Statistical Learning with Interaction Data

Applications to pharmacovigilance and network analysis

Charles BOUVEYRON

Professor of Statistics & Chair in Artificial Intelligence
Director of the Institut 3IA Côte d'Azur
Head of the Maasai research team
INRIA Centre of Université Côte d'Azur

charles.bouveyron@univ-cotedazur.fr
charles.bouveyron@inria.fr – @cbouveyron

Joint works with P. Latouche, M. Corneli,
G. Marchello, D. Liang & A. Fresse

Introduction

Introduction

Statistical learning is nowadays an unavoidable field of AI:

- it allows to model complex phenomena and provides synthetic summaries,
- it provides theoretical guarantees about its performance and behaviour,
- it can be even combined with advance deep learning model (VAE, GAN, ...).

Introduction

Statistical learning is nowadays an unavoidable field of AI:

- it allows to model complex phenomena and provides synthetic summaries,
- it provides theoretical guarantees about its performance and behaviour,
- it can be even combined with advance deep learning model (VAE, GAN, ...).

Despite recent impressive results in supervised learning, many situations remain challenging:

- high-dimensional (p large),
- big or as stream (n large),
- evolutive (evolving phenomenon),
- heterogeneous (categorical, functional, network, interaction data, ...)

Introduction

Statistical learning is nowadays an unavoidable field of AI:

- it allows to model complex phenomena and provides synthetic summaries,
- it provides theoretical guarantees about its performance and behaviour,
- it can be even combined with advance deep learning model (VAE, GAN, ...).

Despite recent impressive results in supervised learning, many situations remain challenging:

- high-dimensional (p large),
- big or as stream (n large),
- evolutive (evolving phenomenon),
- heterogeneous (categorical, functional, network, interaction data, ...)

The understanding of the results is essential:

- in many applications, practitioners are very interested in visualizing the processed data,
- and to get a synthetic summary of the data for better interpretation.

The Maasai research team (team.inria.fr/maasai)

A research team in "core AI", created in 2020:

- 6 permanent researchers, 25 Ph.D. students and postdocs, and 4 engineers,
- located at the Centre Inria of Université Côte d'Azur, in Sophia-Antipolis,

The team focuses on the Models and Algorithms of Artificial Intelligence:

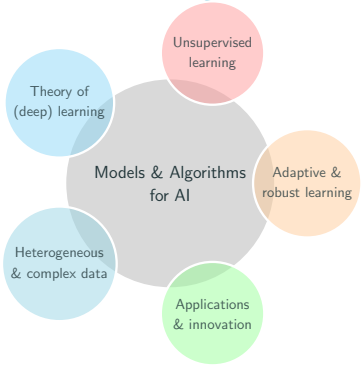


Figure 1: Scientific objectives of Maasai.

The Maasai research team (team.inria.fr/maasai)

A research team in "core AI", created in 2020:

- 6 permanent researchers, 25 Ph.D. students and postdocs, and 4 engineers,
- located at the Centre Inria of Université Côte d'Azur, in Sophia-Antipolis,

The team focuses on the Models and Algorithms of Artificial Intelligence:

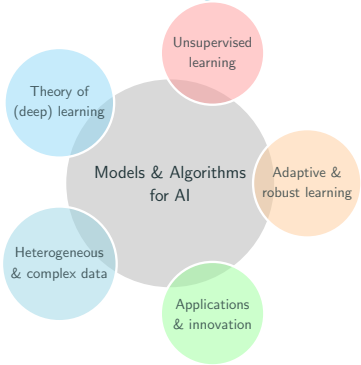
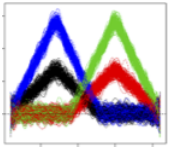
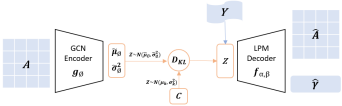
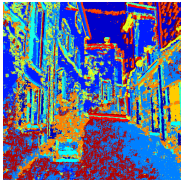


Figure 1: Scientific objectives of Maasai.

A summary of our topics:







Let's start with a question!

Can you guess the relationship between middle-age bishops and the safety surveillance of Covid-19 vaccines?

Let's start with a question!

Can you guess the relationship between middle-age bishops and the safety surveillance of Covid-19 vaccines?

Figure 2: Visualisation of an ecclesiastical network of Bishops of the 5th and 6th century of our era.

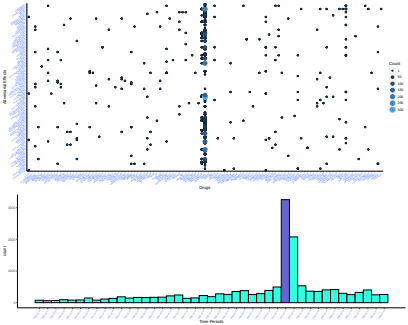
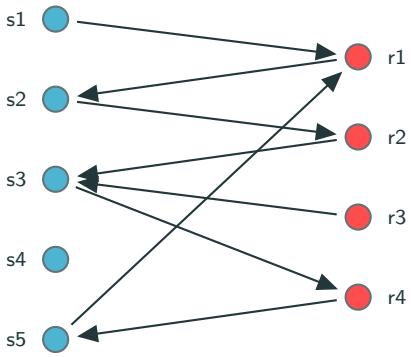


Figure 3: Frequency of monthly declarations of adverse drug reactions from 2010 to 2020.

Dealing with interaction data

Interaction data become ubiquitous for the analysis of:

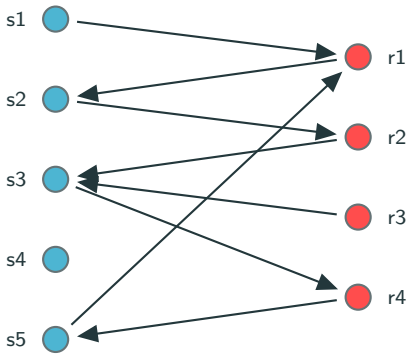
- recommendation systems, pharmacovigilance, ...: bipartite networks / incidence matrices.



