Nonlinear SVM for Multivariate Functional Data

Zheye Yuan advised by Prof. Bing Li

Penn State Statistics

① Background and Objectives

- We have **multiple** functions grouped as a vector, $(f^1, \cdots, f^d)^t$ for input
- The component functions f^i can have **different support** and in general be of differing function spaces
- The original data is **discrete**, but we view them as **functions** for some benefits
- Using this, we want to predict a **binary output**
- We do this by extending the Support Vector Machine (**SVM**)
- We also want to perform this classification **nonlinearly**
- We achieve this by using the Reproducing Kernel Hilbert Space (**RKHS**)
- There are multiple ways to do this with **varying complexities**

③ Examples of Nested Hilbert Spaces

- **The Nested Additive Hilbert Space**
  Construct $\mathcal{M}$ as
  $$\{\phi^1 + \cdots + \phi^d : \phi^i \in \mathcal{M}_i\}$$
  with inner product in $\mathcal{M}$ defined by
  $$\langle \phi, \varphi \rangle_{\mathcal{M}} = \langle \phi^1, \varphi^1 \rangle_{\mathcal{M}_1} + \cdots + \langle \phi^d, \varphi^d \rangle_{\mathcal{M}_d}$$
- **The Nested Multiplicative Hilbert Space**
  Construct $\mathcal{M}$ as
  $$\{\phi^1 \cdot \phi^2 \cdots \phi^{d-1} \cdot \phi^d : \phi^i \in \mathcal{M}_i\}$$
  with inner product in $\mathcal{M}$ defined by
  $$\langle \phi, \varphi \rangle_{\mathcal{M}} = \langle \phi^1, \varphi^1 \rangle_{\mathcal{M}_1} \cdot \langle \phi^2, \varphi^2 \rangle_{\mathcal{M}_2} \cdots \langle \phi^d, \varphi^d \rangle_{\mathcal{M}_d}$$
- **The General Nested Hilbert Space**
  Construct $\mathcal{M}$ as
  $$\{(\phi^1, \phi^2, \cdots, \phi^{d-1}, \phi^d)^t : \phi^i \in \mathcal{M}_i\}$$
  with pseudo inner product in $\mathcal{M}$ defined by
  $$\langle \phi, \varphi \rangle_{\mathcal{M}} = (\langle \phi^1, \varphi^1 \rangle_{\mathcal{M}_1}, \langle \phi^2, \varphi^2 \rangle_{\mathcal{M}_2}, \cdots, \langle \phi^d, \varphi^d \rangle_{\mathcal{M}_d})^t$$
- The last one is **not** really an **inner product** as it returns a vector
- But people have used them (Please ask **Prof. Reimherr**)

② The First and Second Level Hilbert Spaces

- $f^i \in \mathcal{H}_i$, $\mathcal{H} = \mathcal{H}_1 \times \cdots \times \mathcal{H}_d$, with inner product
  $$\langle f, g \rangle_{\mathcal{H}} = \langle f^1, g^1 \rangle_{\mathcal{H}_1} + \cdots + \langle f^d, g^d \rangle_{\mathcal{H}_d}$$
- We have a **kernel** in each component space
  $$\kappa_i(f^i, g^i) = \exp\left(-\gamma_i \|f^i - g^i\|^2_{\mathcal{H}_i}\right)$$
- Inner product in $\mathcal{M}_i$ is determined by
  $$\langle k_i(\cdot, u), k_i(\cdot, v) \rangle_{\mathcal{M}_i} = k_i(u, v)$$
- $\mathcal{M}_i = \operatorname{span}\{k_i(\cdot, u) : u \in \mathcal{H}_i\}$: **RKHS** generated by κ_i
- And
  $$\left\langle \sum_j a_j k_i(\cdot, f_j), \sum_l b_l k_i(\cdot, f_l) \right\rangle_{\mathcal{M}_i} = \sum_j \sum_l a_j b_l k_i(f_j, f_l)$$
- We call $\{\mathcal{H}_i, \mathcal{M}_i\}$ the **nested Hilbert space**

