Unsupervised statistical learning with latent block models on dynamic discrete data

Graphical Models and Clustering Workshop

Giulia MARCHELLO

Equipe PreMeDICaL,
Inria

Joint work with C. Bouveyron & M. Corneli

May 16th 2024,
Imag Montpellier

Outline

Introduction

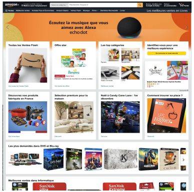
In many applications, statistical learning has to face new needs:

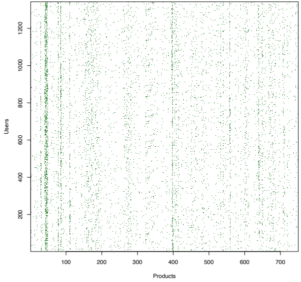
- High-dimensional data,
- Extracting insights from complex datasets,
- Existence of time-dependent patterns.

The challenges of high-dimensional data:

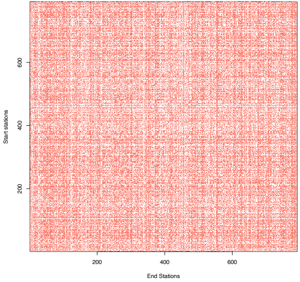
- Curse of dimensionality,
- Computational challenges,
- Sparsity.

Applications

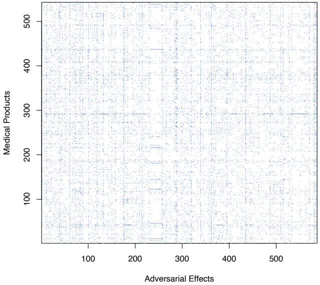




(a) Raw Amazon Fine Food dataset.



(b) Raw London Bike sharing dataset.



(c) Raw pharmacovigilance dataset.

Pharmacovigilance data

The problem in the pharmacovigilance context:

- The method currently used is incomplete,
- The signal detection process is not automated,
- Each Regional Center of Pharmacovigilance (RCPV) has to process a massive amount of data.

Pharmacovigilance data

The problem in the pharmacovigilance context:

- The method currently used is incomplete,
- The signal detection process is not automated,
- Each Regional Center of Pharmacovigilance (RCPV) has to process a massive amount of data.

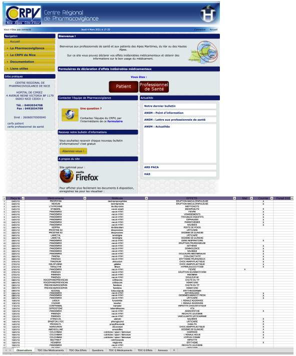
The missions of the RCPV of Nice:

- Detecting safety signals about drugs,
- Answering to questions of health professionals and patients about drugs,
- Promoting the proper use of the medical products.

Pharmacovigilance data

The problem in the pharmacovigilance context:

- The method currently used is incomplete,
- The signal detection process is not automated,
- Each Regional Center of Pharmacovigilance (RCPV) has to process a massive amount of data.



The missions of the RCPV of Nice:

- Detecting safety signals about drugs,
- Answering to questions of health professionals and patients about drugs,
- Promoting the proper use of the medical products.

Our goals:

- Provide useful summaries for medical authorities,
- Identifying possible unexpected phenomena.

The role of unsupervised learning

Various methods to address challenges of massive and high-dimensional data:

- Dimension reduction: data are represented within lower-dimensional subspaces,
- Clustering: grouping similar rows of a data matrix,
- Co-clustering: simultaneously clustering rows and columns of a matrix.

The role of unsupervised learning

Various methods to address challenges of massive and high-dimensional data:

- Dimension reduction: data are represented within lower-dimensional subspaces,
- Clustering: grouping similar rows of a data matrix,
- Co-clustering: simultaneously clustering rows and columns of a matrix.

	1	2	3	4	5	6	7
A	1	0	0	1	0	1	1
B	0	1	0	1	1	1	0
C	1	1	0	1	1	0	1
D	0	1	0	1	1	1	0
E	0	0	1	0	0	1	1
F	1	0	1	1	1	1	1
G	0	0	0	1	0	0	0
H	0	0	1	0	0	1	1
I	0	0	1	1	0	0	1
J	1	0	1	0	1	1	0
K	0	0	1	1	0	1	0
L	1	0	1	1	0	0	1
M	0	0	1	0	0	0	1
N	0	1	1	0	1	0	0
O	1	1	1	1	1	1	1

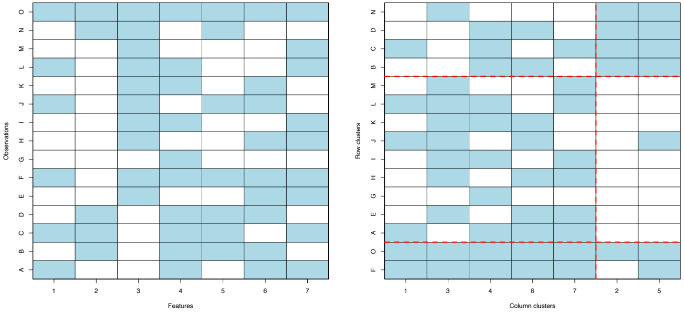


Figure: Incidence matrix and reorganized incidence matrix.

