

The Deep Latent Position Topic Model for Clustering and Representation of Networks with Textual Edges

Rémi Boutin¹, Pierre Latouche² and Charles Bouveyron³

¹ LPSM - Sorbonne Université

² LMBP - Université Clermont Auvergne

³ Maasai team - INRIA, Université Côte d'Azur

Graphical models and Clustering, Montpellier, May 2024



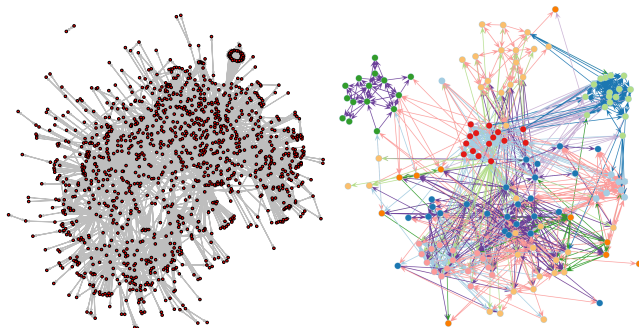
Université
Paris Cité



Introduction

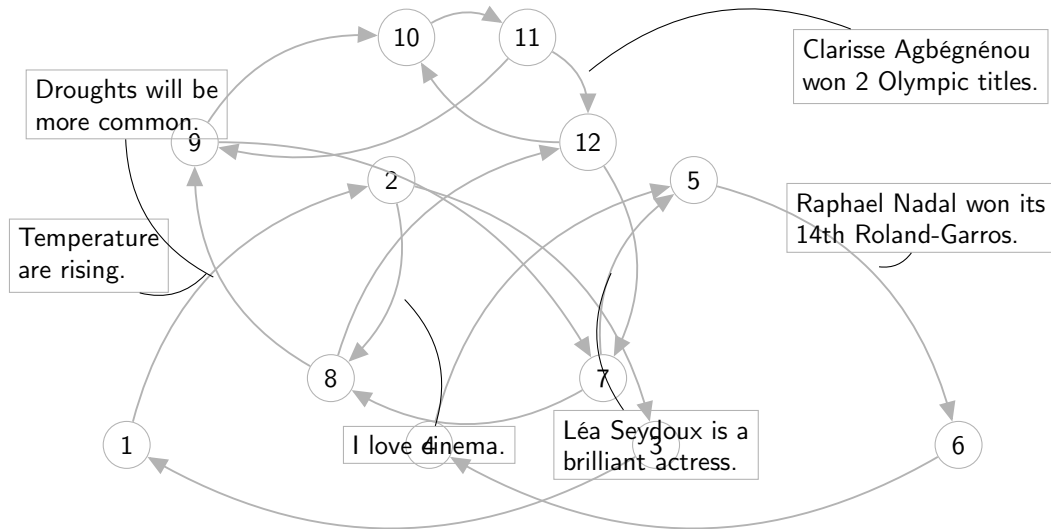
Networks can be observed directly or indirectly from a variety of sources:

- ▶ social websites (Facebook, Twitter, ...),
- ▶ emails (from your Gmail, Clinton's mails, Enron Email data ...),
- ▶ digital/numeric documents (Panama papers, co-authorships, ...),
- ▶ and even archived documents in libraries (digital humanities).

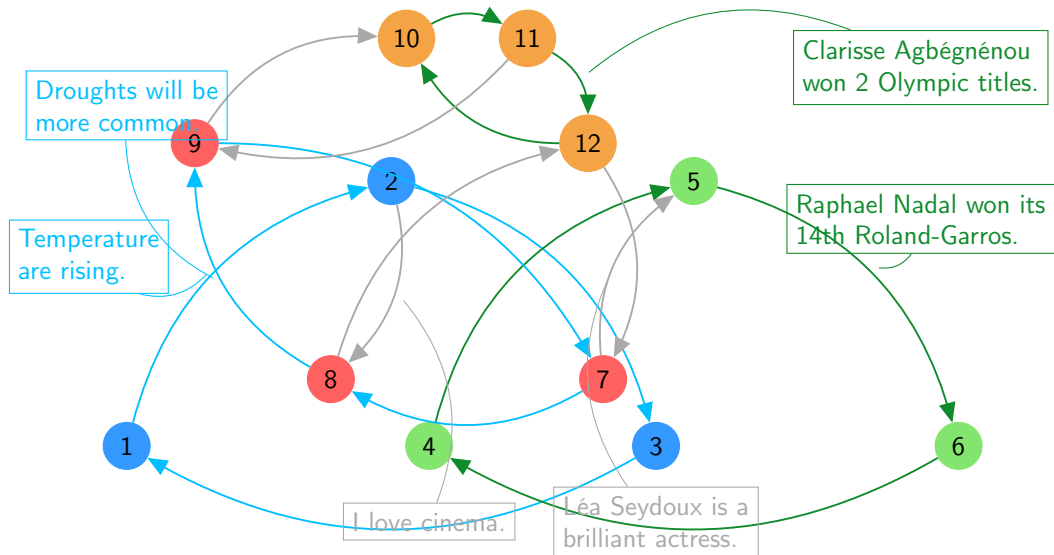


⇒ most of these sources involve text!

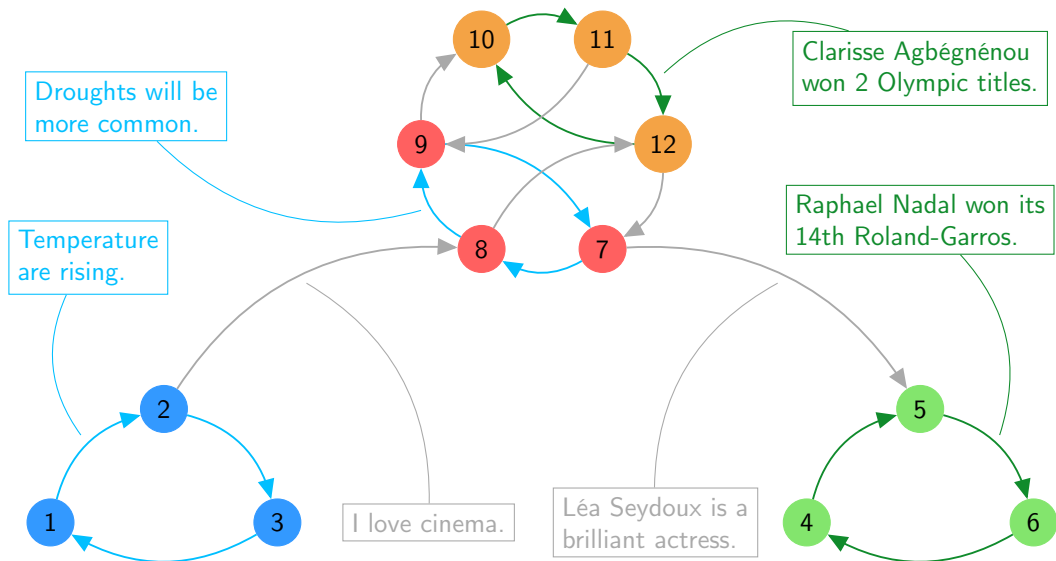
Observed network: difficult to apprehend



