# Strategies for Online Inference of Network Mixture

by

Hugo Zanghi, Franck Picard, Vincent Miele, and Christophe Ambroise

**Research Report No. 14**
**May 2008**

Statistics for Systems Biology Group
Jouy-en-Josas/Paris/Evry, France
http://genome.jouy.inra.fr/ssb/

# Strategies for Online Inference of Network Mixture

**Hugo Zanghi**
HUGO.ZANGHI@EXALEAD.COM
*Exalead, 10 place de la Madeleine,*
*75008 Paris, France*

**Franck Picard**
PICARD@BIOMSERV.UNIV-LYON1.FR

**Vincent Miele**
MIELE@BIOMSERV.UNIV-LYON1.FR
*Laboratoire Biométrie et Biologie Evolutive,*
*UMR CNRS 5558 - Univ. Lyon 1, F-69622, Villeurbanne, France*

**Christophe Ambroise**
AMBROISE@GENOPOLE.CNRS.FR
*Laboratoire Statistique et Génome,*
*UMR CNRS 8071-INRA 1152-UEVE F-91000 Evry, France*

**Editor:**

## Abstract

The statistical analysis of complex networks is a challenging task, as learning structure requires appropriate statistical models and efficient computational procedures. One research has been to develop mixture models for random graphs, and this strategy has been successful in uncovering structure in social or biological networks. The principle of these models is to assume that the distribution of the edge values follows a parametric distribution conditionnaly to a latent structure which is used to catch connectivity patterns. However, these methods suffer from relatively slow estimation procedures, since dependencies are complex and do not necessarily allow for computational simplifications. In this paper, we adapt online estimation strategies developed for the EM algorithm to the case of models for which the probability of the missing data conditionally to the available observations is not tractable. Our work focuses on two methods based for one on the SAEM algorithm for the other on variational methods. A simulation study is carried out to compare the different proposed algorithms with existing approaches, and a real data set based on a sample of the network of the US political websites is studied. We show that our online EM-based algorithms offer a good trade-off between precision and speed for parameter estimation of mixture distributions in the context of random graphs.

## 1. Introduction

Many scientific fields have recently focused on the analysis of networks. Physics, computer sciences, biology and social sciences, all those disciplines now deal with datasets that can be represented as networks, such as power grids, friendship and protein-protein interaction networks. The best example of this “network” revolution is probably the study of the Internet. From such networks, knowledge can be learned by studying their structure or topology. Many strategies have been developed for this purpose, and in this work we focus on model-based methods, *i.e.*, methods which rely on a statistical model that describes the distribution of connections in the network.

Considering a mixture of distributions is a strategy that has focused many attentions (Frank and Harary, 1982; Snijders and Nowicki, 1997; Newman and Leicht, 2007; Daudin et al., 2008). The basics of this strategy is to consider that nodes are spread among an unknown number of connectivity classes which are unknown themselves. Conditionally to the hidden variables (class labels), connections are still independent and Bernoulli distributed, but their marginal distribution is a mixture of Bernoulli distributions with no independence between the connections. Many names have been proposed for this model, and in the following, it will be denoted by MixNet, which is equivalent to the Block Clustering model of Snijders and Nowicki (1997). Block-Clustering for classical binary data can be dated back to the early work of Govaert (Govaert, 1977). In this article, we consider the general form of MixNet for which the conditional distribution of edges belongs to the exponential family.

