3. 任务相关成分分析

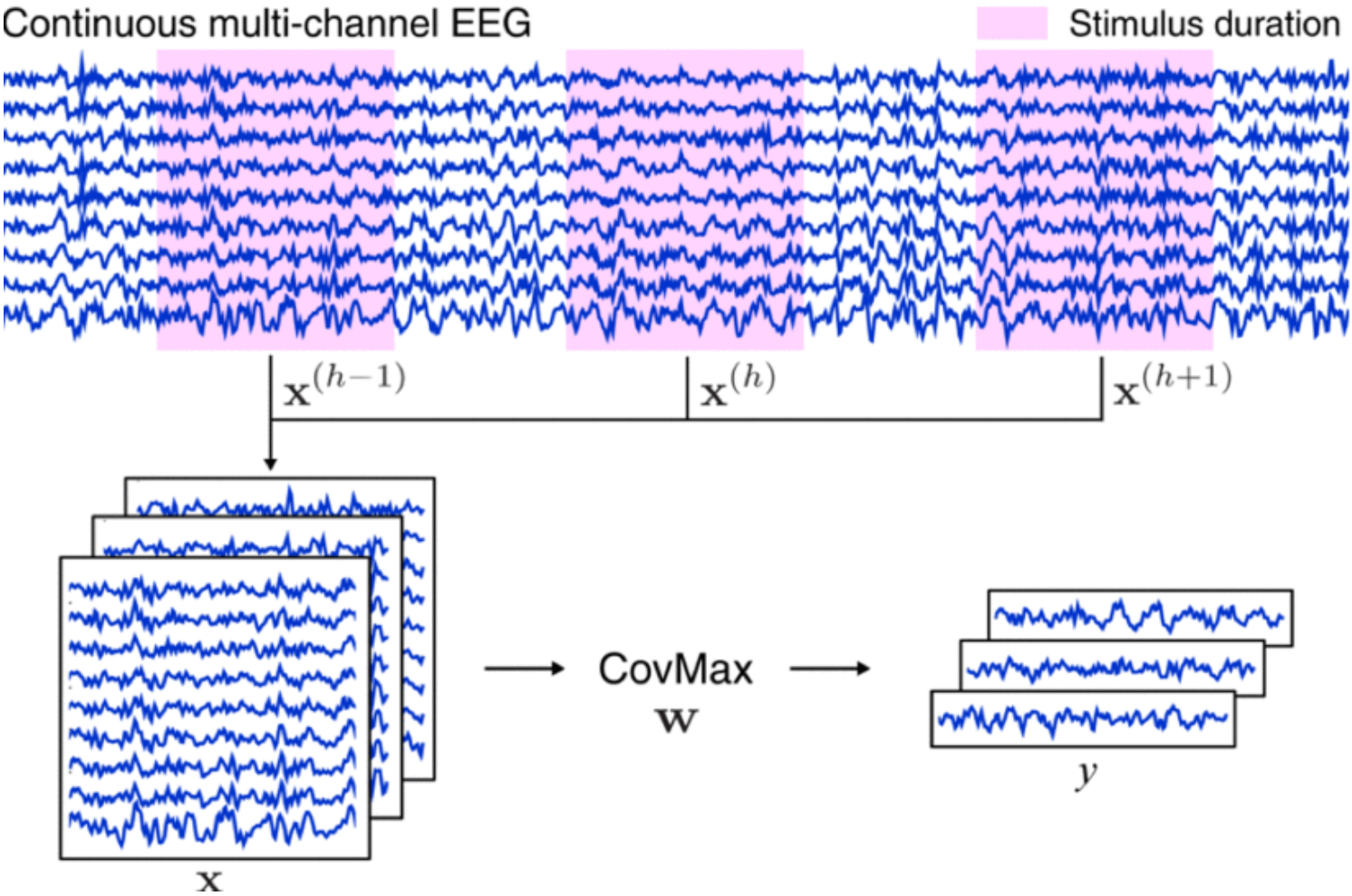
Task-related component analysis, TRCA

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：

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

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

因此，TRCA 的目标函数可写为：

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

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

根据广义瑞利商的结论，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：

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

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

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：

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

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

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：

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

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

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：

$$T_0 = \left(N_c N_p\right) T_\times + \left(N_t - 1\right) T_{M+} \tag{3-1-11}$$

