

# 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\(\)](#)

对于具有  $N_e$  个不同频率刺激的 SSVEP-BCI 系统，频率索引  $k$  的人工构建正余弦模板  $\mathbf{Y}_k$  可表示为：

$$\mathbf{Y}_k = \begin{bmatrix} \sin(2\pi f_k n) \\ \cos(2\pi f_k n) \\ \sin(4\pi f_k n) \\ \cos(4\pi f_k n) \\ \vdots \\ \sin(2N_h \pi f_k n) \\ \cos(2N_h \pi f_k n) \end{bmatrix} \in \mathbb{R}^{(2N_h) \times N_p}, \quad n = \left[ \frac{1}{f_s}, \frac{2}{f_s}, \dots, \frac{N_p}{f_s} \right] \quad (1-1-1)$$

对于单试次多导联 EEG 数据  $\mathbf{X} \in \mathbb{R}^{N_c \times N_p}$  以及假定的所属类别  $k$ ，CCA 的优化目标为一组投影向量  $\hat{\mathbf{U}}_k$  和  $\hat{\mathbf{V}}_k$ ，使得一维信号  $\hat{\mathbf{U}}_k \mathbf{X}$  与  $\hat{\mathbf{V}}_k \mathbf{Y}_k$  之间相关性最大化，其目标函数为：

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

$$\left\{ \begin{array}{l} \mathbf{C}_{\mathbf{X}\mathbf{X}} = \frac{1}{N_p - 1} \mathbf{X} \mathbf{X}^T \in \mathbb{R}^{N_c \times N_c} \\ \mathbf{C}_{\mathbf{Y}_k \mathbf{Y}_k} = \frac{1}{N_p - 1} \mathbf{Y}_k \mathbf{Y}_k^T \in \mathbb{R}^{(2N_h) \times (2N_h)} \\ \mathbf{C}_{\mathbf{X}\mathbf{Y}_k} = \frac{1}{N_p - 1} \mathbf{X} \mathbf{Y}_k^T \in \mathbb{R}^{N_c \times (2N_h)} \\ \mathbf{C}_{\mathbf{Y}_k \mathbf{X}} = \frac{1}{N_p - 1} \mathbf{Y}_k \mathbf{X}^T \in \mathbb{R}^{(2N_h) \times N_c} \end{array} \right. \quad (1-1-3)$$

根据最优化理论，函数 (1-2) 的等效形式为：

$$\begin{cases} \max_{\mathbf{U}_k, \mathbf{V}_k} \mathbf{U}_k \mathbf{C}_{\mathbf{X}\mathbf{Y}_k} \mathbf{V}_k^T \\ s.t. \mathbf{U}_k \mathbf{C}_{\mathbf{X}\mathbf{X}} \mathbf{U}_k^T = \mathbf{V}_k \mathbf{C}_{\mathbf{Y}_k\mathbf{Y}_k} \mathbf{V}_k^T = 1 \end{cases} \quad (1-1-4)$$

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

$$J = \mathbf{U}_k \mathbf{C}_{\mathbf{X}\mathbf{Y}_k} \mathbf{V}_k^T - \frac{1}{2} \lambda (\mathbf{U}_k \mathbf{C}_{\mathbf{X}\mathbf{X}} \mathbf{U}_k^T - 1) - \frac{1}{2} \theta (\mathbf{V}_k \mathbf{C}_{\mathbf{Y}_k\mathbf{Y}_k} \mathbf{V}_k^T - 1) \quad (1-1-5)$$

对函数  $J$  求偏导数并置零 :

$$\begin{cases} \frac{\partial J}{\partial \mathbf{U}_k} = \mathbf{C}_{\mathbf{X}\mathbf{Y}_k} \mathbf{V}_k^T - \lambda \mathbf{C}_{\mathbf{X}\mathbf{X}} \mathbf{U}_k^T = 0 \quad (I) \\ \frac{\partial J}{\partial \mathbf{V}_k} = \mathbf{C}_{\mathbf{Y}_k\mathbf{X}} \mathbf{U}_k^T - \theta \mathbf{C}_{\mathbf{Y}_k\mathbf{Y}_k} \mathbf{V}_k^T = 0 \quad (II) \end{cases} \quad (1-1-6)$$

消元化简后可知  $\lambda = \theta$ 。送佛送到西，咱们来看看到底怎么消元：

$$\begin{cases} \mathbf{U}_k * (I) \rightarrow \mathbf{U}_k \mathbf{C}_{\mathbf{X}\mathbf{Y}_k} \mathbf{V}_k^T - \lambda \mathbf{U}_k \mathbf{C}_{\mathbf{X}\mathbf{X}} \mathbf{U}_k^T = 0 \\ \mathbf{V}_k * (II) \rightarrow \mathbf{V}_k \mathbf{C}_{\mathbf{Y}_k\mathbf{X}} \mathbf{U}_k^T - \theta \mathbf{V}_k \mathbf{C}_{\mathbf{Y}_k\mathbf{Y}_k} \mathbf{V}_k^T = 0 \end{cases} \quad (1-1-7)$$

根据约束条件 (1-1-4) 可知：

$$\lambda = \mathbf{U}_k \mathbf{C}_{\mathbf{X}\mathbf{Y}_k} \mathbf{V}_k^T, \theta = \mathbf{V}_k \mathbf{C}_{\mathbf{Y}_k\mathbf{X}} \mathbf{U}_k^T \quad (1-1-8)$$