The pharmacovigilance data structure

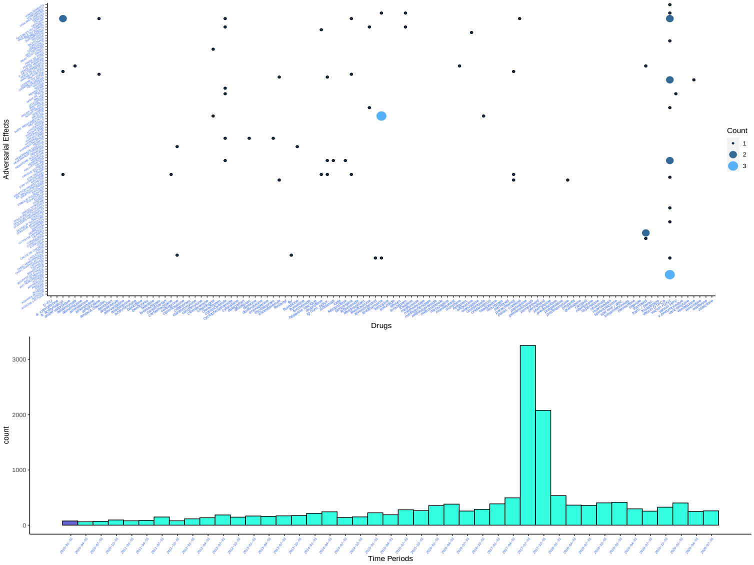


Figure: Evolution of spontaneous reports to RCPV from 2010 to 2020, a small sample is considered.

Time-dependent discrete data

Goals:

- Interpret massive streams of interaction data,
- Summarize dynamic datasets,
- Detect changes in cluster memberships,
- Sparsity modeling.

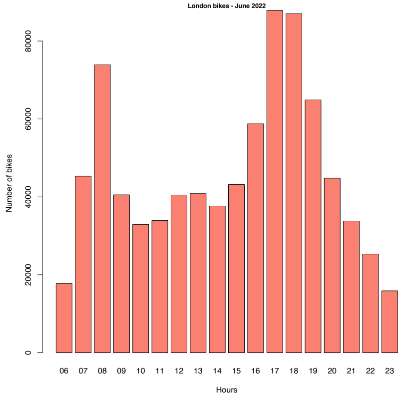


Figure: Histogram of London bikes data along a cumulative day.

Time-dependent discrete data

Goals:

- Interpret massive streams of interaction data,
- Summarize dynamic datasets,
- Detect changes in cluster memberships,
- Sparsity modeling.

Contributions:

- Dynamic co-clustering,
- Sparse dynamic co-clustering,
- Online sparse co-clustering.

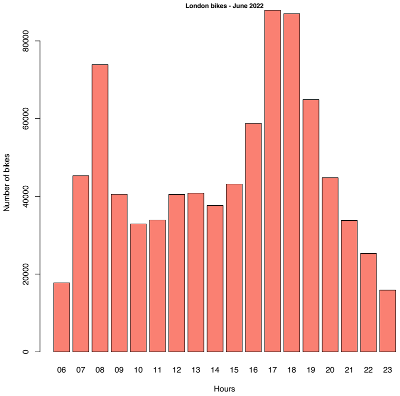


Figure: Histogram of London bikes data along a cumulative day.

Outline

The goal

- Composition of clusters changing along the time,
- Exploit systems of ODEs to model cluster membership over time,
- Enhance sparsity with mixtures of ZIP distributions.

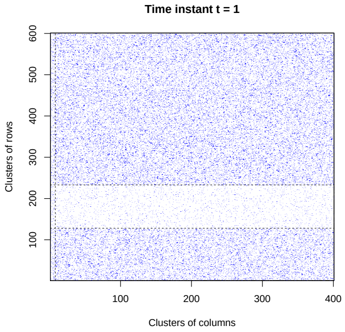


Figure: Example of dynamic co-clustering.

Data and Objectives

The data we consider are organized as follows:

- rows are indexed by $i = 1, \ldots N$;
- columns are indexed by $j = 1, \ldots, M$;
- time instants $t \in [0, T]$ during which N and M are fixed;
- the $N \times M \times T$ tensor $X := \{X_{ij}(t)\}$ contains the number of interactions between any observation and feature pair at any given t.

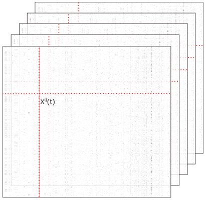


Figure: Data structure.

Data and Objectives

The data we consider are organized as follows:

- rows are indexed by $i = 1, \ldots N$;
- columns are indexed by $j = 1, \ldots, M$;
- time instants $t \in [0, T]$ during which N and M are fixed;
- the $N \times M \times T$ tensor $X := \{X_{ij}(t)\}$ contains the number of interactions between any observation and feature pair at any given t.

