Model-based (co-)clustering in some high-dimensional scenarios

Faicel Chamroukhi

Working Group on Model-Based Clustering summer session
Ann Arbor, July 15-21, 2018
Outline

1. Model-Based Co-Clustering of Multivariate Functional Data
   Joint work with Christophe Biernacki, INRIA-Lille

2. Regularized Mixture-of-Experts for high-dimensional data
   Joint work with Bao Tuyen Huynh, Unicaen, LMNO
Outline

1 Model-Based Co-Clustering of Multivariate Functional Data
   - Motivation
   - Model-based co-clustering
   - Temporal curve segmentation (RHLP)
   - Model-based co-clustering embedding RHLP
   - Conclusion and perspectives

2 Regularized Mixture-of-Experts for high-dimensional data
Functional data are increasingly frequent

[James and Hastie, 2001; James and Sugar, 2003]
[Ramsay and Silverman, 2005]
[Chamroukhi et al., 2010]
[Bouveyron and Jacques, 2011]
[Samé et al., 2011]
[Jacques and Preda, 2014]
[Bouveyron et al., 2018]
[Chamroukhi and Nguyen, 2018]
Clustering of functional data

→ a growing investigation of Model-Based Clustering (MBC) for functional data

Some Reviews on MBC for functional data: [Jacques and Preda, 2014; Chamroukhi and Nguyen, 2018]
Clustering of functional data

→ a growing investigation of Model-Based Clustering (MBC) for functional data

Some Reviews on MBC for functional data: [Jacques and Preda, 2014; Chamroukhi and Nguyen, 2018]

Tecator data set\(^1\): \(n = 240\) spectra with \(m = 100\)

Figure: Original data and clustering results from Chamroukhi [2016b] for the data considered in the same setting as in Hébrail et al. [2010] (six clusters, each cluster is approximated by five linear segments \((R = 5, p = 1)\))

\^[1]\(\text{Tecator data are available at http://lib.stat.cmu.edu/datasets/tecator.}\)
Clustering of functional data

Topex/Poseidon satellite data\(^2\): \(n = 472\) waveforms of \(m = 70\) measured echoes

Figure: Original data and clustering results from Chamroukhi [2016b] with the same setting as in Hébrail et al. [2010]: twenty clusters and a piecewise linear approximation of four segments.

Clustering of functional data

Phonemes data set\(^3\): \(n = 1000\) log-periodograms for \(m = 150\) frequencies

Figure: Original data and clustering results from Chamroukhi [2016b]

\(^3\) 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 \( m \approx 510 \) observations
\( K = 2 \) clusters: operating state without/with possible defect
Outline

1 Model-Based Co-Clustering of Multivariate Functional Data
   - Motivation
   - Model-based co-clustering
   - Temporal curve segmentation (RHLP)
   - Model-based co-clustering embedding RHLP
   - Conclusion and perspectives

2 Regularized Mixture-of-Experts for high-dimensional data
This talk: Multivariate functional data clustering

- Multivariate functional data are increasingly present
- e.g: Data continuously recorded for different subjects from multiple subject’ sensors
  Measurements collected from different network elements (transceivers, cells, sites...):

![Data and Zoom](image)

Figure: An example with $d = 30$ and $n = 20$ daily observations [Ben Slimen et al., 2016].
This talk

Questioning

Clustering of highly multivariate functional data with two guidelines:

- (1) Mathematical guideline: warranty for estimation and selection
- (2) User guideline: keep a user-friendly meaning of the process

Both are important because clustering is a highly risky task…

Proposed answering

(1) Model-based co-clustering with (2) temporal curve segmentation

Novelty corresponds to combining both (1) and (2)
Difference between clustering and co-clustering

- Simultaneous clustering of lines/indiv. ($Z$) and columns/var. ($W$)
- Can be used as a way to reduce dimensionality (var. $\rightarrow W$)

Figure: Binary data set with $n = 500$, $d = 300$, $K = M = 3$
Latent block model for co-clustering

The Latent Block Model [Govaert and Nadif, 2013]

\[
f(X; \Psi) = \sum_{(z,w) \in \mathcal{Z} \times \mathcal{W}} \mathbb{P}(Z, W; \pi, \rho) f(X|Z, W; \theta)
\]

Hypotheses

- The latent variables \( Z \) and \( W \) are independent: \( \mathbb{P}(Z, W) = \mathbb{P}(Z)\mathbb{P}(W) \) and iid:
  \[
  \mathbb{P}(Z) = \prod_i \mathbb{P}(z_i) \text{ with } z_i \sim \text{Multinomial}(\pi_1, \ldots, \pi_K) \text{ where } \pi_k = \mathbb{P}(z_k = k)
  \]
  \[
  \mathbb{P}(W) = \prod_j \mathbb{P}(w_j) \text{ with } w_j \sim \text{Multinomial}(\rho_1, \ldots, \rho_M) \text{ where } \rho_\ell = \mathbb{P}(w_j = \ell)
  \]
- Conditional independence: \( x_{ij}|(z_i, w_j) \perp x_{i'j'}|(z_{i'}, w_{j'}) \)
Latent block model for co-clustering

The Latent Block Model [Govaert and Nadif, 2013]

\[
f(X; \Psi) = \sum_{(z,w) \in Z \times W} P(Z, W; \pi, \rho) f(X | Z, W; \theta)
\]