大家注意  $\lambda = \theta^T$ ，而当我们明确要求优化目标是一维向量的时候，这两位大哥其实都是实数，所以它们相等。之后就是大家在解二元一次方程组时常用的代换消元过程（ $\mathbf{U}_k$  与  $\mathbf{V}_k$  互相替换），我就不再演示了。最终应得到两个特征值方程：

$$\begin{cases} \mathbf{C}_{\mathbf{X}\mathbf{X}}^{-1} \mathbf{C}_{\mathbf{X}\mathbf{Y}_k} \mathbf{C}_{\mathbf{Y}_k\mathbf{Y}_k}^{-1} \mathbf{C}_{\mathbf{Y}_k\mathbf{X}} \mathbf{U}_k^T = \lambda^2 \mathbf{U}_k^T \\ \mathbf{C}_{\mathbf{Y}_k\mathbf{Y}_k}^{-1} \mathbf{C}_{\mathbf{Y}_k\mathbf{X}} \mathbf{C}_{\mathbf{X}\mathbf{X}}^{-1} \mathbf{C}_{\mathbf{X}\mathbf{Y}_k} \mathbf{V}_k^T = \theta^2 \mathbf{V}_k^T \end{cases} \quad (1-1-9)$$

对式 (1-1-9) 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。对所有的假定类别遍历上述过程，基于一维 *Pearson* 相关系数分别计算判别系数并比较大小，确定最终的结果输出  $\hat{k}$ ：

$$\rho_k = \text{corr}(\hat{\mathbf{U}}_k \mathbf{X}, \hat{\mathbf{V}}_k \mathbf{Y}_k), \hat{k} = \arg \max_k \{\rho_k\} \quad (1-1-10)$$

式 (1-1-2) 与 (1-1-9) 请各位务必熟悉，之后部分算法的推导过程中我可能直接从类似前者的形式跳至后者，不再进行中间步骤的详细解说。

## 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\(\)](#)

( 大量数学前置知识警告 )

朋友们我们今天来膜拜 (gank) 澳门大学的内卷发动机 *Chi Man Wong* 和 *Feng Wan* 老师团队。之所以对他们“赞誉有加”，主要有三方面原因：

( 1 ) 算法有用，但只有一点用：他们提出的一系列 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：他们的每篇论文都充分（往死里）挖掘了公开数据集的可用潜力，从 [Benchmark][Benchmark]、[UCSD][UCSD] 再到 [BETA][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：该团队 2020 年发表的一篇关于[空间滤波器构建框架][TRCA-R]的综述性论文就是万恶之源。在其文章中，经常使用怪僻的希腊字母、花体英文字母作为变量名称，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：

$$\bar{\mathbf{X}} = \frac{1}{N_t} \sum_{n=1}^{N_t} \mathbf{X}_i, \mathbf{X}_i \in \mathbb{R}^{N_c \times N_p} \quad (1-3-1)$$

为了凑上自己提出的框架，硬要给你表演一套天书：

$$\oplus_{i=1}^{N_t} \mathbf{X}_i = \begin{bmatrix} \mathbf{X}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{X}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \cdots & \mathbf{X}_{N_t} \end{bmatrix}, \bar{\mathbf{X}} = \frac{1}{N_t} \mathbf{I}_{N_t, N_c} \cdot \left[ \oplus_{i=1}^{N_t} \mathbf{X}_i \right] \cdot \mathbf{I}_{N_t, N_p}^T \quad (1-3-2)$$

组里有些萌新，一板一眼地照着论文公式复现算法，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  $\oplus_{i=1}^{N_t} \mathbf{X}_i$ 、 $\mathbf{I}_{N_t, N_p}$  这样的大矩阵送去后端运算，相当于先做一把电锯去再杀鸡炖汤，能不慢吗？家庭厨房里接上工业用电，能不溢出吗？

不可否认的是，*Chi Man Wong* 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适配了自 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 *Yijun Wang*、*Xiaogang Chen* 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。你延期延掉的是啥还请自己掂量清楚。

言归正传，我们首先有必要介绍一下 *Wong* 提出的统一框架。*Wong* 以及部分研究者喜欢按列展示向量，而本文中向量统一按行排布（我偏不一老子就喜欢按行），因此部分公式可能在形式上与原文有所出入，但本质是一样的：

$$\mathbf{Z} \mathbf{D} \mathbf{P} \mathbf{P}^T \mathbf{D}^T \mathbf{Z}^T \mathbf{W} = \begin{cases} \mathbf{Z} \mathbf{D} \mathbf{D}^T \mathbf{Z}^T \mathbf{W} \mathbf{\Lambda}, & Type I \\ \mathbf{W} \mathbf{\Lambda}, & Type II \end{cases} \quad (1-3-3)$$

这里  $\mathbf{W}$  与  $\mathbf{\Lambda}$  之所以写成矩阵而不是“向量+标量”形式，是因为空间滤波器并不总是将多通道信号压缩至一维，对于需要进行压缩的一般情况，只需取方阵  $\mathbf{Z} \mathbf{D} \mathbf{P} \mathbf{P}^T \mathbf{D}^T \mathbf{Z}^T$  的最大（小）特征值对应的特征向量即可；而当需要保留多个维度（投影子空间）时， $\mathbf{W}$  的最优解为多个特征向量的拼接，拼接顺序以对应特征向量的大小顺序为准。

接下来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 (1-3-3) 中的主要成分进行简单介绍：

$\mathbf{Z}$  是数据（默认按列排布）的集合矩阵，可能是（1）单个数据矩阵；（2）形如  $\oplus_{i=1}^{N_t} \mathbf{X}_i$  的多种数据块对角拼接组成的联合矩阵。一般来说（2）中的对角联合矩阵，在整体公式中需要经过  $\mathbf{I}$  矩阵的变形处理，将其转换为由多个数据块**横向**或**纵向**拼接而成的大矩阵，如式 (1-3-11)；

$\mathbf{D}$  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  $\mathbf{D} = \mathbf{I}$ ；

