c-VEPのニューラルデコーディング

鷲沢研究室 佐藤純一

April 18, 2016

c-VEP BCI

- PN 系列（Pseudo Noise; 擬似雑音系列）を利用
- t-VEP や f-VEP に比べて高い情報転送量（ITR)

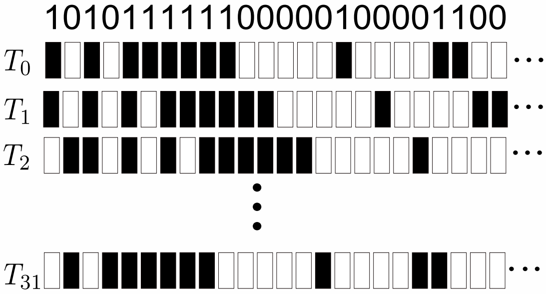

図 **1:** PN 系列と視覚刺激の対応

c-VEP BCI

- 特徴抽出：空間フィルタ
- 識別：テンプレートマッチング

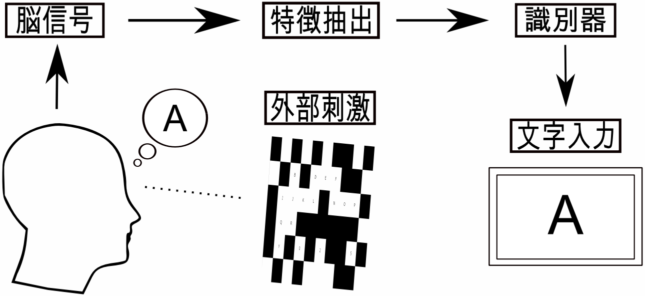

図 **2:** c-VEP BCI の流れ

空間フィルタ

空間フィルタ

- $x_i[t]$：i チャンネルの c-VEP $(i = 1, \dots, I)$

$$y[t] = \sum_{i=1}^{I} w_i x_i[t]$$

従来手法 **CCA** による空間フィルタ

- $\boldsymbol{X}_k$：k 番目の訓練行列
  - (i, n) 要素：$x_i[n]$
- $\boldsymbol{R} = \frac{1}{K} \sum_{k=1}^{K} \boldsymbol{X}_k$
- CCA 空間フィルタは以下の最適化問題における $\boldsymbol{w}_x$

$$\max_{\boldsymbol{w}_x, \boldsymbol{w}_r} \frac{\sum_{k=1}^{K} \boldsymbol{w}_x^\top \boldsymbol{X}_k \boldsymbol{R}^\top \boldsymbol{w}_r}{\sqrt{\sum_{k=1}^{K} \boldsymbol{w}_x^\top \boldsymbol{X}_k \boldsymbol{X}_k^\top \boldsymbol{w}_x \cdot \boldsymbol{w}_r^\top \boldsymbol{R} \boldsymbol{R}^\top \boldsymbol{w}_r}}.$$

CCA 空間フィルタの問題点

- CCA 空間フィルタの重みは訓練データ $\boldsymbol{X}_k$ と $\boldsymbol{R}$ のみで決定
- 提示された PN 系列の情報を用いていない
  - PN 系列の性質（低い自己相関）を持っているとは限らない
- c-VEP の性質を取り入れることで性能を向上させる

- 脳信号からオリジナルの刺激を取り出す
- 本研究では，c-VEP を PN 系列（提示刺激）に近づける
  - 目標変数（target）に PN 系列を用いた最適化

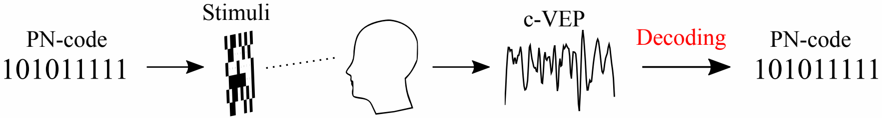

図 **3:** ニューラルデコーディング

時空間の逆フィルタ

- $w_{i,\tau}$: 時空間フィルタの重み
- τ_0 : 視覚刺激から c-VEP の誘発までの時間ラグ
- N_τ : フィルタのタップ長
- 逆フィルタの出力信号 $y[n]$ は以下で表される

$$y[n] = \sum_{i=1}^{I} \sum_{\tau=0}^{N_\tau - 1} w_{i,\tau} x_i[n + \tau + \tau_0]$$

- 逆フィルタを設計するため，$+\tau$ とする

線形逆フィルタ least mean square eror (LMSE)

