Functional Mixtures-of-Experts

Faicel Chamroukhi

with V-H. Hoang, N-T. Phan, G-J. McLachlan

The University of Queensland, School of Mathematics and Physics
Brisbane, Australia, 20 August 2019
Research supported in part by an Ethel Raybould Visiting Fellowship

Faicel Chamroukhi

Functional Mixtures-of-Experts 2/51

Ethel H. Raybould, 1899-1987
Lecturer in Mathematics, 1928-1955

Miss Ethel Harriet Raybould’s association with UQ began when she enrolled at the University to study mathematics in 1919. She graduated in the Faculty of Arts with First Class Honours in Mathematics in March 1927 and was also awarded a Gold Medal for outstanding merit in her final honours examination.

She first began teaching at UQ in 1928 when she was seconded from the Department of Women’s Work at the Central Technical College to fill a temporary vacancy. She continued in that capacity until 1931 when she was appointed to the position of Lecturer in Mathematics permanently. The same year she was awarded a Master of Arts for her thesis on The Transfinite and its Significance in Analysis.

From 1937 to 1939 Miss Raybould took leave to study advanced mathematics at Columbia University in New York. In 1951 she was promoted to Senior Lecturer and in 1955 she retired.

Her association with UQ endures not only because she was its first female lecturer, she was also one of the University’s most generous benefactors, she left a bequest of more than $920,000 to the University when she died in 1987.

The Raybould Lecture Theatre in the Hawken Engineering Building was constructed with part of the Raybould bequest. The balance was used to establish the Raybould Tutorial Fellowship, the Raybould Visiting Fellowship and the Ethel Raybould Prize in Mathematics.

* picture taken by my phone from a poster at one of the UQ buildings.
Outline

1. Introduction
2. Functional Data Analysis Framework
3. Mixture-of-Experts Modeling
4. Functional Mixture-of-Experts (FunME)
5. Statistical Inference
Scientific context

- The data are assumed to represent samples from random variables with unknown probability distributions.
- The area of statistical learning and analysis of complex data.
- **Data**: Complex data $\rightarrow$ heterogeneous, temporal/dynamical, high-dimensional/functional, incomplete,...
- **Objective**: Transform the data into knowledge:
  $\rightarrow$ Reconstruct hidden structure/information, groups/hierarchy of groups, summarizing prototypes, underlying dynamical processes, etc.

Modeling framework

- **Latent variable models**: $f(x|\theta) = \int_z f(x, z|\theta) dz$
  Generative formulation:
  $z \sim q(z|\theta)$
  $x|z \sim f(x|z, \theta)$
  $\rightarrow$ Mixture models: $f(x|\theta) = \sum_{k=1}^{K} \mathbb{P}(z = k)f(x|z = k, \theta_k)$ and extensions.
Mixture models [McLachlan and Peel., 2000]

Mixture modeling framework

- Mixture density: \( f(x|\theta) = \sum_{k=1}^{K} \pi_k f_k(x|\theta_k) \)

- High power for density approximation: [Nguyen et al., 2019] ▶ get pdf here

- Generative model

\[
\begin{align*}
z & \sim \mathcal{M}(1; \pi_1, \ldots, \pi_K) \\
x|z & \sim f(x|\theta_z)
\end{align*}
\]

← learn \( \theta \) from the data
Mixtures and the EM algorithm

Finite Mixture Models [McLachlan and Peel., 2000]

\[ f(x; \theta) = \sum_{k=1}^{K} \pi_k f_k(x; \theta_k) \quad \text{with} \quad \pi_k > 0 \ \forall k \ \text{and} \ \sum_{k=1}^{K} \pi_k = 1. \]

Maximum-Likelihood Estimation

\[ \hat{\theta} \in \arg \max_{\theta} \log L(\theta) \]

log-likelihood:

\[ \log L(\theta) = \sum_{i=1}^{n} \log \left( \sum_{k=1}^{K} \pi_k f_k(x_i; \theta_k) \right). \]

The EM algorithm [Dempster et al., 1977, McLachlan and Krishnan, 2008]

\[ \theta^{new} \in \arg \max_{\theta \in \Omega} \mathbb{E}[\log L_c(\theta)|D, \theta^{old}] \]

complete log-likelihood:

\[ \log L_c(\theta) = \sum_{i=1}^{n} \sum_{k=1}^{K} Z_{ik} \log \left[ \pi_k f_k(x_i; \theta_k) \right] \quad \text{where} \quad Z_{ik} = \mathbb{I}(Z_i = k) \]

Clustering

\[ \hat{z}_i = \arg \max_{1 \leq k \leq K} \mathbb{P}(Z_i = k|x_i; \hat{\theta}), \quad (i = 1, \ldots, n) \]
Mixtures in a high-dimensional setting

- Parsimonious GMMs [Banfield and Raftery, 1993, Celeux and Govaert, 1995]:
  - Eigenvalue decomposition of the covariance mat. $\Sigma_k = \lambda_k D_k A_k D_k^T$.
  - $\lambda_k$ the volume of the $k$th cluster (the amount of space of the cluster).
  - $D_k = (v_{k1}, \ldots, v_{kp})$ orthogonal matrix of eigenvectors $v$ of $\Sigma_k$ determines the orientation of the cluster.
  - $A_k = \text{diag}(\lambda_{k1}, \ldots, \lambda_{kp})/|\Sigma_k|^{1/p}$ a normalized diagonal matrix (its determinant is 1) of the eigenvalues of $\Sigma_k$ arranged in a decreasing order. This matrix is associated with the shape of the cluster.
Mixtures in a high-dimensional setting

for $p > n$:

- **LASSO Regularization** : [Pan and Shen, 2007] [Celeux et al., 2019]
- Mixtures of Factor Analyzers [McLachlan et al., 2003] (or MCFA extension)

$$\Sigma_k = B_k B_k^T + \Lambda_k :$$

$B_k$ is a $p \times q$ (with $q < p$) matrix and $\Lambda_k$ is a diagonal matrix.

$$\rightarrow (B_k B_k^T + \Lambda_k)^{-1} \text{ and } |B_k B_k^T + \Lambda_k| \text{ are calculated in a } q\text{-dimensional space!}$$

$\rightarrow$ Here we consider the case where the data are entire functions : $\{X(t); t \in \mathcal{T}\}$
Outline

1 Introduction

2 Functional Data Analysis Framework

3 Mixture-of-Experts Modeling

4 Functional Mixture-of-Experts (FunME)

5 Statistical Inference
Functional data are increasingly frequent

- Railway time-series trajectories
- Tecator data
- Phonemes curves
- Satellite waveforms
Statistical analysis of functional data