$\mathbf{P}$  为正交投影矩阵，通常满足  $\mathbf{P} = \mathbf{P}^T = \mathbf{P}\mathbf{P}^T = \mathbf{P}^T\mathbf{P}$ 。根据给定的投影方向 ( $\mathbf{T}$ )，可表示为：

$$\begin{cases} \mathbf{P} = \mathbf{T}^T (\mathbf{T}\mathbf{T}^T)^{-1} \mathbf{T} = \mathbf{Q}_T \mathbf{Q}_T^T \\ \mathbf{T} = \mathbf{Q}_T \mathbf{R}_T, \text{ Reduced QR decomposition} \end{cases} \quad (1-3-4)$$

不难发现，该框架的数学本质是一系列广义特征值 (*Generalized eigenvalue problems, GEPs*) 方程，而空间滤波器构建过程中常见的广义瑞利商 (*Generalized Rayleigh quotient*) 问题通常又可以转化为 *GEP* 方程加以求解，因此该框架几乎能够契合现有各种 *SSVEP-BCI* 系统中常见的空间滤波器算法。尽管如此，除了 *Wong* 设计的算法，我基本上不会使用这个框架来展示公式。原因除了之前吐槽过的“数学墙”以外，还有很重要的一点，即凭空创造、设计更好的时域滤波矩阵  $\mathbf{D}$  或正交投影矩阵  $\mathbf{P}$  都是不现实的，想要从数学上证明某个投影空间具有某些数学特性或优势都极具挑战性。我个人更倾向于通过直观物理含义的途径来阐释算法原理，希望通过我的讲解，能够让大实现数学原理、编程实践与性能优化的三者合一，从而更好地掌握算法的精髓、洞察未来发展方向。

在本节前置数学知识的最后，给大家简单介绍一下广义瑞利商及其与 *GEP* 问题的关系。形如式 (1-3-5) 所示的函数称为瑞利商 (*Rayleigh quotient*)，其中  $\mathbf{A}$  为 *Hermitte* 矩阵：

$$f(\omega) = \frac{\omega \mathbf{A} \omega^T}{\omega \omega^T}, \mathbf{A} \in \mathbb{R}^{N \times N}, \omega \in \mathbb{R}^{1 \times N} \quad (1-3-5)$$

一般最优化问题需要求解的是瑞利商的最值，参考式 (1-1-4) 的方式，将式 (1-3-5) 转化为最优化问题的标准描述形式，之后利用 *Lagrangian* 乘子法构建函数  $J$ ：

$$\begin{cases} \max_{\omega} \omega \mathbf{A} \omega^T \\ s.t. \omega \omega^T = 1 \end{cases} \implies J(\omega) = \omega \mathbf{A} \omega^T - \lambda (\omega \omega^T - 1) \quad (1-3-6)$$

对  $J$  求导并置零，最终可得特征值方程：

$$\frac{dJ(\omega)}{d\omega} = 2\mathbf{A}\omega^T - 2\lambda\omega^T = 0 \rightarrow \mathbf{A}\omega^T = \lambda\omega^T \quad (1-3-7)$$

至此可以看出，瑞利商的最值即为方阵  $\mathbf{A}$  最大 (小) 特征值，取最值时的解即为该特征值对应的特征向量。至于广义瑞利商，其形如式 (1-3-8) 所示的函数， $\mathbf{B}$  同样也是 *Hermitte* 矩阵：

$$f(\omega) = \frac{\omega \mathbf{A} \omega^T}{\omega \mathbf{B} \omega^T}, \mathbf{A}, \mathbf{B} \in \mathbb{R}^{N \times N}, \omega \in \mathbb{R}^{1 \times N} \quad (1-3-8)$$

同上进行类似操作，可以得到式 (1-3-9) 所示的 *GEP* 方程。由此可知广义瑞利商的最值即为方阵  $\mathbf{B}^{-1}\mathbf{A}$  最大

(小) 特征值对应的特征向量：

$$\mathbf{A}\boldsymbol{\omega}^T = \lambda \mathbf{B}\boldsymbol{\omega}^T \implies (\mathbf{B}^{-1}\mathbf{A})\boldsymbol{\omega}^T = \lambda \boldsymbol{\omega}^T \quad (1-3-9)$$

接下来我们来看 msCCA 算法。首先给出统一框架（Type I）下的各部分组成：

$$\begin{cases} \mathbf{Z} = \mathbf{I}_{N_e, N_c} \left( \bigoplus_{k=1}^{N_e} \bar{\mathbf{X}}_k \right) \in \mathbb{R}^{N_c \times (N_e N_p)} \\ \mathbf{D} = \mathbf{I}_{N_e N_p} \in \mathbb{R}^{(N_e N_p) \times (N_e N_p)} \\ \mathbf{P} = \mathbf{Q}_y \mathbf{Q}_y^T \in \mathbb{R}^{(N_e N_p) \times (N_e N_p)} \end{cases} \quad (1-3-10)$$

其中：

$$\mathbf{y} = [\mathbf{Y}_1 \quad \mathbf{Y}_2 \quad \cdots \quad \mathbf{Y}_{N_e}] \in \mathbb{R}^{(2N_h) \times (N_e N_p)} \quad (1-3-11)$$

$$\mathbf{Q}_y \mathbf{Q}_y^T = \begin{bmatrix} \mathbf{Q}_{Y_1} \mathbf{Q}_{Y_1}^T & \mathbf{Q}_{Y_1} \mathbf{Q}_{Y_2}^T & \cdots & \mathbf{Q}_{Y_1} \mathbf{Q}_{Y_{N_e}}^T \\ \mathbf{Q}_{Y_2} \mathbf{Q}_{Y_1}^T & \mathbf{Q}_{Y_2} \mathbf{Q}_{Y_2}^T & \cdots & \mathbf{Q}_{Y_2} \mathbf{Q}_{Y_{N_e}}^T \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{Q}_{Y_{N_e}} \mathbf{Q}_{Y_1}^T & \mathbf{Q}_{Y_{N_e}} \mathbf{Q}_{Y_2}^T & \cdots & \mathbf{Q}_{Y_{N_e}} \mathbf{Q}_{Y_{N_e}}^T \end{bmatrix} \quad (1-3-12)$$