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

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

$$O_2 = O_\times \left({N_c}^2 N_p\right) + O_+ \left[\max\left(N_t N_c N_p, {N_c}^2 N_p\right)\right] \tag{3-1-13}$$

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：

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

eTRCA 是基于 TRCA 的集成学习版本，它把各类别 $\hat{\boldsymbol{\omega}}_k \in \mathbb{R}^{1\times N_c}$ 按行拼接在一起组成高维滤波器 $\hat{\boldsymbol{W}}$，通过计算二维 *Pearson* 相关系数完成信号的模式识别：

$$\begin{cases} \hat{\boldsymbol{W}} = \begin{bmatrix} \hat{\boldsymbol{\omega}}_1 \\ \hat{\boldsymbol{\omega}}_2 \\ \vdots \\ \hat{\boldsymbol{\omega}}_{N_e} \end{bmatrix} \in \mathbb{R}^{N_e \times N_c} \\ \\ \rho_k = corr2\left(\hat{\boldsymbol{W}} \bar{\boldsymbol{X}}_k, \hat{\boldsymbol{W}} \boldsymbol{\chi}\right) \end{cases} \tag{3-1-15}$$

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 $\bar{\bar{\boldsymbol{A}}}$、$\bar{\bar{\boldsymbol{B}}}$：