A broad literature:
[James and Hastie, 2001, James and Sugar, 2003]
[Ramsay and Silverman, 2005]
[Ferraty and Vieu, 2006]
[Ramsay et al., 2011]
[Bouveyron and Jacques, 2011]
[Samé et al., 2011]
[Delaigle et al., 2012]
[Jacques and Preda, 2014]
[Bouveyron et al., 2018]
[Qiao et al., 2018]
A review can be found in [Chamroukhi and Nguyen, 2018]

- Functional regression
- Functional classification
- Functional clustering, including model-based
- Functional graphical models
- ...
Classification of functional data

Phonemes data set¹: \( n = 1000 \) log-periodograms for \( m = 150 \) frequencies

¹ Data from http://www.math.univ-toulouse.fr/staph/npfda/, used in Ferraty and Vieu [2003]
Clustering of functional data

Clustering real curves of high-speed railway-switch operations
Data: \( n = 115 \) curves of \( m \approx 510 \) observations
\( K = 2 \) clusters: operating state without/with possible defect
Clustering switch operations

Clustering real curves of high-speed railway-switch operations

Data: $n = 115$ curves of $m \simeq 510$ observations

$K = 2$ clusters: operating state without/with possible defect
Mixture-of-Experts modeling (for vectorial data)

- Data: an observed i.i.d sample of the pair \((X, Y)\) where the response \(Y \in \mathbb{R}\) for the vector of predictors \(X \in \mathbb{R}^p\) is governed by a hidden categorical variable \(Z\)
  
- \(z_i \in [K]\) is the expert label for \((X_i, Y_i)\)

- Mixture of experts (ME) [Jacobs et al., 1991, Jordan and Jacobs, 1994]:

\[
  f(y|x; \Psi) = \sum_{k=1}^{K} \pi_k(x; w) f_k(y|x; \theta_k)
\]

- Gating network (e.g. softmax): \(\pi_k(x; w) = \frac{\exp(w_{k0} + w_k^T x)}{1 + \sum_{\ell=1}^{K-1} \exp(w_{\ell0} + w_\ell^T x)}\)

- Experts network (e.g. Gaussian regressors): \(f_k(y|x; \theta_k) = \phi(y; \mu(x; \beta_k), \sigma_k^2)\) with parametric (non-)linear regression functions \(\mu(x; \beta_k)\)

- Parameter vector \(\Psi = (w^T, \Psi_1^T, \ldots, \Psi_K^T)^T\)

\(\rightarrow\) For a review, see Nguyen and Chamroukhi [2018]
Illustration

Cluster 1
Cluster 2
Cluster 3
Expert mean 1
Expert mean 2
Expert mean 3

Mixing probabilities

Faicel Chamroukhi
Functional Mixtures-of-Experts
Fitting the ME model

Maximum Likelihood Estimation via EM [Dempster et al., 1977, Jacobs et al., 1991]

- MLE: $\hat{\Psi}$ is commonly estimated by maximizing the observed-data log-likelihood:
  
  $$\hat{\Psi}_n \in \arg \max_{\Psi \in \Theta} L(\Psi) \text{ with } L(\Psi) = \sum_{i=1}^{n} \log \sum_{k=1}^{K} \pi_k(x_i; w) f(y_i | x_i; \Psi_k)$$

  $\leftrightarrow$ the EM algorithm

Consider a high-dimensional setting

Looking for sparse models

Regularized MLE of the ME [Khalili, 2010] [Chamroukhi and Huynh, 2019]

$\hat{\Psi}_n \in \arg \max_{\Psi \in \Theta} L(\Psi) - \text{Pen} \lambda(\Psi)$

$\text{Pen} \lambda(\Psi)$

LASSO penalties for experts and the gating network encourages sparse solutions

performs parameter estimation and feature selection

Doesn't apply (directly) to functional data (e.g., functional predictors and/or responses)
Fitting the ME model

Maximum Likelihood Estimation via EM [Dempster et al., 1977, Jacobs et al., 1991]

- MLE: $\hat{\Psi}$ is commonly estimated by maximizing the observed-data log-likelihood:
  \[
  \hat{\Psi}_n \in \arg \max_{\Psi \in \Theta} L(\Psi) \quad \text{with} \quad L(\Psi) = \sum_{i=1}^{n} \log \sum_{k=1}^{K} \pi_k(x_i; w) f(y_i|x_i; \Psi_k)
  \]
  $\hookrightarrow$ the EM algorithm

$\hookrightarrow$ Consider a high-dimensional setting
$\hookrightarrow$ Looking for sparse models

Regularized MLE of the ME [Khalili, 2010] [Chamroukhi and Huynh, 2019]

$\Psi$ is estimated by maximizing a penalized observed-data log-likelihood:

\[
\hat{\Psi}_n \in \arg \max_{\Psi \in \Theta} L(\Psi) - \text{Pen}_\lambda(\Psi)
\]

- $\hookrightarrow$ $\text{Pen}_\lambda(\Psi)$ LASSO penalties for experts and the gating network
- encourages sparse solutions
- performs parameter estimation and feature selection

$\hookrightarrow$ Doesn’t apply (directly) to functional data (e.g functional predictors and/or responses)
Mixtures-of-Experts with functional predictors

- ME to relate functional predictors \( \{X(t) \in \mathbb{R}; t \in \mathcal{T} \subset \mathbb{R} \} \) to a scalar response \( Y \in \mathcal{Y} \subset \mathbb{R} \)
- The inputs \( X(\cdot) \) are data continuously recorded from (multiple) subject’ sensors for some time period

**Figure** – Functional predictors \( X_{ij}(t) \ t \in \mathcal{T}, i = 1, \cdots, n \) and \( j = 1, \ldots, p \).
Mixtures-of-Experts with functional predictors

- ME to relate functional predictors \( \{X(t) \in \mathbb{R}; \; t \in \mathcal{T} \subset \mathbb{R} \} \) to a scalar response \( Y \in \mathcal{Y} \subset \mathbb{R} \).

- The inputs \( X(\cdot) \) are data continuously recorded from (multiple) subject’s sensors for some time period.

\[ X_{ij}(t) \in \mathbb{R}; \; t \in \mathcal{T} \subset \mathbb{R}, \; i = 1, \ldots, n \text{ and } j = 1, \ldots, p. \]

\[ \mathcal{F} \quad \text{Figure} \quad \text{Functional predictors } X_{ij}(t) \; t \in \mathcal{T}, \; i = 1, \ldots, n \text{ and } j = 1, \ldots, p. \]

\[ \leftarrow \quad \text{We first consider univariate functional predictors } (p = 1) \]

- Let \( \{X_i(\cdot), Y_i\}_{i=1}^n \) be a random i.i.d sample from the pair \( \{X(\cdot), Y\} \).
ME for functional predictors and a scalar response

Questioning

Regression, Clustering and classification of observations with functional predictors with three guidelines:

- (1) generative modeling: warranty for estimation and prediction
- (2) deal with high-dimensional setting (sparsity and feature selection)
- (3) User guideline: keep an interpretable fit