不要被这些花里胡哨的公式迷乱了双眼，我们来看看每一步都发生了什么：

$$\mathbf{Z} = \underbrace{\begin{bmatrix} \mathbf{I}_{N_c} & \mathbf{I}_{N_c} & \cdots & \mathbf{I}_{N_c} \end{bmatrix}}_{\mathbb{R}^{N_c \times (N_e N_c)}} \underbrace{\begin{bmatrix} \bar{\mathbf{X}}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{X}}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \bar{\mathbf{X}}_{N_e} \end{bmatrix}}_{\mathbb{R}^{(N_e N_c) \times (N_e N_p)}} = \underbrace{\begin{bmatrix} \bar{\mathbf{X}}_1 & \bar{\mathbf{X}}_2 & \cdots & \bar{\mathbf{X}}_{N_e} \end{bmatrix}}_{\mathbb{R}^{N_c \times (N_e N_p)}} \quad (1-3-13)$$

$$\mathbf{ZD} = \underbrace{\begin{bmatrix} \bar{\mathbf{X}}_1 & \bar{\mathbf{X}}_2 & \cdots & \bar{\mathbf{X}}_{N_e} \end{bmatrix}}_{\mathbb{R}^{N_c \times (N_e N_p)}} \underbrace{\begin{bmatrix} \mathbf{I}_{N_p} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{N_p} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I}_{N_p} \end{bmatrix}}_{\mathbb{R}^{(N_e N_p) \times (N_e N_p)}} = \mathbf{Z} \quad (1-3-14)$$

$$\mathbf{ZDP} = \underbrace{\left[ \sum_{k=1}^{N_e} \bar{\mathbf{X}}_k \mathbf{Q}_{Y_k} \mathbf{Q}_{Y_1}^T \quad \cdots \quad \sum_{k=1}^{N_e} \bar{\mathbf{X}}_k \mathbf{Q}_{Y_k} \mathbf{Q}_{Y_{N_e}}^T \right]}_{\mathbb{R}^{N_c \times (N_e N_p)}} \quad (1-3-15)$$

$$\begin{aligned} \mathbf{ZDP} \mathbf{P}^T \mathbf{D}^T \mathbf{Z}^T &= \sum_{c=1}^{N_e} \left[ \sum_{a=1}^{N_e} \bar{\mathbf{X}}_a \mathbf{Q}_{Y_a} \mathbf{Q}_{Y_c}^T \left( \sum_{b=1}^{N_e} \bar{\mathbf{X}}_b \mathbf{Q}_{Y_b} \mathbf{Q}_{Y_c}^T \right)^T \right] \\ &\xrightarrow{\mathbf{Q}_{Y_c}^T \mathbf{Q}_{Y_c} = \mathbf{I}} N_e \sum_{b=1}^{N_e} \sum_{a=1}^{N_e} \bar{\mathbf{X}}_a \mathbf{Q}_{Y_a} \mathbf{Q}_{Y_b}^T \bar{\mathbf{X}}_b \end{aligned} \quad (1-3-16)$$

$$\mathbf{ZDD}^T \mathbf{Z}^T = \mathbf{ZZ}^T = \underbrace{\begin{bmatrix} \bar{\mathbf{X}}_1 & \bar{\mathbf{X}}_2 & \cdots & \bar{\mathbf{X}}_{N_e} \end{bmatrix}}_{\mathbb{R}^{N_c \times (N_e N_p)}} \underbrace{\begin{bmatrix} \bar{\mathbf{X}}_1 \\ \bar{\mathbf{X}}_2 \\ \vdots \\ \bar{\mathbf{X}}_{N_e} \end{bmatrix}}_{\mathbb{R}^{(N_e N_p) \times N_c}} = \sum_{k=1}^{N_e} \bar{\mathbf{X}}_k \bar{\mathbf{X}}_k^T \quad (1-3-17)$$

可真是费了好一番力气才完成。最终 *GEP* 方程为（仅需一维投影向量）：

$$\left( N_e \sum_{b=1}^{N_e} \sum_{a=1}^{N_e} \bar{\mathbf{X}}_a \mathbf{Q}_{Y_a} \mathbf{Q}_{Y_b}^T \bar{\mathbf{X}}_b^T \right) \boldsymbol{\omega}^T = \lambda \left( \sum_{k=1}^{N_e} \bar{\mathbf{X}}_k \bar{\mathbf{X}}_k^T \right) \boldsymbol{\omega}^T \quad (1-3-18)$$

相信各位很快就会发现，这个公式没有为我们提供任何直白的、一般人能阅读的有效信息。坦白地说，仅靠式 (1-3-18) 设计的滤波器以及相应的模板匹配方法是不完整的，具体原因请各位移步下一节 *ms-eCCA*，我们将从另一个更合理的角度审视这个算法。

不得不说 *Wong* 这一手阉割刀法堪比老黄，先在 *JNE* 上发表精心推导设计的两大 *ms-* 算法，再在 *IEEE TBME* 上发表统一框架，顺带蜻蜓点水一般地，用框架小增小改就套出了这个丐版 *msCCA*，还比老旧的 *itCCA* 强上一截，以此彰显框架的“易用性”，其实根本没这么浅显，套模型套框架也并非研学之道。