④ Support Vector Machine

- **Minimize**
  $$\langle \phi, \phi \rangle_{\mathcal{M}} + \lambda E[1 - Y(\langle \phi, k(\cdot, X) \rangle_{\mathcal{M}} - t)]^+$$
- Y is 1 or -1
- **The sample version written with slack variables**
  $$\begin{aligned} &\text{minimize } \frac{1}{2}\langle \phi, \phi \rangle_{\mathcal{M}} + \lambda \sum_{i=1}^N \xi_i \\ &\text{subject to } Y^i(\langle \phi, k(\cdot, X^i) \rangle_{\mathcal{M}} - t) \geq 1 - \xi_n \\ &\xi_i \geq 0, \ 1 \leq i \leq N \\ &\phi \in \mathcal{M}, \ t \in \mathbb{R}, \ \xi \in \mathbb{R}^N \end{aligned}$$
- Solve above by Lagrangian multipliers, **the Dual**
  $$\begin{aligned} &\text{maximize } \sum_{i=1}^N \alpha_i - \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j Y^i Y^j k(X^i, X^j) \\ &\text{subject to } \sum_{i=1}^N \alpha_i Y^i = 0 \\ &0 \leq \alpha_i \leq \lambda, \ 1 \leq i \leq N \end{aligned}$$

⑤ The Solution

- Solve this to get **classification rule**
  $$\operatorname{sign}\left[\sum_{i=1}^N \alpha_i Y^i k(\cdot, X^i) - t\right]$$
- **Cutoff** t is obtained by solving
  $$Y^j\left(\sum_{i=1}^N \alpha_i Y^i k(X^i, X^j) - t\right) = 1$$
  for an j for which $0 < \alpha_j < \lambda$

⑥ Super Additive Model

- Use **Karhunen Loève** Expansion to write each function as
  $$X_i = \sum_{j=1}^{\infty} \lambda_j e_j \ (\lambda_j \text{ is r.v.})$$
- **Kernel** between vector of functions X and Y
  $$k(X, Y) = \sum_{i=1}^d k_i(\lambda(X_i), \lambda(Y_i))$$
- $\lambda(X_i)$ is the vector of coefficients of the p functions with **largest varying** coefficients
- Super additive because we **further simplify** the additive model by only considering the p most **representative** functions

⑦ The Results

method	EEG	fMRI
add	92/122	107/145*
mult	91/122	98/145
sup-add	82/122	107/145*

Table 1: Performance of the three methods

- fMRI is a rather **imbalanced** data (more negatives)
- Asterisk on two results signify that the **prediction** tendency was the **same** regardless of the true label
- fMRI cannot be explained by an **additive** model while EEG probably can be
- The line in Figure 1 is the **mean** of the **best classification** lines (Not what CS people do)
- Figure 1 shows **good** ideal **separation** in the projections $\sum_{i=1}^N \alpha_i Y^i k(X^i, X^j)$ (EEG mult)
- We want the range of projections to be **large relative to** the overlap of the projections for the two classes
- **Negative** subjects are placed **left** of the line in Figure 2
- Figure 2 shows dominant **correct** prediction on **both sides** of the line
- Figure 3 exhibits a **valley** for the relation between λ and relative distance

⑧ Additional Remarks

- **Rossi** and **Villa** had single function input
- Some models did **not** exhibit a **valley** shape relative distance against λ
- By using **functions** you can **smooth** and retain **continuity**
- I will email **bibliography**: `zxy124@psu.edu`