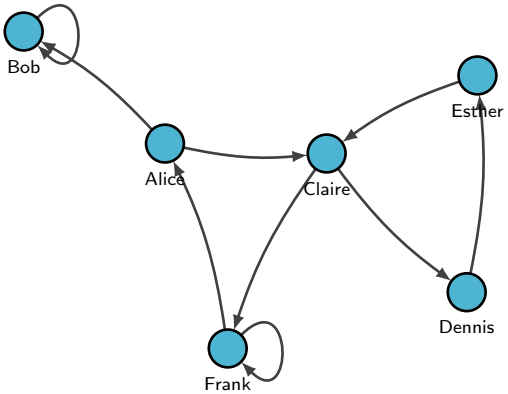
Dealing with interaction data

Interaction data become ubiquitous for the analysis of:

- recommendation systems, pharmacovigilance, ...: bipartite networks / incidence matrices.



- social networks, co-authorship networks, communication networks, ...: networks / adjacency matrices.



Dealing with interaction data

Clustering such interaction data is a recurrent task:

- clustering the nodes of a network $\rightarrow$ detection of influencers, detection of weak signals, ...
- co-clustering of ordinal data $\rightarrow$ recommendation systems, ...
- co-clustering of count data $\rightarrow$ bike sharing systems, traffic modeling, ...

With possibilities to apply it to the field of public health:

- detecting safety signals in adverse drug reaction data (pharmacovigilance),
- model and predict the use of health care services of a hospital.
- understand the role of social networks on adverse drug reaction declaration (pharmacovigilance),
- summarizing medical publication networks to help fighting against a pandemic.

Co-clustering of interaction data streams for Pharmacovigilance

Motivation: providing an automatic tool for Pharmacovigilance

Pharmacovigilance:

- it is the study of adverse reactions to drugs and vaccines,
- it aims at detecting safety signals about drugs,
- this task is done manually nowadays,
- it can be complicated in case of important media coverage.

The Nice RCPV data:

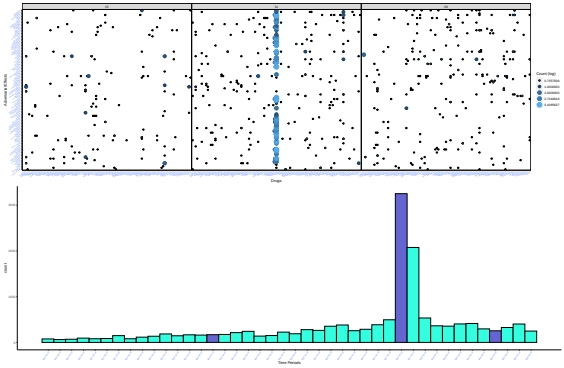


Figure 4: Evolution of spontaneous reports (extract) to RCPV from 2010 to 2020.

Data and objectives

The data we consider are organized as follows:

- rows are indexed by $i = 1, \dots N$;
- columns are indexed by $j = 1, \dots, M$;
- 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.

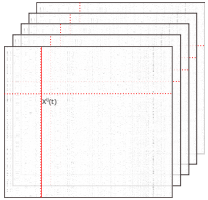


Figure 5: Data structure.

Data and objectives

The data we consider are organized as follows:

- rows are indexed by $i = 1, \dots N$;
- columns are indexed by $j = 1, \dots, M$;
- 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.

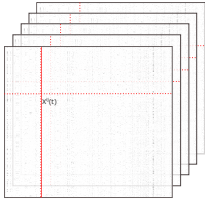


Figure 5: Data structure.

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 zero-inflated dynamic latent block model (ZI-dLBM)

1) Modeling of the row and column clusters:

- At each time instant t, the ith row of $X(t)$ is assigned to an (unobserved) group among Q(t), according to:
$$Z_i(t) \sim \mathcal{M}(1, \alpha(t) := (\alpha_1(t), \ldots, \alpha_Q(t))),$$
where $\alpha_q(t) \geq 0$ and $\sum_{q=1}^{Q} \alpha_q(t) = 1$, for all $t = 0, ..., T$.

- similarly, the jth column of $X(t)$ is assigned to an (unobserved) group among L(t), according to:
$$W_j(t) \sim \mathcal{M}(1, \beta(t) := (\beta_1(t), \ldots, \beta_L(t))),$$
where $\beta_\ell(t) \geq 0$ and $\sum_{\ell=1}^{L} \beta_\ell(t) = 1$, for all $t = 0, ..., T$.

- row assignments $z_i(t)$ are further assumed to be independent from column assignments $w_j(t)$, for all i, j;

The zero-inflated dynamic latent block model (ZI-dLBM)

2) Modeling of a potential extreme sparsity:

- the observed variable $X(t)$ is assumed to be modeled by a mixture of block-conditional Zero-Inflated (ZI) distributions:

$$X_{ij}(t)|Z_i(t) = k, W_j(t) = \ell \quad \sim \quad ZI(\zeta_{k,\ell}, \pi(t)),$$

where:

- ζ is the block-dependent vector of parameters for the distribution $\phi(X_{ij}(t), .)$,
- $\pi(t)$ is the sparsity probability at any given time period t.

The zero-inflated dynamic latent block model (ZI-dLBM)

The zero-inflated distribution is therefore such that:

$$\begin{cases} X_{ij}(t)|Z_i(t), W_j(t) \sim 0 & \text{with probability } \pi(t) \\ X_{ij}(t)|Z_i(t), W_j(t) \sim \phi(X_{ij}(t); \zeta_{Z_i(t), W_j(t)}) & \text{with probability } 1 - \pi(t) \end{cases} \tag{1}$$

The zero-inflated dynamic latent block model (ZI-dLBM)

The zero-inflated distribution is therefore such that:

$$\begin{cases} X_{ij}(t)|Z_i(t), W_j(t) \sim 0 & \text{with probability } \pi(t) \\ X_{ij}(t)|Z_i(t), W_j(t) \sim \phi(X_{ij}(t); \zeta_{Z_i(t), W_j(t)}) & \text{with probability } 1 - \pi(t) \end{cases} \tag{1}$$

The modeling of the data sparsity can be re-formulated by introducing a latent variable $A(t)$:

- such that $A_{ij}(t) \sim \mathcal{B}(\pi(t))$,
- and we therefore get:

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

The zero-inflated dynamic latent block model (ZI-dLBM)

3) Modeling of the dynamic of sparsity and cluster proportions:

The evolving mixing proportion and the sparsity parameter are assumed to be generated by three systems of ODEs, respectively:

- $\frac{d}{dt}a(t) = f_Z(a(t))$, with $\alpha_q(t) = \frac{e^{a_q(t)}}{\sum_{q=1}^{Q} e_q^a(t)}$,
- $\frac{d}{dt}b(t) = f_W(b(t))$, with $\beta_\ell(t) = \frac{e^{b_\ell(t)}}{\sum_{\ell=1}^{L} e_\ell^b(t)}$,
- $\frac{d}{dt}c(t) = f_A(c(t))$, with $\pi(t) = \frac{e^{c(t)}}{e^{c(t)}+e^{(1-c(t))}}$,

The zero-inflated dynamic latent block model (ZI-dLBM)

3) Modeling of the dynamic of sparsity and cluster proportions:

The evolving mixing proportion and the sparsity parameter are assumed to be generated by three systems of ODEs, respectively:

- $\frac{d}{dt}a(t) = f_Z(a(t))$, with $\alpha_q(t) = \frac{e^{a_q(t)}}{\sum_{q=1}^{Q} e^{a}_{q}(t)}$,
- $\frac{d}{dt}b(t) = f_W(b(t))$, with $\beta_\ell(t) = \frac{e^{b_\ell(t)}}{\sum_{\ell=1}^{L} e^{b}_{\ell}(t)}$,
- $\frac{d}{dt}c(t) = f_A(c(t))$, with $\pi(t) = \frac{e^{c(t)}}{e^{c(t)}+e^{(1-c(t))}}$,

Since we work with discrete time points, the dynamic systems reduce to their Euler schemes:

- $a(t+1) = a(t) + f_Z(a(t))$,
- $b(t+1) = b(t) + f_W(a(t))$,
- $c(t+1) = c(t) + f_A(a(t))$.

The zero-inflated dynamic latent block model (ZI-dLBM)

3) Modeling of the dynamic of sparsity and cluster proportions:

The evolving mixing proportion and the sparsity parameter are assumed to be generated by three systems of ODEs, respectively:

- $\frac{d}{dt}a(t) = f_Z(a(t))$, with $\alpha_q(t) = \frac{e^{a_q(t)}}{\sum_{q=1}^{Q} e^{a}_{q}(t)}$,
- $\frac{d}{dt}b(t) = f_W(b(t))$, with $\beta_\ell(t) = \frac{e^{b_\ell(t)}}{\sum_{\ell=1}^{L} e^{b}_{\ell}(t)}$,
- $\frac{d}{dt}c(t) = f_A(c(t))$, with $\pi(t) = \frac{e^{c(t)}}{e^{c(t)}+e^{(1-c(t))}}$,

Since we work with discrete time points, the dynamic systems reduce to their Euler schemes:

- $a(t+1) = a(t) + f_Z(a(t))$,
- $b(t+1) = b(t) + f_W(a(t))$,
- $c(t+1) = c(t) + f_A(a(t))$.

We further assume that the functions f_Z, f_W and f_A can be modeled by three fully connected neural networks.

The zero-inflated dynamic latent block model (ZI-dLBM)

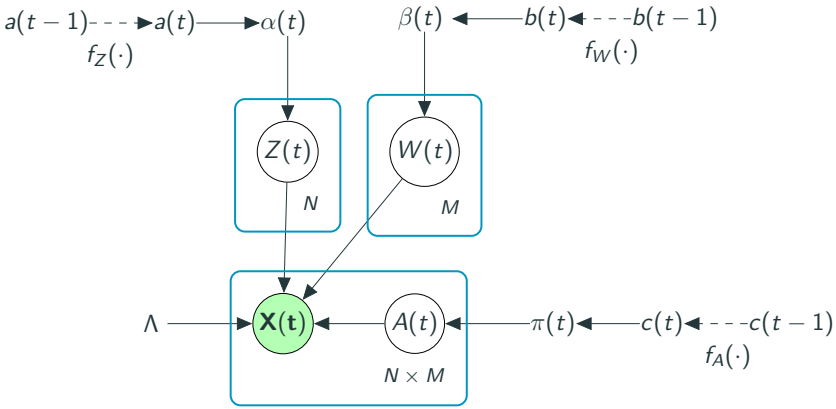


Figure 6: Graphical representation of the ZIP-dLBM model.

The joint distribution (Poisson case)

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

We rely on the Variational-EM algorithm (VEM) for infering the model:

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

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

where:

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

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

The inference: variational assumptions assumptions

We rely on the Variational-EM algorithm (VEM) for infering the model:

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

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

where:

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

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

In order to optimize $\mathcal{L}(q,\theta)$, we further assume that $q(A,Z,W)$ can factorize:

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

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

The inference: the 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 $\theta = (\alpha(t), \beta(t), \pi(t), \Lambda)$.
  - 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 $\alpha(t), \beta(t)$ and $\pi(t)$ are obtained through a stochastic gradient descent optimization process.

The inference: VEM Algorithm

Algorithm 1 VEM-SGD Algorithm (for the Zero-Inflated Poisson distribution)

Require: $X, Q, L, max.iter$, $\alpha(t), \beta(t), \pi(t), \Lambda$ from Initialization.

Initialization of $\tau(t)$ and $\eta(t)$: sampling from $\mathcal{M}(\alpha(t))$ and $\mathcal{M}(\beta(t))$, respectively;

Initialization of $\delta(t)$: matrix of 1, then setting $\delta(t) = 0$ when $X > 0$;

for it $= 1$ to $max.iter$ **do**

VE-Step:

for p $= 1$ to Fixed.Point **do**

Update $\delta(t)$, $\tau(t)$, $\eta(t)$:

$$\delta_{ij}(t) = \frac{\exp(R_{ij}(t))}{(1 + \exp(R_{ij}(t)))},$$

where:

$$R_{ij}(t) = \log(\pi(t)\mathbf{1}_{\{X_{ij}(t)=0\}}) + \sum_{q=1}^{Q}\sum_{\ell=1}^{L}\Big[-\tau_{iq}(t)\eta_{j\ell}(t)X_{ij}(t)\log\Lambda_{q\ell} + \tau_{iq}(t)\eta_{j\ell}(t)\Lambda_{q\ell}\Big] + \log X_{ij}(t)! - \log(1-\pi(t)).$$

$$\tau_{iq}(t) = \frac{1}{D_q}\exp\left(\sum_{j=1}^{M}\sum_{\ell=1}^{L}\Big\{(1-\delta_{ij}(t))\Big[\eta_{j\ell}(t)X_{ij}(t)\log(\Lambda_{q\ell}) - \eta_{j\ell}(t)\Lambda_{q\ell}\Big]\Big\} + \log(\alpha_q(t))\right).$$

$$\eta_{j\ell}(t) = \frac{1}{D_\ell}\exp\left(\sum_{i=1}^{N}\sum_{q=1}^{Q}\Big\{(1-\delta_{ij}(t))\Big[\tau_{iq}(t)X_{ij}(t)\log(\Lambda_{q\ell}) - \tau_{iq}(t)\Lambda_{q\ell}\Big]\Big\} + \log(\beta_\ell(t))\right).$$

with D_q and D_ℓ normalizing constants.