## 1.4 多重刺激扩展 CCA：ms-eCCA

(Multi-stimulus CCA)

[论文链接][*ms-TRCA*] | 代码：[cca.msecca\(\)](#)

顾名思义，ms-eCCA 是 msCCA 的扩展，在模板匹配上类似 eCCA 之于 CCA，此外在扩增的刺激目标选择上有所约束。阅读本节内容之前，建议阅读 3.2 节以了解部分背景知识，因为本节将从另外一个角度解释 ms-eCCA 公式的由来。ms-eCCA 与 ms-eTRCA 是 Wong 在同一篇论文中提出的方法，在发表时间上也先于 msCCA。与 3.2 节所述一致，ms- 技术假定一套滤波器同时适用于目标频率以及周边少数频率的信号，通过合并一定范围内的多类别信号，强行扩增可用训练样本数目。

“合并样本”的操作，既可以理解为不同类别样本模板在时间顺序上的拼接，见式(1-4-2)；也可以理解为不同类别样本各自协方差矩阵的叠加，见式 (1-4-4)。结合 3.2 节 ms-(e)TRCA 与 1.1 节 CCA 的相关内容，ms-eCCA 的目标函数可表示为：

$$\hat{U}_k, \hat{V}_k = \arg \max_{U_k, V_k} \frac{Cov(U_k \mathbf{Z}_k, \mathbf{Y}_k V_k)}{\sqrt{Var(U_k \mathbf{Z}_k)} \sqrt{Var(V_k \mathbf{Y}_k)}} = \arg \max_{U_k, V_k} \frac{U_k \mathbf{C}_{\mathbf{Z}_k \mathbf{Y}_k} V_k^T}{\sqrt{(U_k \mathbf{C}_{\mathbf{Z}_k \mathbf{Z}_k} U_k^T)} \sqrt{(V_k \mathbf{C}_{\mathbf{Y}_k \mathbf{Y}_k} V_k^T)}} \quad (1-4-1)$$

$$\begin{cases} \mathbf{Z}_k = [\bar{\mathbf{X}}_{k-m} & \bar{\mathbf{X}}_{k-m+1} & \cdots & \bar{\mathbf{X}}_{k+n}] \in \mathbb{R}^{N_c \times [(m+n+1)N_p]} \\ \mathbf{Y}_k = [\mathbf{Y}_{k-m} & \mathbf{Y}_{k-m+1} & \cdots & \mathbf{Y}_{k+n}] \in \mathbb{R}^{N_c \times [(m+n+1)N_t N_p]} \end{cases} \quad (1-4-2)$$

$$\mathbf{Y}_k = \begin{bmatrix} \sin(2\pi f_k \mathbf{t} + \phi_k) \\ \cos(2\pi f_k \mathbf{t} + \phi_k) \\ \vdots \\ \sin(2\pi N_h f_k \mathbf{t} + N_h \phi_k) \\ \cos(2\pi N_h f_k \mathbf{t} + N_h \phi_k) \end{bmatrix} \in \mathbb{R}^{(2N_h) \times N_p}, \quad \mathbf{t} = \begin{bmatrix} \frac{1}{f_s} & \frac{2}{f_s} & \cdots & \frac{N_p}{f_s} \end{bmatrix} \quad (1-4-3)$$

$$\begin{cases} \mathbf{C}_{\mathbf{Z}_k \mathbf{Z}_k} = \frac{1}{N_p - 1} \sum_{i=-n}^m \bar{\mathbf{X}}_{k+i} \bar{\mathbf{X}}_{k+i}^T \in \mathbb{R}^{N_c \times N_c} \\ \mathbf{C}_{\mathbf{Y}_k \mathbf{Y}_k} = \frac{1}{N_p - 1} \sum_{i=-n}^m \mathbf{Y}_{k+i} \mathbf{Y}_{k+i}^T \in \mathbb{R}^{(2N_h) \times (2N_h)} \\ \mathbf{C}_{\mathbf{Z}_k \mathbf{Y}_k} = \frac{1}{N_p - 1} \sum_{i=-n}^m \bar{\mathbf{X}}_{k+i} \mathbf{Y}_{k+i}^T \in \mathbb{R}^{N_c \times (2N_h)} \\ \mathbf{C}_{\mathbf{Y}_k \mathbf{Z}_k} = \frac{1}{N_p - 1} \sum_{i=-n}^m \mathbf{Y}_{k+i} \bar{\mathbf{X}}_{k+i}^T \in \mathbb{R}^{(2N_h) \times N_c} \end{cases} \quad (1-4-4)$$

