

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**或者html、md等文件本地查看。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。

更新进展：今天写了TRCA部分的公式说明（2022/7/18）

近期计划：编写eCCA、msCCA的说明内容。

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
χ	EEG测试数据矩阵
X	EEG训练数据矩阵
x	EEG训练数据序列
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
X^i, x^i	第 i 试次或第 i 导联数据，详见各部分具体说明
X_k, x_k	第 k 类别数据
\bar{X}_k, \bar{x}_k	类别样本中心，由 X_k 或 x_k 按试次叠加平均获得
$\bar{\bar{X}}, \bar{\bar{x}}$	总体样本中心，由 \bar{X}_k 或 \bar{x}_k 按类别叠加平均获得
f_s	EEG 信号采样率
$\omega, U, V \dots$	低维空间滤波器
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 $\mathbf{X}_k^i \in \mathbb{R}^{N_c \times N_p}$ ，其对应频率的人工构建正余弦模板 $\mathbf{Y}_k \in \mathbb{R}^{(2N_h) \times N_p}$ 可表示为：

$$\mathbf{Y}_k = \begin{pmatrix} \sin(2\pi f n) \\ \cos(2\pi f n) \\ \sin(4\pi f n) \\ \cos(4\pi f n) \\ \dots \\ \sin(2N_h \pi f n) \\ \cos(2N_h \pi f n) \end{pmatrix}, n = [\frac{1}{f_s}, \frac{2}{f_s}, \dots, \frac{N_p}{f_s}] \quad (1-1)$$

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

$$\hat{\mathbf{U}}_k^i, \hat{\mathbf{V}}_k^i = \arg \max_{\mathbf{U}_k^i, \mathbf{V}_k^i} \frac{Cov(\mathbf{U}_k^i \mathbf{X}_k^i, \mathbf{V}_k^i \mathbf{Y}_k)}{\sqrt{Var(\mathbf{U}_k^i \mathbf{X}_k^i)} \sqrt{Var(\mathbf{V}_k^i \mathbf{Y}_k)}} = \arg \max_{\mathbf{U}_k^i, \mathbf{V}_k^i} \frac{\mathbf{U}_k^i \mathbf{C}_{\mathbf{X}\mathbf{Y}} \mathbf{V}_k^{iT}}{\sqrt{\mathbf{U}_k^i \mathbf{C}_{\mathbf{X}\mathbf{X}} \mathbf{U}_k^{iT}} \sqrt{\mathbf{V}_k^i \mathbf{C}_{\mathbf{Y}\mathbf{Y}} \mathbf{V}_k^{iT}}} \quad (1-2)$$

$$\begin{cases} \mathbf{C}_{\mathbf{X}\mathbf{X}} = \frac{1}{N_p - 1} \mathbf{X}_k^i \mathbf{X}_k^{iT} \in \mathbb{R}^{N_c \times N_c} \\ \mathbf{C}_{\mathbf{Y}\mathbf{Y}} = \frac{1}{N_p - 1} \mathbf{Y}_k \mathbf{Y}_k^T \in \mathbb{R}^{(2N_h) \times (2N_h)} \\ \mathbf{C}_{\mathbf{X}\mathbf{Y}} = \frac{1}{N_p - 1} \mathbf{X}_k^i \mathbf{Y}_k^T \in \mathbb{R}^{N_c \times (2N_h)} \\ \mathbf{C}_{\mathbf{Y}\mathbf{X}} = \frac{1}{N_p - 1} \mathbf{Y}_k \mathbf{X}_k^{iT} \in \mathbb{R}^{(2N_h) \times N_c} \end{cases} \quad (1-3)$$

根据最优化理论，函数 (1-2) 的等效形式为：

$$\begin{cases} \max_{U_k^i, V_k^i} U_k^i C_{XY} V_k^{iT} \\ s.t. U_k^i C_{XX} U_k^{iT} = V_k^i C_{YY} V_k^{iT} = 1 \end{cases} \quad (1-4)$$

利用Lagrangian乘子法构建多元函数 $J(U_k^i, V_k^i, \lambda, \theta)$:

$$J = U_k^i C_{XY} V_k^{iT} - \frac{1}{2} \lambda (U_k^i C_{XX} U_k^{iT} - 1) - \frac{1}{2} \theta (V_k^i C_{YY} V_k^{iT} - 1) \quad (1-5)$$