STBM/ETSBM results: difficult to represent



Our goal with Deep-LPTM



Notations

- ▶ i and j will refer to **nodes**.
- ▶ q and r will refer to **clusters**.
- ▶ k will refer to **topics**.
- ▶ $\beta_k \in \Delta_V$: **a topic** over the V words.
- ▶ Q : the **number of clusters**.
- ▶ K : the **number of topics**.
- ▶ N : the **number of nodes**.
- ▶ M : the **number of edges**.
- ▶ $\text{softmax}(x) = (\sum_{k=1}^K e^{x_k})^{-1} (e^{x_1}, \dots, e^{x_K})$,
 $\forall x \in \mathbb{R}^K$.
- ▶ $\mathbf{A} \in \mathcal{M}_{N \times N}(\{0, 1\})$: the binary adjacency matrix,
 $A_{ij} = 1$ if i is connected to j .
- ▶ $\mathbf{W} = (W_{ij})_{ij}$: the documents,
 W_{ij} the document sent from i to j .

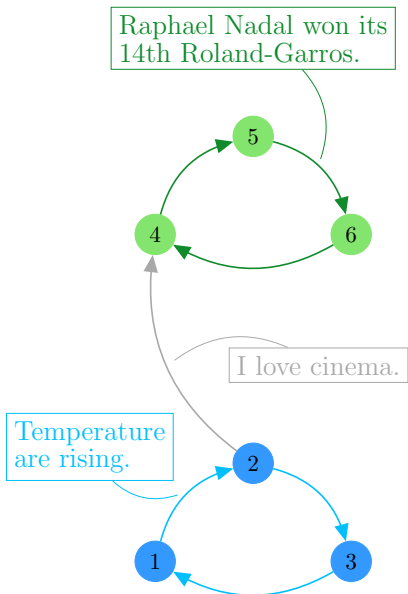


Table of contents

1. Introduction
2. Generative model
3. Inference and model selection
4. Evaluation of Deep LPTM on synthetic data
5. Real world use-case: Deep LPTM applied to the ENRON dataset
6. Conclusion

Node generation

Based on the latent position cluster model¹, C_i the cluster membership of node i for all $i \in \{1, \dots, M\}$

$$C_i \stackrel{i.i.d}{\sim} \mathcal{M}_Q(1, \pi). \quad (1)$$

where Q corresponds to the number of clusters.

The latent vector representing node i , denoted Z_i , is assumed to be Gaussian:

$$Z_i \mid C_{iq} = 1 \sim \mathcal{N}_p(\mu_q, \sigma_q^2 I_p). \quad (2)$$

Denoting $\eta_{ij} := \kappa - \|Z_i - Z_j\|$, the probability for node i to be connected to node j is

$$P(A_{ij} = 1 \mid Z_i, Z_j, \kappa) = \frac{1}{1 + e^{-\eta_{ij}}}. \quad (3)$$

¹Handcock et al. (2007).

Node generation

Based on the latent position cluster model¹, C_i the cluster membership of node i for all $i \in \{1, \dots, M\}$

$$C_i \stackrel{i.i.d}{\sim} \mathcal{M}_Q(1, \pi). \quad (1)$$

where Q corresponds to the number of clusters.

The latent vector representing node i , denoted Z_i , is assumed to be Gaussian:

$$Z_i \mid C_{iq} = 1 \sim \mathcal{N}_p(\mu_q, \sigma_q^2 I_p). \quad (2)$$

Denoting $\eta_{ij} := \kappa - \|Z_i - Z_j\|$, the probability for node i to be connected to node j is

$$P(A_{ij} = 1 \mid Z_i, Z_j, \kappa) = \frac{1}{1 + e^{-\eta_{ij}}}. \quad (3)$$

¹Handcock et al. (2007).

Node generation

Based on the latent position cluster model¹, C_i the cluster membership of node i for all $i \in \{1, \dots, M\}$

$$C_i \stackrel{i.i.d}{\sim} \mathcal{M}_Q(1, \pi). \quad (1)$$

where Q corresponds to the number of clusters.

The latent vector representing node i , denoted Z_i , is assumed to be Gaussian:

$$Z_i \mid C_{iq} = 1 \sim \mathcal{N}_p(\mu_q, \sigma_q^2 I_p). \quad (2)$$

Denoting $\eta_{ij} := \kappa - \|Z_i - Z_j\|$, the probability for node i to be connected to node j is

$$P(A_{ij} = 1 \mid Z_i, Z_j, \kappa) = \frac{1}{1 + e^{-\eta_{ij}}}. \quad (3)$$

¹Handcock et al. (2007).

Text generation in Deep-LPTM²

1. $Y_{ij} \mid A_{ij} C_{iq} C_{jr} = 1 \sim \mathcal{N}_K(m_{qr}, s_{qr}^2 I_K),$
2. $\theta_{ij} = \text{softmax}(Y_{ij}),$ proportions of topic in documents sent from i to j
3. $W_{ij} \mid A_{ij} = 1, \theta_{ij} \sim \mathcal{M}_V(M_{ij}, \theta_{ij}^\top \beta),$ where $\beta = (\beta_1 \cdots \beta_K)^\top \in \mathcal{M}_{K \times V}(\mathbb{R})$ is the vocabulary matrix and

$$\theta_{ij}^\top \beta = \sum_{k=1}^K \theta_{ijk} \beta_k, \in \mathbb{R}^V.$$

Each topic k , represented by $\beta_k \in \Delta_V$, is obtained by computing:

$$\beta_k = \text{softmax}(\rho^\top \alpha_k),$$

- ▶ $\rho \in \mathcal{M}_{L \times V}(\mathbb{R})$ L -dimensional word embeddings
- ▶ $\alpha = (\alpha_1 \cdots \alpha_K) \in \mathcal{M}_{L \times K}(\mathbb{R})$ L -dimensional topic embeddings

²based on Dieng et al. (2020).