Hypotheses

- The latent variables \( Z \) and \( W \) are independent: \( P(Z, W) = P(Z)P(W) \) and iid:
  \[
P(Z) = \prod_i P(z_i) \text{ with } z_i \sim \text{Multinomial}(\pi_1, \ldots, \pi_K) \text{ where } \pi_k = P(z_k = k)
\]
  \[
P(W) = \prod_j P(w_j) \text{ with } w_j \sim \text{Multinomial}(\rho_1, \ldots, \rho_M) \text{ where } \rho_\ell = P(w_j = \ell)
\]
- Conditional independence: \( x_{ij} | (z_i, w_j) \perp x_{i'j'} | (z_{i'}, w_{j'}) \)

\[\leftarrow\] binary data: binary [Govaert and Nadif, 2003, 2008; Keribin et al., 2012],
\[\leftarrow\] categorical data: multinomial [Keribin et al., 2014]
\[\leftarrow\] continuous data: Gaussian [Lomet, 2012; Govaert and Nadif, 2013]
\[\leftarrow\] functional data: functional PCA + Gaussian, see further [Ben Slimen et al., 2016]
Inference for the latent block model

Inference of the latent block model

- variational block EM (VBEM) for maximum likelihood estimation and fuzzy co-clustering [Govaert and Nadif, 2006, 2008].


- Bayesian inference [Keribin et al., 2012, 2014]: Bayesian latent block mixtures for binary data and categorical data & a variational Bayesian inference and Gibbs sampling.

- Number of blocks estimation: ICL criterion [Lomet, 2012; Keribin et al., 2014]
Package blockcluster on the cloud

massiccc.lille.inria.fr

Massive Clustering with Cloud Computing
Clustering of heterogeneous data with missing values.
Hosted in the cloud. No installation or configuration required.
Upload your data, and get results straight away.

How MASSICCC Platform works
MASSICCC Platform lets you upload your data and select from a range of analysis tools to extract meaningful information. No need to install any software or configure some tools. You’ll see results straight away.

- **Upload your data securely**
  Upload all or part of your data to MASSICCC Platform and run multiple algorithms on them. You will have full control on the data you upload and only you will be able to access them.

- **Focus on the data**
  Minimum configuration is required to use our algorithms since we'll select the most sensible options by default based on the data you provide. No scientific background is required to start working and get results. Advanced configuration options are available if you need specific functions.
Functional data notation

- Data: (discretized) values of underlying smooth functions, not just vectors
- Data: A sample of $n$ heterogeneous univariate curves $(x_1, y_1), \ldots, (x_n, y_n)$
- $(x_i, y_i)$ consists of $m_i$ observations $y_i = (y_{i1}, \ldots, y_{imi})$ observed at the independent covariates, (e.g., time $t$ in time series), $(x_{i1}, \ldots, x_{imi})$
Functional data modeling: “classical” approach

[Ramsay and Silverman, 2005] and many others

- Step 1: \((x, y)\) decomposed into a finite basis of function (B-spline...) : \(Y_i(t) \approx \sum_{r=1}^{d} c_{ir} \phi_r(x_i(t))\) with \(c\) estimated by OLS
- Step 2: functional principal components analysis (PCA) which is performed as a usual PCA of the basis expansion coefficients \(c\) using a metric defined by the inner products between the basis functions
- Step 3: set a probability distribution on \(c\), typically Gaussian

It defines a distribution on \(c\) instead of \(y\)…
Functional data modeling: regression RHLP

Alternatively, use a segmentation via generative piecewise polynomial regression modeling of \( f(y|x) \) [Chamroukhi et al.]

\[ \rightarrow \text{Regression with Hidden Logistic Process (RHLP)} \]
\[ \rightarrow \text{See formula later} \]

It gives a distribution on \( y \) and also a meaningful segmentation of the curve.
RHL for modeling different types of functions

Original Impedance spectrum
Approximated Impedance spectrum

Faicel Chamroukhi
Model-based (co-)clustering in some high-dimensional scenarios
Package mixtcomp on the cloud

massiccc.lille.inria.fr

Massive Clustering with Cloud Computing

Clustering of heterogeneous data with missing values. Hosted in the cloud. No installation or configuration required. Upload your data, and get results straight away.

How MASSICCC Platform works

MASSICCC Platform lets you upload your data and select from a range of analysis tools to extract meaningful information. No need to install any software or configure some tools. You'll see results straight away.

- **Upload your data securely**
  Upload all or part of your data to MASSICCC Platform and run multiple algorithms on them. You will have full control on the data you upload and only you will be able to access them.

- **Focus on the data**
  Minimum configuration is required to use our algorithms since we'll select the most sensible options by default based on the data you provide. No scientific background is required to start working and get results. Advanced configuration options are available if you need specific functions.
Multivariate functional data co-clustering

[Chamroukhi and Biernacki, 2017]

- Data: $\mathbf{Y} = (y_{ij})$ a data sample matrix of $n$ individuals defined on a set $I$ and $d$ continuous functional variables defined on a set $J$.

- Each variable $y_{ij}$ is an univariate curve $y_{ij} = (y_{ij}(t_1), \ldots, y_{ij}(t_{T_{ij}}))$ of $T_{ij}$ observations $y(t) \in \mathbb{R}$ linked to covariates $\mathbf{x}_{ij} = (x_{ij}(t_1), \ldots, x_{ij}(t_{T_{ij}}))$ at the points $(t_1, \ldots, t_{T_{ij}})$, typically a sampling time.
Embedding RHLP in co-clustering

[Chamroukhi and Biernacki, 2017]

- Functional Latent Block Model for Co-clustering:

\[
f(Y|X; \Psi) = \sum_{(z,w) \in \mathcal{Z} \times \mathcal{W}} \mathbb{P}(Z; \pi) \mathbb{P}(W; \rho) f(Y|X, Z, W; \theta)
= \sum_{(z,w) \in \mathcal{Z} \times \mathcal{W}} \prod_{i,k} \pi_{zk} \prod_{j,\ell} \rho_{\ell}^{w_{j\ell}} \prod_{i,j,k,\ell} f(y_{ij}|x_{ij}; \theta_{k\ell})^{z_{ik}w_{j\ell}}. \]

with parameter vector \( \Psi = (\pi^T, \rho^T, \theta^T)^T \), where \( \pi = (\pi_1, \ldots, \pi_K)^T \), \( \rho = (\rho_1, \ldots, \rho_M)^T \), and \( \theta = (\theta_{11}^T, \ldots, \theta_{k\ell}^T, \ldots, \theta_{K\ell}^T)^T \).
Embedding RHLP in co-clustering