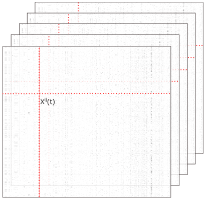


Figure: Data structure.

We aim at estimating:

- The latent variables for the clustering of rows and columns into Q and L groups,
- A latent variable for modeling the evolving sparsity of the data.

The Zip-dLBM

- Multinomial random variables to represent the membership to clusters:
  - $Z_i(t) \sim \mathcal{M}(1, \alpha(t) := (\alpha_1(t), \ldots, \alpha_Q(t)))$,
  - $W_j(t) \sim \mathcal{M}(1, \beta(t) := (\beta_1(t), \ldots, \beta_L(t)))$.
- Zero-Inflated Poisson distribution to model the data:
  - $X_{ij}(t)|Z_i(t), W_j(t) \sim ZIP(\Lambda_{Z_i(t),W_j(t)}, \pi(t))$.

    where:
    - Λ: block-dependent Poisson intensity parameter,
    - $\pi(t)$: sparsity at any given time period.

$$\begin{cases} X_{ij}(t)|Z_i(t), W_j(t) \sim \delta_0(X_{ij}(t)) & \text{with probability } \pi(t) \\ X_{ij}(t)|Z_i(t), W_j(t) \sim \mathcal{P}(\Lambda_{Z_i(t),W_j(t)}) & \text{with probability } 1 - \pi(t) \end{cases} \tag{1}$$

The Zip-dLBM

- Multinomial random variables to represent the membership to clusters:
  - $Z_i(t) \sim \mathcal{M}(1, \alpha(t) := (\alpha_1(t), \ldots, \alpha_Q(t)))$,
  - $W_j(t) \sim \mathcal{M}(1, \beta(t) := (\beta_1(t), \ldots, \beta_L(t)))$.
- Zero-Inflated Poisson distribution to model the data:
  - $X_{ij}(t)|Z_i(t), W_j(t) \sim ZIP(\Lambda_{Z_i(t),W_j(t)}, \pi(t))$.
    where:
    - Λ: block-dependent Poisson intensity parameter,
    - $\pi(t)$: sparsity at any given time period.

$$\begin{cases} X_{ij}(t)|Z_i(t), W_j(t) \sim \delta_0(X_{ij}(t)) & \text{with probability } \pi(t) \\ X_{ij}(t)|Z_i(t), W_j(t) \sim \mathcal{P}(\Lambda_{Z_i(t),W_j(t)}) & \text{with probability } 1-\pi(t) \end{cases} \tag{1}$$

- To model the data sparsity we introduce: $A_{ij}(t) \sim \mathcal{B}(\pi(t))$:

$$\begin{cases} X_{ij}(t)|Z_i(t), W_j(t) \sim \delta_0(X_{ij}(t)) & \text{if } A_{ij}(t) = 1 \\ X_{ij}(t)|Z_i(t), W_j(t) \sim \mathcal{P}(\Lambda_{Z_i(t),W_j(t)}) & \text{if } A_{ij}(t) = 0 \end{cases} \tag{2}$$

The Zip-dLBM

- The evolving mixing proportion and the sparsity parameter are assumed to be generated by three systems of ODEs.
- We discretize the dynamic systems by making use of their Euler scheme:
  - $a(t+1) = a(t) + f_Z(a(t)),$ with $\alpha_q(t) = \frac{e^{a_q(t)}}{\sum_{q=1}^{Q} e_q^a(t)},$
  - $b(t+1) = b(t) + f_W(b(t)),$ with $\beta_\ell(t) = \frac{e^{b_\ell(t)}}{\sum_{\ell=1}^{L} e_\ell^b(t)},$
  - $c(t+1) = c(t) + f_A(c(t)),$ with $\pi(t) = \frac{e^{c(t)}}{e^{c(t)}+e^{(1-c(t))}}.$

The Zip-dLBM

- The evolving mixing proportion and the sparsity parameter are assumed to be generated by three systems of ODEs.
- We discretize the dynamic systems by making use of their Euler scheme:
  - $a(t+1) = a(t) + f_Z(a(t))$, with $\alpha_q(t) = \frac{e^{a_q(t)}}{\sum_{q=1}^{Q} e^{a}_{q}(t)}$,
  - $b(t+1) = b(t) + f_W(b(t))$, with $\beta_\ell(t) = \frac{e^{b_\ell(t)}}{\sum_{\ell=1}^{L} e^{b}_{\ell}(t)}$,
  - $c(t+1) = c(t) + f_A(c(t))$, with $\pi(t) = \frac{e^{c(t)}}{e^{c(t)}+e^{(1-c(t))}}$.
- Where f_Z, f_W and f_A are three fully connected neural networks.