简而言之，在空间滤波器构建上，ms-eCCA 与 ms-(e)TRCA 的思路一致，即把不同类别信号按时间维度顺次拼接，



以起到数据扩增的作用。类比 1.1 节推导过程可知，式 (1-4-1) 所对应的两个 *GEP* 方程为：

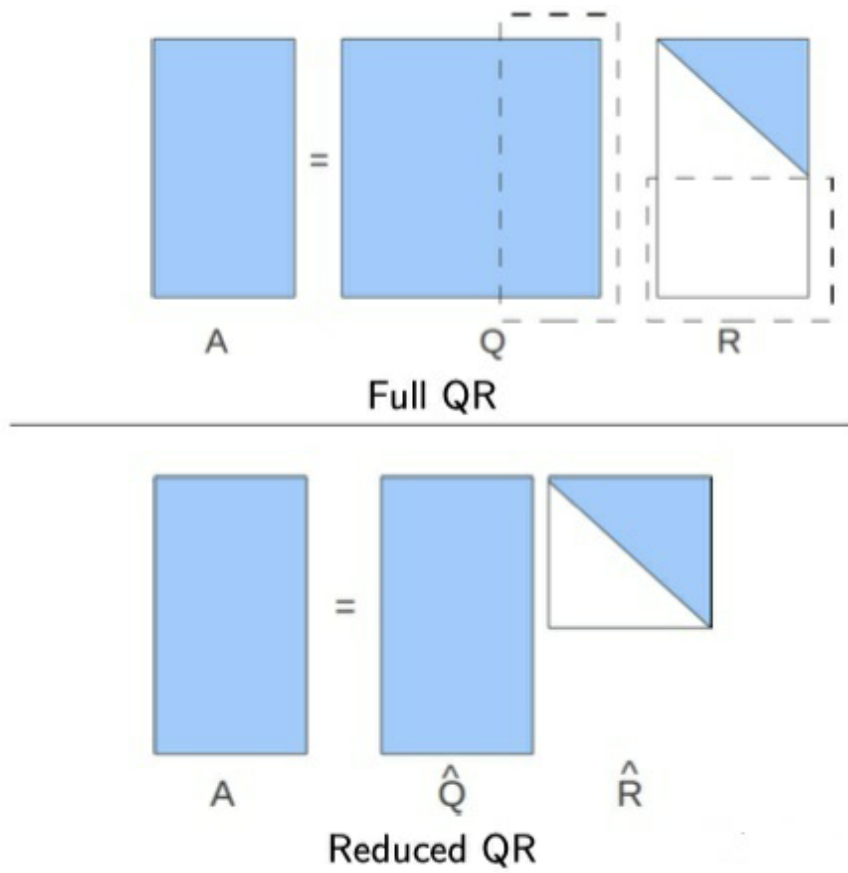
$$\begin{cases} \mathbf{C}_{\mathbf{Z}_k \mathbf{Z}_k}^{-1} \mathbf{C}_{\mathbf{Z}_k \mathbf{Y}_k} \mathbf{C}_{\mathbf{Y}_k \mathbf{Y}_k}^{-1} \mathbf{C}_{\mathbf{Y}_k \mathbf{Z}_k} \mathbf{U}_k = \lambda^2 \mathbf{U}_k^T & (I) \\ \mathbf{C}_{\mathbf{Y}_k \mathbf{Y}_k}^{-1} \mathbf{C}_{\mathbf{Y}_k \mathbf{Z}_k} \mathbf{C}_{\mathbf{Z}_k \mathbf{Z}_k}^{-1} \mathbf{C}_{\mathbf{Z}_k \mathbf{Y}_k} \mathbf{V}_k = \theta^2 \mathbf{V}_k^T & (II) \end{cases} \quad (1-4-5)$$

首先来看方程组中的 (I)：

$$\begin{aligned} & \left( \sum_k \bar{\mathbf{X}}_k \bar{\mathbf{X}}_k^T \right)^{-1} \left( \sum_a \bar{\mathbf{X}}_a \mathbf{Y}_a^T \right) \left( \sum_c \mathbf{Y}_c \mathbf{Y}_c^T \right)^{-1} \left( \sum_b \mathbf{Y}_b \bar{\mathbf{X}}_b^T \right) \mathbf{U}_k^T = \lambda^2 \mathbf{U}_k^T \\ \Rightarrow & \left( \sum_a \bar{\mathbf{X}}_a \mathbf{Y}_a^T \right) \left( \sum_c \mathbf{Y}_c \mathbf{Y}_c^T \right)^{-1} \left( \sum_b \mathbf{Y}_b \bar{\mathbf{X}}_b^T \right) \mathbf{U}_k^T = \lambda^2 \left( \sum_k \bar{\mathbf{X}}_k \bar{\mathbf{X}}_k^T \right) \mathbf{U}_k^T \\ \Rightarrow & \left[ \sum_b \sum_a \bar{\mathbf{X}}_a \mathbf{Y}_a^T \left( \sum_c \mathbf{Y}_c \mathbf{Y}_c^T \right)^{-1} \mathbf{Y}_b \bar{\mathbf{X}}_b^T \right] \mathbf{U}_k^T = \lambda^2 \left( \sum_k \bar{\mathbf{X}}_k \bar{\mathbf{X}}_k^T \right) \mathbf{U}_k^T \quad (1-4-6) \end{aligned}$$

到这一步我们不难发现，其结果已经与上一节末尾的式 (1-3-18) 形式非常相似，仅剩的问题在于正交投影  $\mathbf{Q}_k \mathbf{Q}_k^T$  与正余弦矩阵  $\mathbf{Y}_k$  之间的关系。

因此我们有必要讲解（插播）一下何为 QR 分解 (*QR decomposition*)。QR 分解又称正交三角分解，通常用于求解矩阵的特征值与特征向量。QR 分解的作用是将实（复）非奇异矩阵  $\mathbf{A}$  转化为正交（酉）矩阵  $\mathbf{Q}$  与实（复）非奇异上三角矩阵  $\mathbf{R}$  的乘积。从操作结果来看，QR 矩阵分为全分解 (*full decomposition*) 与约化分解 (*reduced decomposition*) 两种，二者的差异见下图：