Text generation in Deep-LPTM²

1. $Y_{ij} \mid A_{ij} C_{iq} C_{jr} = 1 \sim \mathcal{N}_K(m_{qr}, s_{qr}^2 I_K)$,
2. $\theta_{ij} = \text{softmax}(Y_{ij})$, proportions of topic in documents sent from i to j
3. $W_{ij} \mid A_{ij} = 1, \theta_{ij} \sim \mathcal{M}_V(M_{ij}, \theta_{ij}^\top \beta)$, where $\beta = (\beta_1 \cdots \beta_K)^\top \in \mathcal{M}_{K \times V}(\mathbb{R})$ is the vocabulary matrix and

$$\theta_{ij}^\top \beta = \sum_{k=1}^K \theta_{ijk} \beta_k, \in \mathbb{R}^V.$$

Each topic k , represented by $\beta_k \in \Delta_V$, is obtained by computing:

$$\beta_k = \text{softmax}(\rho^\top \alpha_k),$$

- ▶ $\rho \in \mathcal{M}_{L \times V}(\mathbb{R})$ L -dimensional word embeddings
- ▶ $\alpha = (\alpha_1 \cdots \alpha_K) \in \mathcal{M}_{L \times K}(\mathbb{R})$ L -dimensional topic embeddings

²based on Dieng et al. (2020).

Text generation in Deep-LPTM²

1. $Y_{ij} \mid A_{ij} C_{iq} C_{jr} = 1 \sim \mathcal{N}_K(m_{qr}, s_{qr}^2 I_K)$,
2. $\theta_{ij} = \text{softmax}(Y_{ij})$, proportions of topic in documents sent from i to j
3. $W_{ij} \mid A_{ij} = 1, \theta_{ij} \sim \mathcal{M}_V(M_{ij}, \theta_{ij}^\top \beta)$, where $\beta = (\beta_1 \cdots \beta_K)^\top \in \mathcal{M}_{K \times V}(\mathbb{R})$ is the vocabulary matrix and

$$\theta_{ij}^\top \beta = \sum_{k=1}^K \theta_{ijk} \beta_k, \in \mathbb{R}^V.$$

Each topic k , represented by $\beta_k \in \Delta_V$, is obtained by computing:

$$\beta_k = \text{softmax}(\rho^\top \alpha_k),$$

- ▶ $\rho \in \mathcal{M}_{L \times V}(\mathbb{R})$ L -dimensional word embeddings
- ▶ $\alpha = (\alpha_1 \cdots \alpha_K) \in \mathcal{M}_{L \times K}(\mathbb{R})$ L -dimensional topic embeddings

²based on Dieng et al. (2020).

To summarise: Deep-LPTM graphical model

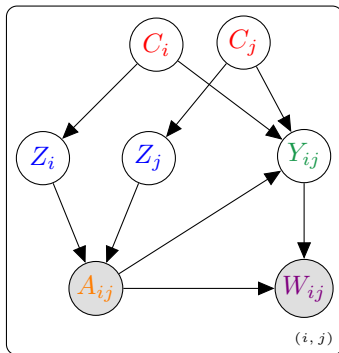


Figure: Deep-LPTM graphical representation.

- ▶ C_i node cluster membership
- ▶ Z_i node latent representation
- ▶ Y_{ij} text latent representation

⇒ Node cluster memberships bridge the gap between textual data and node representation.

- ▶ C_i node cluster membership
- ▶ Z_i node latent representation
- ▶ Y_{ij} text latent representation

Marginal likelihood

Denoting Θ the set of all model parameters,

$$\log p(\mathbf{A}, \mathbf{W} \mid \Theta) = \log \left(\sum_{\mathbf{C}} \int_{\mathbf{Z}} \int_{\mathbf{Y}} p(\mathbf{A}, \mathbf{W}, \mathbf{C}, \mathbf{Z}, \mathbf{Y} \mid \Theta) d\mathbf{Z} d\mathbf{Y} \right). \quad (4)$$

This quantity is not tractable since the sum over all configurations requires to compute Q^N terms. Besides, it involves integrals that cannot be computed analytically.

→ **Variational inference** for approximation purposes.

- ▶ C_i node cluster membership
- ▶ Z_i node latent representation
- ▶ Y_{ij} text latent representation

Marginal likelihood

Denoting Θ the set of all model parameters,

$$\log p(\mathbf{A}, \mathbf{W} \mid \Theta) = \log \left(\sum_{\mathbf{C}} \int_{\mathbf{Z}} \int_{\mathbf{Y}} p(\mathbf{A}, \mathbf{W}, \mathbf{C}, \mathbf{Z}, \mathbf{Y} \mid \Theta) d\mathbf{Z} d\mathbf{Y} \right). \quad (4)$$

This quantity is not tractable since the sum over all configurations requires to compute Q^N terms. Besides, it involves integrals that cannot be computed analytically.

→ **Variational inference** for approximation purposes.

- ▶ C_i node cluster membership
- ▶ Z_i node latent representation
- ▶ Y_{ij} text latent representation

Marginal likelihood

Denoting Θ the set of all model parameters,

$$\log p(\mathbf{A}, \mathbf{W} \mid \Theta) = \log \left(\sum_{\mathbf{C}} \int_{\mathbf{Z}} \int_{\mathbf{Y}} p(\mathbf{A}, \mathbf{W}, \mathbf{C}, \mathbf{Z}, \mathbf{Y} \mid \Theta) d\mathbf{Z} d\mathbf{Y} \right). \quad (4)$$

This quantity is not tractable since the sum over all configurations requires to compute Q^N terms. Besides, it involves integrals that cannot be computed analytically.

→ **Variational inference** for approximation purposes.