- **RHLP** [Chamroukhi et al., 2009]: model the conditional data distribution for each block $k\ell$, assuming that each functional variable $y_{ij}$ is governed by an $S_{k\ell}$-state hidden process of $y_{ij}$:

$$f(y_{ij} | x_{ij}; \theta_{k\ell}) = \prod_{t=1}^{T_{ij}} \sum_{r=1}^{S_{k\ell}} \alpha_{k\ell r}(t; \xi_{k\ell}) \mathcal{N}(y_{ij}(t); \beta_{k\ell r}^T x_{ij}(t), \sigma_{k\ell r}^2)$$

where the dynamical weights $\alpha'$s are given by the multinomial logistic:

$$\alpha_{k\ell r}(t; \xi_{k\ell}) = \frac{\exp (\xi_{k\ell 0} + \xi_{k\ell 1} t)}{1 + \sum_{r'=1}^{S_{k\ell}-1} \exp (\xi_{k\ell 0} + \xi_{k\ell 1} t)}.$$

Can be seen as a generative piecewise polynomial regression model where the transition points are smoothly controlled by logistic weights

$\leftrightarrow$ a particular mixture-of-experts model [Jacobs et al., 1991; Jordan and Jacobs, 1994]/(parametric) mixture of regressions with predictor-dependent mixing proportions [Young and Hunter, 2010]
Block mean curve approximation and segmentation

- **Approximation**: a prototype mean curve

\[
y_t | (z_i, w_j) \approx \hat{y}_t = \mathbb{E}[Y(t) | z_i, w_j, x(t); \Psi] = \sum_{s=1}^{S_{kl}} \alpha_{k\ell r}(t; \hat{\xi}_{k\ell}) \hat{\beta}_{k\ell r}^T x_i(t)
\]

\[\hookrightarrow\text{A smooth and flexible approximation thanks to the logistic weights}\]

- **Curve segmentation**:

\[
\hat{h}_t | (z_i, w_j) = \arg\max_{1 \leq s \leq S_{kl}} \mathbb{E}[H_t | z_i, w_j, x_{ij}(t); \hat{\xi}] = \arg\max_{1 \leq k \leq K} \alpha_{k\ell r}(t; \hat{\xi}_{k\ell})
\]
The complete-data log-likelihood:

\[
\log L_c(\Psi) = \log f(Y, Z, W, H|X; \Psi) \\
= \sum_{i,k} z_{ik} \log \pi_k + \sum_{j,\ell} w_{j\ell} \log \rho_{\ell} \\
+ \sum_{i,j,k,\ell,t,r} z_{ik} w_{j\ell} h_{tr} \log \left[ \alpha_{k\ell r}(t; \xi_{k\ell}) \mathcal{N} \left( y_{ij}(t); \beta_{k\ell r}^T x_{ij}(t), \sigma_{k\ell r}^2 \right) \right]
\]

where \((h_{tr}; t = 1, \ldots, T_{ij}, r = 1, \ldots, S_{k\ell})\) is a binary variable indicating from which state the observation \(y_{ij}(t)\) within the block cluster \(k\ell\) is originated.
The E-Step computes the expected complete-data log-likelihood, given the observed curves \((X, Y)\), and the current parameter estimation \(\Psi^{(q)}\)

\[
Q(\Psi, \Psi^{(q)}) = \mathbb{E} \left[ \log L_c(\Psi) \mid X, Y; \Psi^{(q)} \right] = \sum_{i,k} \mathbb{P}(z_{ik} = 1 \mid y_{ij}, x_{ij}) \log \pi_k + \sum_{j,\ell} \mathbb{P}(w_{j\ell} = 1 \mid y_{ij}, x_{ij}) \log \rho_{\ell} + \sum_{i,j,k,\ell,t,r} \mathbb{P}(z_{ik}w_{j\ell} = 1 \mid y_{ij}, x_{ij}) \mathbb{P}(h_{tr} = 1 \mid z_{ik}, w_{j\ell}, y_{ij}(t), x_{ij}(t)) \times \\
\log \left[ \alpha_{k\ell}(t; \xi_{k\ell}) \mathcal{N} \left( y_{ij}(t); \beta_{k\ell}^T x_{ij}(t), \sigma_{k\ell}^2 \right) \right]
\]

\(\mapsto\) Requires the calculation of the posterior joint distribution \(\mathbb{P}(z_{ik}w_{j\ell} = 1 \mid y_{ij}, x_{ij})\)

\(\mapsto\) does not factorize due to the conditional dependence on the observed curves of the row and the column labels


\(\mapsto\) We adopt this variational approximation in our context
Variational block EM algorithm

\[ P(z_{ik} w_{j\ell} = 1 | y_{ij}, x_{ij}) \approx P(z_{ik} = 1 | y_{ij}, x_{ij}) \times P(w_{j\ell} = 1 | y_{ij}, x_{ij}) \]
Variational block EM algorithm

\[ P(z_{ik}w_{j\ell} = 1 | y_{ij}, x_{ij}) \approx P(z_{ik} = 1 | y_{ij}, x_{ij}) \times P(w_{j\ell} = 1 | y_{ij}, x_{ij}) \]

**Initialization:** start from an initial solution at iteration \( q = 0 \), and then alternate at the \((q + 1)\)th iteration between the following variational E- and M- steps until convergence:

**VE Step** Estimate the variational approximated posterior memberships:

1. \( \tilde{z}^{(q+1)}_{ik} \propto \pi_k^{(q)} \exp \left( \sum_{j,\ell,t,r} \tilde{w}^{(q)}_{j\ell} \tilde{h}^{(q)}_{tr} \log \left[ \alpha_{k\ell r}(t; \xi^{(q)}_{k\ell}) \mathcal{N} \left( y_{ij}(t); \beta^{T(q)}_{k\ell r} x_{ij}(t), \sigma^{(q)^2}_{k\ell r} \right) \right] \right) \)

2. \( \tilde{w}^{(q+1)}_{j\ell} \propto \rho_{\ell}^{(q)} \exp \left( \sum_{i,k,t,r} \tilde{z}^{(q)}_{ik} \tilde{h}^{(q)}_{tr} \log \left[ \alpha_{k\ell r}(t; \xi^{(q)}_{k\ell}) \mathcal{N} \left( y_{ij}(t); \beta^{T(q)}_{k\ell r} x_{ij}(t), \sigma^{(q)^2}_{k\ell r} \right) \right] \right) \)

3. \( \tilde{h}^{(q+1)}_{tr} \propto \alpha_{k\ell r}^{(q)}(t; \xi^{(q)}_{k\ell}) \mathcal{N} \left( y_{ij}(t); \beta^{(q)^T}_{k\ell r} x_{ij}(t), \sigma^{(q)^2}_{k\ell r} \right) \)

where:

- \( \tilde{z}_{ik} = P(z_{ik} = 1 | y_{ij}, x_{ij}) \),
- \( \tilde{w}_{j\ell} = P(w_{j\ell} = 1 | y_{ij}, x_{ij}) \),
- \( \tilde{h}_{tr} = P(h_{tr} = 1 | z_i, w_j, y_{ij}(t), x_{ij}(t)) \)
Variational block EM algorithm

**M Step** update the parameters estimates $\theta^{(q+1)}$ given the estimated posterior memberships at the current iteration $q + 1$:

1. $\pi_k^{(q+1)} = \frac{\sum_i \tilde{z}_{ik}^{(q+1)}}{n}$

2. $\rho_{\ell}^{(q+1)} = \frac{\sum_j \tilde{w}_{j\ell}^{(q+1)}}{d}$
Variational block EM algorithm

**M Step** update the parameters estimates $\theta^{(q+1)}$ given the estimated posterior memberships at the current iteration $q + 1$:

1. $\pi_{k}^{(q+1)} = \frac{\sum_i \tilde{z}_{ik}^{(q+1)}}{n}$

2. $\rho_{\ell}^{(q+1)} = \frac{\sum_j \tilde{w}_{j\ell}^{(q+1)}}{d}$

The update of each block parameters $\theta_{k\ell}$ consists in a weighted version of the RHLP updating rules:

3. $\xi_{k\ell}^{(new)} = \xi_{k\ell}^{(old)} - \left[ \frac{\partial^2 F(\xi_{k\ell})}{\partial \xi_{k\ell} \partial \xi_{k\ell}^T} \right]^{-1} \left. \frac{\partial F(\xi_{k\ell})}{\partial \xi_{k\ell}} \right|_{\xi_{k\ell} = \xi_{k\ell}^{(old)}} \frac{\partial F(\xi_{k\ell})}{\partial \xi_{k\ell}} |_{\xi_{k\ell} = \xi_{k\ell}^{(old)}}$ which is the IRLS maximisation of $F(\xi_{k\ell}) = \sum_{i,j,t} \tilde{z}_{ik}^{(q)} \tilde{w}_{j\ell}^{(q)} \tilde{h}_{tr}^{(q)} \log \alpha_{k\ell r}(t; \xi_{k\ell})$ w.r.t $\xi_{k\ell}$. 

Faicel Chamroukhi
Model-based (co-)clustering in some high-dimensional scenarios 29/58
**Variational block EM algorithm**

**M Step** update the parameters estimates $\theta^{(q+1)}$ given the estimated posterior memberships at the current iteration $q + 1$:

1. $$\pi_k^{(q+1)} = \frac{\Sigma_i \tilde{z}_{ik}^{(q+1)}}{n}$$
2. $$\rho_{\ell}^{(q+1)} = \frac{\Sigma_j \tilde{w}_{j\ell}^{(q+1)}}{d}$$

The update of each block parameters $\theta_{k\ell}$ consists in a weighted version of the RHLP updating rules:

3. $$\xi_{k\ell}^{(new)} = \xi_{k\ell}^{(old)} - \left[ \frac{\partial^2 F(\xi_{k\ell})}{\partial \xi_{k\ell} \partial \xi_{k\ell}^T} \right]^{-1} \frac{\partial F(\xi_{k\ell})}{\partial \xi_{k\ell}} \bigg|_{\xi_{k\ell} = \xi_{k\ell}^{(old)}}$$
   which is the IRLS maximisation of
   $$F(\xi_{k\ell}) = \Sigma_{i,j,t} \tilde{z}_{ik}^{(q)} \tilde{w}_{j\ell}^{(q)} \tilde{h}_{tr}^{(q)} \log \alpha_{k\ell r}(t; \xi_{k\ell}) \text{ w.r.t } \xi_{k\ell}.$$  

The regression parameters updates consist in analytic WLS problems:

4. $$\beta_{k\ell r}^{(q+1)} = \left[ \Sigma_{i,j} \tilde{z}_{ik}^{(q)} \tilde{w}_{j\ell}^{(q)} X_{ij}^T \Lambda_{ijkr}^{(q)} X_{ij} \right]^{-1} \Sigma_{i,j} \tilde{z}_{ik}^{(q)} \tilde{w}_{j\ell}^{(q)} X_{ij}^T \Lambda_{ijkr}^{(q)} y_{ij}$$
5. $$\sigma_{k\ell r}^{2(q+1)} = \frac{\Sigma_{i,j} \tilde{z}_{ik}^{(q)} \tilde{w}_{j\ell}^{(q)} \| \sqrt{\Lambda_{ijkr}^{(q)}} (y_{ij} - X_{ij} \beta_{k\ell r}^{(q+1)}) \|^2}{\Sigma_{i,j} \tilde{z}_{ik}^{(q)} \tilde{w}_{j\ell}^{(q)} \text{ trace}(\Lambda_{ijkr}^{(q)})}$$
   where $X_{ij}$ is the design matrix for the $i$th curve, $\Lambda_{ijkr}^{(q)}$ is the diagonal matrix whose diagonal elements are the posterior segment memberships $\{\tilde{h}_{ijtr}^{(q)}; t = 1, \ldots, T_{ij}\}$.  

Faicel Chamroukhi

Model-based (co-)clustering in some high-dimensional scenarios 29/58
It is also possible to use the Classification EM (CEM) approximation of EM [Celeux and Govaert, 1992].

**Parameter estimation by an SEM algorithm: SEM-FLBM**

- The SEM algorithm [Celeux and Diebolt, 1985] allows to overcome some drawbacks of the variational-EM algorithm, including its sensitivity to starting values; SEM does not use an approximation.
- Eg. SEM for latent block models for categorical data [Keribin et al., 2012, 2014]
- The formulas of VEM-FLBM and SEM-FLBM are essentially the same, except that we incorporate a stochastic step consisting of sampling binary indicator variables $z_{ik}$, $w_{j\ell}$ and $h_{tr}$ according to $\tilde{z}_{ik}$, $\tilde{w}_{j\ell}$ and $\tilde{h}_{tr}$. 
Conclusion and perspectives

Conclusion

- A full generative framework for the cluster analysis and segmentation of high-dimensional non-stationary functional data
- The model inference can be performed by a variational EM algorithm or SEM

Perspectives

- Numerical experiments
- Package
Outline

1. Model-Based Co-Clustering of Multivariate Functional Data

2. Regularized Mixture-of-Experts for high-dimensional data
   - Mixture-of-Experts (MoE) Modeling and MLE
   - Regularized MLE of the MoE
   - Proposed EM algorithm with block coordinate ascent
   - Experimental study
Context

- Heterogeneous regression data \((x, y) \rightarrow \text{underlying unknown partition } z\)
- Data issued from non-linear regression function \(f(y|x)\)

Modeling framework

Mixture-of-experts/(parametric) mixture of regressions with predictor-dependent mixing proportions:

\[
p(y_i|x_i) = \sum_{z_i} \mathbb{P}(z_i|x_i)p(y_i|x_i, z_i),
\]
Mixture-of-Experts (MoE) modeling framework

- Observed pairs of data \((x, y)\) where the response \(y \in \mathbb{R}\) for the predictors \(x \in \mathbb{R}^p\) governed by a hidden categorical random variable \(Z\)

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

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

- 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 \(\mu(x; \beta_k)\)

- Non-normal MoE, for data with atypical observations, and with possible heavy tailed and asymmetric distributions: [Chamroukhi, 2016a, 2017; Nguyen and Chamroukhi, 2018]

- Parameter vector \(\theta = (w^T, \theta_1^T, \ldots, \theta_K^T)^T\)
**Standard MLE of the MoE model**

- MLE: $\theta$ is commonly estimated by maximizing the observed-data log-likelihood:

$$\hat{\theta}_n \in \arg \max_{\theta \in \Theta} L(\theta)$$

with

$$L(\theta) = \ln f((x_1, y_1), \ldots, (x_n, y_1); \theta) = \sum_{i=1}^{n} \ln \sum_{k=1}^{K} \pi_k(x_i; w) f(y_i | x_i; \theta_k).$$

$\leftrightarrow$ the EM algorithm (Dempster et al. [1977])

$\hookrightarrow$ Consider a high-dimensional setting

Looking for a sparse models

$\text{Regularized MLE of the MoE}$

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

$$\hat{\theta}_n \in \arg \max_{\theta \in \Theta} PL(\theta)$$

with

$$PL(\theta) = L(\theta) - \text{Pen}(\theta)$$

$\text{Pen}(\theta)$ should encourage sparsity

parameter estimation and selection problem
Standard MLE of the MoE model

- MLE: $\theta$ is commonly estimated by maximizing the observed-data log-likelihood:

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

with

\[
L(\theta) = \ln f((x_1, y_1), \ldots, (x_n, y_1); \theta) = \sum_{i=1}^{n} \ln \sum_{k=1}^{K} \pi_k(x_i; w) f(y_i|x_i; \theta_k).
\]

$\hookrightarrow$ the EM algorithm (Dempster et al. [1977])

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

Regularized MLE of the MoE

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

\[
\hat{\theta}_n \in \arg \max_{\theta \in \Theta} PL(\theta)
\]

with $PL(\theta) = L(\theta) - \text{Pen}(\theta)$

- $\hookrightarrow$ Pen($\theta$) should encourage sparsity
- parameter estimation and selection problem
Proposed Regularized Mixture of Experts model

\[
\text{Pen}(\theta) = \sum_{k=1}^{K} \lambda_k \| \beta_k \|_1 + \sum_{k=1}^{K-1} \gamma_k \| w_k \|_1 + \frac{\rho}{2} \| w_k \|_2^2
\]