Proposed answering

(1) Mixture modeling (Mixture-of-Experts model) (2) regularization to encourage sparse solutions (3) Functional regression, classification and clustering

Main modeling guidelines

- Functional generalized linear models [James, 2002, Müller and Stadtmüller, 2005] (including FLR)
- Functional linear regression (FLR) (anf FGLM) that’s interpretable FLiRTI [James et al., 2009]
The experts are formulated as functional regression models (see eg. James [2002])

\[ Y_i = \beta_{z_i,0} + \int_{\mathcal{T}} X_i(t)\beta_{z_i}(t)dt + \varepsilon_i, \quad i = 1, \ldots, n, \tag{1} \]

\( z_i \in [K] \) is the unknown expert label for \((X_i(\cdot), Y_i)\)
\( \beta_{z_i,0} \in \mathbb{R} \) is an unknown intercept coefficient of functional LR \( z_i \)
\( \{\beta_{z_i}(t) \in \mathbb{R}; t \in \mathcal{T}\} \) is the unknown function of parameters of functional expert \( z_i \)
\( \varepsilon_i \overset{iid}{\sim} \mathcal{N}(0, \sigma^2_{z_i}) \) with \( \sigma^2_{z_i} \in \mathbb{R}^+ \) the variance of expert \( z_i \)
Stochastic representation of the FunME model

**Functional experts network**

- The experts are formulated as functional regression models (see eg. James [2002])

\[ Y_i = \beta_{z_i,0} + \int_{\mathcal{T}} X_i(t)\beta_{z_i}(t)dt + \varepsilon_i, \quad i = 1, \ldots, n, \]  

\( z_i \in [K] \) is the unknown expert label for \((X_i(.), Y_i)\)

\( \beta_{z_i,0} \in \mathbb{R} \) is an unknown intercept coefficient of functional LR \( z_i \)

\( \{\beta_{z_i}(t) \in \mathbb{R}; t \in \mathcal{T}\} \) is the unknown function of parameters of functional expert \( z_i \)

\( \varepsilon_i \iid \mathcal{N}(0, \sigma^2_{z_i}) \) with \( \sigma^2_{z_i} \in \mathbb{R}^+ \) the variance of expert \( z_i \)

**Functional gating network**

- Multinomial logistic (softmax) functional gated network : For \( z = 1, \cdots, K - 1 \):

\[ h_z(X(t), t \in \mathcal{T}) = \log \left\{ \frac{\mathbb{P}(Z = z|X(t), t \in \mathcal{T})}{\mathbb{P}(Z = K|X(t), t \in \mathcal{T})} \right\} = \alpha_{z,0} + \int_{\mathcal{T}} X(t)\alpha_z(t)dt \]

\[ \mathbb{P}(Z = z|X(t), t \in \mathcal{T}) = \frac{\exp (\alpha_{z,0} + \int_{\mathcal{T}} X(t)\alpha_z(t)dt)}{1 + \sum_{z' = 1}^{K-1} \exp (\alpha_{z',0} + \int_{\mathcal{T}} X(t)\alpha_{z'}(t)dt)}, \]  

\( \alpha_{z,0} \in \mathbb{R} \) is an unknown intercept parameter

\( \{\alpha_z(t) \in \mathbb{R}; t \in \mathcal{T}\} \) is the unknown function of parameters of gating network \( z \)
Representation of the functional predictors

\[ Y_i = \beta_{z_i,0} + \int_{\mathcal{T}} X_i(t)\beta_{z_i}(t)dt + \varepsilon_i, \quad i = 1, \ldots, n, \]

\[ h_z(X(t), t \in \mathcal{T}) = \alpha_{z,0} + \int_{\mathcal{T}} X(t)\alpha_z(t)dt. \]

- Estimating the coefficient functions \( \alpha(.) \) and \( \beta(.) \) is a high-dimensional problem needs approximation for dimensionality reduction
- Two main approaches: i) basis representation ii) functional PCA (FPCA) [Ramsay and Silverman, 2005]
Representation of the functional predictors

\[ Y_i = \beta_{z_i,0} + \int_{T} X_i(t)\beta_{z_i}(t)dt + \varepsilon_i, \quad i = 1, \ldots, n, \]

\[ h_z(X(t), t \in T) = \alpha_{z,0} + \int_{T} X(t)\alpha_z(t)dt. \]

- Estimating the coefficient functions \( \alpha(.) \) and \( \beta(.) \) is a high-dimensional problem needs approximation for dimensionality reduction.
- Two main approaches : i) basis representation ii) functional PCA (FPCA) [Ramsay and Silverman, 2005]

Here we represent the functional data by using a basis expansion:

\[ X_i(t) = \sum_{j=1}^{r} x_{ij} b_j(t) = \mathbf{x}_i^\top \mathbf{b}_r(t), \quad (3) \]

- \( \mathbf{b}_r(t) = (b_1(t), b_2(t), \ldots, b_r(t))^\top \) is an \( r \)-dimensional basis ((B-)spline, Wavelet,..)
- \( \mathbf{x}_i = (x_{i1}, \ldots, x_{ir})^\top \) can be seen as the vector representation of \( X_i(.) \)
Representation of the functional predictors

\[ Y_i = \beta_{z_i,0} + \int_{\mathcal{T}} X_i(t) \beta_{z_i}(t) dt + \varepsilon_i, \quad i = 1, \ldots, n, \]

\[ h_z(X(t), t \in \mathcal{T}) = \alpha_{z,0} + \int_{\mathcal{T}} X(t) \alpha_z(t) dt. \]

- Estimating the coefficient functions \( \alpha(.) \) and \( \beta(.) \) is a high-dimensional problem \( \hookrightarrow \) needs approximation for dimensionality reduction
- Two main approaches : i) basis representation ii) functional PCA (FPCA) [Ramsay and Silverman, 2005]

\[ \hookrightarrow \text{ Here we represent the functional data by using a basis expansion :} \]

\[ X_i(t) = \sum_{j=1}^{r} x_{ij} b_j(t) = \mathbf{x}_i^\top \mathbf{b}_r(t), \quad (3) \]

- \( \mathbf{b}_r(t) = (b_1(t), b_2(t), \ldots, b_r(t))^\top \) is an \( r \)-dimensional basis ((B-)spline, Wavelet,..)
- \( \mathbf{x}_i = (x_{i1}, \ldots, x_{ir})^\top \) can be seen as the vector representation of \( X_i(.) \)

Here the \( X \)'s are directly observed. We later consider the case when they are not.