一般情况下，我们需要的是约化 QR 分解结果。计算方法有很多，这里给出其中一种方便理解与实操的方法——*Gram-Schmidt* 正交化。首先我们获取矩阵  $\mathbf{A} \in \mathbb{R}^{m \times n}$  的列向量组  $\mathbf{A}(:, j)$ ，之后单位化  $\mathbf{A}(:, 1)$ ，并将其作为第一个正交基，对后续列向量依次进行投影分解：

$$\begin{cases} \mathbf{A}(:, 1) = r_{1,1} \mathbf{q}_1 \\ \mathbf{A}(:, 2) = r_{2,1} \mathbf{q}_1 + r_{2,2} \mathbf{q}_2 \\ \dots \\ \mathbf{A}(:, n) = r_{n,1} \mathbf{q}_1 + r_{n,2} \mathbf{q}_2 + \dots + r_{n,n} \mathbf{q}_n \end{cases} \quad (1-4-7)$$

$$\mathbf{A} = [\mathbf{q}_1 \quad \mathbf{q}_2 \quad \dots \quad \mathbf{q}_n] \begin{bmatrix} r_{1,1} & r_{1,2} & \dots & r_{1,n} \\ 0 & r_{2,2} & \dots & r_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & r_{n,n} \end{bmatrix} = \mathbf{Q}_A \mathbf{R}_A \quad (1-4-8)$$

我们在 -R 技术、ms- 技术乃至未来将更新的 TDCA 算法中都经常见到正交投影  $\mathbf{P}_k = \mathbf{Q}_k \mathbf{Q}_k^T$  的身影，我们来仔细研究一下这个矩阵。已知  $\mathbf{Y}_k^T = \mathbf{Q}_k \mathbf{R}_k$ ，对照式 (1-4-8) 不难看出：

(1)  $\mathbf{Q}_k$  本质上就是  $\mathbf{Y}_k^T$ ，二者在数值上存在一定的系数比例。因为对于正余弦模板矩阵而言，其列向量本来就是正交的，只是不满足单位化而已；

(2)  $\mathbf{R}_k$  是个对角阵，主对角线上的系数绝对值均相等，且为  $\mathbf{Y}_k^T$  中任一列向量的内积平方根， $\mathbf{R}_k$  的唯一作用就是将  $\mathbf{Q}_k$  压缩至单位化水平。这一点很容易证明：

$$\mathbf{Y}_k^T \mathbf{Y}_k = \mathbf{Q}_k (\mathbf{R}_k \mathbf{R}_k^T) \mathbf{Q}_k^T = r_{1,1}^2 \mathbf{Q}_k \mathbf{Q}_k^T \quad (1-4-9)$$

又因为式 (1-3-4) 指出， $\mathbf{P} = \mathbf{T}^T (\mathbf{T} \mathbf{T}^T)^{-1} \mathbf{T} = \mathbf{Q}_T \mathbf{Q}_T^T$ ，我们知道  $\mathbf{Q}_k \mathbf{Q}_k^T = \mathbf{Y}_k^T (\mathbf{Y}_k \mathbf{Y}_k^T)^{-1} \mathbf{Y}_k$ ，接下来我们证明这个矩阵能够起到正交投影的作用。

关于正交投影，从泛函的角度而言可以给出如下定义：令  $\mathbf{H}$  为向量空间， $\mathbf{M}$  是  $\mathbf{H}$  内的  $n$  维子空间。若对于  $\mathbf{H}$  中的向量  $\mathbf{x}$ ：

$$\exists \hat{\mathbf{x}} \in \mathbf{M}, \text{ s.t. } \forall \mathbf{y} \in \mathbf{M}, \langle \mathbf{x} - \hat{\mathbf{x}}, \mathbf{y} \rangle = 0 \quad (1-4-10)$$

则称  $\hat{\mathbf{x}}$  是  $\mathbf{x}$  在子空间  $\mathbf{M}$  上的投影。对于某一频率、各次谐波的正余弦信号张成的向量空间  $\mathbf{Y}_k$ ，矩阵中每一个行向量都是该空间内的一个向量。我们来看投影过程：

$$\begin{aligned} \mathbf{X} \mathbf{Y}_k^T (\mathbf{Y}_k \mathbf{Y}_k^T)^{-1} \mathbf{Y}_k &= \hat{\mathbf{X}} \implies \mathbf{X} \mathbf{Y}_k^T = \hat{\mathbf{X}} \mathbf{Y}_k^T \\ \therefore (\mathbf{X} - \hat{\mathbf{X}}) \mathbf{Y}_k^T &= 0 \implies \langle \mathbf{X} - \hat{\mathbf{X}}, \mathbf{Y}_k \rangle = 0 \end{aligned} \quad (1-4-11)$$

由上可知，经过投影后的信号  $\hat{\mathbf{X}}$  位于向量空间  $\mathbf{Y}_k$  内，即纯化了 EEG 信号中与正余弦刺激相关的成分，尽管这个成分也许并不能完全覆盖刺激诱发信号的共性或个性特征。结合此前关于 QR 分解的描述，我们知道  $\mathbf{Q}_k \mathbf{Q}_k^T$  与  $\mathbf{Y}_k^T \mathbf{Y}_k$  的差别仅在于一个数值平衡，起到该效果的部分显然就是  $(\mathbf{Y}_k \mathbf{Y}_k^T)^{-1}$  了：

$$\begin{aligned} \langle \mathbf{Y}_k(i, :), \mathbf{Y}_k(i, :) \rangle &= \|\mathbf{Y}_k(i, :)\|_2^2 \\ \|\mathbf{Y}_k(i, :)\|_2^2 &= \|\mathbf{Y}_k(j, :)\|_2^2, \forall i, j \in [1, 2N_h] \\ (\mathbf{Y}_k \mathbf{Y}_k^T)^{-1} &= \begin{bmatrix} \frac{1}{\|\mathbf{Y}_k(1, :)\|_2^2} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{\|\mathbf{Y}_k(2N_h, :)\|_2^2} \end{bmatrix} = \frac{\mathbf{I}_{2N_h}}{\|\mathbf{Y}_k(i, :)\|_2^2} \end{aligned} \quad (1-4-12)$$