Inference

- ▶ C_i node cluster membership
- ▶ Z_i node latent representation
- ▶ Y_{ij} text latent representation

The **variational inference** consists in splitting the likelihood in two terms. For any distribution $R(\mathbf{C}, \mathbf{Z}, \mathbf{Y})$,

$$\log p(\mathbf{A}, \mathbf{W} \mid \Theta) = \mathcal{L}(R(\cdot); \Theta) + \text{KL}(R(\cdot) \parallel p(\mathbf{C}, \mathbf{Z}, \mathbf{Y} \mid \mathbf{A}, \mathbf{W})), \quad (5)$$

where

$$\mathcal{L}(R(\cdot); \Theta) = \mathbb{E}_R \left[\log \frac{p(\mathbf{A}, \mathbf{W}, \mathbf{C}, \mathbf{Z}, \mathbf{Y} \mid \Theta)}{R(\mathbf{C}, \mathbf{Z}, \mathbf{Y})} \right]. \quad (6)$$

- ▶ C_i node cluster membership
- ▶ Z_i node latent representation
- ▶ Y_{ij} text latent representation

The **variational inference** consists in splitting the likelihood in two terms. For any distribution $R(\mathbf{C}, \mathbf{Z}, \mathbf{Y})$,

$$\log p(\mathbf{A}, \mathbf{W} \mid \Theta) = \mathcal{L}(R(\cdot); \Theta) + \text{KL}(R(\cdot) \parallel p(\mathbf{C}, \mathbf{Z}, \mathbf{Y} \mid \mathbf{A}, \mathbf{W})), \quad (5)$$

where

$$\mathcal{L}(R(\cdot); \Theta) = \mathbb{E}_R \left[\log \frac{p(\mathbf{A}, \mathbf{W}, \mathbf{C}, \mathbf{Z}, \mathbf{Y} \mid \Theta)}{R(\mathbf{C}, \mathbf{Z}, \mathbf{Y})} \right]. \quad (6)$$

Inference

Assumptions regarding the variational distributions:

$$R(\mathbf{C}, \mathbf{Z}, \mathbf{Y} \mid \mathbf{A}, \mathbf{W}) = R(\mathbf{C})R(\mathbf{Z} \mid \mathbf{A})R(\mathbf{Y} \mid \mathbf{A}, \mathbf{W}),$$

- ▶ C_i node cluster membership
- ▶ Z_i node latent representation
- ▶ Y_{ij} text latent representation

Assumptions regarding the variational distributions:

$$R(\mathbf{C}, \mathbf{Z}, \mathbf{Y} \mid \mathbf{A}, \mathbf{W}) = R(\mathbf{C})R(\mathbf{Z} \mid \mathbf{A})R(\mathbf{Y} \mid \mathbf{A}, \mathbf{W}),$$

$$R(\mathbf{C}) = \prod_{i=1}^N R_{\tau_i}(C_i) = \prod_{i=1}^N \mathcal{M}_Q(C_i; 1, \tau_i),$$

- ▶ C_i node cluster membership
- ▶ Z_i node latent representation
- ▶ Y_{ij} text latent representation

Assumptions regarding the variational distributions:

- ▶ C_i node cluster membership
- ▶ Z_i node latent representation
- ▶ Y_{ij} text latent representation

$$R(\mathbf{C}, \mathbf{Z}, \mathbf{Y} \mid \mathbf{A}, \mathbf{W}) = R(\mathbf{C})R(\mathbf{Z} \mid \mathbf{A})R(\mathbf{Y} \mid \mathbf{A}, \mathbf{W}),$$

$$R(\mathbf{C}) = \prod_{i=1}^N R_{\tau_i}(C_i) = \prod_{i=1}^N \mathcal{M}_Q(C_i; 1, \tau_i),$$

$$R(\mathbf{Z} \mid \mathbf{A}) = \prod_{i=1}^N R_{\phi_Z}(Z_i \mid \mathbf{A}) = \prod_{i=1}^N \mathcal{N}_p(Z_i; \mu_{\phi_Z}(\mathbf{A})_i, \sigma_{\phi_Z}^2(\mathbf{A})_i I_p),$$

Assumptions regarding the variational distributions:

- ▶ C_i node cluster membership
- ▶ Z_i node latent representation
- ▶ Y_{ij} text latent representation

$$R(\mathbf{C}, \mathbf{Z}, \mathbf{Y} \mid \mathbf{A}, \mathbf{W}) = R(\mathbf{C})R(\mathbf{Z} \mid \mathbf{A})R(\mathbf{Y} \mid \mathbf{A}, \mathbf{W}),$$

$$R(\mathbf{C}) = \prod_{i=1}^N R_{\tau_i}(C_i) = \prod_{i=1}^N \mathcal{M}_Q(C_i; 1, \tau_i),$$

$$R(\mathbf{Z} \mid \mathbf{A}) = \prod_{i=1}^N R_{\phi_Z}(Z_i \mid \mathbf{A}) = \prod_{i=1}^N \mathcal{N}_p(Z_i; \mu_{\phi_Z}(\mathbf{A})_i, \sigma_{\phi_Z}^2(\mathbf{A})_i I_p),$$

$$R(\mathbf{Y} \mid \mathbf{A}, \mathbf{W}) = \prod_{i \neq j} R_{\phi_Y}(Y_{ij} \mid W_{ij})^{A_{ij}} = \prod_{i \neq j} \mathcal{N}_K(Y_{ij}; \mu_{\phi_Y}(W_{ij}), \text{diag}(\sigma_{\phi_Y}^2(W_{ij})))^{A_{ij}},$$

where $(\mu_{\phi_Z}, \sigma_{\phi_Z}^2)$ are the outputs of the encoder of a **variational graph auto encoder**³ and $(\mu_{\phi_Y}, \sigma_{\phi_Y}^2)$ the outputs of **ETM encoder**.

³Kipf, Welling (2016).

Details about VGAE⁴

Denoting $\tilde{\mathbf{A}} = \mathbf{D}^{-1/2}(\mathbf{A} + \mathbf{I}_N)\mathbf{D}^{-1/2}$, the graph convolutional network can be summarised as

⁴Kipf, Welling (2016).

Denoting $\tilde{\mathbf{A}} = \mathbf{D}^{-1/2}(\mathbf{A} + \mathbf{I}_N)\mathbf{D}^{-1/2}$, the graph convolutional network can be summarised as

$$\begin{aligned}\mu_\phi(\mathbf{A}) &= \tilde{\mathbf{A}} \operatorname{ReLU}(\tilde{\mathbf{A}}\mathbf{\Omega}_0)\mathbf{\Omega}_\mu, \\ \log \sigma_\phi^2(\mathbf{A}) &= \tilde{\mathbf{A}} \operatorname{ReLU}(\tilde{\mathbf{A}}\mathbf{\Omega}_0)\mathbf{\Omega}_\sigma,\end{aligned}$$