- Lasso penalty for the experts \(\rightarrow\) encourage a sparse solution
- The elastic net penalty (Zou and Hastie [2005]) for the gating network:
  - \(\rightarrow\) reduce the norm of the estimated values of the gating network parameters by using the \(L_2\) penalties;
  - \(\rightarrow\) the Lasso penalty to recover a sparse solution
- The convexity of \(L_1\) and \(L_2\) penalties have also advantageous numerical properties.
- If the correlation between the features is high, one can add \(L_2\) penalties for the expert network.
Regularized MLE via an EM algorithm

- The penalized log-likelihood function:

\[ PL(\theta) = L(\theta) - \sum_{k=1}^{K} \lambda_k \| \beta_k \|_1 - \sum_{k=1}^{K-1} \gamma_k \| w_k \|_1 - \frac{\rho}{2} \| w_k \|_2^2 \]

- The penalized complete-data log-likelihood function:

\[ PL_c(\theta) = L_c(\theta) - \sum_{k=1}^{K} \lambda_k \| \beta_k \|_1 - \sum_{k=1}^{K-1} \gamma_k \| w_k \|_1 - \frac{\rho}{2} \| w_k \|_2^2 \]

with

\[ L_c(\theta) = \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} \log [\pi_k (x_i; w) f(y_i | x_i; \theta_k)] \]

such that \( z_{ik} = 1 \) iff \( z_i = k \) (the data pair \((x_i, y_i)\) originates from expert \(k\)
Parameter estimation for RMoE

Khalili’s method [Khalili, 2010]:

- Approximates the $L_1$ penalty function in a some neighborhood by an $\varepsilon$-local quadratic function
  \[ \eta|t| \approx \eta|t_0| + \frac{\eta}{2(|t_0| + \varepsilon)} (t^2 - t_0^2). \]
  \(\leftrightarrow\) Almost surely none of the components will be exactly zero.

- Needs using a threshold to recover the zero coefficients
  \(\leftrightarrow\) The size of threshold affects the degree of sparsity of the solution.

- The Newton-Raphson algorithm is used to update the M-step of the EM algorithm.
  \(\leftrightarrow\) This approach still require computing the inverse matrix.

In our proposal:

- A block EM algorithm with coordinate ascent algorithm to estimate the parameters:
  \(\leftrightarrow\) Exact $L_1$ penalty regularization;
  \(\leftrightarrow\) Avoids computing matrix inversion;
  \(\leftrightarrow\) Avoids using a threshold to recover the zero coefficients.
Block EM algorithm with coordinate ascent

E-step

- Compute the conditional expectation of the penalized complete-data log-likelihood

\[ Q(\theta; \theta^{(q)}) = \mathbb{E} \left[ PL_c(\theta) | \mathcal{D}; \theta^{(q)} \right] \]

\[ = \sum_{i=1}^{n} \sum_{k=1}^{K} \tau_{ik}^{(q)} \log [\pi_k(x_i; w) f_k(y_i | x_i; \theta_k)] \]

\[ - \sum_{k=1}^{K} \lambda_k \| \beta_k \|_1 - \sum_{k=1}^{K-1} (\gamma_k \| w_k \|_1 - \frac{\rho}{2} \| w_k \|_2^2). \]

↩ Calculate the posterior component probabilities:

\[ \tau_{ik}^{(q)} = \mathbb{P}(Z_i = k | y_i, x_i; \theta^{(q)}) = \frac{\pi_k(x_i; w^{(q)}) N(y_i; \beta_k^{(q)} + x_i^T \beta_k^{(q)}, \sigma_k^{(q)2})}{\sum_{l=1}^{K} \pi_l(x_i; w^{(q)}) N(y_i; \beta_l^{(q)} + x_i^T \beta_l^{(q)}, \sigma_l^{(q)2})}. \]

↩ As in standard MoE
Block EM algorithm with coordinate ascent (cont.)

M-step

- Maximizing the $Q$ function: $\theta^{(q+1)} \in \arg\max_{\theta} Q(\theta; \theta^{(q)})$ with

$$Q(\theta; \theta^{(q)}) = Q(w; \theta^{(q)}) + Q(\beta, \sigma; \theta^{(q)}),$$

where

$$Q(w; \theta^{(q)}) = \sum_{i=1}^{n} \sum_{k=1}^{K} \tau_{ik}^{(q)} \log \pi_k(x_i; w) - \sum_{k=1}^{K-1} \left( \gamma_k \| w_k \|_1 - \frac{\rho}{2} \| w_k \|_2^2 \right), \quad (1)$$

$\leftrightarrow$ a weighted regularized multiclass logistic regression problem

and

$$Q(\beta, \sigma; \theta^{(q)}) = \sum_{k=1}^{K} \sum_{i=1}^{n} \tau_{ik}^{(q)} \log \mathcal{N}(y_i; \beta_{k0} + x_i^T \beta_k, \sigma_k^2) - \sum_{k=1}^{K} \lambda_k \| \beta_k \|_1 \quad (2)$$

$\leftrightarrow K$ independent weighted LASSO problems
Updating the gating network parameters

- Coordinate ascent algorithm to update $w$ Tseng [1988, 2001]
- $w_{kj}$ is updated by maximizing the component $(k, j)$ of (1) given by

$$Q(w_{kj}; \theta^{(q)}) = \begin{cases} 
F(w_{kj}; \theta^{(q)}) - \gamma_k w_{kj}, & \text{if } w_{kj} > 0 \ (F_1) \\
F(0; \theta^{(q)}) & \text{if } w_{kj} = 0 \\
F(w_{kj}; \theta^{(q)}) + \gamma_k w_{kj}, & \text{if } w_{kj} < 0 \ (F_2)
\end{cases}$$

$$F(w_{kj}; \theta^{(q)}) = \sum_{i=1}^{n} \tau_{ik}^{(q)} (w_{k0} + w_k^T x_i) - \sum_{i=1}^{n} \log \left( 1 + \sum_{l=1}^{K-1} e^{w_{l0} + w_l^T x_i} \right) - \frac{\rho}{2} w_{kj}^2. \ (3)$$