- $s[n]$：PN 系列
  - 直流成分除去のため中心化（平均を 0 にする）
  - 脳信号とサンプリング周波数を合わせるためアップサンプル
  - $s[n] \in \{-0.5, +0.5\}$, $n = 0, \ldots, N-1$
- $y[n]$ と $s[n]$ の 2 乗誤差を最小化

$$\min_{\boldsymbol{w}} \sum_{n=0}^{N-1} (y[n] - s[n])^2.$$

正規方程式

- $\boldsymbol{X} \in \mathbb{R}^{N \times M}$: EEG, N はサンプル数，$M = N_\tau N_{\mathsf{ch}}$ は特徴量の次元数
- $y^{(n)} = \sum_{m=0}^{M-1} w_m x_m^{(n)}$ とおくと，目的関数は以下で表される

$$J(w) = \frac{1}{N} \sum_{n=0}^{N-1} \left(\sum_{m=0}^{M-1} w_m x_m^{(n)} - s^{(n)} \right)^2$$

- $k = 0, \ldots, M-1$ のとき，k で偏微分して最小化する

$$\frac{\partial J(w)}{\partial w_k} = \frac{2}{N} \sum_{n=0}^{N-1} (\sum_{m=0}^{M-1} w_m x_m^{(n)} - s^{(n)}) x_k^{(n)} = 0$$

$$\sum_{n=0}^{N-1} \sum_{m=0}^{M-1} x_k^{(n)} x_m^{(n)} w_m - \sum_{n=0}^{N-1} x_k^{(n)} s^{(n)} = 0$$

- したがって

$$\boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{w} - \boldsymbol{X}^\top \boldsymbol{s} = 0$$

$$\boldsymbol{w} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{s}$$

線形逆フィルタ lasso

- N_τ を大きくすると w の要素数が増加し過学習しやすい
- l_1 ノルム正則化で疎な解を得ることができる

$$\min_{\boldsymbol{w}} \sum_{n=0}^{N-1} (y[n] - s[n])^2 + \alpha \sum_{i=1}^{I} \sum_{\tau=0}^{N_\tau - 1} |w_{i,\tau}|,$$

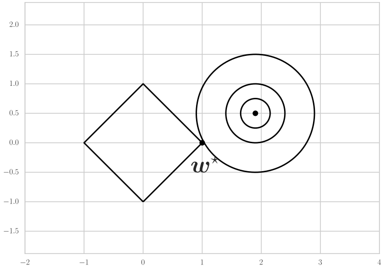

図 **4:** 誤差関数の等高線と lasso の制約条件

非線形逆フィルタ

- c-VEP は PN 系列の非線形変換
  - PN 系列による刺激が脳内を通過して発生
- 線形モデルでは表現に限界がある

→ 非線形モデルによりニューラルデコーディングを行う

線形モデル

- 線形回帰（LMSE，lasso）
- ロジスティック回帰

非線形モデル

- ニューラルネットワーク
- カーネル回帰
- Support Vector Machine (SVM)

ニューラルネットワーク（NN）

- 入力層，中間層，出力層をもつ
- 情報が入力から出力に伝搬する構造をもつ
- この研究では 4 層（2 つの中間層）をもつ NN を利用

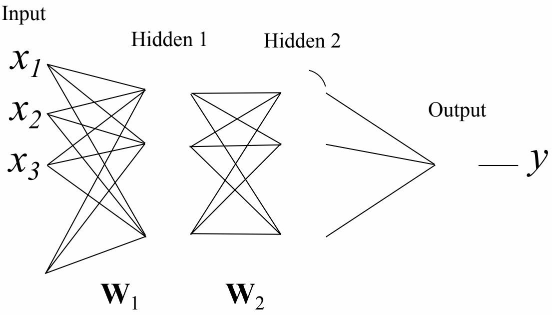

図 **5:** 4 層ニューラルネットワーク

NN による非線形逆フィルタ

- J_1，J_2，J_3：入力層，中間層 1，中間層 2 の次元
- $\boldsymbol{W}_1 \in \mathbb{R}^{J_2 \times J_1}$，$\boldsymbol{W}_2 \in \mathbb{R}^{J_3 \times J_2}$，$\boldsymbol{w}_3 \in \mathbb{R}^{J_3 \times 1}$：
  中間層 1，中間層 2 の重み行列，出力層の重みベクトル
- $\boldsymbol{b}_1 \in \mathbb{R}^{J_2}$，$\boldsymbol{b}_2 \in \mathbb{R}^{J_3}$，$b_3 \in \mathbb{R}$：各層のバイアス
- 中間層の活性化関数に以下の ReLU を用いる

$$f(u) = \max(u, 0)$$

- 出力層の活性化関数に $\tanh$ を用いる
  - 値域 $(-1, 1)$

$$g(u) = \tanh(u)$$

NN による非線形逆フィルタ

- 中間層と出力層の出力はそれぞれ以下の式で表される

$$\boldsymbol{z}_n^1 = f(\boldsymbol{W}_1 \boldsymbol{x}_n + \boldsymbol{b}_1)$$

$$\boldsymbol{z}_n^2 = f(\boldsymbol{W}_2 \boldsymbol{z}_n^1 + \boldsymbol{b}_2)$$