end for

M-Step:

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

$$\hat{\Lambda}_{q\ell} = \frac{\sum_{i,j,t}\Big\{\tau_{iq}(t)\eta_{j\ell}(t)\Big(X_{ij}(t) - \delta_{ij}(t)X_{ij}(t)\Big)\Big\}}{\sum_{i,j,t}\Big\{\tau_{iq}(t)\eta_{j\ell}(t)\Big(1-\delta_{ij}(t)\Big)\Big\}}.$$

for epoch $= 1$ to Epochs **do**

Update $\hat{\alpha}(t), \hat{\beta}(t), \hat{\pi}(t)$:

Loss Evaluation;

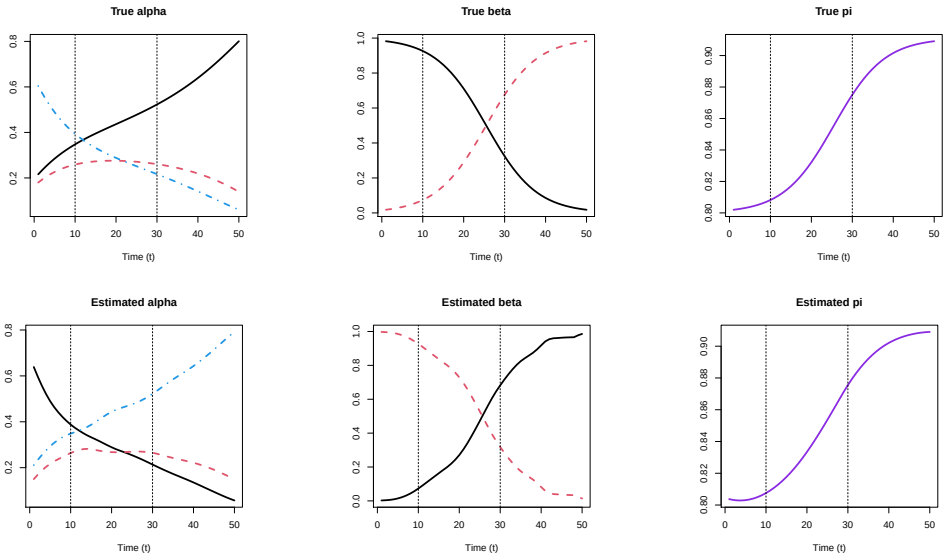
Algorithm backpropagation;

Numerical optimization with SGD.

end for

end for

Introductory example



Introductory example

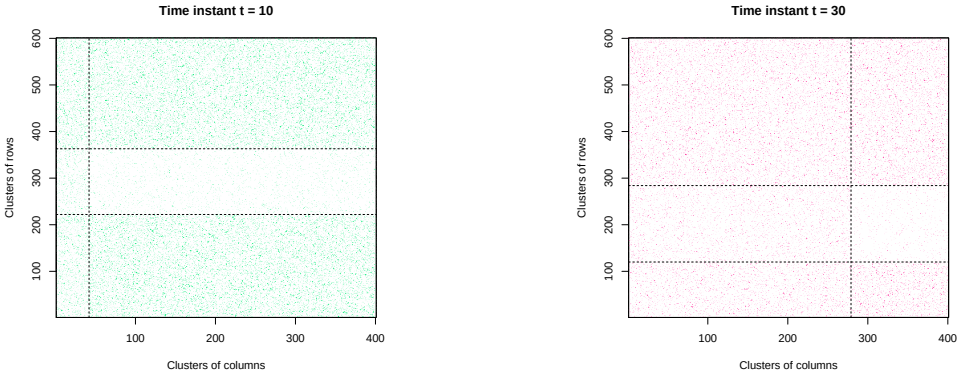
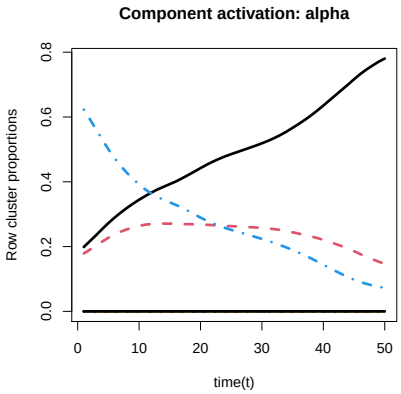
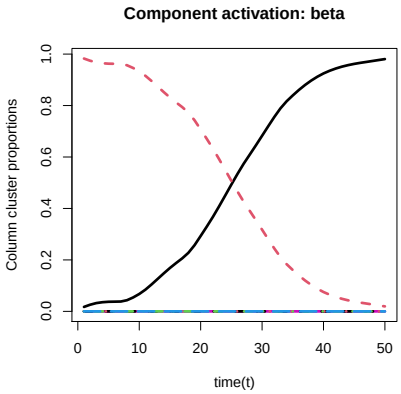


Figure 8: Reorganized incidence matrices at time instants t =10,30, according to the estimates $\hat{Z}$ and $\hat{W}$.

Model Selection example





Model selection experiment

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

Pharmacovigilance: analysis of adverse drug reactions

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;
- several channels (e.g. website form, email, etc);
- time horizon of 7 years (trimester as unity measure);
- 27 754 notifications in the dataset;
- only drugs and ADRs notified more than 20 times are considered;
- the resulting dataset contains 236 drugs, 324 ADRs and 29 trimesters.

