \sum_{i=1}^{N_t} Cov(\boldsymbol{\omega}_k \boldsymbol{X}_k^i, \boldsymbol{\omega}_k \boldsymbol{X}_k^j)}{\sum_{i=1}^{N_t} Var(\boldsymbol{\omega}_k \boldsymbol{X}_k^i)} = \underset{\boldsymbol{\omega}_k}{\arg\max} \frac{\boldsymbol{\omega}_k \boldsymbol{S}_k {\boldsymbol{\omega}_k}^T}{\boldsymbol{\omega}_k \boldsymbol{Q}_k {\boldsymbol{\omega}_k}^T} \tag{3-2}$$

$$\begin{cases} \boldsymbol{S}_k = \sum_{j=1, j \neq i}^{N_t} \sum_{i=1}^{N_t} \boldsymbol{X}_k^i {\boldsymbol{X}_k^j}^T, i \neq j \\ \\ \boldsymbol{Q}_k = \sum_{i=1}^{N_t} \boldsymbol{X}_k^i {\boldsymbol{X}_k^i}^T \end{cases} \tag{3-3}$$

根据广义瑞丽商 (*Generalized Rayleigh quotient*) 的结论，上述目标函数的单维度最优解即为方阵 ${\boldsymbol{Q}_k}^{-1} \boldsymbol{S}_k$ 的**最大特征值对应的特征向量**。接下来对TRCA的目标函数作进一步分析：

$$\boldsymbol{S}_k = \sum_{j=1}^{N_t} \sum_{i=1}^{N_t} \boldsymbol{X}_k^i {\boldsymbol{X}_k^j}^T - \boldsymbol{Q}_k = {N_t}^2 \bar{X}_k {\bar{X}_k}^T - \boldsymbol{Q}_k = \boldsymbol{S}_k^{'} - \boldsymbol{Q}_k \tag{3-4}$$

$$\frac{\boldsymbol{\omega}_k \boldsymbol{S}_k {\boldsymbol{\omega}_k}^T}{\boldsymbol{\omega}_k \boldsymbol{Q}_k {\boldsymbol{\omega}_k}^T} = \frac{\boldsymbol{\omega}_k \boldsymbol{S}_k^{'} {\boldsymbol{\omega}_k}^T}{\boldsymbol{\omega}_k \boldsymbol{Q}_k {\boldsymbol{\omega}_k}^T} - 1 \tag{3-5}$$

相比于直接计算 $\boldsymbol{S}_k$，经由 $\boldsymbol{S}_k^{'}$ 替换或计算得到 $\boldsymbol{S}_k$ 能够大幅提升运算速度。其原因如下：将单次浮点数相乘与相加设为两种单位操作，其耗时分别为 $T_{\times}$ 和 T_{+}，对应时间复杂度分别为 $O_{\times}$ 与 O_{+}。则针对 $\boldsymbol{X}_k^i$ 执行一次矩阵乘法 $\boldsymbol{X}_k^i {\boldsymbol{X}_k^i}^T$ 或矩阵加法 $\boldsymbol{X}_k^i + \boldsymbol{X}_k^j$ 所需的理论运行时间 $T_{M\times}$、T_{M+} 分别为：

$$\begin{cases} T_{M\times} = ({N_c}^2 N_p) T_{+} + [{N_c}^2 (N_p - 1)] T_{\times} \\ \\ T_{M+} = (N_c N_p) T_{+} \end{cases} \tag{3-6}$$

对于具有 $\mathbb{R}^{N_t \times N_c \times N_p}$ 维度的训练数据张量 $\boldsymbol{X}_k$，求解 $\boldsymbol{S}_k$ 的总计理论时间 T_1 与时间复杂度 O_1 分别为：

$$\begin{cases} T_1 = N_t (N_t - 1) T_{M\times} + [N_t (N_t - 1) - 1] T_{M+} \\ \\ O_1 = O_{\times} ({N_t}^2 {N_c}^2 N_p) + O_{+} ({N_t}^2 {N_c}^2 N_p) \end{cases} \tag{3-7}$$