$$y_n = g(\boldsymbol{w}_3^\top \boldsymbol{z}_n^2 + b_3)$$

- $\boldsymbol{w}$ を $\boldsymbol{w}_1, \boldsymbol{W}_2, \boldsymbol{w}_3, \boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3$ の成分（M 個）をもつベクトルとする
- パラメータは 2 乗誤差最小化によって決定される

$$\min_{\boldsymbol{w}} E(\boldsymbol{w}) = \sum_{n=0}^{N-1} (s[n] - y_n)^2$$

勾配降下法

- $E(\boldsymbol{w})$ は凸関数ではなく，極小点は一般に多数存在する
- $E(\boldsymbol{w})$ の局所的な極小点 $\boldsymbol{w}$ を求める
- $E(\boldsymbol{w})$ の勾配は以下で定義される

$$\nabla E(\boldsymbol{w}) \equiv \frac{\partial E}{\partial \boldsymbol{w}} = \left[\frac{\partial E}{\partial w_1} \cdots \frac{\partial E}{\partial w_M}\right]^\top$$

- 現在の重みを $\boldsymbol{w}^{(t)}$，動かした後の重みを $\boldsymbol{w}^{(t+1)}$ とする
- $\boldsymbol{w}$ の初期値を決め，勾配降下法は以下のように重み $\boldsymbol{w}$ を更新する

$$\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} - \epsilon \nabla E$$

- ϵ は学習率（learning rate）であり，更新量の大きさを定める

確率的勾配降下法（SGD; stochastic gradient descent）

- 訓練サンプル $n = 1, \ldots, N$ のうち一部だけをつかってパラメータを更新する
- 1 つのミニバッチ（少数の訓練集合）を $\mathcal{D}_t$ とする
  - t 回目の更新毎にサンプル集合が変わる
- $\mathcal{D}_t$ の全サンプルに対する誤差を計算し，パラメータを更新する

$$E_t(\boldsymbol{w}) = \frac{1}{N_t} \sum_{n \in \mathcal{D}_t} E_n(\boldsymbol{w})$$

- ここで，$N_t = |\mathcal{D}_t|$ はミニバッチのサンプル数
- SGD を使うと局所的な極小解にトラップされるリスクが低減する
  - 目的関数 $E_t(\boldsymbol{w})$ が更新ごとに異なるため

誤差逆伝搬法 (1)

- ユニット i から j へ入力の重み付き線形和を u_j とおく

$$u_j = \sum_i w_{ji} z_i$$

- 活性化関数 $h(\cdot)$ を用いたとき，j での出力は

$$z_j = h(u_j)$$

- 誤差 $\delta_j \equiv \frac{\partial E}{\partial u_j}$ を導入する
- 偏微分の連鎖法則から以下の逆伝搬公式が得られる

$$\delta_j = h'(u_j) \sum_k w_{kj} \delta_k$$

誤差逆伝搬法 (2)

誤差逆伝搬法

1. 入力ベクトル $\boldsymbol{x}_n$ を $u_j = \sum_i w_{ji} z_i$, $z_j = h(u_j)$ で順伝搬させる
2. 出力ユニットの誤差 $\delta_k = \partial E_n / \partial u_k$ を求める
3. $\delta_j = h'(u_j) \sum_k w_{kj} \delta_k$ で δ を逆伝搬させる
4. $\frac{\partial E_n}{\partial w_{ji}} = \delta_j z_i$ で微分を計算する

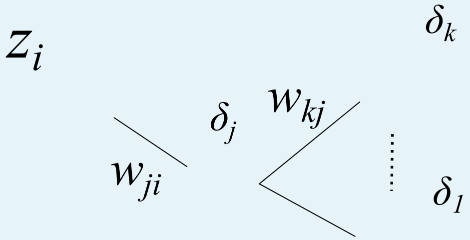

図 **6:** 逆伝搬

誤差逆伝搬法 (3)

- $f(u) = \max(0, u)$ は $u = 0$ で微分不可能であるが，$u = 0$ において $[0, 1]$ の任意の値を置くことで微分可能なものとする（劣微分）

$$f'(u) = \begin{cases} 1 \ (u \leq 0) \\ 0 \ (\text{otherwise}) \end{cases}$$

- $g(u) = \tanh(u)$ のとき

$$g'(u) = 1 - g(u)^2$$

Dropout

- 多層ニューラルネットのユニットを確率的に選別して学習する
- 中間層と入力層のユニットを確率 p でランダムに選出する
- ユニットの選出は重みを更新するたびに行う
- 学習終了後はすべてのユニットを使って順伝搬計算を行う
    - ドロップアウトの対象ユニットは重みを p 倍する
- 学習時にネットワークの自由度を下げることで，過学習を避ける
- ネットワークを独立に訓練し，推論時に結果を平均するのと同じ効果がある