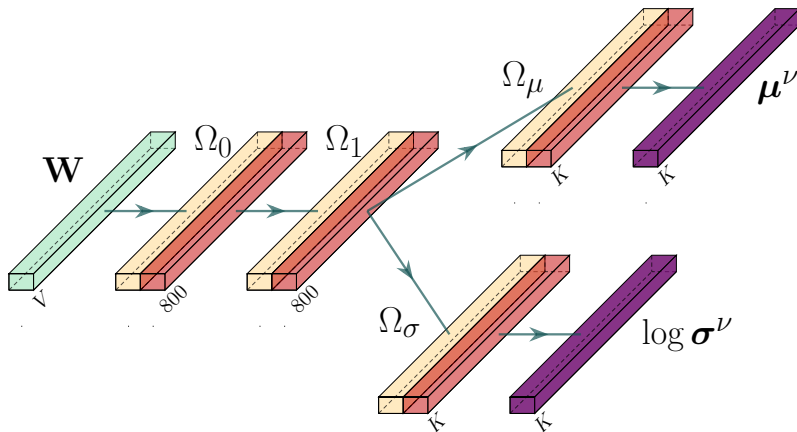
where

- ▶ $\operatorname{ReLU}(x) = (\max(0, x_1), \dots, \max(0, x_F))$ if $x \in \mathbb{R}^F$,
- ▶ $\mathbf{\Omega}_0 \in \mathcal{M}_{N \times D}(\mathbb{R})$ with $D = 64$ in all the experiments we carried out,
- ▶ $\mathbf{\Omega}_\mu, \mathbf{\Omega}_\sigma \in \mathcal{M}_{D \times (Q-1)}(\mathbb{R})$.

⁴Kipf, Welling (2016).

Neural network encoding the documents

Figure: Representation of the neural network mapping the documents to the variational parameters.



Thanks to the previous assumptions, the ELBO is given by:

$$\begin{aligned}\mathcal{L}(R(\cdot); \boldsymbol{\alpha}, \boldsymbol{\rho}) = & \mathbb{E}_R [\log p(\mathbf{W} \mid \mathbf{C}, \mathbf{A}, \mathbf{Y}, \boldsymbol{\alpha}, \boldsymbol{\rho})] \\ & + \mathbb{E}_R [\log p(\mathbf{Y})] - \mathbb{E}_R [\log R(\mathbf{Y})] \\ & + \mathbb{E}_R [\log p(\mathbf{A} \mid \mathbf{Z}, \kappa)] \\ & + \mathbb{E}_R [\log p(\mathbf{Z})] - \mathbb{E}_R [\log R(\mathbf{Z})] \\ & + \mathbb{E}_R [\log p(\mathbf{C} \mid \pi)] - \mathbb{E}_R [\log R(\mathbf{C})] .\end{aligned}$$

Proposition

The parameters of the node embedding distributions maximising the ELBO are given by:

$$\mu_q = \frac{1}{N_q} \sum_{i=1}^N \tau_{iq} \mu_{\phi_Z}(\mathbf{A})_i, \quad (7)$$

$$\sigma_q^2 = \frac{1}{pN_q} \sum_{i=1}^N \tau_{iq} \left(\|\mu_{\phi_Z}(\mathbf{A})_i - \mu_q\|_2^2 + p\sigma_{\phi_Z}^2(\mathbf{A})_i \right), \quad (8)$$

where $N_q = \sum_{i=1}^N \tau_{iq}$ is the posterior mean of the number of nodes in cluster q .

Proposition

The parameters of the edge embedding distributions maximising the ELBO are given by:

$$m_{qr} = \frac{1}{N_{qr}} \sum_{i,j=1}^N A_{ij} \tau_{iq} \tau_{jr} \mu_{\phi_Y}(W_{ij}), \quad (9)$$

$$s_{qr}^2 = \frac{1}{KN_{qr}} \sum_{i,j=1}^N A_{ij} \tau_{iq} \tau_{jr} \left[\|\mu_{\phi_Y}(W_{ij}) - m_{qr}\|_2^2 + \sum_{k=1}^K \sigma_{\phi_Y}^2(W_{ij})_k \right], \quad (10)$$

where $N_{qr} = \sum_{i,j=1}^N A_{ij} \tau_{iq} \tau_{jr}$ denotes the expected number of documents sent from cluster q to cluster r under the approximated posterior distribution.

Optimisation: the clusters encapsulate both information

Proposition

The variational node cluster membership probability τ_{iq} maximising the ELBO is given by:

$$\tau_{iq} \propto \gamma_q \exp \left\{ -\text{KL}_q^{\mathbf{Z}_i} - \sum_{j \neq i} \sum_{r=1}^Q \left(A_{ij} \tau_{jr} \text{KL}_{qr}^{\mathbf{Y}_{ij}} + A_{ji} \tau_{jr} \text{KL}_{rq}^{\mathbf{Y}_{ji}} \right) \right\},$$

where

$$\begin{aligned} \text{KL}_q^{\mathbf{Z}_i} &= \text{KL} \left(\underbrace{\mathcal{N}_p(\mu_{\phi_Z}(\mathbf{A})_i, \sigma_{\phi_Z}^2(\mathbf{A})_i \mathbf{I}_p)}_{\text{variational distribution of node embedding}} \parallel \underbrace{\mathcal{N}_p(\mu_q, \sigma_q^2 \mathbf{I}_p)}_{\text{distribution of cluster } q \text{ embedding}} \right), \\ \text{KL}_{qr}^{\mathbf{Y}_{ij}} &= \text{KL} \left(\underbrace{\mathcal{N}_K(\mu_{\phi_Y}(\mathbf{W}_{ij}), \text{diag}(\sigma_{\phi_Y}^2(\mathbf{W}_{ij})))}_{\text{variational distribution of edge embedding}} \parallel \underbrace{\mathcal{N}_K(m_{qr}, s_{qr}^2 \mathbf{I}_K)}_{\text{distribution of document embedding sent from cluster } q \text{ to } r} \right). \end{aligned}$$

Optimisation of the encoders

The parameters of the **graph convolutional network encoder** as well as the parameters of the **encoder of the neural topic model** are optimised using a MC estimate of the gradient obtained ... **it cannot be done directly !**

The reparametrisation trick⁵

How to compute the gradient $\frac{\partial}{\partial \phi_Y} \mathcal{L}(R(\cdot); \alpha, \rho)$?

$$\frac{\partial}{\partial \phi_Y} \mathcal{L}(R(\cdot); \alpha, \rho) = \frac{\partial}{\partial \phi_Y} \mathbb{E}_R [\log p(\mathbf{W} \mid \mathbf{C}, \mathbf{A}, \mathbf{Y}, \alpha, \rho)] - \frac{\partial}{\partial \phi_Y} \overbrace{\text{KL}(R(\mathbf{Y}) \mid p(\mathbf{Y}))}^{\text{analytical form}}.$$

Since $R(\cdot)$ depends on ϕ_Y , we cannot interchange the derivative and the integral in the term on the left-hand side.