\( \hookrightarrow \) The \( x_{ij} \)'s can be computed explicitly by \( x_{ij} = \int_{\mathcal{T}} X_i(t) b_j(t) dt \) for \( j = 1, \ldots, r \) and \( \mathbf{x}_i = (x_{i1}, \ldots, x_{ir})^\top. \)
Representation of the functional gating network

Functional linear predictor for the gating network defined as:

\[
h_z(X(t), t \in T) = \log \left\{ \frac{\mathbb{P}(Z = z|X(t), t \in T)}{\mathbb{P}(Z = K|X(t), t \in T)} \right\} = \alpha_{z,0} + \int_T X(t) \alpha_z(t) dt
\]

The function \( \alpha_z(t) \) is represented similarly as for \( X \) function by

\[
\alpha_z(t) = \sum_{j=1}^{q} \zeta_{z,j} b_j(t) = \zeta_z^\top b_q(t)
\]

(4)

where

- \( b_q(t) = (b_1(t), \ldots, b_q(t))^\top \) is a \( q \)-dimensional basis (of the same type as \( X \)).
- \( \zeta_z = (\xi_{z,1}, \xi_{z,2}, \ldots, \xi_{z,q})^\top \) is the vector of logistic regression coefficients
Representation of the functional gating network

Then the functional linear predictor \( h_z(X_i) \) for \( i = 1, \ldots, n \) is represented as

\[
h_z(X_i(t), t \in \mathcal{T}; \alpha) = \alpha_{z_i,0} + \int_{\mathcal{T}} X_i(t) \alpha_{z_i}(t) dt = \alpha_{z_i,0} + \int_{\mathcal{T}} x_i^\top b_r(t) b_q(t) \xi_{z_i} dt \\
= \alpha_{z_i,0} + x_i^\top \left( \int_{\mathcal{T}} b_r(t) b_q(t) dt \right) \xi_{z_i} \\
= \alpha_{z_i,0} + \xi_{z_i}^\top r_i,
\]

where

- \( x_i = (x_{i,1}, \ldots, x_{i,r})^\top \)
- \( r_i = \left( \int_{\mathcal{T}} b_r(t) b_q(t) dt \right)^\top x_i \)
Then the functional linear predictor $h_z(X_i)$ for $i = 1, \ldots, n$ is represented as

$$
h_z(X_i(t), t \in T; \alpha) = \alpha_{z,i,0} + \int_T X_i(t)\alpha_{z,i}(t)dt = \alpha_{z,i,0} + \int_T x_i^\top b_r(t)b_q^\top(t)\zeta_{z,i}dt
$$

$$
= \alpha_{z,i,0} + x_i^\top \left( \int_T b_r(t)b_q^\top(t)dt \right) \zeta_{z,i}
$$

$$
= \alpha_{z,i,0} + \zeta_{z,i}^\top r_i,
$$

where

- $x_i = (x_{i,1}, \ldots, x_{i,r})^\top$
- $r_i = \left( \int_T b_r(t)b_q(t)^\top dt \right)^\top x_i$

The FunME gating network (2) is then now phrased as

$$
h_{z,i}(X_i; \xi) = \alpha_{z,i,0} + \zeta_{z,i}^\top r_i
$$

$$
\pi_k(r_i; \xi) = \frac{\exp \{\alpha_{k,0} + \zeta_k^\top r_i\}}{1 + \sum_{k'=1}^{K-1} \exp \{\alpha_{k',0} + \zeta_{k'}^\top r_i\}}
$$

(5)

where $\xi = ((\alpha_{1,0}, \zeta_1^\top), \ldots, (\alpha_{K-1,0}, \zeta_{K-1}^\top))^\top \in \mathbb{R}^{(K-1) \times (q+1)}$ is the unknown parameter vector of the gating network, to be estimated.
Representation of the functional experts

\[ Y_i = \beta_{z_i,0} + \int_T X_i(t) \beta_{z_i}(t) dt + \varepsilon_i, \quad i = 1, \ldots, n. \]

- The coefficient function \( \beta_z(\cdot) \) is represented by the following expansion:

\[ \beta_z(t) = \sum_{j=1}^{p} \eta_{z,j} b_j(t) + e(t) = \eta_z^\top b_p(t) + e(t) \]

- \( b_p(t) = (b_1(t), b_2(t), \ldots, b_p(t))^\top \) is a \( p \)-dimensional basis \(((B-)spline, \text{Wavelet,}..)\)
- \( \eta_z = (\eta_{z,1}, \eta_{z,2}, \ldots, \eta_{z,p})^\top \) is the vector of regression coefficients
- \( e(t) \stackrel{iid}{\sim} \mathcal{N}(0, \sigma^2_e), \quad e(\cdot) \perp X_i \)'s and represents the approximation error of \( \beta_z(t) \) by linear projection \( b_p(t) \top \eta_z \).
Representation of the functional experts

The functional linear expert regressor \( z \) is then represented as:

\[
Y_i = \beta_{zi,0} + \int_{T} X_i(t) \beta_{zi}(t) dt + \varepsilon_i = \beta_{zi,0} + \int_{T} x_i^\top b_r(t) \left( b_p^\top(t) \eta_{zi} + e_i(t) \right) dt + \varepsilon_i \\
= \beta_{zi,0} + x_i^\top \left( \int_{T} b_r(t) b_p^\top(t) dt \right) \eta_{zi} + \int_{T} X_i(t) e(t) dt + \varepsilon_i \\
= \beta_{zi,0} + \eta_{zi}^\top x_i + \varepsilon_i + \int_{T} X_i(t) e(t) dt
\]

where

- \( x_i = (x_i, 1, \ldots, x_i, r)^\top \)
- \( x_i = \left( \int_{T} b_r(t) b_p(t)^\top dt \right)^\top x_i \)
- \( \varepsilon_i^* = \varepsilon_i + \int_{T} X_i(t) e(t) dt \sim \mathcal{N}(0, \sigma_{zi}^2) \).
Representation of the functional experts

The functional linear expert regressor $z$ is then represented as:

$$
Y_i = \beta_{zi,0} + \int_T X_i(t) \beta_{zi}(t) dt + \epsilon_i = \beta_{zi,0} + \int_T x_i^T b_r(t) \left( b_p^T(t) \eta_{zi} + e_i(t) \right) dt + \epsilon_i
$$

$$
= \beta_{zi,0} + x_i^T \left( \int_T b_r(t) b_p^T(t) dt \right) \eta_{zi} + \int_T X_i(t) e(t) dt + \epsilon_i
$$

$$
= \beta_{zi,0} + \eta_{zi}^T x_i + \epsilon_i + \int_T X_i(t) e(t) dt
$$

where

- $x_i = (x_{i,1}, \ldots, x_{i,r})^T$
- $x_i = \left( \int_T b_r(t) b_p(t)^T dt \right)^T x_i$
- $\epsilon_i^* = \epsilon_i + \int_T X_i(t) e(t) dt \sim \mathcal{N}(0, \sigma_{zi}^2)$.