実験

環境

- 20~22 歳までの健康な男性 5 人
- EEG は 16ch，600Hz で計測し，120Hz にダウンサンプリング
- PN 系列を 60Hz から 120Hz にアップサンプリング
- 180 サンプルでトレーニング，288 サンプルでテスト

手法

1. CCA 空間フィルタ（従来手法）
2. 線形時空間逆フィルタ LMSE,
3. 線形時空間逆フィルタ lasso,
4. ニューラルネットによる非線形逆フィルタ

評価基準

1. 入力の正確さ: classification accuracy
2. デコードの復元度：相関係数

デコード結果

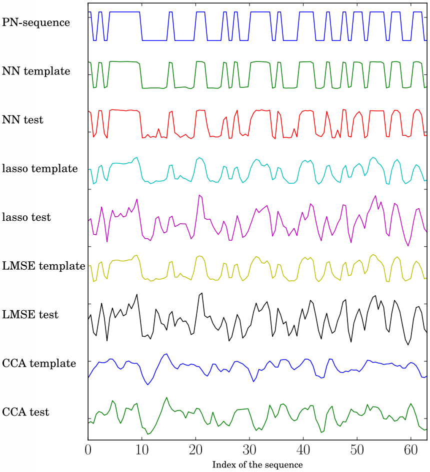

図 **7:** 被験者 1 のデコード波形

識別率

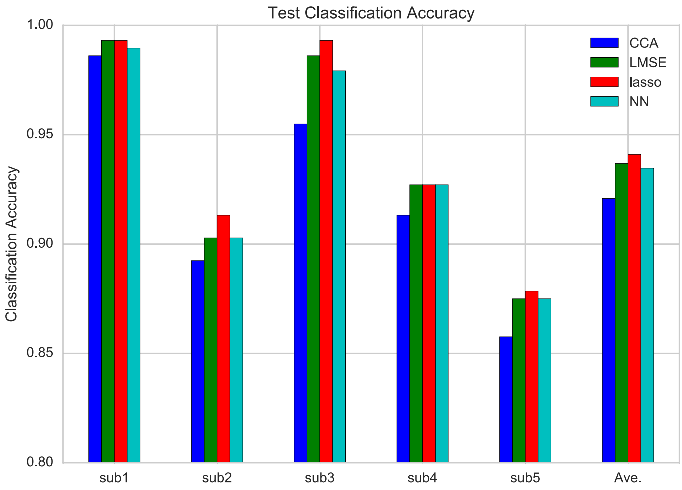

図 **8:** 被験者 1~5 の識別率

相関係数

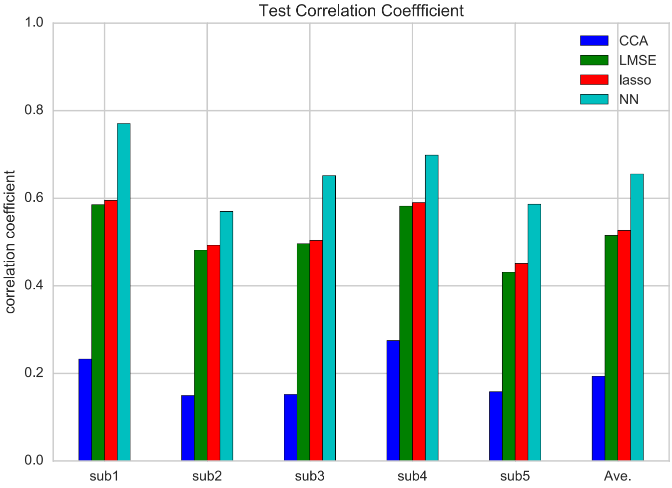

図 **9:** 被験者 1~5 の $s[n]$ と $y[n]$ の相関係数

考察

- デコード性能の改善しても必ずしも識別率は向上しない
  - NN が最も高い相関
  - lasso が最も高い識別率
- 今回は NN は相関を最大化したが，識別率最大化でパラメータを選べば改善すると考えられる

まとめと今後の課題

まとめ

- 線形，非線形の逆フィルタの提案
    - LMSE
    - lasso
    - Neural Network
- 提案法はいずれも高精度にデコードした
- CCA 空間フィルタよりも高い識別率，PN 系列との高い相関を示した

今後の課題

- PN 系列以外の系列を利用
- Reccurent Neural Network の適用
    - 再帰的構造をもつニューラルネットワーク