The reparametrisation trick removes this dependency by sampling $\epsilon \sim \mathcal{N}_K(0, \mathbf{I}_K)$ and taking $Y_{ij} = \mu_{ij}^{\phi_Y}(\mathbf{A}) + \sigma_{ij}^{\phi_Y}(\mathbf{A})\epsilon$, such that the following holds:

$$\begin{aligned} \frac{\partial}{\partial \phi_Y} \mathbb{E}_R [\log p(\mathbf{W} \mid \mathbf{C}, \mathbf{A}, \mathbf{Y}, \alpha, \rho)] &= \frac{\partial}{\partial \phi_Y} \mathbb{E}_\epsilon [\mathbb{E}_C [\log p(\mathbf{W} \mid \mathbf{C}, \mathbf{A}, \mathbf{Y}, \alpha, \rho)]] \\ &= \mathbb{E}_\epsilon \left[\frac{\partial}{\partial \phi_Y} \mathbb{E}_C [\log p(\mathbf{W} \mid \mathbf{C}, \mathbf{A}, \mathbf{Y}, \alpha, \rho)] \right]. \end{aligned}$$

A Monte-Carlo estimate of this last expression can now be computed.

⁵Kingma, Welling (2014); Rezende et al. (2014).

Model selection

Our criterion:

- ▶ C_i node cluster membership
- ▶ Z_i node latent representation
- ▶ Y_{ij} text latent representation

$$\log p(\mathbf{A}, \mathbf{W}, \mathbf{C}, \mathbf{Z}, \mathbf{Y} \mid \mathcal{M}, Q, K, P) = \log \int_{\theta} p(\mathbf{A}, \mathbf{W}, \mathbf{C}, \mathbf{Z}, \mathbf{Y} \mid \theta, \mathcal{M}, Q, K, P) p(\theta) d\theta.$$

→ this quantity is intractable. Therefore, we estimate it using a BIC-like approximation.

Proposed estimate:

$$\text{IC2L}(\mathcal{M}, Q, K, P, \hat{\mathbf{C}}, \hat{\mathbf{Z}}, \hat{\mathbf{Y}}) = \max_{\theta} \log p(\mathbf{A}, \mathbf{W}, \hat{\mathbf{C}}, \hat{\mathbf{Z}}, \hat{\mathbf{Y}} \mid \theta, \mathcal{M}, Q, K, P) - \Omega(\mathcal{M}, Q, K, P),$$

with $\hat{\mathbf{C}}$, $\hat{\mathbf{Z}}$ and $\hat{\mathbf{Y}}$ the **maximum-a-posteriori** estimates, and $\Omega(\mathcal{M}, Q, K, P)$ the **penalty** from BIC-like approximations.

Our criterion:

- ▶ C_i node cluster membership
- ▶ Z_i node latent representation
- ▶ Y_{ij} text latent representation

$$\log p(\mathbf{A}, \mathbf{W}, \mathbf{C}, \mathbf{Z}, \mathbf{Y} \mid \mathcal{M}, Q, K, P) = \log \int_{\theta} p(\mathbf{A}, \mathbf{W}, \mathbf{C}, \mathbf{Z}, \mathbf{Y} \mid \theta, \mathcal{M}, Q, K, P) p(\theta) d\theta.$$

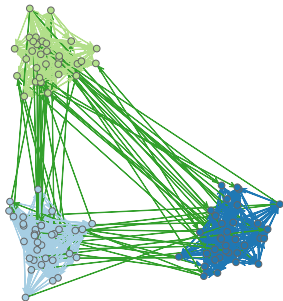
→ this quantity is intractable. Therefore, we estimate it using a BIC-like approximation.

Proposed estimate:

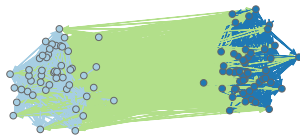
$$\text{IC2L}(\mathcal{M}, Q, K, P, \hat{\mathbf{C}}, \hat{\mathbf{Z}}, \hat{\mathbf{Y}}) = \max_{\theta} \log p(\mathbf{A}, \mathbf{W}, \hat{\mathbf{C}}, \hat{\mathbf{Z}}, \hat{\mathbf{Y}} \mid \theta, \mathcal{M}, Q, K, P) - \Omega(\mathcal{M}, Q, K, P),$$

with $\hat{\mathbf{C}}$ $\hat{\mathbf{Z}}$ and $\hat{\mathbf{Y}}$ the **maximum-a-posteriori** estimates, and $\Omega(\mathcal{M}, Q, K, P)$ the **penalty** from BIC-like approximations.

Scenario A



Scenario B



Scenario C

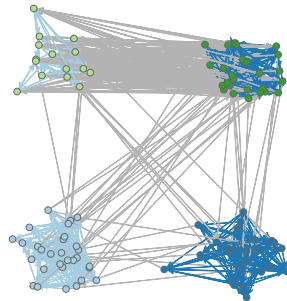


Figure: Networks sampled from each scenario. The node colours denote the node cluster memberships and the edge colours denote the majority topic in the corresponding documents.

Simulations - Scenario A, B and C

Table: Presentation of our three scenarii to evaluate our model.

	Scenario A	Scenario B	Scenario C
Q (clusters)	3	2	4
K (topics)	4	3	3
Communities	3	1	3
π_{qr} (community probability) $\eta = 0.25, \epsilon = 0.01$	$\begin{pmatrix} \eta & \epsilon & \epsilon \\ \epsilon & \eta & \epsilon \\ \epsilon & \epsilon & \eta \end{pmatrix}$	$\begin{pmatrix} \eta & \eta \\ \eta & \eta \end{pmatrix}$	$\begin{pmatrix} \eta & \epsilon & \epsilon & \epsilon \\ \epsilon & \eta & \epsilon & \epsilon \\ \epsilon & \epsilon & \eta & \eta \\ \epsilon & \epsilon & \eta & \eta \end{pmatrix}$
Topic between q and r	$\begin{pmatrix} t_1 & t_4 & t_4 \\ t_4 & t_2 & t_4 \\ t_4 & t_4 & t_3 \end{pmatrix}$	$\begin{pmatrix} t_1 & t_3 \\ t_3 & t_2 \end{pmatrix}$	$\begin{pmatrix} t_1 & t_3 & t_3 & t_3 \\ t_3 & t_2 & t_3 & t_3 \\ t_3 & t_3 & t_1 & t_3 \\ t_3 & t_3 & t_3 & t_2 \end{pmatrix}$
Sufficient information to find the clusters	Network	Topic	Network & Topics