The FunME expert (1) can thus be expressed as

$$
Y_i = \beta_{zi,0} + \eta_{zi}^T x_i + \epsilon_i^*, \quad i = 1, \ldots, n, \quad (7)
$$

and we have $f(y_i|x_i(\cdot), z_i = k; \theta_k) = \phi(y_i; \beta_{k,0} + \eta_k^T x_i, \sigma_k^2)$ where $\theta_k = (\beta_{k,0}, \eta_k^T, \sigma_k^2)^T \in \mathbb{R}^{p+2}$ is the unknown parameter vector of expert density $k$. 
FunME model

The Functional ME model

Combining (5) and (7), the resulting FunME distribution is defined by

$$f(y_i | X_i ; \Psi) = \sum_{k=1}^{K} \pi_k(r_i; \xi) \phi(y_i; \beta_{k,0} + \eta_k^T x_i, \sigma_k^*^2)$$

(8)

where $\pi_k(r_i; \xi) = \exp\{\alpha_{k,0} + \zeta_k^T r_i\}/\left[1 + \sum_{k'=1}^{K-1} \exp\{\alpha_{k',0} + \zeta_{k'}^T r_i\}\right]$ and $\Psi = (\xi^T, \theta_1^T, \ldots, \theta_K^T)^T$ the unknown parameter vector of the model

Model fitting

Since it is a mixture-of-experts model, then $\Psi$ can be estimated by:

- Regularized ML to encourage sparsity (eg. lasso penalty [Tibshirani, 1996])
- Regularized ML (lasso-type regularization) on the derivatives of the $\alpha(\cdot)$ and $\beta(\cdot)$ function, by relying on the FLiRTI methodology [James et al., 2009]
1) FunME and MLE via the EM algorithm

Maximum-Likelihood Estimation

\[ \hat{\Psi} \in \arg \max_{\Psi} \log L(\Psi) \]

log-likelihood: \[ \log L(\Psi) = \sum_{i=1}^{n} \log \sum_{k=1}^{K} \pi_k(X_i; \xi) \phi(y_i; \beta_{k,0} + \eta_k^T x_i, \sigma_k^2) \]
1) FunME and MLE via the EM algorithm

Maximum-Likelihood Estimation

$$\hat{\Psi} \in \arg \max_{\Psi} \log L(\Psi)$$

log-likelihood: $$\log L(\Psi) = \sum_{i=1}^{n} \log \sum_{k=1}^{K} \pi_k(X_i; \xi) \phi(y_i; \beta_k,0 + \eta_k^\top x_i, \sigma_k^*)$$

The EM algorithm [Dempster et al., 1977]

$$\Psi^{new} \in \arg \max_{\Psi \in \Omega} \mathbb{E}[\log L_c(\Psi)|\{X_i,Y_i\}_{i=1}^{n},\Psi^{old}]$$

complete log-likelihood:

$$\log L_c(\Psi) = \sum_{i=1}^{n} \sum_{k=1}^{K} Z_{ik} \log \left[ \pi_k(r_i; \xi) \phi(y_i; \beta_k,0 + \eta_k^\top x_i, \sigma_k^*) \right] \text{ where}$$

$$Z_{ik} = 1_{\{z_i=k\}}, \ k = 1, \ldots, K$$
1) FunME and MLE via the EM algorithm

**Maximum-Likelihood Estimation**

\[ \hat{\Psi} \in \arg \max_{\Psi} \log L(\Psi) \]

log-likelihood:

\[ \log L(\Psi) = \sum_{i=1}^{n} \log \sum_{k=1}^{K} \pi_k(X_i; \xi) \phi(y_i; \beta_k, 0 + \eta_k^\top x_i, \sigma_k^*{2}) \]

**The EM algorithm [Dempster et al., 1977]**

\[ \Psi^{new} \in \arg \max_{\Psi \in \Omega} \mathbb{E}[\log L_c(\Psi)|\{X_i, Y_i\}_{i=1}^{n}, \Psi^{old}] \]

complete log-likelihood:

\[ \log L_c(\Psi) = \sum_{i=1}^{n} \sum_{k=1}^{K} Z_{ik} \log \left[ \pi_k(r_i; \xi) \phi(y_i; \beta_k, 0 + \eta_k^\top x_i, \sigma_k^*{2}) \right] \]

where

\[ Z_{ik} = 1_{\{z_i = k\}}, \ k = 1, \ldots, K \]

**Clustering, Regression**

- Expert label:
  \[ \hat{z}_i = \arg \max_{1 \leq k \leq K} \mathbb{E}(Z_{ik}|X_i; \hat{\Psi}), \quad (i = 1, \ldots, n) \]

- Expert’s mean function:
  \[ \hat{y}_i|\{X_i, \hat{z}_i = k\} = \hat{\beta}_k, 0 + \hat{\eta}_k^\top x_i, \quad (i = 1, \ldots, n; \ k = 1, \ldots, K) \]

- FunME mean function:
  \[ \hat{y}_i = \sum_{k=1}^{K} \pi_k(\hat{\beta}_k, 0 + \hat{\eta}_k^\top x_i), \quad (i = 1, \ldots, n) \]
ML parameter estimation via EM (FunME-EM)

The E-Step

Compute the expectation of the complete-data log-likelihood, given the observed data \(\{x_i(\cdot), y_i\}_{i=1}^n\), using the current parameter vector \(\Psi^{(s)}\):

\[
Q(\Psi; \Psi^{(s)}) = \mathbb{E}\left[ \log L_c(\Psi) | \{x(\cdot), y\}_{i=1}^n; \Psi^{(s)} \right]
\]

\[
= \sum_{i=1}^n \sum_{k=1}^K \tau_{ik}^{(s)} \log \left[ \pi_k(x_i(\cdot); \xi) \phi(y_i; \beta_{0,k} + \eta_k^\top x_i, \sigma^*_{k}^2) \right], \tag{9}
\]

where \(\tau_{ik}^{(s)} = \phi(y_i; \beta_{0,k}^{(s)} + x_i^\top \eta_k^{(s)}, \sigma^2_{k}^{(s)}) / f(y_i|x_i; \Psi^{(s)})\), is the probability that the pair \((x_i(t), y_i)\) is generated by the \(k\)th expert.
ML parameter estimation via EM (FunME-EM)

The E-Step

Compute the expectation of the complete-data log-likelihood, given the observed data 
\{x_i(\cdot), y_i\}_{i=1}^n, using the current parameter vector \(\Psi^{(s)}\):

\[
Q(\Psi; \Psi^{(s)}) = \mathbb{E} \left[ \log L_c(\Psi) | \{x(\cdot), y\}_{i=1}^n; \Psi^{(s)} \right]
\]

\[
= \sum_{i=1}^n \sum_{k=1}^K \tau_{ik}^{(s)} \log \left[ \pi_k(x_i(\cdot); \xi) \phi(y_i; \beta_k, 0 + \eta_k^\top x_i, \sigma^2_k) \right],
\]  
(9)

where \(\tau_{ik}^{(s)} = \phi(y_i; \beta_{0,k}^{(s)} + x_i^\top \eta_k^{(s)}, \sigma^2_k)/f(y_i|x_i; \Psi^{(s)})\), is the probability that the pair \((x_i(t), y_i)\) is generated by the \(k\)th expert.