Univariate Newton-Raphson algorithm

- $F_1$ and $F_2$ are smooth univariate concave functions in $w_{kj}$. Univariate Newton-Raphson algorithm can be used to update $w_{kj}$

$$w_{kj}^{(s+1)} = w_{kj}^{(s)} - \left( \frac{\partial^2 F(w_{kj}; \theta^{(q)})}{\partial^2 w_{kj}} \right)^{-1} \left|_{w_{kj}^{(s)}} \left( \frac{\partial F(w_{kj}; \theta^{(q)})}{\partial w_{kj}} \right) - \gamma_k \text{sign}(w_{kj}) \right|_{w_{kj}^{(s)}},$$

where $\frac{\partial^2 F(w_{kj}; \theta^{(q)})}{\partial^2 w_{kj}}$ and $\frac{\partial F(w_{kj}; \theta^{(q)})}{\partial w_{kj}}$ have closed-form.
Updating the expert parameters

M-step (cont.)

- Update $\beta_{kj}$ using coordinate ascent algorithm with soft-thresholding operator

$$
\beta_{kj}^{[s+1]} = S_{\lambda_k, \sigma_k(q)^2} \frac{\sum_{i=1}^{n} \tau_{ik}^{(q)} r_{ikj} x_{ij}}{\sum_{i=1}^{n} \tau_{ik} x_{ij}},
$$

where $r_{ikj} = y_i - \beta_{kj}^{[s]} x_{ij}$, $[S_{\gamma}(u)]_j = \text{sign}(u_j)(|u_j| - \gamma)_+$ and $(x)_+ = \max\{x, 0\}$ in the $s$th loop of the coordinate ascent algorithm.

$$
\beta_{k0}^{[s+1]} = \frac{\sum_{i=1}^{n} \tau_{ik}^{(q)} (y_i - x_i^\top \beta_k^{[s+1]})}{\sum_{i=1}^{n} \tau_{ik}^{(q)}}.
$$
Updating the expert parameters

M-step (cont.)

- Update $\beta_{kj}$ using coordinate ascent algorithm with soft-thresholding operator

$$\beta_{kj}^{[s+1]} = S_k \sigma_k^{(q)2} \left( \sum_{i=1}^{n} \tau_{ik}^{(q)} r_{ikj}^{[s]} x_{ij} \right) / \sum_{i=1}^{n} \tau_{ik}^{(q)}x_{ij},$$

where $r_{ikj}^{[s]} = y_i - \beta_{k0}^{[s]} - \beta_{kj}^{[s]} T x_i + \beta_{kj}^{[s]} x_{ij}$, $[S_\gamma(u)]_j = \text{sign}(u_j)(|u_j| - \gamma)_+$ and $(x)_+ = \max\{x, 0\}$ in the $s$th loop of the coordinate ascent algorithm.

$$\beta_{k0}^{[s+1]} = \sum_{i=1}^{n} \tau_{ik}^{(q)} (y_i - x_i^\top \beta_{kj}^{[s+1]}) / \sum_{i=1}^{n} \tau_{ik}^{(q)}.$$

- Rerun the E-step, keep

$$(w_{k0}^{(q+2)}, w_k^{(q+2)}) = (w_{k0}^{(q+1)}, w_k^{(q+1)}); \quad (\beta_{k0}^{(q+2)}, \beta_{k}^{(q+2)}) = (\beta_{k0}^{(q+1)}, \beta_{k}^{(q+1)}),$$

and update $\sigma_k^{2(q+2)}$ as follows

$$\sigma_k^{2(q+2)} = \sum_{i=1}^{n} \tau_{ik}^{(q+1)} (y_i - \beta_{k0}^{(q+2)} - x_i^\top \beta_{k}^{(q+2)})^2 / \sum_{i=1}^{n} \tau_{ik}^{(q+1)}.$$
Simulation study

Simulation protocol

- $\mathbf{x} \sim \mathcal{N}(0; \Sigma)$ with $\text{corr}(x_{ij}, x_{ij'}) = 0.5|j-j'|$; $K = 2$
- Sample size: $n = 300$, 100 different data sets;
- The regression coefficients:
  
  $$(\beta_{10}, \beta_1)^T = (0, 0, 1.5, 0, 0, 0, 1)^T; \sigma_1 = 1$$
  $$(\beta_{20}, \beta_2)^T = (0, 1, -1.5, 0, 0, 2, 0)^T; \sigma_2 = 1$$
  $$(w_{10}, w_1)^T = (1, 2, 0, 0, -1, 0, 0)^T; \sigma_3 = 1$$

Considered approaches for comparison

- The standard MoE;
- MoE+$L_2$ (MoE with $L_2$ penalties in the gating network);
- MoE-BIC (MoE with model selection using BIC criterion - 100 submodels);
- MIXLASSO (MLR with Lasso penalties) (see Khalili and Chen [2007]);

Evaluation criteria

- The sensitivity/specificity (sparsity);
- The parameter estimation (density estimation);
- The misclassification error: Adjust rand index - ARI (clustering).
Sensitivity/specificity result

- **Sensitivity** ($S_1$): proportion of correctly estimated zero coefficients;
- **Specificity** ($S_2$): proportion of correctly estimated nonzero coefficients.