Pharmacovigilance: analysis of adverse drug reactions

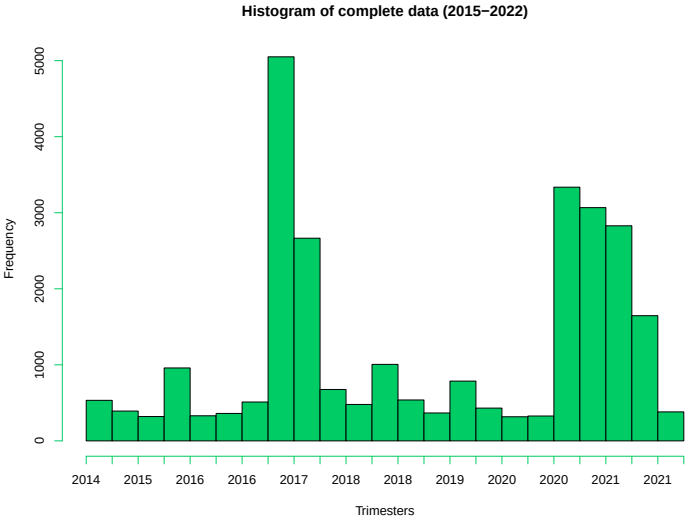


Figure 9: Frequency of declarations received by the pharmacovigilance center from January 2015 to March 2022, sorted by month.

Pharmacovigilance: analysis of adverse drug reactions

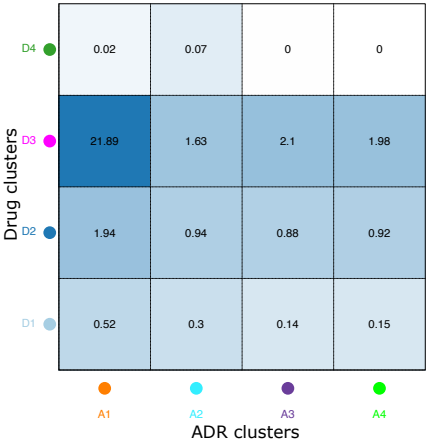


Figure 10: Estimated Poisson intensities, each color represents a different drug (ADR) cluster.

Pharmacovigilance: analysis of adverse drug reactions

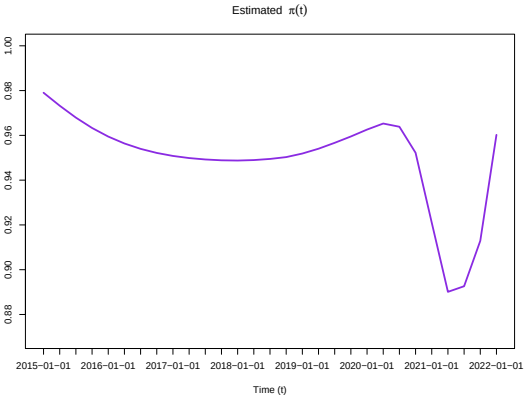
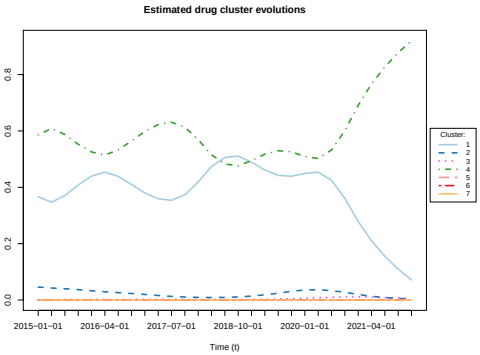


Figure 11: Evolution of the estimates $\hat{\pi}$.

Pharmacovigilance: analysis of adverse drug reactions



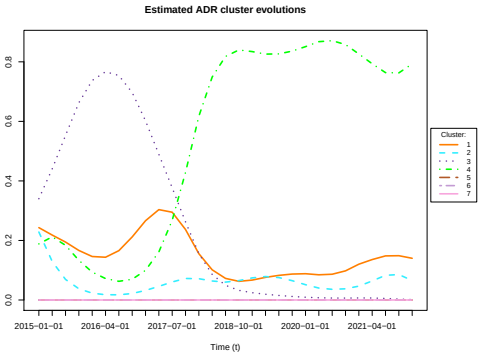


Figure 12: Evolution of the estimates $\hat{\alpha}$ and $\hat{\beta}$.

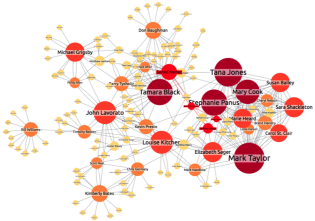
The deep latent position model for network clustering

Introduction: context and motivation

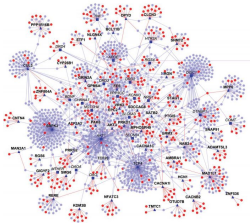
Networks are specific interaction data where receivers are the same individuals as the senders:

- **Nodes** (or vertices) represent an object or entity such as a person, an email, etc.
- **Links** (or edges) represent connections between nodes such as "follow", "send to",etc.

social network



email network



protein-protein interaction network

Introduction: related work on network clustering

Statistical models:

- Based on probabilistic generation:
  - **SBM** (Nowicki et al., 2001),
  - **OSBM** (Latouche et al., 2011),
  - **STBM** (Bouveyron et al., 2018), etc.

- Based on latent positions:
  - **LPM** (Hoff et al., 2002),
  - **LPCM** (Handcock et al., 2007), etc.

$\Longrightarrow$ limitations:

- challenging inference procedure
- scaling difficulties

Introduction: related work on network clustering

Statistical models:

- Based on probabilistic generation:
  - **SBM** (Nowicki et al., 2001),
  - **OSBM** (Latouche et al., 2011),
  - **STBM** (Bouveyron et al., 2018), etc.
- Based on latent positions:
  - **LPM** (Hoff et al., 2002),
  - **LPCM** (Handcock et al., 2007), etc.

$\Longrightarrow$ limitations:

- challenging inference procedure
- scaling difficulties

Deep learning approaches:

- Based on VAE architecture:
  - **VGAE** (Kipf et al., 2016),
  - **ARVGA** (Pan et al. 2018),
  - **DGLFRM** (Mehta et al., 2019), etc.