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

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

别被式 (3-1-17) 一层又一层的叠加操作唬住了，在 *ndarray* 的基础上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，运算速度会有显著提升：

```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，即式 (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 $\boldsymbol{\omega}_k \left(\sum_{i=1}^{N_t} \boldsymbol{X}_k^i \boldsymbol{X}_k^{i\,T} \right) \boldsymbol{\omega}_k{}^T$ 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 多重刺激 TRCA：ms-(e)TRCA

(Multi-stimulus (e)TRCA)

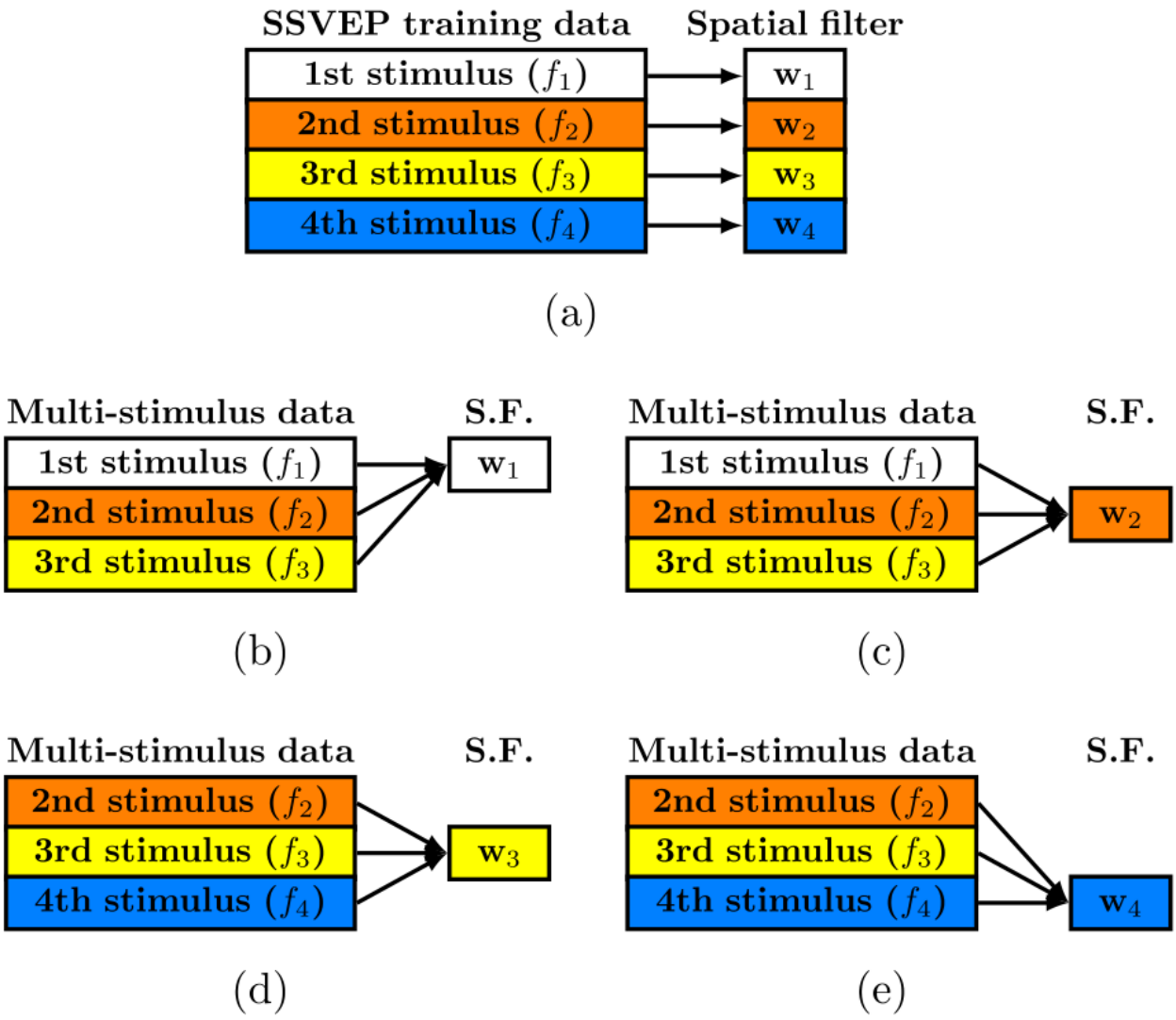
论文链接 | 代码：trca.ms_etrca()

该算法是我们提到的第一种关于TRCA的改进算法，与 msCCA、ms-eCCA 一样出自 *Wong* 的之手，行文风格也一以贯之。由 3.1 节末尾分析可知，TRCA 类型的算法需要较多的训练数据以保证协方差矩阵估计的准确性，在原文中亦有相关描述：

> ***To guarantee good performance, the number of calibration trials for each visual stimulus cannot be small for the eCCA method and the eTRCA method; otherwise, their recognition accuracies would decrease dramatically . . . A major reason is that the estimate of covariance matrices in the CCA-based (or the TRCA-based) methods could become unreliable in the case of small training data so that the resulting spatial filters may not be***

> *accurate. As a matter of fact, the algorithms relying on the estimator of covariance matrix also suffer from insufficient training data problem . . .*

然而对于具有较多指令集（32、40 或更多）的 SSVEP-BCI 系统而言，为每个指令获取足量训练试次（$N_t > 10$）在时间上通常是不允许的。而且刺激时间如果过长，诱发信号的质量会由于受试者累积的视觉疲劳而急剧下降，不利于系统实际应用。因此，*Wong* 提出了一种学习方法，从目标 A 的相邻刺激（B、C 等）的诱发信号中学习 A 的特征。以一个简单的 4 指令集系统为例，ms- 技术的学习流程见下图（原文 fig.1）：



原文对于这一套流程的可行性做了连篇累牍的证明，在此仅提炼其最终结论：

去除 0.14 s 的人眼视觉通路延迟之后，SSVEP 采集信号与闪烁刺激相位基本一致、不同刺激间的相位差也与闪烁刺激相位差基本一致。

听君一席话，如听一席话是吧？这不是 SSVEP 信号处理的常规步骤吗？此外关于 Benchmark 数据集的刺激参数设定（频率、相位等），他们也特地写了三个**方程**来说明，总之一通操作着实看得人头昏脑胀。鉴于本人有幸曾

经听过 *Wong* 的线上报告，视频里他只消三言两语便把 ms-eTRCA 的原理描述得鞭辟入里，所以我相信这种反人类的数学语言其实出自 *Feng Wan* 老师的手笔。

接下来我们先看看原文中给出的公式。ms-eTRCA 与 ms-TRCA 的关系类似 eTRCA 之于 TRCA，因此前两者的目标函数是共通的：

$$\hat{\boldsymbol{\omega}}_k = \arg\max_{\boldsymbol{\omega}_k} \frac{\boldsymbol{\omega}_k \boldsymbol{A}_k {\boldsymbol{A}_k}^T {\boldsymbol{\omega}_k}^T}{\boldsymbol{\omega}_k \boldsymbol{B}_k {\boldsymbol{B}_k}^T {\boldsymbol{\omega}_k}^T} \tag{3-2-1}$$

