猕猴手指移动神经解码线性时不变模型的时间相关性研究

冯景义 [1] 吴海锋 [1] 曾 玉 [1]

摘 要 利用猕猴运动皮层神经元峰电位数信号估计其手指移动位置是一神经解码问题, 通常采用时不变线性模型 (Time-invariant linear model, TILM) 来解决. 本文分析了传统 TILM 模型的时间相关性问题, 依据猕猴手指移动位置的连续性特点, 采用一种新的模型去解码其手指移动位置, 称之为卷积空间模型 (Convolution space model, CSM). 与传统的模型相比, 卷积空间模型不但将当前时刻的状态与前一个时刻建立了相关, 而且与前多个时刻的状态也有相关. 在实验中, 利用公开数据来评判本文方法的解码性能, 实验结果表明, 传统方法的解码误差要大于 CSM 模型的方法, 因此 CSM 模型具有更好的解码准确性.

关键词 神经解码, 猕猴手指移动, CSM 模型, TILM 模型

引用格式 冯景义, 吴海锋, 曾玉. 猕猴手指移动神经解码线性时不变模型的时间相关性研究. 自动化学报, 2021, **47**(2): 442−452

DOI 10.16383/j.aas.c180098

开放科学 (资源服务) 标识码 (OSID):

Time Correlation of Time-invariant Linear Models in Neural Decoding for the Macaque′s Moving Finger

FENG Jing-Yi[1] WU Hai-Feng[1] ZENG Yu[1]

Abstract It is a neural decoding problem to estimate the position of a macaque′s moving finger through neuron spike signals in motor cortex, which is usually solved by a time-invariant linear model (TILM). This paper analyzes the temporal correlation of the traditional TILM model. According to the continuity characteristics of the position of a macaque' s moving finger, a new model is adopted to decode the finger movement, which is called CSM (Convolution space model). Compared with traditional decoding models, the CSM model can express that a state at the current time will be related to states at multiple previous times, rather than only one previous time. In experiments, we use the public data to evaluate the decoding performance of our method. The experimental results show that the CSM model has lower decoding errors than traditional methods and thus has better decoding accuracy.

Key words Neural decoding, macaque′s moving finger, CSM model, TILM model

Citation Feng Jing-Yi, Wu Hai-Feng, Zeng Yu. Time correlation of time-invariant linear models in neural decoding for the macaque′s moving finger. *Acta Automatica Sinica*, 2021, **47**(2): 442−452

通过对神经回路感知外部世界和产生行为的研究, 可揭示大脑的工作机制和规律, 也可让人体增强感知外部世界和控制外部世界的能力. 在神经编码中, 例如, 对听力障碍者来说, 人工耳蜗可将声音信号编码为计算系统可处理的数字信号, 通过刺激听觉神经使得患者具有感知外界声音的能力[1]. 在神经解码中, 对于伤残人士来说, 可直接通过大脑运动皮层的神经元峰电位数信号去操控外界设备的移动, 如移动鼠标[2−3], 机械手臂[4−5] 等, 以及截瘫患者的上肢和手[6] 等, 使其拥有一定的外部世界控制能力, 目前康复机器人与智能辅助系统的研究取得了不错的进展[7].

神经编码将外部世界映射至脑活动[8], 把从脑区获得的神经元峰电位信号进行分类处理, 其处理后

收稿日期 2018-02-12 录用日期 2018-08-28
Manuscript received February 12, 2018; accepted August 28, 2018
国家自然科学基金(61762093), 云南省科技厅第十七批省中青年学术和技术带头人 (2014HB019), 云南省高校科技创新团队, 云南省重点应用和基础研究基金 (2018FA036), 云南省教育厅科学研究基金项目 (2018Y106), 云南民族大学研究生创新基金科研项目 (2018YJCXS175) 资助
Supported by National Natural Science Foundation of China (61762093), The 17th batches of Young and Middleaged Leaders in Academic and Technical Reserved Talents Project of Yunnan Province (2014HB019), The Program for Innovative Research Team (in Science and Technology) in University of Yunnan Province, The Key Applied and Basic Research Foundation of Yunnan Province (2018FA036), The Science Research Fund Program in Education Department of Yunnan Province (2018Y106), The Graduate Student Innovation Fund Research Project in Yunnan Minzu University (2018YJCXS175)
本文责任编委 秦涛
Recommended by Associate Editor QIN Tao
1. 云南民族大学电气信息工程学院 昆明 650504
1. Department of Information Engineering, Yunnan Minzu University, Kunming 650504

的峰电位信号需与肌肉或骨骼对外部世界产生的活动建立一一对应关系. 神经解码是一种对神经编码的逆过程, 从脑活动中解析出人体对外部世界的动作, 例如, 通过分类后的峰电位信号去预测或估计身体的运动过程, 而本文所研究的猕猴手指移动位置估计是一个典型的神经编解码问题. 首先完成神经编码, 将神经记录仪器所采集的某一时刻猕猴运动皮层神经元峰电位数与该时刻猕猴手指移动位置建立对应关系. 由于该问题已经在文献 [9−14] 中做了详细讨论, 因此本文不再将其作为研究重点. 其次完成神经解码, 通过已经与猕猴手指移动建立对应关系的神经元峰电位数去估计猕猴手指移动位置信息, 而本文的研究重点将集中在如何更好地解决神经解码问题上.

较早的猕猴手指移动的编码问题在文献 [15] 中进行了介绍, 该文献发现猕猴的上部肢体的运动方向与其脑区的运动皮层中单个神经元的峰电位信号存在着相应关系; Vargas-Irwin 等实现了猕猴机器手臂的三维运动轨迹的重建, 证实利用局部的神经集群发放信号可以解析出丰富的运动信息[16]; O′Doherty 等在猕猴的初级感觉皮层上第一次实现了带有触觉反馈的闭环的脑机接口系统[17]. 在传统的解码方法中, 较早采用的是独立线性 (Linear) 方法[18−19], 该方法利用的是线性时不变模型 (Time-invariant linear model, TILM), 将每一时刻运动状态值与该时刻记录的峰电位数信号看作时不变的正比关系, 其最大优点是易于实施且计算简单, 但缺点是把运动轨迹每一时刻的状态值均看成了一个独立过程, 因此预测轨迹的准确度较低.

目前, 为了解决该问题, 常采用的方法是状态空间模型 (State space model, SSM). 在神经科学领域中, Velliste 等通过 SSM 模型解码猕猴的运动皮层神经元峰电位数信号, 实现了对四自由度假肢的控制[20]; Shanechi 等通过构建 SSM 模型和应用最优反馈控制模型解码出了猴子运动的轨迹状态[21]; Chang 等基于卡尔曼滤波 (Kalman-filtering, KF) 方法, 设计出了同时考虑 “手控” 和 “脑控” 过程的算法, 加快了脑控的实现[22]. 薛明龙等通过 SSM 模型, 采用了一种无监督积分卡尔曼滤波解码模型 (Unsupervised Cabuture-Kalman-filtering decoding, UCKD) 解码出了猕猴手指移动轨迹的位置[23−24]; Hotson 等基于递归贝叶斯估计 (Recursive Bayes estimation, RBE), 通过整合环境传感器信息提高了脑机接口系统的解码效果[25]. 最近, 浙江大学李宏宝还实现了猕猴手臂规避障碍过程中的运动前区皮层的神经解码[26], 以及张毅等利用 BRCSP (Bagging regularized common spatial pattern) 算法实现左右手运动想象的脑电信号去控制智能轮椅完成了 “8” 字形路径实验[27]. 从 SSM 的观测方程看, 仍是一种 TILM 模型, 但同独立线性方法相比, SSM 未把手指移动轨迹看成是一个时间上独立的过程, 而是将当前时刻的移动状态与前一个时刻相关联, 因此估计精确度有了较大提高.

本文通过对传统 TILM 模型的时间相关性进行研究, 从 SSM 模型出发, 将每一时刻的手指移动状态值与之前多个时刻的峰电位数的簇向量进行相关, 推导出了另一种 TILM 模型. 因为该模型是把猕猴手指移动轨迹的位置表示为一组神经元峰电位信号的簇向量与一组常系数的卷积, 故称之为卷积空间模型 (Convolutional space model, CSM). 为了训练该模型, 采用最小二乘和最陡梯度下降等常规方法来得到模型参数, 同时分析了时间相关性对这些方法的影响. 在实验中, 采用一组公开的实测数据对本文的时间相关性问题进行验证, 实验结果表明, CSM 模型的训练算法所解码的手指移动位置信息要比传统的 SSM 模型有较小的解码误差.