$\Longrightarrow$ limitations:

- rely on an external algorithm (e.g. k-means) for clustering;
- without taking into account edge features;
- use an simple inner product as decoder.

Deep latent position model (deepLPM)

Extending the idea of graph VAE, we propose the deepLPM model here:

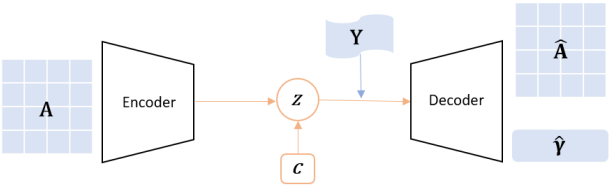
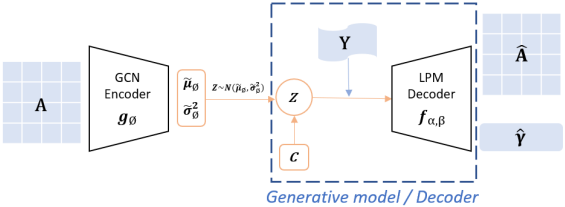


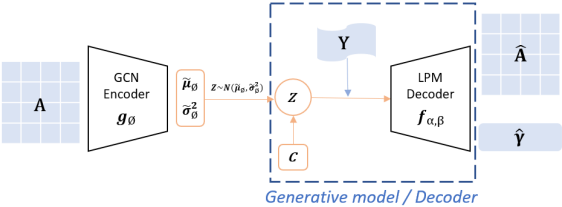
Figure 13: A deep-learning-like model view of deepLPM.

- take into account edge features
- use a more general latent position-based decoder
- build an end-to-end clustering model

The generative model for deepLPM



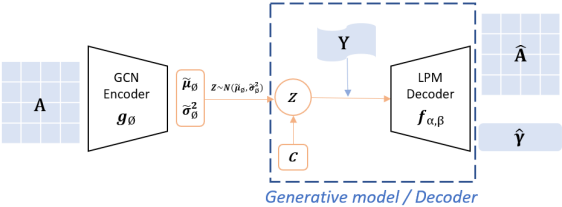
The generative model for deepLPM



First, each node is assigned to a cluster via a random variable c_i encoding its cluster membership:

$$c_i \overset{iid}{\sim} \mathcal{M}(1, \pi), \quad \text{with} \quad \pi \in [0,1]^K, \ \sum_{k=1}^{K} \pi_k = 1. \tag{8}$$

The generative model for deepLPM



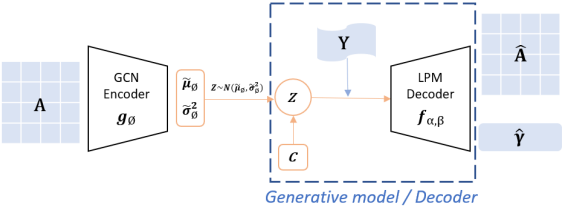
First, each node is assigned to a cluster via a random variable c_i encoding its cluster membership:

$$c_i \overset{iid}{\sim} \mathcal{M}(1, \pi), \quad \text{with} \quad \pi \in [0,1]^K, \ \sum_{k=1}^{K} \pi_k = 1. \tag{8}$$

Then, conditionally to its cluster membership, a latent embedding vector z_i is generated:

$$z_i | (c_{ik} = 1) \sim \mathcal{N}(\mu_k, \sigma_k^2 I_P), \quad \text{with} \quad \sigma_k^2 \in \mathbb{R}^{+*}. \tag{9}$$

The generative model for deepLPM



Finally, the probability of a connection between nodes i and j is modeled by

$$A_{ij}|z_i, z_j \sim \mathcal{B}(f_{\alpha,\beta}(z_i, z_j)), \tag{10}$$

with

$$f_{\alpha,\beta}(z_i, z_j) = \sigma(\alpha + \beta^T y_{ij} - ||z_i - z_j||^2). \tag{11}$$

where $f_{\alpha,\beta}$ is a decoding neural network parametrized by α and β. Moreover, σ is the logistic sigmoid function and y_{ij} is the covariate of the edge connecting i with j.

Model inference: a variational auto-encoding procedure

Denoting by $\Theta = \{\pi, \mu_k, \sigma_k^2, \alpha, \beta\}$, we want to maximize the integrated log-likelihood:

$$\log p(A|\Theta) = \log \int_Z \sum_C p(A, Z, C|\Theta) dZ, \tag{12}$$

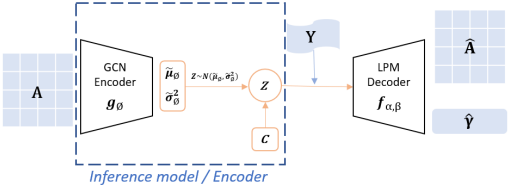
with respect to Θ.

Since Equation (12) is not tractable, we use a variational approach to approximate it

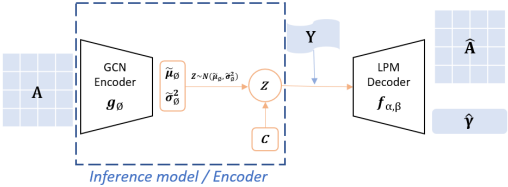
$$\log p(A|\Theta) = \underbrace{\mathcal{L}(q(Z, C); \Theta)}_{ELBO} + D_{KL}(q(Z, C)||p(Z, C|A, \Theta)). \tag{13}$$

where D_{KL} denotes the Kullback-Leibler divergence.

Model inference: variational assumptions



Model inference: variational assumptions

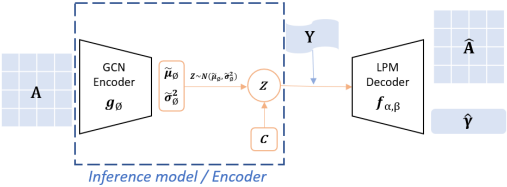


Assumption 1.

$$q(Z) = \prod_{i=1}^{N} q(z_i) = \prod_{i=1}^{N} \mathcal{N}(z_i; \tilde{\mu}_\phi(\overline{A})_i, \tilde{\sigma}_\phi{}^2(\overline{A})_i I_P), \tag{14}$$

where $[\tilde{\mu}_\phi(\overline{A}), \log \tilde{\sigma}_\phi{}^2(\overline{A})] = g_\phi(\overline{A})$, g_ϕ is a two-layer GCN encoder parametrized by ϕ. $\overline{A} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ kipf2016semi is the normalized adjacency matrix.