分子分母中的协方差矩阵与 (e)TRCA 略有差异，具体如下：

$$\begin{cases} \boldsymbol{A}_k = \begin{bmatrix} \bar{\boldsymbol{X}}_{k-m} & \bar{\boldsymbol{X}}_{k-m+1} & \cdots & \bar{\boldsymbol{X}}_{k+n} \end{bmatrix} \in \mathbb{R}^{N_c \times [(m+n+1)N_p]} \\ \boldsymbol{B}_k = \begin{bmatrix} \boldsymbol{\chi}_{k-m} & \boldsymbol{\chi}_{k-m+1} & \cdots & \boldsymbol{\chi}_{k+n} \end{bmatrix} \in \mathbb{R}^{N_c \times [(m+n+1)N_t N_p]} \end{cases} \tag{3-2-2}$$

$$\boldsymbol{\chi}_k = \begin{bmatrix} \boldsymbol{X}_k^1 & \boldsymbol{X}_k^2 & \cdots & \boldsymbol{X}_k^{N_t} \end{bmatrix} \in \mathbb{R}^{N_c \times (N_t N_p)} \tag{3-2-3}$$

可以看出，*multi-stimulus* 技术的本质就是把目标刺激前 m、后 n 个不同频率的刺激信号**顺次拼接**起来，在时间维度上对训练数据进行扩增，其范围为 d（含自身），具体拼接个数（m、n）依不同情况各有一番规定。

（1）关于 m 与 n 的大小分配（一般情况）：若 d 为奇数，则前向与后向扩增范围相等；若 d 为偶数，则前向扩增比后向多一位，即有：

$$\begin{cases} m = n = \dfrac{1}{2}(d-1), \quad d = 2n+1 | n \in \mathbb{N}^+ \\ m = \dfrac{1}{2}d, \ n = \dfrac{1}{2}d - 1, \quad d = 2n | n \in \mathbb{N}^+ \end{cases} \tag{3-2-4}$$

（2）假设刺激目标 k 处于相对**靠前**的位置，由于前向没有足够类别的信号用于拼接，因此需向后顺延扩增位数。例如 $d = 5, k = 2$，应向后顺延一位（$m = 1, n = 3$）；若 $d = 6, k = 2$，则向后顺延两位（$m = 1, n = 4$）。综上可总结出式 (3-2-7)：

$$\begin{cases} m = k - 1 \\ \\ n = d - k \end{cases}, \ k \in \left[1, \ \frac{1}{2}d\right] \tag{3-2-5}$$

（3）假设刺激目标 k 处于**中部**位置，即 (1) 中所述的“一般情况”，则有式 (3-2-8)：

$$\begin{cases} m = \left[\frac{1}{2}d\right] \\ \\ n = d - \left[\frac{1}{2}d\right] - 1 \end{cases}, \ k \in \left(\left[\frac{1}{2}d\right], \ N_e - \left(d - \left[\frac{1}{2}d\right]\right)\right) \tag{3-2-6}$$

（4）假设刺激目标 k 位于**尾部**位置，此时与（2）相反，需向前顺延扩增位数，即有式 (3-2-9)：

$$\begin{cases} m = d - 1 - (N_e - k) \\ \\ n = N_e - k \end{cases}, \ k \in \left[N_e - \left(d - \left[\frac{1}{2}d\right] - 1\right), N_e\right] \tag{3-2-7}$$

好了，我们再回过头去看式 (3-2-3)，该式分子、分母中的协方差矩阵可以通过与式 (3-1-5) 联动，进一步改写为如下形式：

$$\begin{cases} \boldsymbol{A}_k {\boldsymbol{A}_k}^T = \sum_{i=-m}^{n+1} \bar{\boldsymbol{X}}_{k+i} {\bar{\boldsymbol{X}}_{k+i}}^T = \frac{1}{{N_t}^2} \sum_{i=-m}^{n+1} \boldsymbol{S}'_{k+i} \\ \\ \boldsymbol{B}_k {\boldsymbol{B}_k}^T = \sum_{i=-m}^{n+1} \sum_{j=1}^{N_t} \boldsymbol{X}_i^j {\boldsymbol{X}_i^j}^T = \sum_{i=-m}^{n+1} \boldsymbol{Q}_{k+i} \end{cases} \tag{3-2-8}$$