The M-Step

- Update the value of the parameter vector \(\Psi\) by \(\Psi^{(s+1)} = \arg\max_{\Psi} Q(\Psi; \Psi^{(s)})\)
- Separate maximizations w.r.t the gating network and the experts network

\[
\xi^{(s+1)} = \arg\max_{\xi} \left\{ Q(\xi; \Psi^{(s)}) = \sum_{i=1}^n \sum_{k=1}^K \tau_{ik}^{(s)} \log \pi_k(x_i(\cdot); \xi) \right\}
\]  
(10)

\[
\theta_k^{(s+1)} = \arg\max_{\theta_k} \left\{ Q(\theta_k; \Psi^{(s)}) = \sum_{i=1}^n \tau_{ik}^{(s)} \log \phi(y_i; \beta_k, 0 + \eta_k^\top x_i, \sigma^2_k) \right\}
\]  
(11)
2) Regularized MLE via an EM-lasso algorithm

$\rightarrow p \gg n$ to ensure a good approximation of $\beta_z(t)$ by $\eta_z^T b_p(t)$ (tradeoff between smoothness of the functional predictor and complexity of the estimation problem.)

Regularized Maximum-Likelihood Estimation

$\hat{\Psi} \in \arg \max_{\Psi} \log L(\Psi) - \text{Pen}_{\lambda, \chi}(\Psi)$

log-likelihood : $\log L(\Psi) = \sum_{i=1}^{n} \log \sum_{k=1}^{K} \pi_k(X_i; \xi) \phi(y_i; \beta_k, 0 + \eta_k^T x_i, \sigma_k^*)$
2) Regularized MLE via an EM-lasso algorithm

$\rightarrow p \gg n$ to ensure a good approximation of $\beta_z(t)$ by $\eta^T_z b_p(t)$ (tradeoff between smoothness of the functional predictor and complexity of the estimation problem.)

Regularized Maximum-Likelihood Estimation

$$\hat{\Psi} \in \arg \max_{\Psi} \log L(\Psi) - \text{Pen}_{\lambda, \chi}(\Psi)$$

log-likelihood: $\log L(\Psi) = \sum_{i=1}^n \log \sum_{k=1}^K \pi_k(X_i; \xi) \phi(y_i; \beta_k, 0 + \eta_k^T x_i, \sigma^*_k^2)$

The EM-lasso algorithm

$$\Psi^{new} \in \arg \max_{\Psi \in \Omega} \mathbb{E}[\log L^c_{\lambda, \chi}(\Psi) | \{X_i, Y_i\}^{n}_{i=1}, \Psi^{old}]$$

complete log-likelihood:

$$\log L^c_{\lambda, \chi}(\Psi) = \sum_{i=1}^n \sum_{k=1}^K Z_{ik} \log \left[ \pi_k(X_i; \xi) \phi(y_i; \beta_k, 0 + \eta_k^T x_i, \sigma^*_k^2) \right] - \text{Pen}_{\lambda, \chi}(\Psi)$$

Lasso regularization

$$\text{Pen}_{\lambda, \chi}(\Psi) = \lambda \sum_{k=1}^K \| \eta_k \|_1 + \chi \sum_{k=1}^{K-1} \| \xi_k \|_1$$ (12)

where $\lambda$ and $\chi$ are positive real values representing tuning parameters.
Regularized MLE via EM-lasso (FunME-EMlasso)

The EM-lasso algorithm for FunME

- E-Step : unchanged
- M-Step : $\Psi^{(s+1)} = \arg\max_{\Psi} \left\{ Q_{\lambda,\chi}(\Psi; \Psi^{(s)}) = Q(\Psi; \Psi^{(s)}) - \text{Pen}_{\lambda,\chi}(\Psi) \right\}$

Updating the expert’ network parameters

$\theta_k^{(s+1)} \in \arg\max_{\theta_k} Q_{\lambda}(\theta_k; \Psi^{(s)})$ with

$$Q_{\lambda}(\theta_k; \Psi^{(s)}) = \sum_{i=1}^{n} \tau_{ik}^{(s)} \log \phi(y_i; \beta_k, 0 + \eta_k^T x_i, \sigma_k^*) - \lambda \sum_{j=1}^{p} |\eta_{kj}|,$$

- $\Rightarrow$ A weighted LASSO problem for the $\eta_k$'s
- $\Rightarrow$ Apply the LASSO machinery
- $\Rightarrow$ the update of $\sigma_k^*$ is a weighted variant of the standard univariate Gaussian regression
Updating the gating network parameters

\( \xi^{(s+1)} \in \arg \max_\xi \; Q_\chi(\xi; \Psi^{(s)}) \) with

\[
Q_\chi(\xi; \Psi^{(s)}) = \sum_{i=1}^{n} \sum_{k=1}^{K} \tau_{ik}^{(s)} \log \pi_k(r_i; \xi) - \chi \sum_{k=1}^{K-1} q \sum_{j=1}^{q} |\xi_{kj}|
\]

\[
= \sum_{i=1}^{n} \left( \sum_{k=1}^{K-1} \tau_{ik}^{(s)} \left( \alpha_{k,0} + \xi_k^\top r_i \right) - \log \left( 1 + \sum_{k' = 1}^{K-1} \exp \{ \alpha_{k',0} + \xi_{k'}^\top r_i \} \right) \right) - \chi \sum_{k=1}^{K-1} q \sum_{j=1}^{q} |\xi_{kj}|,\]

→ A weighted version of the regularized multinomial logistic problem (e.g. [Mousavi and Sørensen, 2017])

- There is no closed-form solution
- we then use a Newton-Raphson with Coordinate Ascent updates of the gating network coefficients \( \xi_{kj} \).
Coordinate Ascent for the gating network