Model inference: variational assumptions



Assumption 1.

$$q(Z) = \prod_{i=1}^{N} q(z_i) = \prod_{i=1}^{N} \mathcal{N}(z_i; \tilde{\mu}_\phi(\overline{A})_i, \tilde{\sigma}_\phi{}^2(\overline{A})_i I_P), \tag{14}$$

where $[\tilde{\mu}_\phi(\overline{A}), \log \tilde{\sigma}_\phi{}^2(\overline{A})] = g_\phi(\overline{A})$, g_ϕ is a two-layer GCN encoder parametrized by ϕ. $\overline{A} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ kipf2016semi is the normalized adjacency matrix.

Assumption 2.

$$q(C) = \prod_{i=1}^{N} \mathcal{M}(c_i; 1, \gamma_i), \quad \text{with} \quad \sum_{k=1}^{K} \gamma_{ik} = 1, \tag{15}$$

where γ_{ik} represents the variational probability that node i is in cluster k.

Model inference: a joint optimization

With the above assumptions, $\mathcal{L}$(ELBO) can be further developed as

$$\mathcal{L} = \underbrace{[\sum_{i \neq j} A_{ij} \log \eta_{ij} + (1 - A_{ij}) \log(1 - \eta_{ij})]}_{\textit{implicit optimization}} - \underbrace{\sum_{i=1}^{N} \sum_{k=1}^{K} \gamma_{ik} D_{KL}(\mathcal{N}(\tilde{\mu}_{\phi}(\overline{A})_i, \tilde{\sigma}_{\phi}{}^2(\overline{A})_i I_P) || \mathcal{N}(\mu_k, \sigma_k^2 I_P))}_{\textit{explicit optimization + implicit optimization}} + \underbrace{\sum_{i=1}^{N} \sum_{k=1}^{K} \gamma_{ik} \log(\frac{\pi_k}{\gamma_{ik}})}_{\textit{explicit optimization}},$$

where $\eta_{ij} = \sigma(\alpha + \beta^T y_{ij} - ||z_i - z_j||^2)$.

Model inference: a joint optimization

With the above assumptions, $\mathcal{L}$(ELBO) can be further developed as

$$\mathcal{L} = \underbrace{[\sum_{i \neq j} A_{ij} \log \eta_{ij} + (1 - A_{ij}) \log(1 - \eta_{ij})]}_{\textit{implicit optimization}} - \underbrace{\sum_{i=1}^{N} \sum_{k=1}^{K} \gamma_{ik} D_{KL}(\mathcal{N}(\tilde{\mu}_\phi(\overline{A})_i, \tilde{\sigma}_\phi{}^2(\overline{A})_i I_P) || \mathcal{N}(\mu_k, \sigma_k^2 I_P))}_{\textit{explicit optimization + implicit optimization}} + \underbrace{\sum_{i=1}^{N} \sum_{k=1}^{K} \gamma_{ik} \log(\frac{\pi_k}{\gamma_{ik}})}_{\textit{explicit optimization}},$$

where $\eta_{ij} = \sigma(\alpha + \beta^T y_{ij} - ||z_i - z_j||^2)$.

Algorithm 2 Estimation of deepLPM

Input: adjacency matrix A, edge features Y
pretrain_model = pretrain(A, 50 epochs) ▷ pre-training to save initial weights of encoder/decoder
while $\mathcal{L}$ increases **do**
  $\tilde{\mu}_\phi, \tilde{\sigma}_\phi^2 = \text{GCN}(A)$
  explicit optimization (closed formulas):
  update $\hat{\gamma}_{ik}$, $\hat{\pi}_k, \hat{\mu}_k, \hat{\sigma}_k^2$
  calculate loss $-\mathcal{L}$
  implicit optimization (SGD):
  update encoder parameter ϕ and decoder parameters α, β

Numerical experiments: scenario A

Scenario A: 3 communities simulated according to **LPCM**, the mean of each cluster is set to

$$\begin{cases} \mu_1 = [0, 0] \\ \mu_2 = [1.5 * \delta, 1.5 * \delta] \\ \mu_3 = [-1.5 * \delta, 1.5 * \delta] \end{cases}$$

where $\delta \in [0.2, 0.95]$.

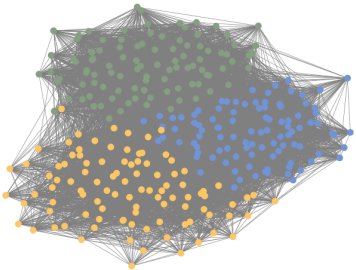


Figure 14: Network simulated on scenario A.

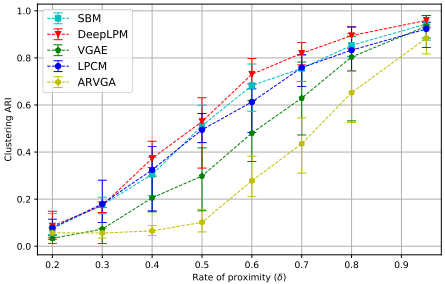


Figure 15: Clustering ARI with different proximity rate δ in Sc.A.

Numerical experiments: scenario B

Scenario B: 1 cluster with large external connectivity and 2 communities with high internal connectivity based on **SBM**

$$\Pi = \begin{pmatrix} b & a & a \\ a & a & b \\ a & b & a \end{pmatrix}$$

where $a = 0.25$, $b = 0.01 + (1 - \delta^{'}) * (a - 0.01)$, with $\delta^{'} \in [0.2, 1.0]$

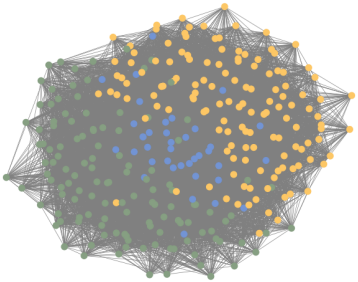


Figure 16: Network simulated on scenario B.

