常见 SSVEP 信号处理算法（空间滤波器）

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 的部分说明、吐槽（2022/7/23）
近期计划：看心情更新。

教资这科目二比科目一难多了呀，还得是科目三生物好学。

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）

---

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() **| 2022/7/10**

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：

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

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：

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

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

根据最优化理论，函数 (1-2) 的等效形式为：

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

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

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

对函数 J 求偏导数并置零、化简：

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

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

对式 (1-1-7) 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。基于一维 *Pearson* 相关系数，对单试次测试数据 $\boldsymbol{\chi}$ 与不同频率的正余弦模板分别滤波后，计算判别系数并比较大小，确定最终的结果输出 $\hat{k}$：

$$\rho_k = corr(\hat{\boldsymbol{U}}_k\boldsymbol{\chi}, \hat{\boldsymbol{V}}_k\boldsymbol{Y}_k) \tag{1-1-8}$$

$$\hat{k} = \arg\max_k\{\rho_k | k = 1, 2, ..., N_e\} \tag{1-1-9}$$

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() **| 2022/7/16**

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：

$$Cov(\boldsymbol{\omega}_k \boldsymbol{X}_k^i, \boldsymbol{\omega}_k \boldsymbol{X}_k^j) = \frac{1}{N_p - 1} \boldsymbol{\omega}_k \boldsymbol{X}_k^i {\boldsymbol{X}_k^j}^T {\boldsymbol{\omega}_k}^T, i \neq j \tag{3-1-1}$$

$$Var(\boldsymbol{\omega}_k \boldsymbol{X}_k^i) = Cov(\boldsymbol{\omega}_k \boldsymbol{X}_k^i, \boldsymbol{\omega}_k \boldsymbol{X}_k^j) = \frac{1}{N_p - 1} \boldsymbol{\omega}_k \boldsymbol{X}_k^i {\boldsymbol{X}_k^i}^T {\boldsymbol{\omega}_k}^T \tag{3-1-2}$$

因此，TRCA 的目标函数可写为：

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

$$\boldsymbol{S}_k = \sum_{j=1, j\neq i}^{N_t} \sum_{i=1}^{N_t} \boldsymbol{X}_k^i {\boldsymbol{X}_k^j}^T, \ \boldsymbol{Q}_k = \sum_{i=1}^{N_t} \boldsymbol{X}_k^i {\boldsymbol{X}_k^i}^T \tag{3-1-4}$$

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：

$$\boldsymbol{S}_k = \sum_{j=1}^{Nt}\sum_{i=1}^{Nt}\boldsymbol{X}_k^i\boldsymbol{X}_k^{j\,T} - \boldsymbol{Q}_k = {N_t}^2\bar{X}_k{\bar{X}_k}^T - \boldsymbol{Q}_k = \boldsymbol{S}_k^{'} - \boldsymbol{Q}_k \tag{3-1-5}$$

$$\frac{\boldsymbol{\omega}_k\boldsymbol{S}_k{\boldsymbol{\omega}_k}^T}{\boldsymbol{\omega}_k\boldsymbol{Q}_k{\boldsymbol{\omega}_k}^T} = \frac{\boldsymbol{\omega}_k\boldsymbol{S}_k^{'}{\boldsymbol{\omega}_k}^T}{\boldsymbol{\omega}_k\boldsymbol{Q}_k{\boldsymbol{\omega}_k}^T} - 1 \tag{3-1-6}$$

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：

$$T_{M\times} = ({N_c}^2N_p)T_+ + [{N_c}^2(N_p-1)]T_\times \tag{3-1-7}$$

$$T_{M+} = (N_cN_p)T_+ \tag{3-1-8}$$

对于具有 $\mathbb{R}^{Nt\times Nc\times Np}$ 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：

$$T_1 = N_t(N_t-1)T_{M\times} + [N_t(N_t-1)-1]T_{M+} \tag{3-1-9}$$

$$O_1 = O_\times({N_t}^2{N_c}^2N_p) + O_+({N_t}^2{N_c}^2N_p) \tag{3-1-10}$$

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：

$$T_0 = (N_cN_p)T_\times + (N_t-1)T_{M+} \tag{3-1-11}$$

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

$$T_2 = T_0 + T_{M\times} \tag{3-1-12}$$

$$O_2 = O_\times({N_c}^2N_p) + O_+(\max\{N_tN_cN_p, {N_c}^2N_p\}) \tag{3-1-13}$$

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：

$$\rho_k = corr(\hat{\boldsymbol{\omega}}_k \bar{\boldsymbol{X}}_k, \hat{\boldsymbol{\omega}}_k \boldsymbol{\chi}) \tag{3-1-14}$$