而使用 $\boldsymbol{S}_k^{'}$ 时，首先计算按试次平均后的个体模板 $\bar{\boldsymbol{X}}_k$，其理论运行时间 T_0 为：

$$T_0 = (N_c N_p) T_{\times} + (N_t - 1) T_{M+} \tag{3-9}$$

$\boldsymbol{S}_k^{'}$ 的总计理论计算时间 T_2 与时间复杂度 O_2 分别为：

$$\begin{cases} T_2 = T_0 + T_{M\times} \\ \\ O_2 = O_{\times} ({N_c}^2 N_p) + O_{+} (\max\{N_t N_c N_p, {N_c}^2 N_p\}) \end{cases} \tag{3-10}$$

对比 O_1 与 O_2 可见，样本数量越多，采用该种替换方法与原始情况所产生的偏差越小、速度提升越大。

综上所述，通过训练数据获取当前类别专属的空间滤波器 $\hat{\boldsymbol{\omega}}_k$ 以及信号模板 $\hat{\boldsymbol{\omega}}_k \bar{\boldsymbol{X}}_k$，基于一维*Pearson*相关系数公式，对单试次测试数据 $\boldsymbol{\chi}$ 应用空间滤波后与模板信号计算判别系数：

$$\rho_k = corr(\hat{\boldsymbol{\omega}}_k \bar{\boldsymbol{X}}_k, \hat{\boldsymbol{\omega}}_k \boldsymbol{\chi}) \tag{3-11}$$

eTRCA是基于TRCA的集成学习版本，它把各类别 $\hat{\boldsymbol{\omega}}_k \in \mathbb{R}^{1 \times N_c}$ 按行拼接在一起，在空间维度上扩增了信号模板：

$$\begin{cases} \hat{\boldsymbol{W}} = [{\boldsymbol{\omega}_1}^T, {\boldsymbol{\omega}_2}^T, ..., {\boldsymbol{\omega}_{N_e}}^T]^T \in \mathbb{R}^{N_e \times N_c} \\ \\ \rho_k = corr2(\hat{\boldsymbol{W}} \bar{\boldsymbol{X}}_k, \hat{\boldsymbol{W}} \boldsymbol{\chi}) \end{cases} \tag{3-12}$$

笔者认为，eTRCA虽然性能更为强劲，但该算法可能存在原理性缺陷：容易产生冗余成分。在刺激目标较多时，全类别集成并无必要。具体研究工作正在进行中。

至此我们有必要再回顾一下TRCA的目标函数：

（1）分子中 $\boldsymbol{\omega}_k \bar{X}_k {\bar{X}_k}^T {\boldsymbol{\omega}_k}^T$ 的本质为"**滤波后特征信号的能量**"。训练样本数目越多，叠加平均操作获取的信号模板质量越高，即随机信号成分削减越充分。而且分子能够决定目标函数的最终优化上限。

（2）分母 $\boldsymbol{\omega}_k (\sum_{i=1}^{Nt} \boldsymbol{X}_k^i {\boldsymbol{X}_k^i}^T) {\boldsymbol{\omega}_k}^T$ 的本质为"**滤波后各试次信号能量之和**"。

（3）结合上述两点可见，TRCA的性能优越是原理性的，其结构相当完善。唯一的缺陷在于训练样本数目：当 N_t 较小时，由（1）可知优化目标将产生无法弥补的偏差。因此后续关于TRCA的改进，大多针对少样本下获取更稳健的信号模板估计入手，我们将在(e)TRCA-R、sc-(e)TRCA等算法中观察到这一倾向。

3.2 正余弦扩展TRCA：(e)TRCA-R

论文链接 | 代码：trca.etrca_r()

3.3 多重刺激TRCA：ms-(e)TRCA

(Multi-stimulus (e)TRCA)

论文链接 | 代码：trca.ms_etrca()

3.4 相似度约束TRCA：sc-(e)TRCA

(Similarity-constrained (e)TRCA)

论文链接 | 代码：trca.sc_etrca()

3.5 组TRCA：gTRCA

(Group TRCA)

论文链接 | 代码：trca.gtrca()

3.6 交叉相关性TRCA：xtrca

(Cross-correlation TRCA)

论文链接 | 代码：trca.xtrca()

x. 其它早期算法

x.1 最小能量组合：other.mec()

Minimun energy combination, MEC

x.2 最大对比度组合：other.mcc()

Maximun contrast combination, MCC