1 相关工作

在猕猴手指移动神经解码中, SSM 模型是目前较为流行的一种解码方法, 其状态方程和观测方程可分别表示为[28−31]

$$y_k = h(y_{k-1}) + \omega_k \tag{1a}$$

$$\boldsymbol{s}_k = \boldsymbol{f}(y_k) + \boldsymbol{v}_k \tag{1b}$$

其中, y_k 为时刻 k 的手指移动位置信息, $k = 0, 1, \cdots, K-1, K$ 为数据采样点长度, $\boldsymbol{s}_k$ 为 $N_e \times 1$ 的列向量, 表示为时刻 k 的 N_e 个电极上所采集的神经元峰电位数, $h(\cdot)$ 表示状态方程函数, $\boldsymbol{f}(\cdot)$ 表示观测方程函数, ω_k 为零均值, 方差为 σ^2 的高斯白噪声[23], $\boldsymbol{v}_k = [v_{0k}, v_{1k}, \cdots, v_{N_e-1,k}]^{\mathrm{T}}$ 是高斯白噪声矢量[23], 其均值为 0, 方差矩阵 $\boldsymbol{R}_v$ 为对角线是 $[\sigma^2_{0k}, \sigma^2_{1k}, \cdots, \sigma^2_{N_e-1,k}]$ 的对角矩阵.

求解式 (1) 的状态空间方程通常采用逐次状态估计方法, 可利用卡尔曼滤波 (KF)[22, 29], 粒子滤波[8, 28] 和 RBE[25] 等方法. 相比于将 $\{y_k, k = 0, 1, \cdots, K-1\}$ 看作是独立的 Linear 方法[8], 与前一个状态相关的 SSM 模型所解码得出的手指移动轨迹曲线则更加平滑和抖动较小.

再者, SSM 模型方法需要计算出该模型的 $h(\cdot)$ 和 $\boldsymbol{f}(\cdot)$ 函数, 一种常用的模型是将这些函数看作线性并利用训练数据得到, 此时, 式 (1) 变为

$$y_k = y_{k-1} + \omega_k \tag{2a}$$

$$\boldsymbol{s}_k = \boldsymbol{a}_k y_k + \boldsymbol{b}_k + \boldsymbol{v}_k \tag{2b}$$

其中, $\boldsymbol{a}_k$, $\boldsymbol{b}_k$ 系数为 $N_e \times 1$ 的列向量. 由于该系数为时不变, 故式 (2b) 可认为仍是一种 TILM 模型.

另外, 有人采用一种无监督的 UCKD 模型[23] 把式 (1) 表示成了两组状态空间方程式, 一组采用无监督算法来预测手指移动位置, 另一组用来求解 $\boldsymbol{f}(\cdot)$ 函数, 其过程并未引入训练数据部分.

从以上基于式 (1) ~ (2) 的 SSM 类方法看, 其均认为当前时刻的手指移动位置状态 y_k 仅与前一个时刻状态 y_{k-1} 相关, 在下一节中, 我们将从时间相关性角度来分析 SSM 模型的合理性.

2 猕猴手指移动解码与时间相关性问题

猕猴手指移动位置的编解码过程, 具体编解码过程如图 1 所示. 在相关采集设备齐全的情况下, 选用一只正常健康的猕猴, 其手指能够自由运动的活动区域长为 L, 宽为 B.

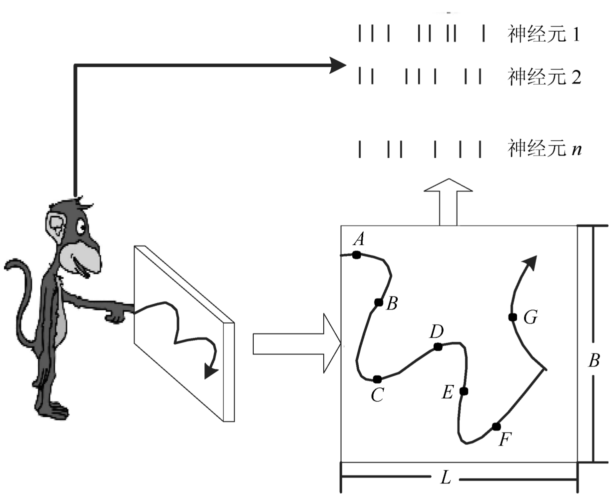


图 1 猕猴手指移动轨迹编码

Fig. 1 Macaque finger movement track coding

首先, 训练猕猴, 在活动区域内会出现亮点作为目标物, 不同时刻目标物出现在活动区域的不同位置, 猕猴会用手指依次移向目标物, 经过反复训练多次, 最终猕猴能够独立地完成此项任务为止. 其次, 若目标物出现的周期为 Δt, 则在猕猴手部运动相关的活动脑区中植入的神经电极阵列将采集到第 $k\Delta t$, $k=0,1,\cdots,K-1$ 时刻的信息, 即每一时刻的 N_e 个神经元峰电位数信号 $\boldsymbol{s}_k$. 然后, 在每一时刻还需要记录猕猴手指移动的位置信息, 如第 $k\Delta t$ 时刻手指移动到 A 点, 在记录神经元峰电位数信号 $\boldsymbol{s}_k$ 的同时, 也要记录手指移动到 A 点的 X 和 Y 轴坐标值 y_k, 依次记录到 B、C、D、E $\cdots\cdots$ 的坐标位置值 y_k. 最后, 猕猴手指移动过程中共采集到了 K 组数据, 即分别是 K 组神经元峰电位数信号 $\boldsymbol{s}_k$ 和位置坐标 y_k 具有相互对应关系. 我们所要做的神经解码技术就是通过 $\boldsymbol{s}_k$ 获取 $\hat{y}_k$, 尽量使得估计 $\hat{y}_k$ 与真实 y_k 达到一致.

从猕猴手指移动过程看, 其移动位置轨迹 $\{y_k, k=0,1,\cdots,K-1\}$ 应是一个连续过程, 而基于 SSM 模型的传统解码方法仅将当前时刻的位置状态与前一个时刻的状态相关. 其实, 猕猴手指移动过程本身是一个具有一定速度和方向的连续过程, 因此, 其每个时刻的移动状态应该与以前若干个时刻的状态存在一定联系, 若仅将当前时刻的状态与前一个时刻建立相关性, 可能不一定是最优的. 所以, 本文尝试将当前时刻的运动状态与前若干个时刻的运动状态建立一定的联系, 进而去寻找一种在时间相关性上更优的解码模型.

3 线性时不变模型的时间相关性研究

3.1 CSM 模型

本小节将从时间相关性角度出发, 将当前时刻的位置状态与之前若干个时刻相关, 以克服 SSM 模型中描述手指移动轨迹连续性的不足. 在该模型的推导中, 将从 SSM 模型入手, 得到了一种具有卷积形式的空间模型, 其可以较为严格地确定和说明噪音的分布和特性.

假定本文模型中时刻 k 的位置状态 y_k 与前 P 个时刻 $k, k-1,\cdots,k-P+1$ 相关, 则由 (2b) 可得, 在第 $k-p$ 时刻的位置坐标

$$y_{k-p}=\boldsymbol{w}_p^{\mathrm{T}}\boldsymbol{s}_{k-p}+b'_p+\omega'_p,\quad p=0,1,\cdots,P-1 \tag{3}$$

其中 $\boldsymbol{w}_p^{\mathrm{T}}=\boldsymbol{a}_{k-p}^{\dagger}$ 表示为系数行向量, $(\cdot)^{\mathrm{T}}$ 为共轭矩阵, $b'_p=-\boldsymbol{w}_p^{\mathrm{T}}\boldsymbol{b}_{k-p}$, $\omega'_p=-\boldsymbol{w}_p^{\mathrm{T}}\boldsymbol{v}_{k-p}$ 仍为零均值高斯白噪声 (见附录 A).