一般来说，非零常系数是不影响矩阵特征值分解结果的。所以我们看 ms-(e)TRCA 的目标函数式，它就是把不同频率信号对应的 (e)TRCA 目标函数的分子、分母各自相加组成新的分式。再直白一点，就是“**把多个频率的信号当一个频率去训练**”，强行增加了可用样本数目。

我们有一点需要注意，根据文章里网格筛选的结果（原文 Fig.3），d 的范围并非是越大越好，在 (e)TRCA 算法上体现得尤为明显。根据本人测试经验，扩增至全类别时（$d = N_e$），ms-TRCA（此时大家共用一套滤波器也就谈不上集成了）往往还不如 eTRCA 效果好，当然显著优于 TRCA。换句话说，在 TRCA 的内核（目标函数）上简单粗暴地扩增协方差矩阵估计，虽然可能起到一定的效果，但是扩增方案以及参数选择仍然需要仔细调整。

说来也令人感慨，ms- 的思路不可谓不简单，但是 *Wong* 等人之所以成功，一方面是因为敢想敢做，另一方面也要归功于砌墙的数学功底，能够把简单的内核包装成高大上的复杂操作，让人一眼完全看不透其内在关联。

3.3 正余弦扩展 TRCA：(e)TRCA-R

论文链接 | **代码：trca.etrca_r()**

该算法依旧出自 *Wong* 的手笔。按照其提出的设计框架，-R 技术就是将原本为单位阵的空间投影矩阵替换为正余弦信号张成的投影空间 $\boldsymbol{\mathcal{P}}$，与之类似的算法还有 MsetCCA1-R（未来更新）。在讲解 (e)TRCA-R 之前，我们先来观

察 (e)TRCA 的目标函数在统一框架（ 1.3 节式 (1-3-4)，**Type I** ）下的各部分组成：

$$\begin{cases} \boldsymbol{\mathcal{Z}} = \boldsymbol{\mathcal{I}}_{N_t,N_c}\left(\oplus_{i=1}^{N_t}\boldsymbol{X}_k^i\right) \in \mathbb{R}^{N_c\times(N_tN_p)} \\ \boldsymbol{\mathcal{D}} = \boldsymbol{I}_{N_tN_p} \in \mathbb{R}^{(N_tN_p)\times(N_tN_p)} \\ \boldsymbol{\mathcal{P}} = \boldsymbol{\mathcal{I}}_{N_t,N_p}{}^T\boldsymbol{\mathcal{I}}_{N_t,N_p} \in \mathbb{R}^{(N_tN_p)\times(N_tN_p)} \end{cases} \tag{3-3-1}$$

为了更清楚地让大家明白这个框架到底干了什么事，我们来依次画一下各步骤的展开形态：

$$\boldsymbol{\mathcal{Z}} = \underbrace{\begin{bmatrix}\boldsymbol{I}_{N_c} & \boldsymbol{I}_{N_c} & \cdots & \boldsymbol{I}_{N_c}\end{bmatrix}}_{\mathbb{R}^{N_c\times(N_tN_c)}} \underbrace{\begin{bmatrix}\boldsymbol{X}_k^1 & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{X}_k^2 & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{X}_k^{N_t}\end{bmatrix}}_{\mathbb{R}^{(N_tN_c)\times(N_tN_p)}} = \underbrace{\begin{bmatrix}\boldsymbol{X}_k^1 & \boldsymbol{X}_k^2 & \cdots & \boldsymbol{X}_k^{N_t}\end{bmatrix}}_{\mathbb{R}^{N_c\times(N_tN_p)}} \tag{3-3-2}$$

$$\boldsymbol{\mathcal{Z}}\boldsymbol{\mathcal{D}} = \underbrace{\begin{bmatrix}\boldsymbol{X}_k^1 & \boldsymbol{X}_k^2 & \cdots & \boldsymbol{X}_k^{N_t}\end{bmatrix}}_{\mathbb{R}^{N_c\times(N_tN_p)}} \underbrace{\begin{bmatrix}\boldsymbol{I}_{N_p} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_{N_p} & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{I}_{N_p}\end{bmatrix}}_{\mathbb{R}^{(N_tN_p)\times(N_tN_p)}} = \boldsymbol{\mathcal{Z}} \tag{3-3-3}$$