The Zip-dLBM

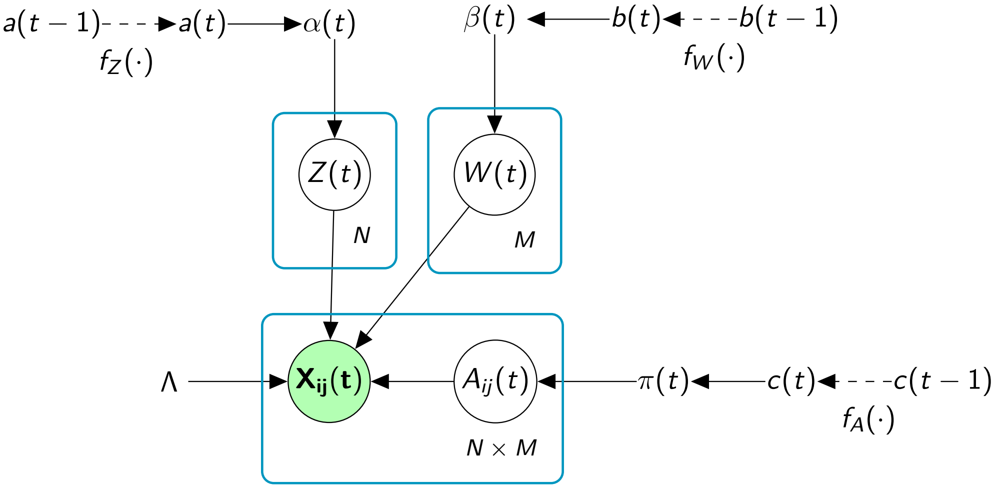


Figure: Graphical representation of Zip-dLBM.

The joint distribution

Given $\theta = (\Lambda, \alpha, \beta, \pi)$, we can compute the likelihood of the complete data:

$$p(X, Z, W, A|\theta) = p(X|Z, W, A, \Lambda, \pi)p(A \mid \pi)p(Z|\alpha)p(W|\beta) \tag{3}$$

where:

$$p(X|A, Z, W, \Lambda, \pi) = \prod_{i=1}^{N}\prod_{j=1}^{M}\prod_{t=1}^{T} \mathbf{1}_{\{X_{ij}(t)=0\}}^{A_{ij}(t)} \left\{ \left(\frac{\Lambda_{Z_i(t)W_j(t)}^{X_{ij}(t)}}{X_{ij}(t)!} \exp(-\Lambda_{Z_i(t)W_j(t)}) \right)^{(1-A_{ij}(t))} \right\}, \tag{4}$$

$$p(A|\pi) = \prod_{i=1}^{N}\prod_{j=1}^{M}\prod_{t=1}^{T} \pi(t)^{A_{ij}(t)} \Big(1 - \pi(t)\Big)^{(1-A_{ij}(t))}, \tag{5}$$

$$p(Z|\alpha) = \prod_{i=1}^{N}\prod_{q=1}^{Q}\prod_{t=1}^{T} \alpha_q(t)^{Z_{iq}(t)}, \tag{6}$$

$$p(W|\beta) = \prod_{j=1}^{M}\prod_{\ell=1}^{L}\prod_{t=1}^{T} \beta_\ell(t)^{W_{j\ell}(t)}. \tag{7}$$

The inference: Variational assumptions assumptions

Goal: maximization of the log-likelihood with respect to the model parameters.

- We rely on the Variational-EM algorithm (VEM).

Given a variational distribution $q(\cdot)$:

$$\log p(X|\theta) = \mathcal{L}(q;\theta) + KL(q(.)||p(.|X,\theta)),$$

where:

$$\begin{aligned}\mathcal{L}(q,\theta) &= \sum_Z \sum_W \sum_A q(Z,W,A) \log \frac{p(X,A,Z,W|\theta)}{q(Z,W,A)} \\ &= E_{q(A,Z,W)}\left[\log \frac{p(X,A,Z,W|\theta)}{q(A,Z,W)}\right].\end{aligned}$$

$$KL(q(.)||p(.|X,\theta)) = -\sum_Z \sum_W \sum_A q(Z,W,A) \log \frac{p(Z,W,A|X,\theta)}{q(Z,W,A)}.$$

The inference: Variational assumptions assumptions

Goal: maximization of the log-likelihood with respect to the model parameters.

- We rely on the Variational-EM algorithm (VEM).

Given a variational distribution $q(\cdot)$:

$$\log p(X|\theta) = \mathcal{L}(q;\theta) + KL(q(.)||p(.|X,\theta)),$$

where:

$$\mathcal{L}(q,\theta) = \sum_Z \sum_W \sum_A q(Z,W,A) \log \frac{p(X,A,Z,W|\theta)}{q(Z,W,A)}$$
$$= E_{q(A,Z,W)}\left[\log \frac{p(X,A,Z,W|\theta)}{q(A,Z,W)}\right].$$

$$KL(q(.)||p(.|X,\theta)) = -\sum_Z \sum_W \sum_A q(Z,W,A) \log \frac{p(Z,W,A|X,\theta)}{q(Z,W,A)}.$$