将式 (2a) 代入 (3) 中, 可得

$$\left\{\begin{array}{l} y_k=\boldsymbol{w}_0^{\mathrm{T}}s_k+b'_0+\omega'_0\\ y_k=\boldsymbol{w}_1^{\mathrm{T}}s_{k-1}+b'_1+\omega''_1\\ \qquad\vdots\\ y_k=\boldsymbol{w}_{P-1}^{\mathrm{T}}s_{k-P+1}+b'_{P-1}+\omega''_{P-1}\end{array}\right. \tag{4}$$

其中, $\omega''_p=\omega'_p+\omega_{k-1}+\omega_{k-2}+\cdots+\omega_{k-p}$, $p=0,1,\cdots,P-1$ 仍为高斯白噪声 (见附录 A). 将式 (4) 中各式累加, 有

$$y_k=\sum_{p=0}^{P-1}\boldsymbol{w'}_p^{\mathrm{T}}\boldsymbol{s}_{k-p}+B+\Omega_k \tag{5}$$

其中 $\boldsymbol{w'}_p^{\mathrm{T}}=\boldsymbol{w}_p^{\mathrm{T}}/P$ 表示为权向量, $B=(b'_0+b'_1+\cdots+b'_{P-1})/P$ 表示为偏置. $\Omega_k=(\omega'_0+\omega''_1+\omega''_2+\cdots\omega''_{P-1})/P$, 仍为零均值高斯白噪声 (见附录 A).

注意到 $\Sigma_{p=0}^{P-1}\boldsymbol{w'}_p^{\mathrm{T}}\boldsymbol{s}_{k-p}$ 中 $\boldsymbol{w'}_k$ 与 $\boldsymbol{s}_k$ 均为向量, 令

$$\boldsymbol{w}'_p = [w'_{0p}, w'_{1p}, \cdots, w'_{N_e-1,p}]^{\mathrm{T}}$$
$$\boldsymbol{s}_{k-p} = [s_{0,k-p}, s_{1,k-p}, \cdots, s_{N_e-1,k-p}]^{\mathrm{T}}$$

则有

$$y_k = \sum_{p=0}^{P-1}\sum_{n=0}^{N_e-1} w'_{np}s_{n,k-p} + B + \Omega_k \tag{6}$$

式 (6) 表明了, 运动状态位置信息 y_k 是神经元峰电位数向量 $\boldsymbol{s}_k$ 与权向量 $\boldsymbol{w}'_k$ 的二维卷积形式, 见图 2. 从图 2 可看出, 该二维卷积模型应该更加适合处理神经元峰电位数信号中的多维形式.

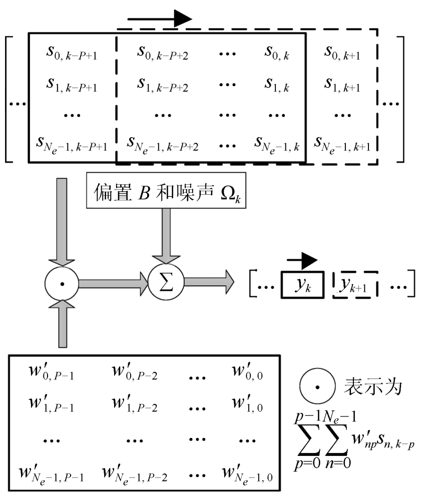


图 2　二维卷积空间模型示意图

Fig. 2　Two dimensional convolution space model

进一步简化, 令

$$\boldsymbol{W} = [\boldsymbol{w}'^{\mathrm{T}}_0, \boldsymbol{w}'^{\mathrm{T}}_1, \cdots, \boldsymbol{w}'^{\mathrm{T}}_{P-1}, B]^{\mathrm{T}}$$
$$\boldsymbol{S}_k = [\boldsymbol{s}^{\mathrm{T}}_k, \boldsymbol{s}^{\mathrm{T}}_{k-1}, \cdots, \boldsymbol{s}^{\mathrm{T}}_{k-P+1}, 1]^{\mathrm{T}}$$

则有

$$y_k = \boldsymbol{S}^{\mathrm{T}}_k\boldsymbol{W} + \Omega_k \tag{7}$$

式 (7) 进一步表明运动状态位置 y_k 是神经元峰电位数矩阵 $\boldsymbol{S}_k$ 与权矩阵 $\boldsymbol{W}$ 的内积并加上高斯白噪声 Ω_k. 从式 (6) ~ (7) 的卷积模型中看到, y_k 不仅与时刻 k 的峰电位 $\boldsymbol{s}_k$ 相关, 还与 $\boldsymbol{s}_{k-P+1}$, $\boldsymbol{s}_{k-P+2}$, $\cdots$, $\boldsymbol{s}_{k-1}$ 相关. 另外, 值得注意的是, 模型中并没有考虑位置信息的时间相关性, y_k 并没有与 y_{k-1}, y_{k-2}, $\cdots$, 等相关.

3.2　时间相关性对模型训练的影响

接下来, 需要对前建立的模型进行训练, 通过训练数据求解出模型的权值参数 $\boldsymbol{W}$. 从第 3.1 节可知, CSM 模型所解决的时间相关性问题, 主要是引入的参数 P 将当前时刻的轨迹状态与前若干个时刻的状态变化建立了相关性. 当 $P = 1$ 时, 模型为时间独立的线性模型, 故本节将重点分析和讨论参数 $P > 1$ 时, 对该模型训练过程的影响.

3.2.1　时间相关性对训练方法的影响

在 TILM 模型中, 若噪声为零均值高斯白噪声条件下, 则可采用最小二乘法 (Least squares, LS) 和最陡梯度下降 (Gradient descent algorithm, GDA) 等训练该模型. 由于 CSM 模型中出现的噪声都是高斯白噪声 (附录 A), 故可采用 LS, 表示为[32]

$$\hat{\boldsymbol{W}} = \bar{\boldsymbol{S}}^{\dagger}\boldsymbol{Y} \tag{8}$$

其中 $\bar{\boldsymbol{S}} = [\boldsymbol{S}_P, \boldsymbol{S}_{P+1}, \cdots, \boldsymbol{S}_K]^{\mathrm{T}}$, $\boldsymbol{Y} = [y_P, y_{P+1}, \cdots, y_K]^{\mathrm{T}}$, $\hat{\boldsymbol{W}}$ 是 $\boldsymbol{W}$ 的估计矩阵.

由于 CSM 模型的参数 $P > 1$, 因此在 LS 算法中, 式 (8) 中的观测矩阵 $\bar{\boldsymbol{S}}$ 的列数将由原来的 $N_e + 1$ 变化为 $PN_e + 1$. 观测矩阵的维度变化将直接影响到模型训练的复杂度.

另外, 可采用批处理的递归最小二乘法 (Recursive Least Square, RLS) 去训练 CSM 模型, 表示为[32]

$$e_k = y_k - \boldsymbol{S}^{\mathrm{T}}_k\tilde{\boldsymbol{W}}_{k-1} \tag{9}$$

$$\boldsymbol{K}_k = \frac{\boldsymbol{P}_{k-1}\boldsymbol{S}_k}{\lambda + \boldsymbol{S}^{\mathrm{T}}_k\boldsymbol{P}_{k-1}\boldsymbol{S}_k} \tag{10}$$

$$\tilde{\boldsymbol{W}}_k = \tilde{\boldsymbol{W}}_{k-1} + \boldsymbol{K}_k e_k \tag{11}$$

$$\boldsymbol{P}_k = \frac{1}{\lambda}[\boldsymbol{P}_{k-1} - \boldsymbol{K}_k\boldsymbol{S}^{\mathrm{T}}_k\boldsymbol{P}_{k-1}] \tag{12}$$

其中, $\tilde{\boldsymbol{W}}_k$ 是第 k 时刻的迭代权值, $\boldsymbol{K}_k$ 是第 k 时刻的增益向量, $\boldsymbol{P}_k$ 是第 k 时刻的逆相关矩阵, $0 < \lambda < 1$, 表示遗忘因子.

同样, (9) ~ (12) 中观测矩阵 $\boldsymbol{S}_k$ 的列数也由原来的 $N_e + 1$ 变化为 $PN_e + 1$.

最后, 梯度下降法 (Gradient descent algorithm, GDA) 同样是一种常用的批处理模型训练算法, 表示为[32]

$$\tilde{\boldsymbol{W}}_k = \tilde{\boldsymbol{W}}_{k-1} - 2\mu e_k\boldsymbol{S}_k \tag{13}$$

其中, μ 是迭代步长. 与 RLS 算法类似, 其观测矩阵仍采用列数为 $PN_e + 1$ 的矩阵 $\boldsymbol{S}_k$.