For each expert $k$, for $j = 1, \ldots, p$:

\[
\zeta_{k,j}^{(t+1)} = S \left( \frac{\sum_{i=1}^{n} w_{ik} r_{ij} (\tilde{h}_i^{(t)} - \tilde{z}_i^{(t)}); \chi}{\sum_{i=1}^{n} w_{ik} r_{ij}^2} \right) \sum_{i=1}^{n} w_{ik} r_{ij}^2
\]

\[
= S \left( R_j^T W_k^{(t)} (\tilde{h}_i^{(t)} - \tilde{z}_i^{(t)}); \chi \right) / (R_j^T W_k^{(t)} R_j) \quad (13)
\]

where

- $\tilde{h}_i^{(s)} = \alpha_{k,0}^{(s)} + r_i^T \zeta_k + (\tau_{ik}^{(s)} - \pi_k(r_i; \xi^{(s)}))/w_{ik}$ is the working response
- $\tilde{z}_i^{(s)} = \alpha_{k,0}^{(s)} + r_i^T \zeta_k - r_{ij} \zeta_{k,j}^{(t+1)}$; fitted value excluding the contribution from $\zeta_{k,j}$
- $w_{ik} = \pi_k(r_i; \xi^{(t)})(1 - \pi_k(r_i; \xi^{(t)}))$
- $W_k^{(t)} = \text{diag}(w_{ik}, \ldots, w_{nk})$ and $R_j$ is the $j$th column of $R = (r_1, \ldots, r_n)^T$
- $S(\cdot)$ is a soft-thresholding operator defined by $S(u, \chi) = \text{sign}(u)(|u| - \chi)_+$ and $(x)_+$ a shorthand for $\max\{x, 0\}$

For $\alpha_{k,0}$, the update is given by

\[
\alpha_{k,0}^{(t+1)} = \frac{\sum_{i=1}^{n} w_{ik} (\tilde{h}_i^{(t)} - r_i^T \zeta_k^{(t)})}{\sum_{i=1}^{n} w_{ik}} = W_k^{(q)} (\tilde{h}_i^{(t)} - R \zeta_i^{(t)}) / \text{trace}(W_k^{(q)})
\]
Coordinate Ascent for the expert network

For each expert $k$, for $j = 1, \ldots, p$:

$$
\eta_{k,j}^{(q+1)} = S \left( \sum_{i=1}^{n} \tau_{ik}^{(s)} (y_i - \beta_{k0}^{(s)} - x_i^T \beta_k^{(s)} + \eta_{k,j}^{(q)} x_{ij}); \lambda \sigma_k^{(s)^2} \right) / \sum_{i=1}^{n} \tau_{ik}^{(s)} x_{ij}^2
$$

$$
= S \left( X_j^T W_k^{(q)} r_{kj}^{(q)}; \lambda \sigma_k^{(s)^2} \right) / (X_j^T W_k^{(q)} X_j)
$$

(14)

where $X_j$ is the $j$th column of the design matrix $X = (x_1, \ldots, x_n)^T$,

$W_k^{(q)} = \text{diag}(\tau_{1k}^{(q)}, \ldots, \tau_{nk}^{(q)})$,

$r_{kj}^{(q)} = y - \beta_{k0}^{(s)} 1_n - X \beta_k^{(q)} + \beta_{kj}^{(q)} X_j$ is the residual without the contribution of the $j$th coefficient

$S(u, \eta) := \text{sign}(u)(|u| - \eta)_+$ is the soft-thresholding operator with $(.)_+ = \max\{., 0\}$.

$$
\beta_{k,0}^{(s+1)} = \frac{\sum_{i=1}^{n} \tau_{ik}^{(s)} (y_i - x_i^T \eta_k^{(s)})}{\sum_{i=1}^{n} \tau_{ik}^{(s)}} = W_k^{(q)} (y - X \eta_k^{(q)}) / \text{trace}(W_k^{(q)}),
$$

(15)

$$
\sigma_k^{2(s+1)} = \frac{\sum_{i=1}^{n} \tau_{ik}^{(s)} \left(y_i - \beta_{k,0}^{(s+1)} - x_i^T \eta_k^{(s+1)}\right)^2}{\sum_{i=1}^{n} \tau_{ik}^{(s)}}
$$

$$
= \frac{\not{\sum_{i=1}^{n} \tau_{ik}^{(s)}}}{\sum_{i=1}^{n} \tau_{ik}^{(s)}} \left\| \sqrt{W_k^{(s+1)}} (y - \beta_{k0}^{(s+1)} 1_n - X \eta_k^{(s+1)}) \right\|_2^2 / \text{trace}(W_k^{(q)})
$$

(16)
Example
3) FunME by regularizing functional derivatives

- For FunME-LASSO regularization described previously, there is no actually reason that the functions $\beta(.)$ and $\alpha(.)$ be sparse.
- So regularizing the parameter vectors representing these functions has no obvious interpretability.

→ FLiRTI methodology [James et al., 2009] offers an interpretable and sparse fit for functional linear regression.

- Regularization is performed on the the derivatives of the coefficient function, rather than on the parameters of the function.
- We rely on FLiRTI methodology for the regression functions $\beta_{zi}(t)$ (and $\alpha_{zi}(t)$)

FLiRTI: determine whether the $d$th derivative of $\beta_{zi}(t)$ is zero or not at each point $t_j$.

→ can produce a highly interpretable estimate for $\beta_{zi}(t)$ curves:
- $\beta_{zi}^{(0)}(t) = 0$ implies that $X(t)$ has no effect on $Y$ at $t$
- $\beta_{zi}^{(1)}(t) = 0$ means that $\beta_{zi}(t)$ is constant at $t$,
- $\beta_{zi}^{(0)}(t) = 1$ shows that $\beta_{zi}(t)$ is a linear function of $t$, etc.
Let $D^d$ be the $d$th finite difference operator defined recursively as

$D^1 b(t_j) = p[b(t_j) - b(t_{j-1})],$

$D^2 b(t_j) = D[Db(t_j)] = p^2[b(t_j) - 2b(t_{j-1}) + b(t_{j-2})],$

$D^d b(t_j) = D[D^{d-1} b(t_j)].$

$D^d b(t_j)$ is an approximation for $b^{(d)}(t_j) = [b_1^{(d)}(t_j), \ldots, b_p^{(d)}(t_j)]^\top$

$A_p = [D^d b(t_1), D^d b(t_2), \ldots, D^d b(t_p)]^\top$ (the approximate derivative matrix)

Let $\gamma_{z_i} = A_p \eta_{z_i}$

If $\beta_{z_i}^{(d)}(t) = 0$ over a large regions of $t$ for some $d$, then $\gamma_{z_i}$ is sparse. Then $\gamma_{z_i} = [\gamma_{z_i,1}, \ldots, \gamma_{z_i,p}]^\top$ provides a sparse estimate for $[\beta_{z_i}^{(d)}(t_1), \ldots, \beta_{z_i}^{(d)}(t_p)]^\top$.

**FLiRTI for the expert’ network of FunME**

$Y_i = \beta_{z_i,0} + \eta_{z_i}^\top x_i + \varepsilon_i^* = \beta_{z_i,0} + (A_p^{-1} \gamma_{z_i})^\top x_i + \varepsilon_i^*$

$= \beta_{z_i,0} + (x_i^\top A_p^{-1}) \gamma_{z_i} + \varepsilon_i^*$

$= \beta_{z_i,0} + v_i^\top \gamma_{z_i} + \varepsilon_i^*.$

and we now have $\theta_k = (\beta_k,0, \gamma_k^\top, \sigma_k^2)\top$ parameter vector of expert density $k$
FLiRTI for the gating network of FunME

- Let $\omega_k = A_q \zeta_k$ where $A_q = [D^d b(t_1), D^d b(t_2), \ldots, D^d b(t_q)]^\top$
  $\implies$ we get $\zeta_k = A_q^{-1} \omega_k$.