<table>
<thead>
<tr>
<th>Method</th>
<th>Expert 1</th>
<th>Expert 2</th>
<th>Gate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$S_1$</td>
<td>$S_2$</td>
<td>$S_1$</td>
</tr>
<tr>
<td>MoE</td>
<td>0.000</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>MoE+$L_2$</td>
<td>0.000</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>MoE-BIC</td>
<td>0.920</td>
<td>1.000</td>
<td>0.930</td>
</tr>
<tr>
<td>MIXLASSO</td>
<td>0.775</td>
<td>1.000</td>
<td>0.693</td>
</tr>
<tr>
<td><strong>Our MoE-Lasso+$L_2$</strong></td>
<td>0.700</td>
<td>1.000</td>
<td>0.803</td>
</tr>
</tbody>
</table>

Table: Sensitivity ($S_1$) and specificity ($S_2$) results.

- MoE and MoE+$L_2$ could not be considered as model selection methods since their sensitivity equal zero.
- MIXLASSO can detect the zero coefficients in the experts. However, this model has a poor result when clustering the data.
- The MoE-Lasso+$L_2$ model can detect the zero coefficients in the experts and the gating network.
Parameter estimation for expert 1

\[(\beta_{10}, \beta_1)^T = (0, 0, 1.5, 0, 0, 0, 1)^T.\]
Parameter estimation for expert 2

\( (\beta_{20}, \beta_{21})^T = (0, 1, -1.5, 0, 0, 2, 0)^T. \)
Parameter estimation for gating network

- \((w_{10}, w_1)^T = (1, 2, 0, 0, -1, 0, 0)^T\).
Result for data clustering

<table>
<thead>
<tr>
<th>Model</th>
<th>MoE</th>
<th>MoE+$L_2$</th>
<th>MoE-BIC</th>
<th>MoE-Lasso + $L_2$</th>
<th>MIXLASSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>C. rate</td>
<td>89.57%(\pm 1.65)%</td>
<td>89.62%(\pm 1.63)%</td>
<td>90.05%(\pm 1.65)%</td>
<td>89.46%(\pm 1.76)%</td>
<td>82.89%(\pm 1.92)%</td>
</tr>
</tbody>
</table>

Table: clustering accuracy results (correct classification rate and Adjusted Rand Index).

Remarks

- MoE-BIC provides the best results. However, it is hard to apply BIC in reality especially for high dimensional data, since this involves a huge collection of model candidates.
- MIXLASSO can detect zero coefficients in the experts, but it provides a poor result when clustering data.
- MoE-Lasso+$L_2$ can detect zero coefficients in the model and provide a competitive result with MoE, MoE-$L_2$ in term of clustering, although it also causes bias to the non-zero coefficients.
Applications to real data sets

For real data sets, we calculate the mean squared error between the response variable $Y$ with its prediction $\hat{Y}$, where

$$\hat{Y} = \sum_{k=1}^{K} \pi_k(x; \hat{w})(\hat{\beta}_k + x^T\hat{\beta}_k).$$

Housing data: 13 features, 506 observations, $K = 2$.

<table>
<thead>
<tr>
<th></th>
<th>MoE</th>
<th>MoE-Lasso + $L_2$ (Khalili)</th>
<th>MoE-Lasso + $L_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSE</td>
<td>0.1544($\pm$0.577)</td>
<td>0.2044($\pm$0.709)</td>
<td>0.1989($\pm$0.619)</td>
</tr>
</tbody>
</table>

Table: Results for Housing data set.

Baseball salary data: 32 features, 337 observations, $K = 2$.

<table>
<thead>
<tr>
<th></th>
<th>MoE</th>
<th>MoE-Lasso + $L_2$</th>
<th>MIXLASSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSE</td>
<td>0.2625($\pm$0.758)</td>
<td>0.2821($\pm$0.633)</td>
<td>1.1858($\pm$2.792)</td>
</tr>
</tbody>
</table>

Table: Results for Baseball salaries data set.
The proximal Newton method

- We recently improve the proposed algorithm by using the proximal Newton method (Lee et al. [2006], Lee et al. [2014] and Friedman et al. [2010]) for updating the gating network parameters.

- The idea of the proximal Newton method:
  - Approximate the smooth part of $Q(w; \theta^{(q)})$ with its local quadratic form;
  - Use coordinate ascent with soft-thresholding operator to solve the resulting approximated convex optimization problem;
  - Combine with backtracking line search to update $w$. 

Extension result for proximal Newton method

- Coordinate ascent algorithm (CA) VS proximal Newton (PN) method:

<table>
<thead>
<tr>
<th>Criteria</th>
<th>MoE-Lasso + $L_2$ (CA)</th>
<th>MoE-Lasso + $L_2$ (PN)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C.Rate</td>
<td>89.46%(\pm 1.76)%</td>
<td>89.53%(\pm 1.65)%</td>
</tr>
<tr>
<td>$PL(\theta)$ value</td>
<td>$-558.140_{(12.99)}$</td>
<td>$-558.410_{(13.03)}$</td>
</tr>
</tbody>
</table>

Table: Simulation results.

- Application of the proximal Newton algorithm to the residential building data set: 107 features, 372 observations, $K = 3$.

| Proximal Newton         | 0.0120_{(0.879)}         |

Table: Results for residential building data set.
Conclusion

- We propose a regularized MoE which does not require using approximations as in standard MoE regularization.

- A blockwise EM algorithm with coordinate ascent algorithm is proposed to monotonically maximize the RMoE objective function.

- The updating of the gating network for some situations is time consuming since we don’t have a closed-form.

- The algorithm has been improved by using proximal Newton method to update the gating network, which has a closed-form update for each parameter and improve the running time.

- Future work: Estimation and feature selection for hierarchical MoE and MoE with discrete data, ...
References I

Y. Ben Slimen, S. Allio, and J. Jacques. Model-Based Co-clustering for Functional Data. HAL preprint hal-01422756, December 2016. URL https://hal.inria.fr/hal-01422756.


References II


Thank you for your attention!