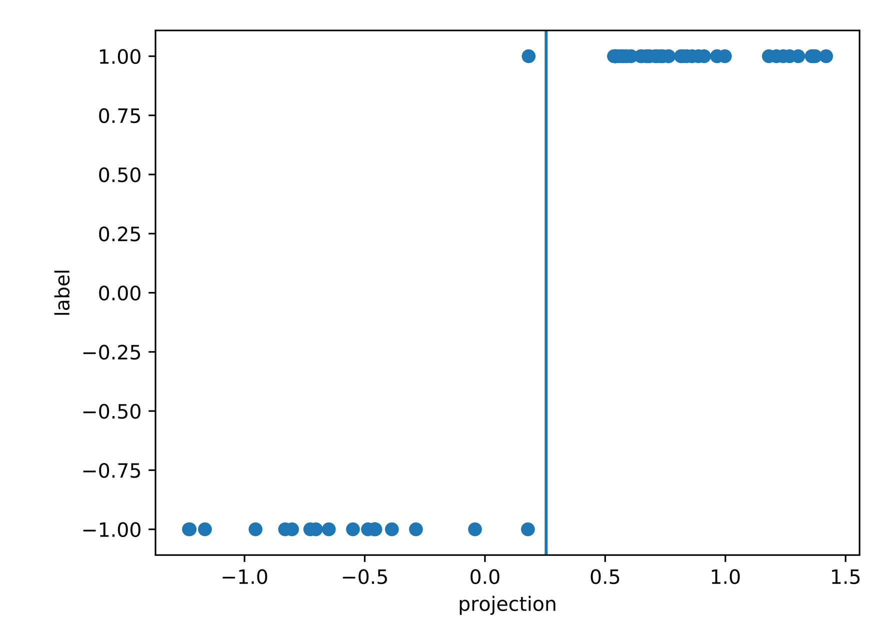

Figure 1: Labels against projected values for training data

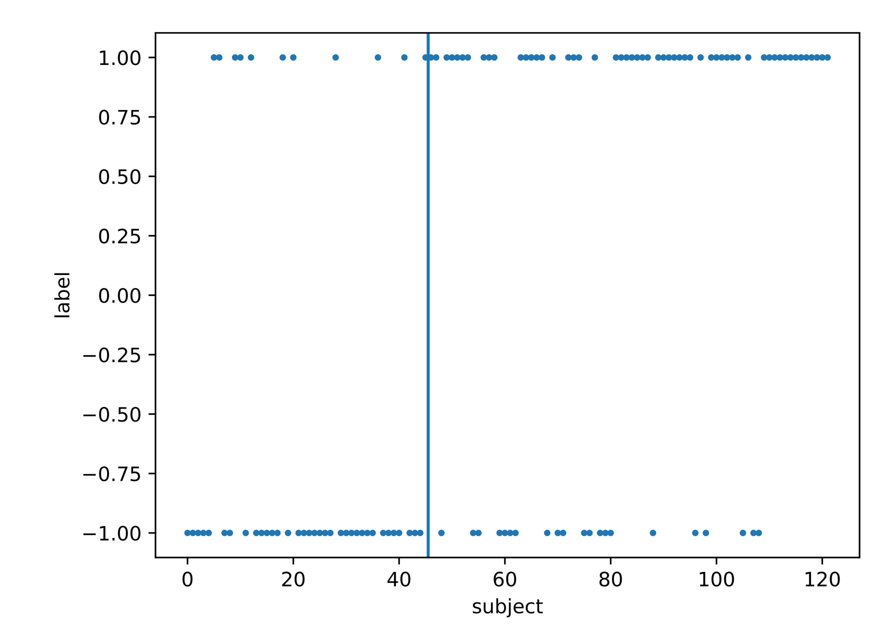

Figure 2: Predicted labels for each subject

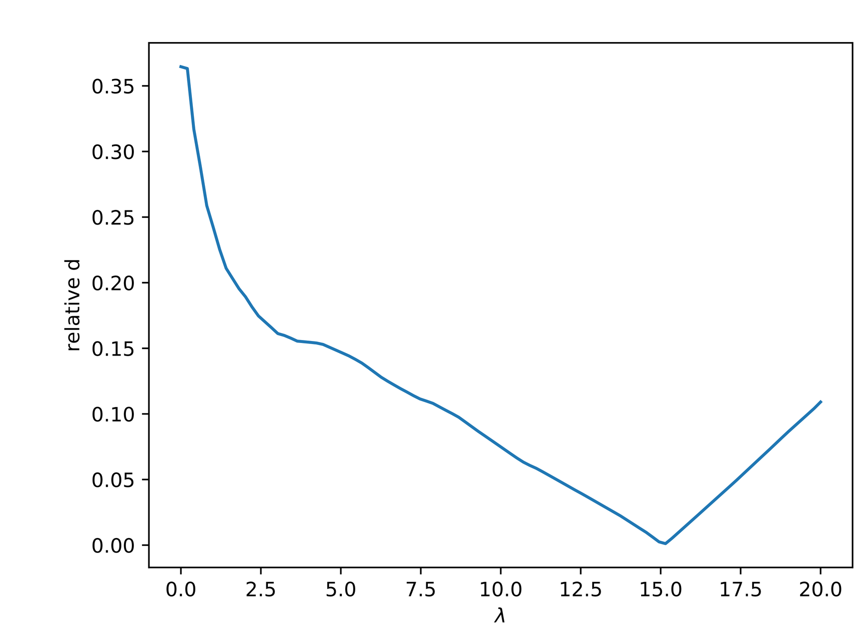

Figure 3: Relative distance against λ