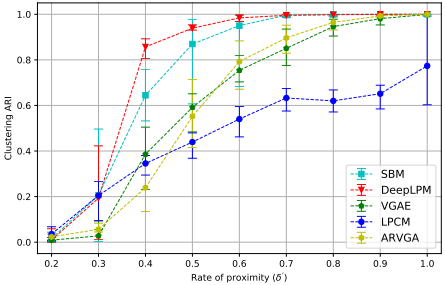


Figure 17: Clustering ARI with different proximity rate $\delta^{'}$ in Sc.B.

Model selection

A key element of our model is to be able to automatically determine P and K thanks to the auto-penalization of the deep encoder (Kingma et al., 2016; Dai et al., 2017).

Model selection

A key element of our model is to be able to automatically determine P and K thanks to the auto-penalization of the deep encoder (Kingma et al., 2016; Dai et al., 2017).

1) Fix $K = 3$, vary $P \in \{2, 4, 8, 16, 32\}$:

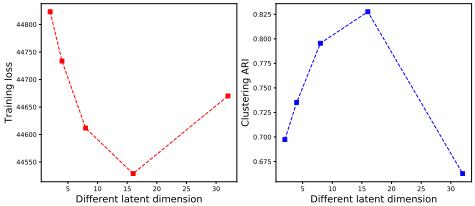


Figure 18: Averaged training loss (-ELBO) and clustering ARI 50 networks based on scenario B.

Model selection

A key element of our model is to be able to automatically determine P and K thanks to the auto-penalization of the deep encoder (Kingma et al., 2016; Dai et al., 2017).

1) Fix $K = 3$, vary $P \in \{2, 4, 8, 16, 32\}$:

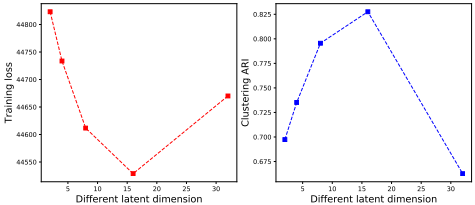


Figure 18: Averaged training loss (-ELBO) and clustering ARI 50 networks based on scenario B.

2) Fix $P = 16$, vary $K \in [2, 6]$:

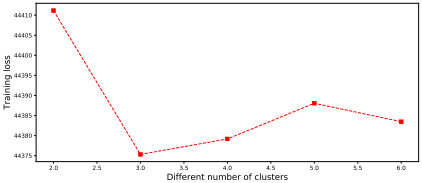


Figure 19: Averaged training loss (-ELBO) on 50 synthetic data in scenario B.

Application to Digital Humanities: analysis of a medieval network

The considered data (Lamassé et al., 2014) report the ecclesiastical councils that took place in Merovingian Gaul during the 5th and 6th centuries.

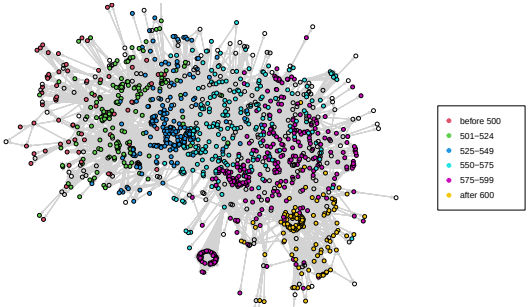


Figure 20: Visualisation of the ecclesiastical network, highlighting the temporality of the relationships.

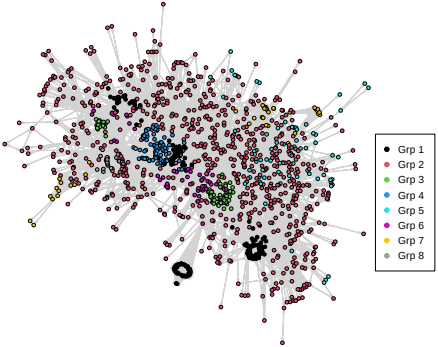


Figure 21: Visualisation of 8 cluster partitions with covariates on medieval data.

Conclusion

The ZI-dLBM model:

- a model for the co-clustering of sparse evolving count data matrices,
- an interesting tool to summarize massive pharmacovigilance data and detect patterns,
- we plan to extend this model to the online setup to handle streams of ADR declarations.

C. Bouveyron, M. Corneli and G. Marchello, *A Deep Dynamic Latent Block Model for the Co-clustering of Zero-Inflated Data Matrices*, Preprint HAL 03800210, Université Côte d'Azur, 2022

G. Marchello, A. Fresse, M. Corneli and C. Bouveyron, *Co-clustering of evolving count matrices in pharmacovigilance with the dynamic latent block model,* Statistics & Computing, in press, 2022.

Conclusion

The ZI-dLBM model:

- a model for the co-clustering of sparse evolving count data matrices,
- an interesting tool to summarize massive pharmacovigilance data and detect patterns,
- we plan to extend this model to the online setup to handle streams of ADR declarations.

C. Bouveyron, M. Corneli and G. Marchello, *A Deep Dynamic Latent Block Model for the Co-clustering of Zero-Inflated Data Matrices*, Preprint HAL 03800210, Université Côte d'Azur, 2022

G. Marchello, A. Fresse, M. Corneli and C. Bouveyron, *Co-clustering of evolving count matrices in pharmacovigilance with the dynamic latent block model,* Statistics & Computing, in press, 2022.

The deepLPM model:

- a (deep) latent model to represent and cluster network data with covariates,
- we applied it for visualizing and clustering a historical network,
- we plan to extend this model to handle text data as covariates on the nodes.

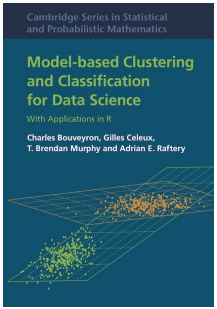
C. Bouveyron, M. Corneli, P. Latouche D. Liang, *Clustering by Deep Latent Position Model with Graph Convolutional Network*, Preprint HAL n°03629104, Université Côte d'Azur, 2022.

Conclusion

"Ce qui est simple est toujours faux.
Ce qui ne l'est pas est inutilisable."

Paul Valéry

A couple of announcements



Available at Cambridge University Press

PDF at `math.unice.fr/~cbouveyr/MBCbook`

Several open positions

at Institut 3IA Côte d'Azur!

`3ia.univ-codeazur.eu/apply`