Noise in the Hard Scenario

- ▶ node i (j resp.) in cluster q (r resp.)
- ▶ topic proportion $\theta_{ij}^* = (0, \dots, 0, 1, 0, \dots, 0)$ with 1 on the corresponding topic
- ▶ $\zeta = 0$: pure topic, $\zeta = 1$: uniform distribution over topics
- ▶ $\eta = 0.1$ instead of 0.25

$$\theta_{ij} = (1 - \zeta)\theta_{ij}^* + \zeta * \left(\frac{1}{K}, \dots, \frac{1}{K}\right)^\top. \quad (11)$$

Simulations - Detailed example with three communities

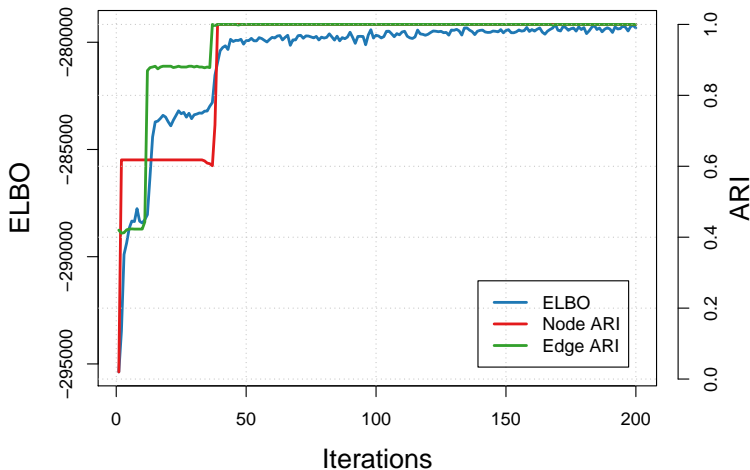


Figure: Evolution of the ARIs and the ELBO during the iterations of the optimisation procedure.

Simulations - Detailed example with three communities

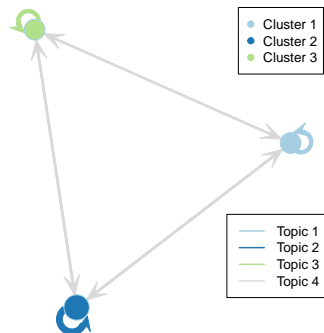


Figure: The meta-network obtained with Deep-LPTM on a scenario A .

	Topic 1	Topic 2	Topic 3	Topic 4
1	cancer	black	princess	seats
2	cell	hole	birth	david
3	occur	gravity	charlotte	political
4	genes	light	cambridge	lost
5	cancers	shadow	queen	kingdom
6	due	credit	granddaughter	black
7	mutations	event	duchess	party
8	radiation	disc	palace	part
9	princess	princess	london	resentment
10	include	horizon	great	united

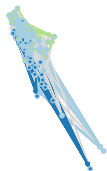
Table: Topics of the model in Scenario A Easy, represented by the 10 most probable words per topic.

Simulations - Detailed example with three communities

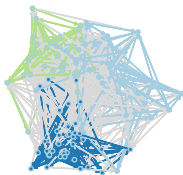
Iteration 1



Iteration 333



Iteration 666



Iteration 1000

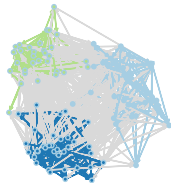


Figure: Evolution of the node embeddings during training.

IC2L model selection results for different triplets (K, P, Q)

	$K = 2$	$K = 3$	$K = 4$	$K = 5$	$K = 6$
$Q = 2$	0	0	0	0	0
$Q = 3$	0	0	0	0	0
$Q = 4$	0	10	0	0	0
$Q = 5$	0	0	0	0	0
$Q = 6$	0	0	0	0	0

Table: Number of times a triplet (K, P, Q) is associated with the highest IC2L over 10 graphs simulated according to Scenario C ($Q^* = 4$ and $K^* = 3$). All the models with the highest IC2L value correspond to $P = 2$. Therefore, only the table corresponding to this value is shown.

		Scenario A	Scenario B	Scenario C
Easy	ETSBM	0.99 ± 0.03	1.00 ± 0.00	0.96 ± 0.04
	ETSBM - PT	1.00 ± 0.00	1.00 ± 0.00	0.96 ± 0.05
	Deep-LPTM	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
	Deep-LPTM - PT	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
Hard	ETSBM	0.96 ± 0.10	0.90 ± 0.30	0.72 ± 0.25
	ETSBM - PT	0.99 ± 0.01	1.00 ± 0.00	0.74 ± 0.21
	Deep-LPTM	0.99 ± 0.02	1.00 ± 0.00	0.89 ± 0.15
	Deep-LPTM - PT	1.00 ± 0.01	1.00 ± 0.00	0.85 ± 0.18

Table: ARI of the node clustering over 10 graphs in three scenarios for the two levels of difficulty Easy and Hard. Deep-LPTM, as well as ETSBM, are presented with and without pre-trained embeddings (denoted PT)

Context: ENRON was an American gas selling company in North America. In December 2001, the company filed for the largest bankruptcy at that time. The emails of the company were made public by the federal energy regulatory commission (FERC).

Preprocessing of the dataset:

- ▶ We kept the emails sent between September and December 2001.
- ▶ Concatenation of emails sent from one account to the other.
- ▶ Number of employees (= nodes): 149.
- ▶ Number of documents (= edges): 1,200 documents from 21,000 emails.
- ▶ IC2L was computed for $Q \in \{5, 7, 10\}$, $K \in \{3, 5, 7, 10\}$, $P \in \{2, 4, 8, 16\}$. The highest value was obtained for:

$$\hat{Q}_{\text{IC2L}}, \hat{K}_{\text{IC2L}}, \hat{P}_{\text{IC2L}} = (7, 10, 2)$$

Real world example: ENRON email dataset

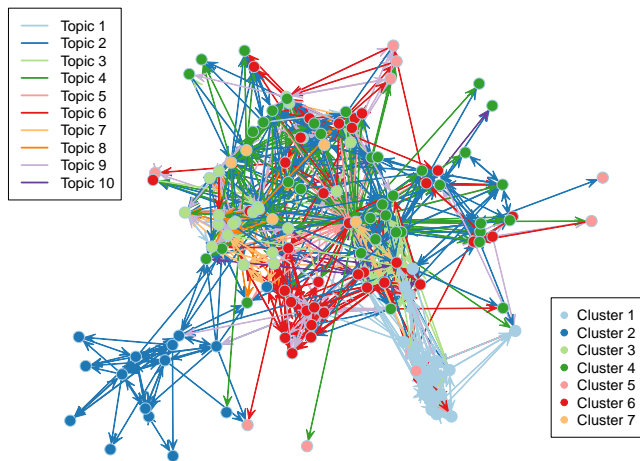


Figure: Deep-LPTM representation of Enron email network. The node cluster memberships are denoted by the colour of the nodes and the majority topic in the documents are denoted by the colour of the edges.