这里需要额外指出， $\|\mathbf{Y}_k(i, :)\|_2^2$  的数值与正余弦信号的频率或相位是无关的（这一点可以自行编程测试，不再证明），且人工构建正余弦信号的峰峰值均为 2（即幅值相等），因此该范数只与数据长度有关。换句话说在每一批数据中，对任意  $k$  该范数均相等：

$$\left( \sum_c \mathbf{Y}_c \mathbf{Y}_c^T \right)^{-1} = \frac{\mathbf{I}_{2N_h}}{N_e \|\mathbf{Y}_k(i, :)\|_2^2} \quad (1-4-13)$$

因此，式 (1-4-6) 可简化为：

$$\begin{aligned} \frac{1}{N_e} \left[ \sum_b \sum_a \bar{\mathbf{X}}_a \frac{\mathbf{Y}_a^T}{\|\mathbf{Y}_k(i, :)\|_2} \frac{\mathbf{Y}_b}{\|\mathbf{Y}_k(i, :)\|_2} \bar{\mathbf{X}}_b^T \right] \mathbf{U}_k^T &= \lambda^2 \left( \sum_k \bar{\mathbf{X}}_k \bar{\mathbf{X}}_k^T \right) \mathbf{U}_k^T \\ \Rightarrow \left( \frac{1}{N_e} \sum_b \sum_a \bar{\mathbf{X}}_a \mathbf{Q}_{\mathbf{Y}_a} \mathbf{Q}_{\mathbf{Y}_b}^T \bar{\mathbf{X}}_b^T \right) \mathbf{U}_k^T &= \lambda^2 \left( \sum_k \bar{\mathbf{X}}_k \bar{\mathbf{X}}_k^T \right) \mathbf{U}_k^T \end{aligned} \quad (1-4-14)$$

这样我们就来到了式 (1-3-18)。条条大路通罗马，不是吗？但比起式 (1-4-14) 这种惊险刺激的过山车，我个人还是更喜欢式 (1-4-5) 这种平静祥和的旅途。

对比可知，上一节中提到的 msCCA 只是式 (1-4-5) 中的  $(I)$ 。尽管它可以进行模板匹配，但是其背后的设计思维显然不止于此，这也是我们通过统一框架无法了解到的信息。式 (1-4-1) 指出，msCCA 的训练目标应当是为训练数据与正余弦信号（二者均为合并样本）各自寻找一组投影向量以满足后续需求，而不是单方面优化 EEG 数据。

结合上述关于投影矩阵相关的说明，我们可以剔除  $\mathbf{C}_{\mathbf{Y}_k \mathbf{Y}_k}^{-1}$  以进一步简化  $(I)$ ：

$$\mathbf{C}_{\mathbf{Z}_k \mathbf{Z}_k}^{-1} \mathbf{C}_{\mathbf{Z}_k \mathbf{Y}_k} \mathbf{C}_{\mathbf{Y}_k \mathbf{Z}_k} \mathbf{U}_k = \lambda^2 \mathbf{U}_k^T \quad (1-4-15)$$

我们来看看这个 GEP 方程对应的广义瑞利商形式：

$$\hat{\mathbf{U}}_k = \arg \max_{\mathbf{U}_k} \frac{\mathbf{U}_k \mathbf{C}_{\mathbf{Z}_k \mathbf{Y}_k} \mathbf{C}_{\mathbf{Y}_k \mathbf{Z}_k} \mathbf{U}_k^T}{\mathbf{U}_k \left( \sum_k \bar{\mathbf{X}}_k \bar{\mathbf{X}}_k^T \right) \mathbf{U}_k^T} \quad (1-4-16)$$

这个方程的分母是**滤波后合并样本的能量**，但是分子就很有意思了，显然  $\mathbf{U}_k \mathbf{C}_{\mathbf{Z}_k \mathbf{Y}_k}$  并不能表示某种信号，因为它的长度已经与采样信号不一样了，这一点在所有 CCA 系列算法中其实都存在。目前我还无法给它一个合理的物理解释，如果各位观众姥爷有高见，希望邮件联系我指点迷津。 $(I)$  的问题到此为止了，可  $(II)$  真不是一位善茬，无论如何都无法缩减运算量，只得就此作罢。最后总结一下 ms-eCCA 的空间滤波器构建函数（GEP 方程）以及模板匹配所需的步骤：

$$\begin{cases} \mathbf{C}_{\mathbf{Z}_k \mathbf{Z}_k}^{-1} \mathbf{C}_{\mathbf{Z}_k \mathbf{Y}_k} \mathbf{C}_{\mathbf{Y}_k \mathbf{Z}_k} \mathbf{U}_k = \lambda^2 \mathbf{U}_k^T & (I) \\ \mathbf{C}_{\mathbf{Y}_k \mathbf{Y}_k}^{-1} \mathbf{C}_{\mathbf{Y}_k \mathbf{Z}_k} \mathbf{C}_{\mathbf{Z}_k \mathbf{Z}_k}^{-1} \mathbf{C}_{\mathbf{Z}_k \mathbf{Y}_k} \mathbf{V}_k = \theta^2 \mathbf{V}_k^T & (II) \end{cases} \quad (1-4-17)$$

$$\rho_k = \sum_{i=1}^2 \text{sign}(\rho_{k,i}) \times \rho_{k,i}^2, \begin{cases} \rho_{k,1} = \text{corr}(\mathbf{U}_k \boldsymbol{\chi}, \mathbf{V}_k \mathbf{Y}_k) \\ \rho_{k,2} = \text{corr}(\mathbf{U}_k \boldsymbol{\chi}, \mathbf{U}_k \bar{\mathbf{X}}_k) \end{cases} \quad (1-4-18)$$

## 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\(\)](#)

---