In order to optimize this lower bound $\mathcal{L}(q,\theta)$ we assume that $q(A,Z,W)$ can be factorized:

$$q(Z,W,A) = q(Z)q(W)q(A) = \prod_{i=1}^{N}\prod_{j=1}^{M}\prod_{t=1}^{T} q(A_{ij}(t)) \prod_{i=1}^{N}\prod_{t=1}^{T} q(Z_i(t)) \prod_{j=1}^{M}\prod_{t=1}^{T} q(W_j(t))$$

$$= \prod_{i=1}^{N}\prod_{j=1}^{M}\prod_{t=1}^{T} \delta_{ij}(t)^{A_{ij}(t)}(1-\delta_{ij}(t))^{1-A_{ij}(t)} \prod_{i=1}^{N}\prod_{q=1}^{Q}\prod_{t=1}^{T} \tau_{iq}(t)^{Z_{iq}(t)} \prod_{j=1}^{M}\prod_{\ell=1}^{L}\prod_{t=1}^{T} \eta_{j\ell}(t)^{W_{j\ell}(t)}.$$

The inference: Lower Bound

$\mathcal{L}(q,\theta)$ can be finally expressed as:

$$\mathcal{L}(q,\theta) = \sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{j=1}^{M}\Bigg\{\delta_{ij}(t)\log(\pi(t)\mathbf{1}_{\{X_{ij}(t)=0\}}) + (1-\delta_{ij}(t))\Bigg[\log(1-\pi(t))+$$

$$+\sum_{q=1}^{Q}\sum_{\ell=1}^{L}\Big\{\tau_{iq}(t)\eta_{j\ell}(t)X_{ij}(t)\log\Lambda_{q\ell} - \tau_{iq}(t)\eta_{j\ell}(t)\Lambda_{q\ell}\Big\}\Bigg] - (1-\delta_{ij}(t))\log(X_{ij}(t)!)\Bigg\}+$$

$$+\sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{q=1}^{Q}\tau_{iq}(t)\log(\alpha_q(t))+\sum_{t=1}^{T}\sum_{j=1}^{M}\sum_{\ell=1}^{L}\eta_{j\ell}(t)\log(\beta_\ell(t))-\sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{q=1}^{Q}\tau_{iq}(t)\log\tau_{iq}(t)+$$

$$-\sum_{t=1}^{T}\sum_{j=1}^{M}\sum_{\ell=1}^{L}\eta_{j\ell}(t)\log(\eta_{j\ell}(t)) - \sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{j=1}^{M}\Big(\delta_{ij}(t)\log(\delta_{ij}(t)) + (1-\delta_{ij}(t))\log(1-\delta_{ij}(t))\Big).$$

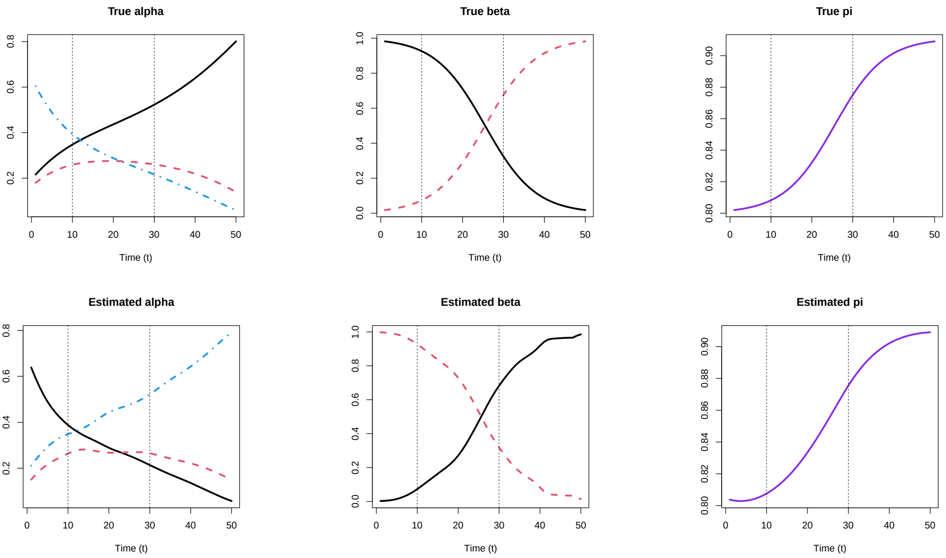
The inference: VEM Algorithm

- VE-Step: Lower bound maximization with respect to $q(A, Z, W)$.
  The optimal sequential updates of the variational distributions are computed through:
  - $\log q^*(A) = E_{W,Z}[\log p(X, A, Z, W \mid \theta)]$
  - $\log q^*(Z) = E_{W,A}[\log p(X, A, Z, W \mid \theta)]$
  - $\log q^*(W) = E_{A,Z}[\log p(X, A, Z, W \mid \theta)]$
- M-Step: Lower bound maximization with respect to $\theta = (\alpha, \beta, \pi, \Lambda)$.
  - The derived optimal update of Λ is:

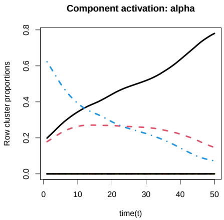
$$\hat{\Lambda}_{q\ell} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{t=1}^{T}\left\{\tau_{iq}(t)\eta_{j\ell}(t)\Big(X_{ij}(t) - \delta_{ij}(t)X_{ij}(t)\Big)\right\}}{\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{t=1}^{T}\left\{\tau_{iq}(t)\eta_{j\ell}(t)\Big(1 - \delta_{ij}(t)\Big)\right\}}$$

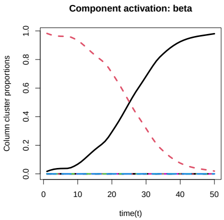
  - The optimal updates of α, β and π are obtained through a stochastic gradient descent optimization process.