对函数 J 求偏导数并置零、化简:

$$\begin{cases} \frac{\partial J}{\partial U_k^i} = C_{XY} V_k^{iT} - \lambda C_{XX} U_k^{iT} = 0 \\ \frac{\partial J}{\partial V_k^i} = C_{YX} U_k^{iT} - \theta C_{YY} V_k^{iT} = 0 \end{cases} \quad (1-6)$$

$$\begin{cases} C_{XX}^{-1} C_{XY} C_{YY}^{-1} C_{YX} U_k^i = \lambda^2 U_k^i \\ C_{YY}^{-1} C_{YX} C_{XX}^{-1} C_{XY} V_k^i = \theta^2 V_k^i \end{cases} \quad (1-7)$$

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)

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 $\mathbf{X}_k^i, \mathbf{X}_k^j \in \mathbb{R}^{N_c \times N_p}$ (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：

$$\begin{cases} Cov(\omega_k \mathbf{X}_k^i, \omega_k \mathbf{X}_k^j) = \frac{1}{N_p - 1} \omega_k \mathbf{X}_k^i \mathbf{X}_k^{jT} \omega_k^T, i \neq j \\ Var(\omega_k \mathbf{X}_k^i) = Cov(\omega_k \mathbf{X}_k^i, \omega_k \mathbf{X}_k^j) = \frac{1}{N_p - 1} \omega_k \mathbf{X}_k^i \mathbf{X}_k^{iT} \omega_k^T \end{cases} \quad (3-1)$$

因此，TRCA的目标函数可写为：

$$\hat{\omega}_k = \arg \max_{\omega_k} \frac{\sum_{j=1, j \neq i}^{N_t} \sum_{i=1}^{N_t} Cov(\omega_k \mathbf{X}_k^i, \omega_k \mathbf{X}_k^j)}{\sum_{i=1}^{N_t} Var(\omega_k \mathbf{X}_k^i)} = \arg \max_{\omega_k} \frac{\omega_k \mathbf{S}_k \omega_k^T}{\omega_k \mathbf{Q}_k \omega_k^T} \quad (3-2)$$

$$\begin{cases} \mathbf{S}_k = \sum_{j=1, j \neq i}^{N_t} \sum_{i=1}^{N_t} \mathbf{X}_k^i \mathbf{X}_k^{jT}, i \neq j \\ \mathbf{Q}_k = \sum_{i=1}^{N_t} \mathbf{X}_k^i \mathbf{X}_k^{iT} \end{cases} \quad (3-3)$$

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 $\mathbf{Q}_k^{-1} \mathbf{S}_k$ 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：

$$\mathbf{S}_k = \sum_{j=1}^{N_t} \sum_{i=1}^{N_t} \mathbf{X}_k^i \mathbf{X}_k^{jT} - \mathbf{Q}_k = N_t^2 \bar{\mathbf{X}}_k \bar{\mathbf{X}}_k^T - \mathbf{Q}_k = \mathbf{S}'_k - \mathbf{Q}_k \quad (3-4)$$

$$\frac{\omega_k \mathbf{S}_k \omega_k^T}{\omega_k \mathbf{Q}_k \omega_k^T} = \frac{\omega_k \mathbf{S}'_k \omega_k^T}{\omega_k \mathbf{Q}_k \omega_k^T} - 1 \quad (3-5)$$

相比于直接计算 \mathbf{S}_k ，经由 \mathbf{S}'_k 替换或计算得到 \mathbf{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 \mathbf{X}_k^i 执行一次矩阵乘法 $\mathbf{X}_k^i \mathbf{X}_k^{iT}$ 或矩阵加法 $\mathbf{X}_k^i + \mathbf{X}_k^j$ 所需的理论运行时间 $T_{M_{\times}}$ 、 $T_{M_{+}}$ 分别为：

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

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

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

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

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

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

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

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 $\hat{\omega}_k$ 以及信号模板 $\hat{\omega}_k \bar{\mathbf{X}}_k$ ，基于一维 *Pearson* 相关系数公式，对单试次测试数据 χ 应用空间滤波后与模板信号计算判别系数：

$$\rho_k = \text{corr}(\hat{\omega}_k \bar{\mathbf{X}}_k, \hat{\omega}_k \chi) \quad (3-11)$$

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

$$\begin{cases} \hat{\mathbf{W}} = [\omega_1^T, \omega_2^T, \dots, \omega_{N_e}^T]^T \in \mathbb{R}^{N_e \times N_c} \\ \rho_k = \text{corr2}(\hat{\mathbf{W}} \bar{\mathbf{X}}_k, \hat{\mathbf{W}} \chi) \end{cases} \quad (3-12)$$

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 $\omega_k \bar{\mathbf{X}}_k \bar{\mathbf{X}}_k^T \omega_k^T$ 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 $\omega_k (\sum_{i=1}^{N_t} \mathbf{X}_k^i \mathbf{X}_k^{iT}) \omega_k^T$ 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