3.2.2　时间相关性对训练复杂度的影响

在 CSM 模型中, LS 的 $\bar{\boldsymbol{S}}$ 为一个 $(K - P + 1) \times (PN_e + 1)$ 的矩阵, 则对 $\bar{\boldsymbol{S}}$ 求伪逆及权值则需要 Ln^3 次乘法, 有

$$L = K - P + 1 \tag{14}$$

$$n = PN_e + 1 \tag{15}$$

忽略低阶项和加减负项, LS 关于参数 P 的计算复杂度可表示为 $\mathrm{O}(P^3)$. 其次, RLS 第 k 步 $\tilde{\boldsymbol{W}}_k$

的求解需要 n^3+2n^2+n 次乘除, 总共要完成 L 步, 如果 $\tilde{\boldsymbol{W}}_k$ 需要 T 次循环才能达到收敛, 那么 RLS 需要的乘除次数共为 $TL(n^3+2n^2+n)$, 忽略低阶项和加减负项, 因此 RLS 关于 P 的计算复杂度可表示为 $\mathrm{O}(TP^3)$. 还有, GDA 第 k 步 $\tilde{\boldsymbol{W}}_k$ 的求解需 n^2+2n 次乘除, 共需完成 L 步, 如果 $\tilde{\boldsymbol{W}}_k$ 共需 T 次循环才能达到收敛, 那么 GDA 需要的乘除次数共为 $TL(n^2+2n)$, 因此 GDA 关于参数 P 的计算复杂度可表示为 $\mathrm{O}(TP^2)$.

表 1 给出了 CSM 模型训练关于参数 P 的计算复杂度, 由于传统的时间独立模型中参数 $P=1$, 因此 CSM 模型的复杂度相比于传统模型的复杂度分别增加了 P^3, P^3 和 P^2 倍.

表 1 参数 P 对算法的训练复杂度

Table 1 Training complexity of parameter P for algorithm

CSM 模型算法	LS	RLS	GDA
复杂度	$\mathrm{O}(P^3)$	$\mathrm{O}(TP^3)$	$\mathrm{O}(TP^2)$

3.2.3 时间相关性对训练参数的影响

在 CSM 模型中, 所求卷积核 $\boldsymbol{W}$ 的参数个数为 PN_e+1, 因此, 相较于时间独立模型的参数个数, 其增加了约 P 倍.

图 3 给出 CSM 模型在 4 组实验中的卷积核 $\boldsymbol{W}$ 训练结果, 其中权重大小为 $\boldsymbol{w}'_p, p=1,2,\cdots,P$ 的 2 范数, $P=10$, 训练方法采用 GDA 算法, 4 组实验的设置可参见第 5.2 节. 从图中 4 组实验的训

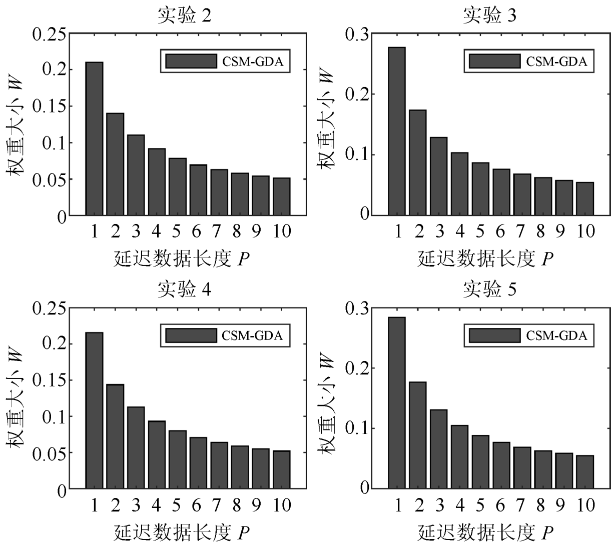


图 3 时间相关性下卷积核权重大小分布

Fig. 3 Convolution kernel weight distribution in time correlation

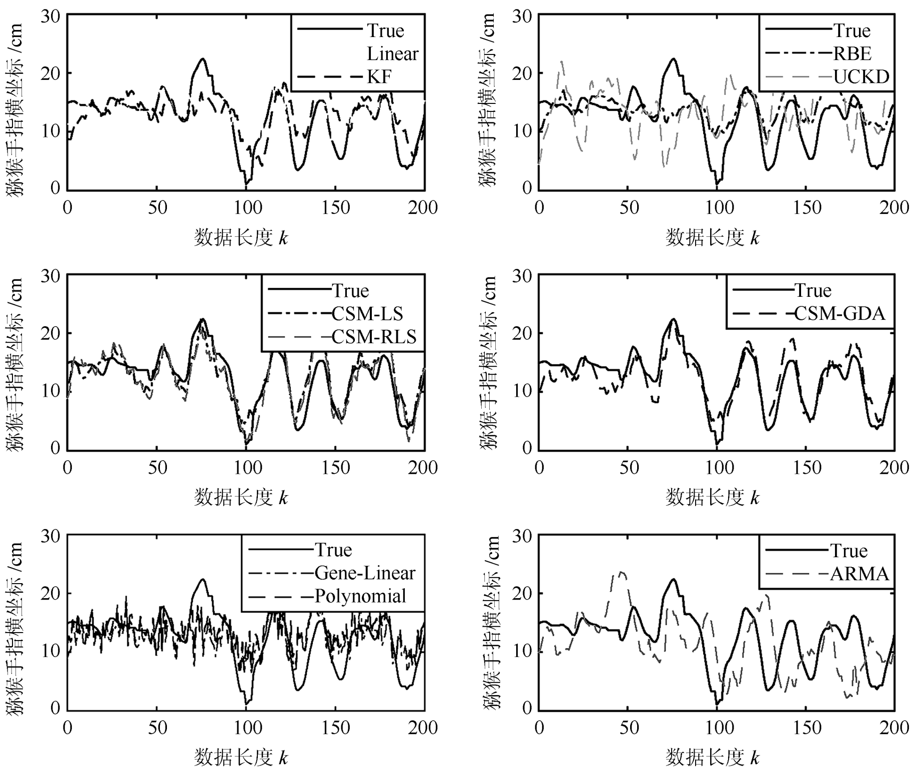


图 4 实验 1 中位置估计与手指移动真实位置曲线

Fig. 4 Position estimation and finger movement real position curve in experiment 1

练结果看, 卷积核的大小随 P 的增大而逐步减小, 这说明与当前时刻越接近的时刻相关性越强. 另外, 随着 P 的增大, 卷积核权重下降逐渐趋于平缓, 特别当 $P > 10$ 时, 权重大小已降至 0.05 以下. 该结果说明, P 的取值并不是越大越好, 因为随着 P 的增大, 其权重对模型时间相关性的贡献已变得非常小.

4　神经解码

完成对 CSM 模型的训练以后, 可将拟解码的神经元峰电位数信号代入到 CSM 模型中, 继而解码出手指移动位置, 如下给出了猕猴手指移动位置神经解码的完整算法.

算法 1. **估计 $\boldsymbol{y}_k$ 的 LS 算法**

输入. 训练数据的峰电位信号 $\boldsymbol{s}_k$, X 或 Y 坐标值 y_k, 拟解码数据的峰电位信号 $\boldsymbol{s}'_k$.

输出. 解码的 X 或 Y 坐标值 y'_k.

1) 预处理: 把神经元峰电位信号 $\boldsymbol{s}_k$ 变为式 (8) 中 $\bar{\boldsymbol{S}}$ 矩阵.

2) 训练: 由式 (8) 求得权值向量 $\hat{\boldsymbol{W}}$.

3) 解码: 将 $\boldsymbol{s}'_k$ 和 $\hat{\boldsymbol{W}}$ 代入式 (5) ~ (7), 求出 y'_k.

算法 2. 批处理 RLS 算法

输入. 训练数据的峰电位信号 $\boldsymbol{s}_k$, X 或 Y 坐标值 y_k, 拟解码数据的峰电位信号 $\boldsymbol{s}'_k$, 数据长度 K 和循环次数 T.

输出. 解码的 X 或 Y 坐标值 y'_k.

初始化. $k = 0$, $t = 0, \mu = 2 \times 10^{-6}$; $\lambda = 0.9999$, $\hat{\boldsymbol{W}}_0 = 0$, $\boldsymbol{P}_0 = \delta^{-1} I$, 其中 $\delta = 1$.

1) 预处理: 把神经元峰电位信号 $\boldsymbol{s}_k$ 变为式 (7) 中 $\boldsymbol{S}_k$ 向量.