Introductory example

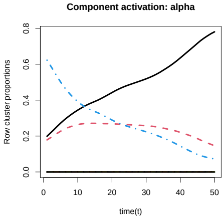


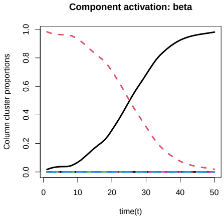
Example on simulated data - Model selection





Example on simulated data - Model selection



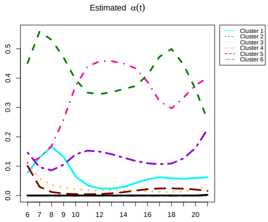


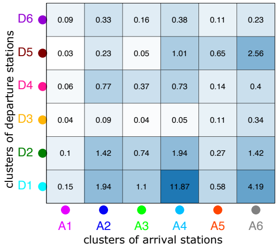
- 50 simulated dataset;
- The maximum of the given Q and L is 10;
- Zip-dLBM succeeds 86% of the time to identify the correct model $(Q = 3,\ L = 2)$.

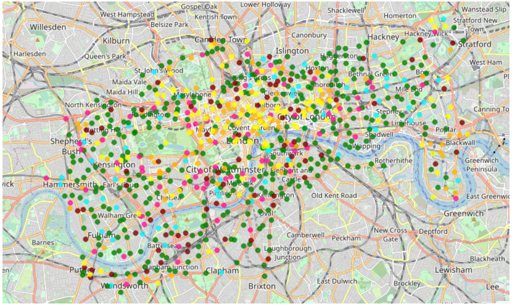
Q/L	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	86	0	0	0	0	0	0	0	0
4	0	2	0	0	0	0	0	0	0	0
5	0	2	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	4	0	0	0	0	0	0	0	0
9	0	2	0	0	0	0	0	0	0	0
10	0	4	0	0	0	0	0	0	0	0

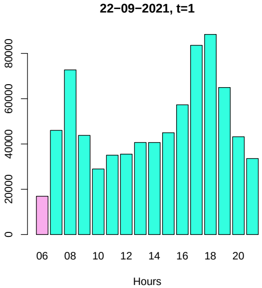
Table: Model selection. Percentage of activated components on 50 simulated datasets. The highlighted cell corresponds to the actual value of Q and L.

Zip-dLBM Appication: London Bikes - Departure Stations

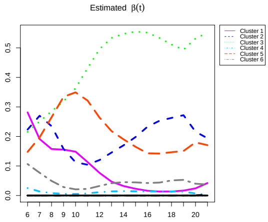






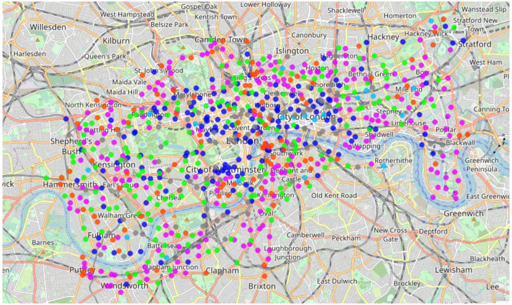


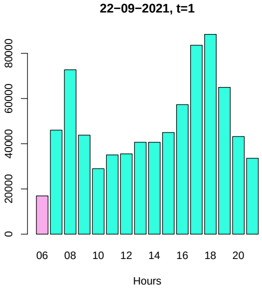
Zip-dLBM Appication: London Bikes - End Stations



clusters of departure stations	A1	A2	A3	A4	A5	A6
D6	0.09	0.33	0.16	0.38	0.11	0.23
D5	0.03	0.23	0.05	1.01	0.65	2.56
D4	0.06	0.77	0.37	0.73	0.14	0.4
D3	0.04	0.09	0.04	0.05	0.11	0.34
D2	0.1	1.42	0.74	1.94	0.27	1.42
D1	0.15	1.94	1.1	11.87	0.58	4.19

clusters of arrival stations





Outline

The online Zip-dLBM

Same assumptions of Zip-dLBM:

- Multinomial random variables to represent the cluster memberships,
- Zero-Inflated Poisson distribution to model the data,
- Time dependent model parameters generated by dynamic systems.

Goal: Real-time simultaneous cluster of observations (rows) and features (columns) of an evolving count data matrix.

- A new inference method to perform online co-clustering,
- LSTM networks on a moving window, $G_d(t)$, to model the dynamic systems,
- Online change point detection.