Real world example: ENRON email dataset

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
tw	ercot	rto	backup	ofo
watson	vepco	steffes	seat	interview
hayslett	ene	christi	location	cycle
donoho	liz	nicolay	test	mmbtu
lindy	dyn	novosel	supplies	usage
geaccone	filename	affairs	building	interviewers
lynn	mws	rtos	floors	fantastic
transwestern	desk	shapiro	mails	super
teb	mw	government	notified	deliveries
lohman	enpower	skilling	seats	dinner
Topic 6	Topic 7	Topic 8	Topic 9	Topic 10
sara	frontier	grigsby	master	edison
shackleton	western	desk	nymex	puc
kim	williams	mike	handling	dwr
ward	dt	taleban	isda	davis
master	project	forces	executed	dasovich
isda	whitt	sheppard	agreement	sce
perlingiere	dth	afghanistan	netting	da
perlingiereenron	enw	holst	multicurrency	state
leathercenter	marathon	gaskill	na	california
shackletonenron	cheyenne	ina	cn	jeff

Figure: The 10 most probable words of each topic according to Deep-LPTM.

Real world example: ETSBM

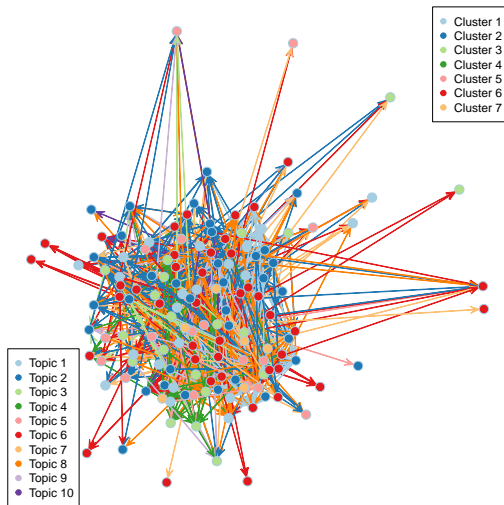


Figure: ETSBM representation of Enron email network. The node cluster memberships are denoted by the colour of the nodes and the majority topic in the documents are denoted by the colour of the edges. 34/38

Real world example: ENRON email dataset

Topic 1

tw
watson
message
gas
mmbtu
capacity
deliveries
original
lynn
socialgas

Topic 6

message
enron
original
jim
ferc
steffes
rto
group
market
energy

Topic 2

enron
message
original
company
mail
november
pmt
amto
fw
trading

Topic 7

backup
plan
seat
work
location
west
enron
day
team
move

Topic 3

mike
message
original
grigsby
desk
john
pmt
october
daily
deals

Topic 8

message
enron
original
gas
november
october
pmt
amto
mail
monday

Topic 4

enron
message
master
corp
original
agreement
attached
october
america
north

Topic 9

day
ofo
gas
cycle
storage
usage
daily
social
mmcf
scheduled

Topic 5






business
interview
enron
friday
phase
interviewers
unit
super
units
dinner

Topic 10

jeff
state
california
edison
power
puc
dasovich
davis
message
original

Conclusion & further work

- ▶ The representation for communities works fine
- ▶ The clustering is efficient in the three studied settings
- ▶ Our model captures meaningful clusters both in terms of connections and topics
- ▶ Combining the block modelling approach with the representation power
- ▶ Improve the graph neural network with latest advancement
- ▶ Incorporate temporal information

-  Dieng, Ruiz, Blei (2020). “Topic modeling in embedding spaces”. In: *Transactions of the Association for Computational Linguistics* 8, pp. 439–453.
-  Handcock, Raftery, Tantrum (2007). “Model-based clustering for social networks”. In: *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 170(2), pp. 301–354.
-  Kingma, Welling (2014). *Auto-Encoding Variational Bayes*. eprint: [1312.6114](#).
-  Kipf, Welling (2016). “Variational graph auto-encoders”. In: *arXiv preprint arXiv:1611.07308*.
-  Rezende, Mohamed, Wierstra (2014). “Stochastic backpropagation and approximate inference in deep generative models”. In: *International conference on machine learning*. PMLR, pp. 1278–1286.

$$\begin{aligned}
 \widehat{\text{IC2L}}(\mathcal{M}, Q, K, P) = & \max_{\kappa} \log p(\mathbf{A} \mid \hat{\mathbf{Z}}, \kappa, \mathcal{M}) - \frac{1}{2} \log(N(N-1)) \\
 & + \max_{\mu, \sigma} \log p(\hat{\mathbf{Z}} \mid \hat{\mathbf{C}}, \mu, \sigma, \mathcal{M}, Q, P) - \frac{QP + Q}{2} \log(N) \\
 & + \max_{\rho, \alpha} \log p(\mathbf{W} \mid \mathbf{A}, \hat{\mathbf{Y}}, \rho, \alpha, \mathcal{M}) - \frac{VL + KL}{2} \log(M) \\
 & + \max_{\mathbf{m}, \mathbf{s}} \log p(\hat{\mathbf{Y}} \mid \mathbf{A}, \hat{\mathbf{C}}, \mathbf{m}, \mathbf{s}, \mathcal{M}, K) - \frac{Q^2 K + Q^2}{2} \log(M) \\
 & + \max_{\gamma} \log p(\hat{\mathbf{C}} \mid \gamma, \mathcal{M}, Q) - \frac{Q-1}{2} \log(N),
 \end{aligned}$$

with $\hat{\mathbf{Z}}$, $\hat{\mathbf{Y}}$ and, $\hat{\mathbf{C}}$ the maximum-a-posteriori estimates, and

$$\begin{aligned}
 \Omega(\mathcal{M}, Q, K, P) = & \frac{1}{2} \log(N(N-1)) \\
 & + \frac{Q(P+2)-1}{2} \log(N) + \frac{L(V+K) + Q^2(K+1)}{2} \log(M).
 \end{aligned}$$