$$\boldsymbol{\mathcal{P}} = \underbrace{\begin{bmatrix}\boldsymbol{I}_{N_p} \\ \boldsymbol{I}_{N_p} \\ \vdots \\ \boldsymbol{I}_{N_p}\end{bmatrix}}_{\mathbb{R}^{(N_tN_p)\times N_p}} \underbrace{\begin{bmatrix}\boldsymbol{I}_{N_p} & \boldsymbol{I}_{N_p} & \cdots & \boldsymbol{I}_{N_p}\end{bmatrix}}_{\mathbb{R}^{N_p\times(N_tN_p)}} = \underbrace{\begin{bmatrix}\boldsymbol{I}_{N_p} & \boldsymbol{I}_{N_p} & \cdots & \boldsymbol{I}_{N_p} \\ \boldsymbol{I}_{N_p} & \boldsymbol{I}_{N_p} & \cdots & \boldsymbol{I}_{N_p} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{I}_{N_p} & \boldsymbol{I}_{N_p} & \cdots & \boldsymbol{I}_{N_p}\end{bmatrix}}_{\mathbb{R}^{(N_tN_p)\times(N_tN_p)}} \tag{3-3-4}$$

$$\boldsymbol{\mathcal{Z}}\boldsymbol{\mathcal{D}}\boldsymbol{\mathcal{P}} = \underbrace{\begin{bmatrix}\boldsymbol{X}_k^1 & \cdots & \boldsymbol{X}_k^{N_t}\end{bmatrix}}_{\mathbb{R}^{N_c\times(N_tN_p)}} \underbrace{\begin{bmatrix}\boldsymbol{I}_{N_p} & \cdots & \boldsymbol{I}_{N_p} \\ \vdots & \ddots & \vdots \\ \boldsymbol{I}_{N_p} & \cdots & \boldsymbol{I}_{N_p}\end{bmatrix}}_{\mathbb{R}^{(N_tN_p)\times(N_tN_p)}} = \underbrace{\begin{bmatrix}\sum_{i=1}^{N_t}\boldsymbol{X}_k^i & \cdots & \sum_{i=1}^{N_t}\boldsymbol{X}_k^i\end{bmatrix}}_{\mathbb{R}^{N_c\times(N_tN_p)}} \tag{3-3-5}$$

$$\boldsymbol{\mathcal{Z}}\boldsymbol{\mathcal{D}}\boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{P}}^T\boldsymbol{\mathcal{D}}^T\boldsymbol{\mathcal{Z}}^T = \underbrace{\begin{bmatrix}\sum_{i=1}^{N_t}\boldsymbol{X}_k^i & \cdots & \sum_{i=1}^{N_t}\boldsymbol{X}_k^i\end{bmatrix}}_{\mathbb{R}^{N_c\times(N_tN_p)}}\underbrace{\begin{bmatrix}\sum_{i=1}^{N_t}\boldsymbol{X}_k^i \\ \vdots \\ \sum_{i=1}^{N_t}\boldsymbol{X}_k^i\end{bmatrix}}_{\mathbb{R}^{(N_tN_p)\times N_c}} = N_t\sum_{j=1}^{N_t}\sum_{i=1}^{N_t}\boldsymbol{X}_k^i{\boldsymbol{X}_k^i}^T \tag{3-3-6}$$