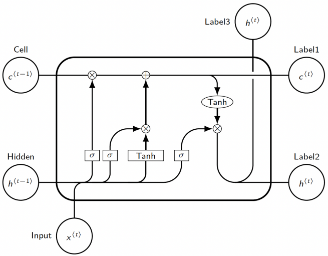


Figure: LSTM neural network.

The online inference

- VE-Step: Lower bound maximization with respect to $q(A, Z, W)$.
  The optimal sequential updates of the variational distributions are computed through:
  - $\log q^*(A) = E_{q(W,Z)}[\log p(X, A, Z, W \mid \theta)]$
  - $\log q^*(Z) = E_{q(W,A)}[\log p(X, A, Z, W \mid \theta)]$
  - $\log q^*(W) = E_{q(A,Z)}[\log p(X, A, Z, W \mid \theta)]$

The online inference

- VE-Step: Lower bound maximization with respect to $q(A, Z, W)$.
  The optimal sequential updates of the variational distributions are computed through:
  - $\log q^*(A) = E_{q(W,Z)}[\log p(X, A, Z, W \mid \theta)]$
  - $\log q^*(Z) = E_{q(W,A)}[\log p(X, A, Z, W \mid \theta)]$
  - $\log q^*(W) = E_{q(A,Z)}[\log p(X, A, Z, W \mid \theta)]$

- M-Step: Lower bound maximization with respect to $\theta = (\Lambda, \alpha(t), \beta(t), \pi(t))$.
  - The optimal update of Λ is:

$$\hat{\Lambda}_{q\ell} = \hat{\Lambda}_{q\ell}^{old} \cdot \frac{D_{q\ell}^{old}}{D_{q\ell}^{old} + D_{q\ell}^{(t)}} + \frac{N_{q\ell}^{(t)}}{D_{q\ell}^{old} + D_{q\ell}^{(t)}},$$

    - $N_{q\ell}^{old}$ and $D_{q\ell}^{old}$ are known at time $t-1$,
    - $N_{q\ell}^{(t)}$ and $D_{q\ell}^{(t)}$ are the current updates at time t.
  - The optimal updates of $\alpha(t), \beta(t)$ and $\pi(t)$ are obtained through:
    - Introduction of a moving window $G_d(t)$,
    - f_A, f_W and f_Z parametrized by LSTMs neural network,
    - loss minimization.

The online inference

Algorithm 1 VEM-SGD Algorithm for Stream Zip-dLBM

Require: $X, \hat{Q}, \hat{L}, Q_{max}, L_{max}, max.iter, G_d(t)$.
  while New observations $X(t)$ come: **do**
    Initialization of $\alpha(t), \beta(t), \pi(t), \Lambda$ with LBM;   *% with $\hat{Q}$, and $\hat{L}$*
    for it = 1 to *max.iter* **do**
      VE-Step:
      for p = 1 to Fixed.Point **do**
        alternatively update $\delta(t)$, $\tau(t)$, $\eta(t)$;   *% fix point eqs*
      end for
      M-Step:

      Update $\theta = (\Lambda, \pi(t), \alpha(t), \beta(t))$.

$$\hat{\Lambda}_{q\ell} = \hat{\Lambda}^{old}_{q\ell} \cdot \frac{D^{old}_{q\ell}}{D^{old}_{q\ell} + D^{(t)}_{q\ell}} + \frac{N^{(t)}_{q\ell}}{D^{old}_{q\ell} + D^{(t)}_{q\ell}}.$$

      Update $\alpha(t), \beta(t), \pi(t)$ *%LSTM on the moving window $t \in G_d(t)$*
    end for
    Discard all the observation before $G_d(t)$
  end while

Figure: Pseudocode of the online inference algorithm.

Example on simulated data

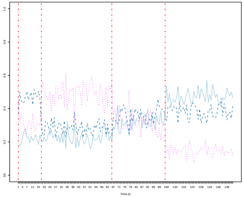


Figure: Simulated α.

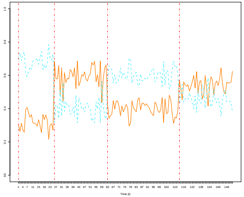


Figure: Simulated β.

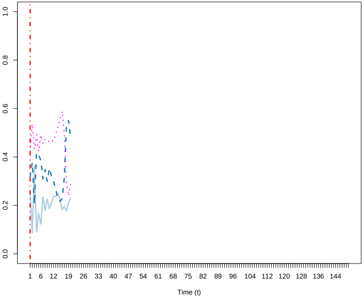


Figure: Real-time evolution of estimated α.

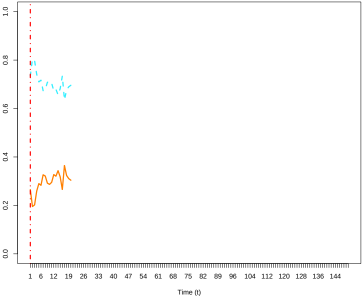


Figure: Real-time evolution of estimated β.