2) 训练: 由式 (9) ~ (12) 计算 $\tilde{\boldsymbol{W}}_k$.

3) 训练: 令 $k = k+1$, 重复步骤 2) ~ 3), 直到 $k = K - 1$.

4) 训练: 令 $\tilde{\boldsymbol{W}}_0 = \tilde{\boldsymbol{W}}_k$, $k = 0$, $t = t + 1$, 重复步骤 2) ~ 4), 直到 $t = T - 1$.

5) 解码: 将 $\boldsymbol{s}'_k$ 和 $\hat{\boldsymbol{W}}_k$ 代入式 (5) ~ (7), 求出 y'_k.

算法 3. **批处理梯度下降算法**

1) 预处理: 把神经元峰电位信号 $\boldsymbol{s}_k$ 变为式 (7) 中 $\boldsymbol{S}_k$ 向量.

2) 训练: 由式 (13) 计算 $\tilde{\boldsymbol{W}}_k$.

3) 训练: 令 $k = k + 1$, 重复步骤 2) ~ 3), 直到 $k = K - 1$.

4) 训练: 令 $\tilde{\boldsymbol{W}}_0 = \tilde{\boldsymbol{W}}_k$, $k = 0$, $t = t + 1$, 重复步骤 2) ~ 4), 直到 $t = T - 1$.

5) 解码: 将 $\boldsymbol{s}'_k$ 和 $\hat{\boldsymbol{W}}_k$. 代入式 (5) ~ (7), 求出 y'_k.

5　实验结果与分析

5.1　数据来源

数据由 Hatsopoulos 实验室[8] 提供, 下载地址为 http://booksite.elsevier.com/0780123838360, 数据采集的具体过程可见第三部分编解码问题描述, 其中相关参数值如下.

1) 猕猴手指活动范围 $L = 25\,\text{cm}$, $B = 18\,\text{cm}$.

2) 猕猴脑部采集电极数 $N_e = 42$.

3) 采样周期 $\Delta t = 70\,\text{ms}$.

4) 数据长度 $K = 3\,101$.

所下载的数据共有两组, 具体描述如下.

5) 数据 1: 由二个数据矩阵组成, 一个是 $K \times N_e$ 的神经元信号特征矩阵; 另一个是 $K \times 2$ 的坐标位置标签矩阵, 其中标签矩阵的第一列为 X 轴, 第二列为 Y 轴, 活动范围为长 L, 宽 B.

6) 数据 2: 格式与数据 1 相同, 同样由二个数据矩阵组成.

其中, 数据 1 和数据 2 均在同等条件下采集. 数据 1 和数据 2 的区别是, 数据 1 中有方向随机的连续性手指移动, 数据 2 中存在有水平或垂直方向维持一段时间的移动.

5.2　实验及参数设置

本实验将先后采用 5 组实验来评判神经解码的性能, 对数据 1 和数据 2 的实验数据经处理分成 5 组实验如下.

实验 1. 对数据 1 采用验证方法为 Holdout 验证[33], 70 % 的数据用于训练, 30 % 的数据用于测试;

实验 2.对数据 1 采用验证方法为 M 折交叉验证[33], 其中 $M = 10$;

实验 3. 对数据 2 采用验证方法为 M 折交叉验证[33], 其中 $M = 10$;

实验 4. 数据 1 作为训练数据, 数据 2 作为测试数据;

实验 5. 数据 2 作为训练数据, 数据 1 作为测试数据.

实验 2 和 3 的 M 折交叉验证中, 误差评判标准采用均方根误差 e_c, 定义如下

$$e_c = \frac{1}{M} \sum_{m=1}^{M} \sqrt{\frac{1}{K'} \sum_{k=0}^{K'-1} (y'_{m,k} - y_{m,k})^2}$$

其中 $y'_{m,k}$ 和 $y_{m,k}$ 分别是第 m 次交叉验证得到的解码值和真实值, K' 为交叉验证的数据长度.

实验 4 和 5 的误差评判标准采用 e_r, 定义如下

$$e_r = \sqrt{\frac{1}{K} \sum_{k=0}^{K-1} (y'_k - y_k)^2}$$

其中 y'_k 和 y_k 分别是测试数据中解码值和真实值,

K 为测试数据的数据长度. 分别采用均方根误差 e_c 和 e_r 来评判解码方法在实验 2 至 5 的泛化能力.

在 5 组实验中, 分别对 Linear[8], KF[22, 29], RBE[25], UCKD[23] 和 CSM 模型方法计算出均方根误差 e_c 和 e_r, 相关参数如下设置.

Linear: 首先用最小二乘法训练式 (3) 的模型, 然后利用测试数据解码移动轨迹位置.

KF: 首先用最小二乘法训练式 (2) 的模型, 然后利用卡尔曼滤波算法解码移动轨迹位置, 相关参数设置, 解码初值 $y'_0 = 10$, 协方差 $P_{y,0|0} = 10$, 状态噪声方差 $\sigma^2 = 0.8$, 观测噪声方差 $\boldsymbol{R}_v = (\bar{\boldsymbol{s}} - \hat{\boldsymbol{s}})^{\rm T}(\bar{\boldsymbol{s}} - \hat{\boldsymbol{s}})/(K-1)$, 其中 $\bar{\boldsymbol{s}} = [\boldsymbol{s}_0, \boldsymbol{s}_1, \cdots, \boldsymbol{s}_{K-1}]^{\rm T}$, $\hat{\boldsymbol{s}} = \boldsymbol{Y}^{\dagger}\hat{\boldsymbol{W}}$.

RBE: X 轴的估计区间为 0.5 : 1 : 24.5 (起始值是 0.5, 间隔是 1, 终点值是 24.5), Y 轴的估计区间为 0.5 : 1 : 14.5.

UCKD: 初始解码值 $y'_0 = 0$, 权值 $\hat{\boldsymbol{w}}_0$ 是均值为零的均匀分布随机信号, 状态噪声方差 $\sigma^2 = 0.8$, 协方差 $P_{y,0|0} = 1$, $\boldsymbol{P}_{\boldsymbol{w},0|0}$ 是 0.05 与单位矩阵的乘积, 权值噪声 $\boldsymbol{R}_{\boldsymbol{w}}$ 是 0.01 与单位矩阵的乘积, 遗忘因子 $\lambda = 0.005$, 最后, 观测噪声方差 $\boldsymbol{R}_v$ 同 KF 算法设置.

CSM: 除以上相关参数以外, 还有, 遗忘因子 $\lambda = 0.9999$, 延迟的时间相关性数据长度 $P = 10$, RLS 的迭代循环次数 T 是 3, GDA 迭代步长 μ 是 2×10^{-6}, GDA 迭代循环次数 T 是 60.

5.3 解码结果分析

图 4 给出了实验 1 中各算法对 X 轴轨迹解码的曲线图, 并从所有测试数据中取 200 个样本点. 从图中得出, 对手指移动位置的轨迹基本都能跟踪上, 当然, 每种算法所表现的性能并不完全相同. Linear 算法解码的轨迹有较多的抖动和毛刺, 如第 80 ～ 100 点附近波动较频繁, 主要原因是 Linear 把手指移动位置看成了一个独立过程, 没有时间前后的相关性. UCKD 算法偏离真实轨迹较大, 如 120 ～ 130 和 150 ～ 160 点出现与真实轨迹反向的情况, 导致误差较大, 是因为该算法采用的是无监督解码, 再者, 该算法解码 X 轴时, 轨迹的正反向有时未能辨识, 但是, 对 Y 轴有较好的辨识度[23]. 采用 SSM 模型的算法是 KF 和 RBE, 解码曲线较为平滑, 抖动较小, 原因是 SSM 模型把当前时刻的状态与前一时刻有了相关性, 但有时解码曲线未能完全反映实际曲线的变化趋势, 如出现急速转向的 40 ～ 60 点附近. 本文采用 CSM 模型的 LS、RLS 和 GDA 算法, 对趋势反转处仍能较好跟踪上真实轨迹, 如 20 ～ 40, 40 ～ 60 和 80 ～ 100 等, 该结果表明当前时刻的状态的确与前若干个时刻的状态有相关性. 另外, 图 4 还分别给出了传统 TILM 模型在 X 轴解码的曲线图, 从图中可以看出, 广义线性 (Generalized linear model, GLM)、二次多项式 (Polynomial)、自回归滑动平均 (Auto-regressive moving average model, ARMA) 三种模型在细节部分总是存在毛刺或抖动情况, 导致总的估计性能较差.