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，通过高维滤波与二维模板匹配完成信号的模式识别：

$$\hat{\boldsymbol{W}} = [{\boldsymbol{\omega}_1}^T, {\boldsymbol{\omega}_2}^T, ..., {\boldsymbol{\omega}_{N_e}}^T]^T \in \mathbb{R}^{N_e\times N_c},\ \rho_k = corr2(\hat{\boldsymbol{W}}\bar{\boldsymbol{X}}_k, \hat{\boldsymbol{W}}\boldsymbol{\chi}) \tag{3-1-15}$$

其中 $corr2()$ 函数本质上就是先把一个二维矩阵按行拉平变成一维序列，之后再计算一维 *Pearson* 系数。关于这个过程，Matlab 的 **corr2()** 函数给出了一种更适合编程的高效运算方法。对于同维度二维矩阵 $\boldsymbol{A} = [a_{ij}]_{m\times n}$，$\boldsymbol{B} = [b_{ij}]_{m\times n}$，计算矩阵中心 $\bar{\boldsymbol{A}}$、$\bar{\boldsymbol{B}}$：

$$\bar{\boldsymbol{A}} = \sum_{j=1}^{m}\sum_{i=1}^{n} a_{ij} = \sum\sum a_{ij},\ \bar{\boldsymbol{B}} = \sum\sum b_{ij} \tag{3-1-16}$$

$$corr2(\boldsymbol{A}, \boldsymbol{B}) = \frac{\sum\sum(a_{ij}-\bar{\boldsymbol{A}})(b_{ij}-\bar{\boldsymbol{B}})}{\sqrt{\sum\sum(a_{ij}-\bar{\boldsymbol{A}})^2}\sqrt{\sum\sum(b_{ij}-\bar{\boldsymbol{B}})^2}} \tag{3-1-17}$$

别被式 (3-1-17) 一层又一层的叠加操作唬住了，使用矩阵运算很容易就能将其复现出来。由于避免了大矩阵变量的维度变换，能够显著加快运算速度：

```python
import numpy as np
def corr2_coef(X, Y):
    """2-D Pearson correlation coefficient
    Args:
        X (ndarray): (m,n)
        Y (ndarray): (m,n)
    Returns:
        coef (float)
    """
    mean_X, mean_Y = X.mean(), Y.mean()  # matrix center
    decen_X, decen_Y = X-mean_X, Y-mean_Y  # decentralized matrix
    numerator = np.einsum('ij->', decen_X*decen_Y)
    denominator_X = np.einsum('ij->', decen_X**2)
    denominator_Y = np.einsum('ij->', decen_Y**2)
    coef = numerator/np.sqrt(denominator_X*denominator_Y)
    return coef
```

论文中关于所谓的集成思想有这样一段描述：

> Since there are N_f individual calibration data corresponding to all visual stimuli, N_f different spatial filters can be obtained. Ideally, they should be similar to each other because the mixing coefficients from SSVEP source signals to scalp recordings could be considered similar within the used frequency range, which indicates the possibility of further improvements by intergrating all spatial filters.

翻译一下，*Nakanishi* 认为 8 - 15.8 Hz 刺激诱发的共计 40 类 SSVEP，彼此之间对应的空间滤波器应当是相似的，因为刺激频段比较窄，诱发脑电的头皮分布模式不至于产生过大的变化。所以根据 8 Hz SSVEP 训练得到的空间滤波器（注意这个“**适用**”应当与 TRCA 的目标函数结合理解），将其应用至其它频率信号时，理论上其目标函数（式 (3-2) ）也能达到比较高的水平，当然这样滤波后的信号质量显然不如“一个萝卜一个坑”，但多多少少能保留相当程度的特征成分。所以将其它类别专属的滤波器用在当前类别上，是变相地扩增了信号模板的空间维度信息。

依我愚见，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，全类别集成似乎并无必要。这一点在 2021 年清华大学 *Liu Binchuan* 发表的 TDCA 算法文章中也有所指出，当然人家是大佬，成果已经产出了。鄙人的具体研究工作仍在进行（~~构思~~）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()

（待完成）

$$\boldsymbol{\mathcal{Z}}^T \boldsymbol{\mathcal{D}}\boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{P}}^T \boldsymbol{\mathcal{D}}^T \boldsymbol{\mathcal{Z}} = \begin{cases} \boldsymbol{\mathcal{Z}}^T \boldsymbol{\mathcal{D}}\boldsymbol{\mathcal{D}}^T \boldsymbol{\mathcal{Z}} \boldsymbol{W} \boldsymbol{\Lambda}, \ TypeI \\ \boldsymbol{W} \boldsymbol{\Lambda}, \ TypeII \end{cases} \tag{3-3-3}$$

这里我不想再去复述他们文章中对各种算法的具体匹配方式，仅在此对式 (3-3-3) 中的主要成分进行简单介绍：

$\boldsymbol{\mathcal{Z}}$ 是数据（默认按列排布）的集合矩阵，可能是单个数据矩阵，也可能是形如 $\bigoplus_{i=1}^{Nt} \boldsymbol{X}_i$ 的多种数据联合；