The gating network probabilities become

$$
\pi_k(v_i; w) = \frac{\exp \{\alpha_{k,0} + \zeta_k^\top r_i\}}{1 + \sum_{k'=1}^{K-1} \exp \{\alpha_{k',0} + \zeta_{k'}^\top r_i\}} = \frac{\exp \{\alpha_{k,0} + v_i^\top \omega_k\}}{1 + \sum_{k'=1}^{K-1} \exp \{\alpha_{k',0} + v_i^\top \omega_{k'}\}}
$$

(17)

with $v_i = r_i^\top A_q^{-1}$ is the new predictor and the new gating network parameter vector $w = ((\alpha_{1,0}, \omega_1^\top), \ldots, (\alpha_{K-1,0}, \omega_{K-1}^\top))^\top$ and $(\alpha_{K-1,0}, \omega_K^\top)^\top$ is a null vector.

The resulting FunME distribution and parameter estimation

$$f(y_i|u_i(.); \Psi) = \sum_{k=1}^{K} \pi_k(v_i; w) \phi(y_i; \beta_{k,0} + \gamma_{k}^\top v_i, \sigma_{k}^{*2})$$

(18)

where $\Psi = (w^T, \Psi_1^T, \ldots, \Psi_K^T)^T$ the unknown parameter vector of the model

$\implies$ Apply the EM-Lasso algorithm developed previously with :
- Predictors : $v_i = x_i^\top A_p^{-1}$ and $v_i = r_i^\top A_q^{-1}$
- Regularization : on $\omega$'s and $\gamma$'s : $\text{Pen}_{\lambda,\chi}(\Psi) = \lambda \sum_{k=1}^{K} \|\gamma_k\|_1 + \chi \sum_{k=1}^{K-1} \|\omega_k\|_1$
Example: Tecator data
Example: Phonemes data \((K=5), \ d=0\)
Example : Phonemes data ($K=5$), $d=1$
Example: Phonemes data ($K=5$), $d=2$
FunME for unobserved predictors

The functional predictors $X_i(t)$ are in general unobserved directly.

\[ X_{i1}(t), X_{i2}(t), \ldots, X_{i9}(t) \]

**Figure** – functional predictors $X_{ij}(t)$ $t \in T$
FunME for unobserved predictors

We rather observe $U_i(t)$ a noisy version of $X_i(t)$

Figure – Noisy functional predictors $U_{ij}(t)$ $t \in \mathcal{T}$
Until now the functional predictors $X_i(t)$ are represented by basis expansion as

\[ X_i(t) = \sum_{j=1}^{r} x_{ij} b_j(t) = \mathbf{x}_i^\top \mathbf{b}_r(t), \]

the coefficients $x_{ij} = \int_{\mathcal{T}} X_i(t) b_j(t) dt$ are unknown since $X_i(t)$ is not observed.

We first model $U_i(t)$ (for a single variable) as

\[ U_i(t) = X_i(t) + \delta_i(t), \quad i = 1, \ldots, n, \quad \delta_i \sim \mathcal{N}(0, \sigma_\delta^2) \]

We assume that the $\delta_i$'s are independent of the $X_i(\cdot)$'s and the $Y_i$'s.

and propose an unbiased estimator of $x_{ij}$ from $U_i(t)$ defined as

\[ \hat{x}_{ij} := \int_{\mathcal{T}} U_i(t) b_j(t) dt. \]

Indeed, we have $\mathbb{E}(\hat{x}_{ij}) = \int_{\mathcal{T}} \mathbb{E}(U_i(t)) b_j(t) dt = \int_{\mathcal{T}} X_i(t) b_j(t) dt = x_{ij}$.

Thus, an estimate $\hat{X}_i(t)$ of $X_i(t)$ can be given as

\[ \hat{X}_i(t) = \hat{\mathbf{x}}_i^\top \mathbf{b}_r(t), \quad i = 1, \ldots, n, \quad (19) \]

with $\hat{\mathbf{x}}_i = (\hat{x}_{i1}, \ldots, \hat{x}_{ir})^\top$.

The previous models/algorithms apply by replacing $\mathbf{x}_i$ by its estimate $\hat{\mathbf{x}}_i$. 
**FunME for classification**

$Y \in [G]$ represents the known group label of the functional predictor $X(\cdot)$.

- **Expert modeling**: functional multinomial logistic distribution

$$
\mathbb{P}(y_i|X_i(\cdot), Z_i = k; \beta) = \prod_{g=1}^{G} \left[ \frac{\exp \left\{ \beta_{kg,0} + \int_T X(t) \beta_{kg}(t)dt \right\}}{1 + \sum_{g'=1}^{G-1} \exp \left\{ \beta_{kg',0} + \int_T X(t) \beta_{kg'}(t)dt \right\}} \right] \mathbb{I}(y_i = g)
$$

$\mapsto$ use the same basis representation for the linear predictors $\beta_{kg,0} + \int_T X(t) \beta_{kg}(t)dt$

- **M-Step**: Newton-Raphson with coordinate ascent

$$
\theta_k^{(t+1)} = \theta_k^{(t)} - \left( \frac{\partial^2 Q_\lambda(\theta_k; \Psi^{(s)})}{\partial \theta_k \partial \theta_k^T} (\theta_k^{(t)}) \right)^{-1} \frac{\partial Q_\lambda(\theta_k; \Psi^{(s)})}{\partial \theta_k} (\theta_k^{(t)})
$$

with $\theta_k = (\theta_{k,1}^T, \ldots, \theta_{k,G-1}^T)$ with $\theta_{k,g} = (\beta_{kg,0}, \eta_{kg}^T)^T \in \mathbb{R}^{p+1}$ for $g \in [G-1]$, be the unknown parameter vector of expert distribution $k$ to be estimated.

- **Bayes (Maximum A Posteriori) rule**:

$$
\hat{y} = \arg \max_{1 \leq y \leq G} \mathbb{P}(Y = y|u; \Psi) = \arg \max_{y=1}^{G} \sum_{k=1}^{K} \pi_k(r; \xi)p(y|x; \theta_k)
$$
Concluding remarks

- A model for heterogeneous data with functional predictors
- The model inference can be performed by the EM algorithm
- Allows to perform feature selection
- Relying on FLiRTI methodology allows to keep the feature selection interpretable
  - Ongoing:
    - BIC-based procedure for model selection
    - Numerical experiments
    - Package (currently codes are written in Matlab and will be made public soon)
    - Extension to the multivariate setting
    - Extension to the case of functional predictors and functional responses


Thank you for your attention!