以上是定性分析, 下面给出定量分析. 表 2 给出各算法对实验 2 至实验 5 中 X 轴的解码误差, 其中实验 2 和 3 的误差计算是 e_c 值, 实验 4 和 5 的误差计算是 e_r. 4 组实验的误差均值由高到低分别为 UCKD、Linear、RBE、KF、GDA、LS 和 RLS, 其中 RLS 算法相比误差最小的传统 KF 算法, 减小约 11.1%. 另外, 分析每一组实验的均方误差, 采用 CSM 模型的算法都会小于或接近于传统算法的误差, 其中在实验 4 中 GDA 的误差为 4.58 cm, 高于传统算法, 说明 GDA 处理这组数据的泛化能力较弱.

另外, 表 2 还给出 Y 轴中各算法的解码误差, 4 组实验结果的误差均值从高到低依次是 UCKD、Linear、RBE、LS、RLS、KF 和 GDA, 此次 KF 算法误差值都略微低于 LS、RLS 算法约 0.02 cm, 但误差值最小的仍然是 CSM 模型的算法. 并且, 对于每组实验, 采用 CSM 模型的算法的均方误差同样小于或接近于传统算法. 只是, GDA 算法相比误差最小的传统 KF 算法, 均方误差仅减小约 5.3%, 不如 X 轴的性能提升. 需注意的是, Y 轴的

表 2 X 轴和 Y 轴 (括号内) 的估计误差 (cm) (保留三位)

Table 2 X-axis and Y-axis (in parentheses) estimated error (cm) (three places reserved)

算法 $X(Y)$	实验 2	实验 3	实验 4	实验 5	平均
Linear	3.813 (2.003)	3.941 (2.513)	4.135 (2.991)	3.919 (2.120)	3.952 (2.407)
KF	3.060 (1.498)	3.908 (2.042)	4.540 (2.939)	3.637 (1.656)	3.786 (2.034)
RBE	3.637 (1.727)	4.699 (2.295)	3.907 (2.400)	3.913 (1.936)	4.015 (2.089)
UCKD	6.596 (4.235)	6.949 (5.504)	7.500 (5.775)	6.220 (4.868)	6.816 (5.096)
CSM-LS	2.959 (1.411)	3.450 (2.154)	3.937 (2.921)	3.109 (1.724)	3.364 (2.052)
CSM-RLS	2.964 (1.411)	3.443 (2.153)	3.957 (2.891)	3.106 (1.726)	3.368 (2.045)
CSM-GDA	2.896 (1.500)	3.270 (2.095)	4.581 (2.406)	3.233 (1.709)	3.495 (1.927)

表 3　二维平面的估计误差 (cm) (保留三位)
Table 3　The estimated error of the two-dimensional plane (cm) (three places reserved)

算法 XY	实验 2	实验 3	实验 4	实验 5	平均
Linear	4.307	4.674	5.103	4.455	4.635
KF	3.407	4.409	5.408	3.996	4.305
RBE	4.026	5.230	4.585	4.366	4.552
UCKD	7.839	8.865	9.466	7.896	8.787
CSM-LS	3.278	4.068	4.902	3.555	3.951
CSM-RLS	3.283	4.061	4.901	3.553	3.950
CSM-GDA	3.261	3.883	5.174	3.656	3.994

泛化能力强于 X 轴, 主要原因是 Y 轴的测试数据和训练数据集的相似度更高. 最后, 表 3 给出二维平面中各算法的解码误差, 与表 2 类似, 误差均值最小的仍是采用 CSM 模型的算法.

图 5 给出 CSM 模型中 P 值对解码性能的影响, 分别给出了实验 2 到 5 中 LS、RLS 和 GDA 的均方误差曲线变化图. 从图中看出, LS 和 RLS 算法的误差曲线几乎处于重合, 在约 $P = 20$ 以后, 误差曲线呈增长趋势. 再者, 在实验 2、实验 3 和实验 5 中, 采用 CSM 模型的 LS、RLS 和 GDA 在参数 $P = 10$ 以前的误差曲线均下降, 直到约 3 cm 左右, 然后趋于稳定, 最后在 $P = 20$ 时又呈增大趋势. 在实验 4 中, GDA 的误差曲线是先减小后增大, 再减小后趋于稳定, 而 LS 和 RLS 是先增大后稳定再增大趋势, 但在约 $P = 10$ 时它们都会处于较小误差. 原因是当 P 过小出现欠拟合, 当 P 过大出现过拟合, 此时有较差的泛化能力. 因此, 当 $P = 10$ 时采用 CSM 模型的算法解码手指移动位置能保证有较小误差.

图 6 给出在 CSM 模型中循环迭代次数 T 值的实验结果, 分别给出 RLS 和 GDA 算法的均方误差曲线图. 从图中看出, GDA 算法的误差曲线随着循环迭代次数的增加, 直到 $T = 10$ 左右, 呈快速下降, 随后下降趋势减缓, 最后在约 $T = 40$ 以后, 误差曲线就会趋于稳定状态. 另一方面, RLS 算法的误差曲线随着循环迭代次数的增加, 不能保证误差曲线下降. 从分析中得出, 采用 CSM 模型的 GDA, 循环迭代次数可最低设置 $T = 40$, 因为之后的误差曲线并不会再下降. 还有, 采用 CSM 模型的 RLS 算法, 由于误差曲线变化不明显, 故可不循环迭代或较少的循环迭代次数.

最后, 为了评估 CSM 模型中三种算法的计算复杂度, 表 4 给出了其在实验 4 中的训练时间, 其中 RLS 算法的循环迭代次数 $T = 3$, GDA 算法的 $T = 60$. PC 电脑操作系统为 Windows 7 旗舰版 64 位 SP1, 处理器为英特尔 Corei5-6400 2.70 GHz 四核, 处理软件为 MATLAB R2017b. 从表 4 可以看出, RLS 算法运行时间最长, GDA 算法次之, LS 算法运行时间最短, 该实验结果与第 3.2.2 节一致.

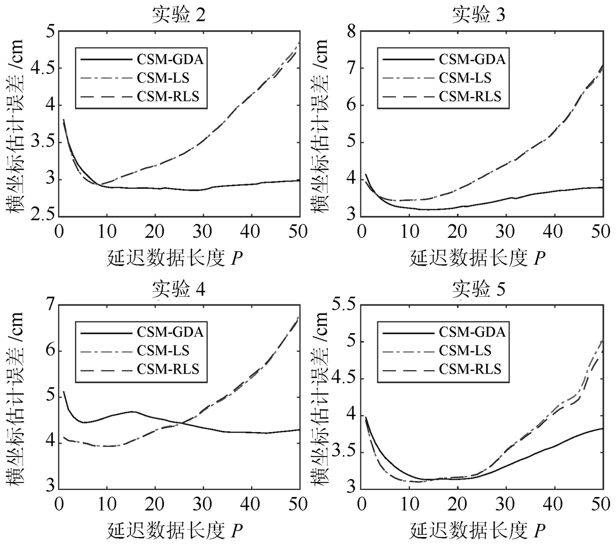


图 5　手指移动横坐标估计误差随延迟数据长度 P 的变化
Fig. 5　Finger movement abscissa estimation error with delay data length P changes

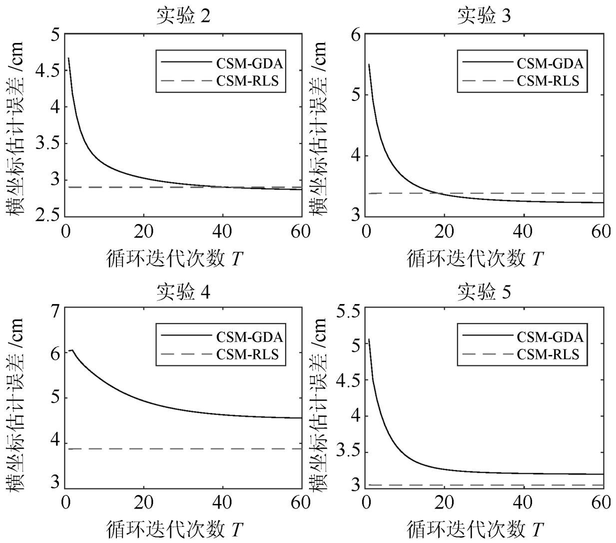


图 6　手指移动横坐标估计误差随迭代次数 T 的变化
Fig. 6　Finger movement abscissa estimation error with the number of iterations cycle T changes

表 4　CSM 模型算法的训练时间 (保留三位)
Table 4　The training time of CSM model algorithm (three places reserved)

CSM 模型算法	LS	RLS	GDA
训练时间 (s)	1.061	23.853	1.285

6 讨论

本文通过对手指移动位置解码模型的时间相关性研究, 采用了 CSM 模型来解码手指移动位置的方法, 并采用了三种算法来训练模型. 与传统模型的解码结果相比, 其解码结果的误差有所降低. 但是, 对于 CSM 模型和训练方法还有以下几点需要进一步进行讨论.