$$\boldsymbol{\mathcal{Z}}\boldsymbol{\mathcal{D}}\boldsymbol{\mathcal{D}}^T\boldsymbol{\mathcal{Z}}^T = \boldsymbol{\mathcal{Z}}\boldsymbol{\mathcal{Z}}^T = \underbrace{\begin{bmatrix}\boldsymbol{X}_k^1 & \boldsymbol{X}_k^2 & \cdots & \boldsymbol{X}_k^{N_t}\end{bmatrix}}_{\mathbb{R}^{N_c\times(N_tN_p)}}\underbrace{\begin{bmatrix}\boldsymbol{X}_k^1 \\ \boldsymbol{X}_k^2 \\ \vdots \\ \boldsymbol{X}_k^{N_t}\end{bmatrix}}_{\mathbb{R}^{(N_tN_p)\times N_c}} = \sum_{i=1}^{N_t}\boldsymbol{X}_k^i{\boldsymbol{X}_k^i}^T \tag{3-3-7}$$

综上所述，仅需一维投影向量的情况下，*GEP* 方程可表示为式 (3-3-8)，忽略常系数影响后可发现该式与 (e)TRCA 的目标函数完全吻合。

$$\left(N_t\sum_{j=1}^{N_t}\sum_{i=1}^{N_t}\boldsymbol{X}_k^i{\boldsymbol{X}_k^i}^T\right)\boldsymbol{\omega} = \lambda\left(\sum_{i=1}^{N_t}\boldsymbol{X}_k^i{\boldsymbol{X}_k^i}^T\right)\boldsymbol{\omega} \tag{3-3-8}$$

在常规 (e)TRCA 中，正交投影矩阵 $\boldsymbol{\mathcal{P}}$ 的本质作用仅是叠加。而在 (e)TRCA-R 中，*Wong* 将其改为了正余弦信号张成的投影空间，其余均与 (e)TRCA 保持一致：

$$\boldsymbol{\mathcal{P}} = \begin{bmatrix}\boldsymbol{Q}_{\boldsymbol{Y}_k} \\ \vdots \\ \boldsymbol{Q}_{\boldsymbol{Y}_k}\end{bmatrix}\begin{bmatrix}\boldsymbol{Q}_{\boldsymbol{Y}_k} & \cdots & \boldsymbol{Q}_{\boldsymbol{Y}_k}\end{bmatrix} = \begin{bmatrix}\boldsymbol{Q}_{\boldsymbol{Y}_k}{\boldsymbol{Q}_{\boldsymbol{Y}_k}}^T & \cdots & \boldsymbol{Q}_{\boldsymbol{Y}_k}{\boldsymbol{Q}_{\boldsymbol{Y}_k}}^T \\ \vdots & \ddots & \vdots \\ \boldsymbol{Q}_{\boldsymbol{Y}_k}{\boldsymbol{Q}_{\boldsymbol{Y}_k}}^T & \cdots & \boldsymbol{Q}_{\boldsymbol{Y}_k}{\boldsymbol{Q}_{\boldsymbol{Y}_k}}^T\end{bmatrix} \tag{3-3-9}$$

注意有 $\boldsymbol{\mathcal{P}} = \boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{P}}^T$，所以 (e)TRCA-R 的 *GEP* 方程可表示为：

$$\left[N_t\sum_{j=1}^{N_t}\sum_{i=1}^{N_t}\left(\boldsymbol{X}_k^i\boldsymbol{Q}_{\boldsymbol{Y}_k}{\boldsymbol{Q}_{\boldsymbol{Y}_k}}^T\right)\left(\boldsymbol{X}_k^j\boldsymbol{Q}_{\boldsymbol{Y}_k}{\boldsymbol{Q}_{\boldsymbol{Y}_k}}^T\right)^T\right]\boldsymbol{\omega}^T = \lambda\left(\sum_{i=1}^{N_t}\boldsymbol{X}_k^i{\boldsymbol{X}_k^i}^T\right)\boldsymbol{\omega}^T \tag{3-3-10}$$

关于投影矩阵的作用，在 ms-eCCA 章节中已有介绍，此处不再赘述。由于 -R 技术在 $\boldsymbol{X}_k^i{\boldsymbol{X}_k^i}^T$ 的过程中插入了额外的矩阵乘法，导致 TRCA 复现过程中的一个重要 trick（通过叠加平均信号的矩阵乘法替代跨试次循环）不再生效，而 **np.einsum()** 函数在面对跨试次矩阵乘法时似乎又存在一些内在的逻辑问题，其运算效率出奇地低，因此目前我推荐的复现方法只有循环。

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()