Outline

Pharmacovigilance data

We consider adverse drug reaction (ADR) data collected by the Regional Center of Pharmacovigilance (RCPV), located in the University Hospital of Nice:

- 2.3 million inhabitants;
- time horizon of 7 years (month as unity measure);
- 39 267 notifications in the dataset;
- we consider only drugs and ADRs notified more than 10 times;
- 419 drugs, 614 ADRs and 87 months
- extreme data sparsity, ranging around 99%.

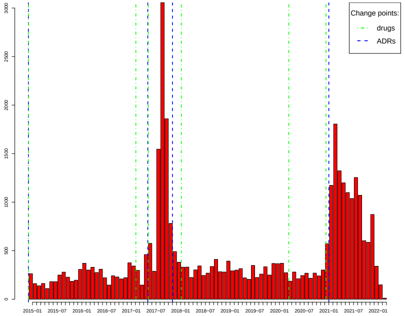


Figure: Histogram of declarations over time, with change points.

Results

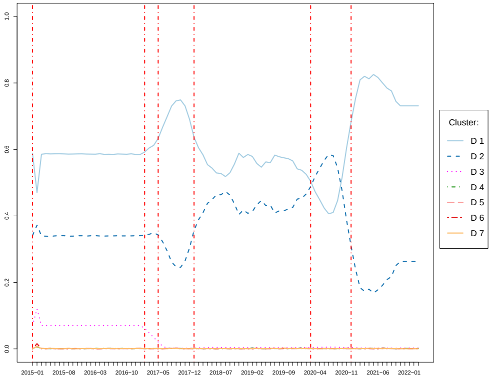


Figure: Evolution of drug clusters proportions.

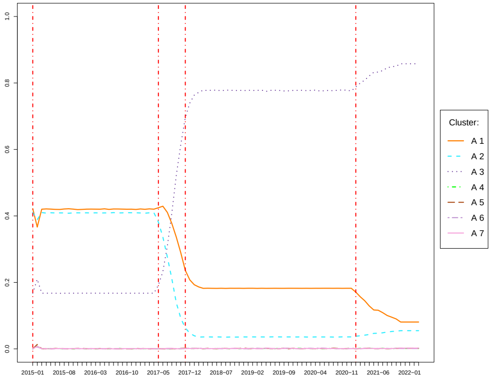


Figure: Evolution of ADR clusters proportions.

Results

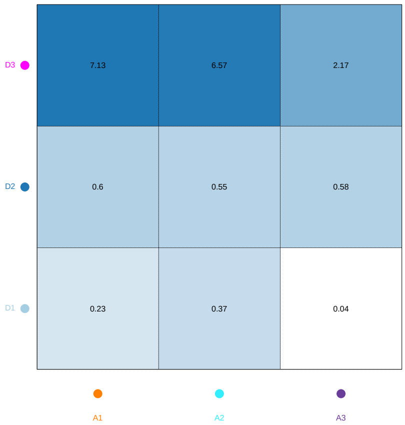


Figure: Estimated Poisson intensity parameter.

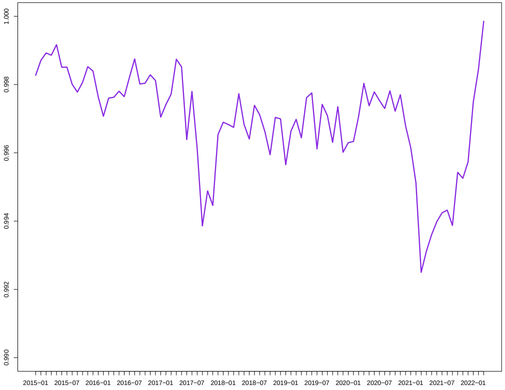


Figure: Evolution of the sparsity estimates $\hat{\pi}$.

Outline

Conclusion

The proposed approach:

- is a dynamic co-clustering method for evolving count matrices,
- it allows to summarize large sets of count data that are observed along the time,
- allows to detect changes in data evolution since observations are allowed to change cluster membership over time,
- the experiment on pharmacovigilance data provided a meaningful segmentation of drugs and adverse drug reactions.

Conclusion

The proposed approach:

- is a dynamic co-clustering method for evolving count matrices,
- it allows to summarize large sets of count data that are observed along the time,
- allows to detect changes in data evolution since observations are allowed to change cluster membership over time,
- the experiment on pharmacovigilance data provided a meaningful segmentation of drugs and adverse drug reactions.

Further works:

- Allow the sparsity parameter $\pi(t)$ to be block-dependent,
- Develop an online model selection method,
- Develop a web platform based for the RCPV. Once implemented, it will regularly run on a center machine, automatically fitting the model to incoming data.

Thank you for your attention!

References:

G. Marchello, M. Corneli, C. Bouveyron. *A Deep Dynamic Latent Block Model for Coclustering of Zero-Inflated Data Matrices*, Journal of Computational and Graphical Statistics (2023).

G. Marchello, A. Destere, M. Corneli, C. Bouveyron. *Deep dynamic co-clustering of count data streams: application to pharmacovigilance*, Preprint (2024).