首先, 对时间相关性分析后, 引入了参数 P 使得需要训练的参数数量增多. 在传统 SSM 模型 (2) 中训练的权值 $\boldsymbol{a}$ 和偏执 $\boldsymbol{b}$ 的数量为 $2N_e$, 在 CSM 模型中该数量变为 PN_e+1. 由于增加了训练参数, 使得训练算法的复杂度增加. 但是, 解码中只是增加算法的乘法步骤, 故训练复杂度对解码影响不大.

再次, 参数 P 的确定会对解码性能产生重要影响. 若 P 选择过大会产生过拟合, 选择过小会欠拟合使误差增大. 为确定 P 的合适值, 可在解码之前, 再设置一个验证集, 以便确定最合适的 P 值.

另外, 从实验数据也可以看到, 本文的 CSM 模型虽然主要性能在 X 轴而非 Y 轴, 但是对二维平面解码的误差性能也有较大改善.

除此之外, 对于批处理的训练方法, RLS 和 GDA 算法需要一定的循环迭代次数. 从实验结果看, RLS 一次循环结果和多次循环结果性能接近, 因此循环次数可设置 1. 对于 GDA 算法, 循环迭代次数需 40 次以上才能达到收敛, 增加了计算复杂度. 然而, 由于前 10 次循环中 GDA 的误差就迅速下降, 虽然后续的循环仍有所下降但已趋向平缓, 因此, 为减少循环时间, 不一定要将循环次数设置至收敛时, $T=10$ 次循环是一个可行的选择. 而且, 参数 P 对 GDA 算法的计算复杂度影响为 $\mathrm{O}(TP^2)$, 相比于其余两种算法的 $\mathrm{O}(P^3)$ 和 $\mathrm{O}(TP^2)$, 增加 T 的值有时并不一定会增加过多的复杂度.

需要注意的是, 本文的 CSM 模型方法本质上仍然是一种有监督的解码方法, 因为算法的解码离不开训练数据集. 其实, 人类的学习过程就是一种无监督, 其可以不依赖于训练标签就可解码出手指移动轨迹, 更符合实际的学习方法. 虽然 UCKD 方法是一种无监督方法, 但令人遗憾的是, 它对于 X 轴的解码效果并不理想. 另外, 还可以考虑半监督或弱监督的解码方法, 即使训练集合的标签信息不完整也能解码出位置信息, 例如仅知道手指移动位置的大概信息而不是确切信息, 这些都可以是以后的一个研究方向.

最后, CSM 模型本身是一个卷积模型, 将近来较为流行的深度卷积神经网络 (Convolutional neural network, CNN) 来进行学习, 从初步试验结果看能够大体跟踪上运动趋势, 然而仍然有一些问题还需解决, 例如网络参数设置、初始化和激活函数等问题. 再者, 目前深度网络中会引入非线性激活函数, 同时时间不变线性模型与递归神经网络 (Recurrent neural network, RNN) 有相关性, 可以考虑把这些特性或方法与本文卷积空间模型进行结合. 需要注意的是, 文中卷积空间模型重点是对输入多维数据的处理, 故可以采用各种算法解决此模型中多维数据的问题, 而递归神经网络本身是可直接调用的算法层面. 神经元峰电位数信号中究竟有哪些特征能够真正影响手指移动位置解码, 这一点并没有一个公认的答案, 而深度卷积神经网络具备自动学习提取目标对象的特征值, 这提供了一个解决本问题可尝试的方向. 例如, 是否可以尝试把本文的解码回归问题转化为分类问题, 因为深度卷积网络更多的应用场合是分类, 这会是一个新的思路. 经过我们的初步实验, 神经网络方面算法处理本文的回归问题, 导致 M 折交叉验证中每次循环迭代产生的估计结果不稳定性很高. 故我们认为, 在后续的研究中, 如果此回归问题转化为分类问题得到解决, 可采用 CNN, RNN 等神经网络算法做更深入的分析.

7 结束语

本文中神经解码就是通过猕猴运动皮层的神经元峰电位信号预测其手指移动位置. 分析了传统解码模型的时间相关性问题, 采用一种具有卷积形式的线性时不变模型, 称为 CSM 模型. 然后, 采用 CSM 模型的 LS、RLS 和 GDA 算法去训练模型的权值参数, 最后去解码出手指移动轨迹. 本文采用的模型使猕猴当前时刻的状态与前若干个时刻的峰电位数有相关性, 将手指移动位置表示为神经元峰电位数的簇向量与一组常系数的卷积.

CSM 模型中, $P=1$ 时, 模型是时间独立的线性模型, $P>1$ 时, 是卷积空间模型. 另外, CSM 为二维卷积, 可用来处理输入多维神经元峰电位数信号. 再者, 该模型主要是针对神经元峰电位数信号的时间相关性, 并未考虑手指移动位置的时间相关性.

在实验中, 利用公开数据, 分别采用 Holdout 和交叉验证给出了 5 组实验结果, 来评判本文方法的解码性能. 从实验结果来看, 与传统方法相比, 本文方法在对 X 轴解码的误差均值上有了约 11.1% 的性能提升. 另外, 实验结果还证实, CSM 模型中当前时刻的状态与前 10 个时刻的状态具有较好的时间相关性, 解码误差较小. 最后, 从实验结果来看, 采用 CSM 模型的 RLS 和 GDA 算法都采用了循环迭代, 其 RLS 算法至少一次循环迭代就能达到收敛, GDA 算法的循环迭代次数较多.

附录A 高斯白噪声证明

由式 (3) 可知

$$\omega'_p = -\boldsymbol{w}_p^{\mathrm{T}} \boldsymbol{v}_{k-p} \tag{A1}$$

其中 $\boldsymbol{v}_k = [v_{0k}, v_{1k}, \cdots, v_{N_e-1,k}]^{\mathrm{T}}$ 是高斯白噪声矢量, 且其均值为 0, 方差矩阵 $\boldsymbol{R}_v$ 为对角线为 $[\sigma_{0k}^2, \sigma_{1k}^2, \cdots, \sigma_{N_e-1,k}^2]$ 的对角矩阵, 因此 ω'_p 是 v_{nk}, $n = 0, 1, \cdots, N_e - 1$ 的线性组合, 也是高斯白噪声. 另外, 由于式 (2) 中 ω_k 是一高斯白噪声, 式 (6) 的

$$\omega''_p = \omega'_p + \omega_{k-1} + \omega_{k-2} + \cdots + \omega_{k-p} \tag{A2}$$

也是白噪声. 那么, 式 (5) 中的

$$\Omega_k = \frac{1}{P}(\omega'_0 + \omega''_1 + \omega''_2 + \cdots + \omega''_{P-1}) \tag{A3}$$

为白噪声 ω'_0 和 ω''_p, $p = 0, 1, \cdots, P-1$ 的线性叠加, 因此 Ω_k 也是高斯白噪声.