$\mathcal{D}$ 是时域滤波矩阵，除了滤波器组技术以外，通常预处理（带通滤波）结束后的数据无需再进行时域滤波，即 $\mathcal{D}=\boldsymbol{I}$；

$\mathcal{P}$ 为正交投影矩阵。

3.3 多重刺激 TRCA：ms-(e)TRCA

(Multi-stimulus (e)TRCA)

论文链接 | **代码：** trca.ms_etrca() | 2022/7/23

朋友们我们今天来膜拜 (~~gank~~) 澳门大学的内卷发动机 *Wong Chiman*。之所以对他“赞誉有加”，主要有三方面原因：

（1）**算法有用，但只有一点用**：他提出的一系列 SSVEP 算法在公开数据集与竞赛数据集中具有极大优势（即样本量不足的情况）。不过在数据样本量充足的情况下，与传统的 (e)TRCA 算法难以拉开差距；

（2）**强悍如斯，地都快耕坏了**：他每篇论文都充分（~~往死里~~）挖掘了公开数据集的可用潜力，从 Benchmark、UCSD 再到 BETA 都能看到他的身影，从 CCA 到 ms-(e)TRCA 各种花里胡哨的算法都测了个遍（~~根本不给人活路~~），低频 SSVEP-BCI 系统的解码被他卷得翻江倒海，再无探索空间。

（3）**故弄玄虚，堆砌数学壁垒**：他 2020 年发表的一篇关于空间滤波器构建框架的综述性论文就是万恶之源。在他的文章中，经常使用怪僻的希腊字母甚至希伯来字母作为变量名称，为了形式简约而把简单的实际操作过程复杂化。例如明明是人畜无害的试次叠加平均：

$$\bar{\boldsymbol{X}}=\frac{1}{N_t}\sum_{n=1}^{N_t}\boldsymbol{X}_i,\ \boldsymbol{X}_i\in\mathbb{R}^{Nc\times Np} \tag{3-3-1}$$

为了凑上自己提出的框架，硬要给你表演一套天书：

$$\begin{cases}\mathcal{I}_{M,N}=[\boldsymbol{I}_N,...,\boldsymbol{I}_N]^T\in\mathbb{R}^{(MN)\times N},\ \boldsymbol{I}_N=diag(1,...,1)\in\mathbb{R}^{N\times N}\\ \bigoplus_{i=1}^{Nt}\boldsymbol{X}_i=\begin{bmatrix}\boldsymbol{X}_1 & \boldsymbol{0} & \cdots & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{X}_2 & \cdots & \boldsymbol{0}\\ \vdots & \vdots & \ddots & \vdots\\ \boldsymbol{0} & \cdots & \cdots & \boldsymbol{X}_{Nt}\end{bmatrix}\\ \bar{\boldsymbol{X}}=\frac{1}{N_t}\mathcal{I}_{Nt,Np}^T\cdot[\bigoplus_{i=1}^{Nt}\boldsymbol{X}_i]\cdot\mathcal{I}_{Nt,Nc}\end{cases} \tag{3-3-2}$$

组里有些萌新，一板一眼地照着他给的公式复现算法，结果训练样本一多，程序不仅运行慢，还动不动就内

存溢出。从原始数据结构中创建 $\bigoplus_{i=1}^{Nt} \boldsymbol{X}_i$、$\boldsymbol{\mathcal{I}}_{Nt,Np}$ 这样的大矩阵送去后端运算，相当于做一把伐木电锯去杀鸡，能不慢吗？家庭厨房里接上工业用电，能不溢出吗？

言归正传。不可否认的是，*Wong Chiman* 及其团队对于 SSVEP 信号解码的研究是成体系的、步骤严谨的，他们提出的空间滤波器框架能够适配自 CCA 以来的各种算法，为后续研究工作打开了思路。更重要的是，他们团队以及清华大学 *Wang Yijun*、*Chen Xiaogang* 等老师带领的团队，都不搞弯道超车，不搞非对称竞争，每一个研究思路都是建立在已有研究基础上，每一步新的收获都会切实体现为文章成果。这样的团队对于学生培养是利好的，学生不用担心梭哈一个老板异想天开的课题而愁于毕业困境。因此再让我跑题一次：**但凡遇到老板鼓吹自己手握多少项目、每年经费多少万、带领多少优秀青年教师团队、手下多少研究生之类的话术，一定要慎之又慎**。你要知道，牛逼吹得不够大是不能吸引上边的人投资的，牛逼吹起来了就是在梭哈你自己的学术生涯与宝贵光阴。老板项目结不了题顶多延期，少赚一点经费，少评一些名声，日子一分都不会难受。你延期延掉的是什么还请自己掂量清楚。

我们来看算法。

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：MEC

(Minimun energy combination)

论文链接 | 代码：other.mec()

x.2 最大对比度组合：MCC

Maximun contrast combination, MCC

论文链接 | 代码：other.mcc()