References

1 Campbell J, Sharma A. Visual cross-modal re-organization in children with cochlear implants. *PLoS One*, 2016, **11**(1): e0147793

2 Wang W, Collinger J L, Degenhart A D, Tyler-Kabara E C, Schwartz A B, Moran D W, et al. An electrocorticographic brain interface in an individual with tetraplegia. *PLoS One*, 2013, **8**(2): e55344

3 Li Yuan-Qing. Application prospect of brain computer interface technology in the field of consciousness disorders. *Chinese Journal of Neurotraumatic Surgery (Electronic Edition)*, 2015, **1**(2): 60−61
(李远清. 脑机接口技术在意识障碍领域应用的前景展望. 中华神经创伤外科电子杂志, 2015, **1**(2): 60−61)

4 Wang Y W, Wang F, Xu K, Zhang Q S, Zhang S M, Zheng X X. Neural control of a tracking task via attention-gated reinforcement learning for brain-machine interfaces. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2015, **23**(5): 458−467

5 Zheng Xiao-Xiang, Wang Yi-Wen, Zhang Shao-Min, Zhang Qiao-Sheng. Decoding grasp movement from monkey premotor cortex for real-time prothetic hand control. *Science and Technology Innovation Herald*, 2016, **13**(12): 167−168
(郑筱祥, 王怡雯, 张韶岷, 张巧生. 猴子 PMd 区脑电解码抓握手势及机械手实时控制. 科技创新导报, 2016, **13**(12): 167−168)

6 Bouton C E, Shaikhouni A, Annetta N V, Bockbrader M A, Friedenberg D A, Nielson D M, et al. Restoring cortical control of functional movement in a human with quadriplegia. *Nature*, 2016, **533**(7602): 247−250

7 Hou Zeng-Guang, Zhao Xin-Gang, Cheng Long, Wang Qi-Ning, Wang Wei-Qun. Recent advances in rehabilitation robots and intelligent assistance systems. *Acta Automatica Sinica*, 2016, **42**(12): 1765−1779
(侯增广, 赵新刚, 程龙, 王启宁, 王卫群. 康复机器人与智能辅助系统的研究进展. 自动化学报, 2016, **42**(12): 1765−1779)

8 Wallisch P, Lusignan M E, Benayoun M D, Baker T L, Dickey A S, Hatsopoulos N G. *MATLAB for Neuroscientists* (Second edition). London: Elsevier, 2014.

9 Kass R E, Eden U T, Brown E N. *Analysis of Neural Data*. New York: Springer, 2014.

10 Hatsopoulos N G, Ojakangas C L, Paninski L, Donoghue J P. Information about movement direction obtained from synchronous activity of motor cortical neurons. *Proceedings of the National Academy of Sciences of the United States of America*, 1998, **95**(26): 15706−15711

11 Hatsopoulos N G, Xu Q Q, Amit Y. Encoding of movement fragments in the motor cortex. *The Journal of Neuroscience*, 2007, **27**(19): 5105−5114

12 Moran D W, Schwartz A B. Motor cortical representation of speed and direction during reaching. *Journal of Neurophysiology*, 1999, **82**(5): 2676−2692

13 Reynaud-Bouret P, Rivoirard V, Grammont F, Tuleau-Malot C. Goodness-of-fit tests and nonparametric adaptive estimation for spike train analysis. *Journal of Mathematical Neuroscience*, 2014, **4**: 3

14 Georgopoulos A P, Kettner R E, Schwartz A B. Primate motor cortex and free arm movements to visual targets in three-dimensional space. II. Coding of the direction of movement by a neuronal population. *Journal of Neurophysiology*, 1988, **8**(8): 2928−2937

15 Georgopoulos A P, Lurito J T, Petrides M, Schwartz A B, Massey J T. Mental rotation of the neuronal population vector. *Science*, 1989, **243**(4888): 234−236

16 Vargas-Irwin C E, Shakhnarovich G, Yadollahpour P, Mislow J M K, Black M J, Donoghue J P. Decoding complete reach and grasp actions from local primary motor cortex population. *Journal of Neuroscience*, 2010, **30**(29): 9659−9669

17 O'Doherty J E, Lebedev M A, Ifft P J, Zhuang K Z, Shokur S, Bleuler H, et al. Active tactile exploration enabled by a brain-machine-brain interface. *Nature*, 2011, **479**(7372): 228−231

18 Serruya M D, Hatsopoulos N G, Paninski L, Fellows M R, Donoghue J P. Instant neural control of a movement signal. *Nature*, 2002, **416**(6877): 141−142

19 Warland D K, Reinagel P, Meister M. Decoding visual information from a population of retinal ganglion cells. *Journal of Neurophysiology*, 1997, **78**(5): 2336−2350

20 Velliste M, Perel S, Spalding M C, Whitford A S, Schwartz A B. Cortical control of a prosthetic arm for self-feeding. *Nature*, 2008, **453**(7198): 1098−1101

21 Shanechi M M, Wornell G W, Williams Z M, Brown E N. Feedback-controlled parallel point process filter for estimation of goal-directed movements from neural signals. *IEEE Transactions on Neural Systems and rehabilitation Engineering*, 2013, **21**(1): 129−140

22 Chang Y H, Chen M, Shanechi M, Carmena J M, Tomlin C. A design of neural decoder by reducing discrepancy between manual control (MC) and brain control (BC). In: Proceedings of the 2014 European Control Conference. Strasbourg, France: IEEE, 2014. 516−521

23 Xue Ming-Long, Wu Hai-Feng, Zeng Yu. Unsupervised CKF decoding for macaque motor cortical spikes. *Acta Automatica Sinica*, 2017, **43**(2): 302−312
(薛明龙, 吴海锋, 曾玉. 无监督的猕猴运动皮层锋电位信号 CKF 解码. 自动化学报, 2017, **43**(2): 302−312)

24 Xue M L, Wu H F, Zeng Y, Yang K. Estimate a macaque′s finger trajectory using unsupervised cubature Kalman filtering decoding. In: Proceedings of the 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC). Xi′an, China: IEEE, 2016. 605−609

25 Hotson G, Smith R J, Rouse A G, Schieber M H, Thakor N V, Wester B A. High precision neural decoding of complex movement trajectories using recursive Bayesian estimation with dynamic movement primitives. *IEEE Robotics and Automation Letters*, 2016, **1**(2): 676−683

26 Li Hong-Bao. PMD representation and decoding of monkey reach plan and execution during obstacle avoidance task [Ph. D. dissertation], Zhejiang University, China, 2017
(李宏宝. 猕猴手臂避障规划与执行过程中背侧运动前区皮层的表征与解码 [博士学位论文], 浙江大学, 中国, 2017)

27 Zhang Yi, Yin Chun-Lin, Cai Jun, Luo Jiu-Fei. Bagging RCSP algorithm for extracting EEG feature. *Acta Automatica Sinica*, 2017, **43**(11): 2044−2050
(张毅, 尹春林, 蔡军, 罗久飞. Bagging RCSP 脑电特征提取算法. 自动化学报, 2017, **43**(11): 2044−2050)

28 Brockwell A E, Rojas A L, Kass R E. Recursive Bayesian decoding of motor cortical signals by particle filtering. *Journal of Neurophysiology*, 2004, **91**(4): 1899−1907

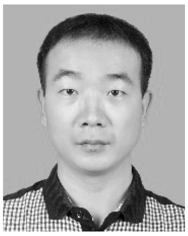
29 Wu W, Shaikhouni A, Donoghue J P, Black M J. Closed-loop neural control of cursor motion using a Kalman filter. In: Proceedings of the 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). San Francisco, CA, USA: IEEE, 2004. 4126−4129

30 Wu W, Gao Y, Bienenstock E, Donoghue J P, Black M J. Bayesian population decoding of motor cortical activity using a Kalman filter. *Neural Computation*, 2006, **18**(1): 80−118

31 Haykin S. *Neural Networks and Learning Machines* (Third edition). Upper Saddle River, NJ: Prentice Hall, 2008.

32 Zhang Xian-Da. *Modern Signal Processing* (Third edition). Beijing: Tsinghua University Press, 2015.
(张贤达. 现代信号处理. 第 3 版. 北京: 清华大学出版社, 2015.)

33 Raschka S, Mirjalili V. *Python Machine Learning* (Second edition). Birmingham, UK: Packt Publishing, 2017.



冯景义 云南民族大学电气信息工程学院硕士研究生. 主要研究方向为神经网络和机器学习.
E-mail: fengjingyione@foxmail.com
(**FENG Jing-Yi** Master student at the School of Electrical and Information Engineering, Yunnan Minzu University. His research interest covers neural network and machine learning.)

吴海锋 云南民族大学教授. 主要研究方向为机器学习, 移动通信和神经信号处理. 本文通信作者.
E-mail: whf5469@gmail.com
(**WU Hai-Feng** Porfessor at Yunnan Minzu University. His research interest covers machine learning, mobile communications and neural signal processing. Corresponding author of this paper.)

曾 玉 云南民族大学助理教授. 主要研究方向为无线网络控制和移动通信.
E-mail: yv.zeng@gmail.com
(**ZENG Yu** Assistant professor at Yunnan Minzu University. Her research interest covers wireless network control, mobile communications.)