When using MixNet one central question is the estimation of the parameters, and the associated optimization strategies. With the new challenge of analyzing complex network structures, one critical aspect of using this method on real data is its speed of execution and its ability to deal with networks made of tens of thousands of nodes, if not more. To this extent, Bayesian strategies are limited as they can handle networks with hundreds of nodes only (Snijders and Nowicki, 1997). One alternative strategy has been to propose heuristic-based algorithms (Newman and Leicht, 2007), and in this work, we focus on non Bayesian maximum likelihood methods as proposed by Daudin et al. (2008). Every proposed strategy face the same difficulty: the distribution $\Pr\{\mathbf{Z}|\mathbf{X}\}$ of the hidden variables $\mathbf{Z}$ (the set of indicator variables for classes of connectivity) conditionally to the observation $\mathbf{X}$ can not be factorized due to conditional dependency. The variational method proposed by Daudin et al. (2008) consists in approximating $\Pr\{\mathbf{Z}|\mathbf{X}\}$ with a mean-field approximation when it can not be computed directly. In this work we also consider a natural strategy based on the Monte Carlo simulation of $\Pr\{\mathbf{Z}|\mathbf{X}\}$, leading to a SAEM algorithm (Delyon et al., 1999).

We address the question of the acceleration of these estimation methods using *on-line* strategies which constitute an efficient alternative to classical batch algorithms when the dataset grows over time. Their application to mixture models have already been studied by many authors (Titterington, 1984; Wang and Zhao, 2002; Liu et al., 2006; MacQueen, 1967). Typical clustering algorithms include the on-line $k$-means algorithm (MacQueen, 1967). More recently Liu et al. (2006) modeled Internet traffic using a recursive EM algorithm for the estimation of Poisson mixture models. In a first section, we present the MixNet model and its associated log-likelihoods. Then we derive *on-line* strategies for SAEM and variational methods in Sections 2 and 3. These accelerations are compared in terms of computational time, parameter estimation and clustering efficiency using simulated datasets. Then the method is illustrated on the complex network of the US political websites.

## 2. A Mixture Model for Networks

### 2.1 Model and Notations

Let us define a random graph $G$, where $\mathcal{V}$ denotes the set of fixed vertices. Random edges are described by a set of random variables $\mathbf{X} = \{X_{ij}, (i,j) \in \mathcal{V}^2\}$ coding for the nature of connection between nodes $i$ and $j$. We suppose that nodes are spread among $Q$ hidden classes and we denote by $Z_{iq}$ the indicator variable such that $\{Z_{iq} = 1\}$ if node $i$ belongs to class $q$. We denote by $\mathbf{Z} = (\mathbf{Z}_1, \dots, \mathbf{Z}_n)$ the vector of classes which is random such that:

$$\mathbf{Z}_i \sim \mathcal{M}(1, \boldsymbol{\alpha} = \{\alpha_1, ..., \alpha_Q\}),$$

with $\boldsymbol{\alpha}$ the vector of proportions for classes.

**Conditional distribution.** MixNet is defined using the conditional distribution of edges given the label of the nodes. $X_{ij}$s are supposed to be conditionnaly independent:

$$\Pr\{\mathbf{X}|\mathbf{Z}; \boldsymbol{\eta}\} = \prod_{ij} \prod_{q,l} \{\Pr\{X_{ij}|Z_{iq}Z_{jl} = 1; \eta_{ql}\}\}^{Z_{iq}Z_{jl}},$$

and $\Pr\{X_{ij}|Z_{iq}Z_{jl} = 1; \eta_{ql}\}$ is supposed to belong to the exponential family, with natural parameter $\eta_{ql}$:

$$\log \Pr\{X_{ij}|Z_{iq}Z_{jl} = 1; \eta_{ql}\} = \eta_{ql}^t h(X_{ij}) - a(\eta_{ql}) + b(X_{ij}),$$

where $h(X_{ij})$ is the vector of sufficient statistics, $a$ a normalizing constant and $b$ a given function. Consequently, the conditional distribution of the graph is also from the exponential family:

$$\log \Pr\{\mathbf{X}|\mathbf{Z}; \boldsymbol{\eta}\} = \sum_{ij,ql} Z_{iq}Z_{jl}\eta_{ql}^t h(X_{ij}) - \sum_{ij,ql} Z_{iq}Z_{jl} a(\eta_{ql}) + \sum_{ij} b(X_{ij}).$$

Numerous classical distributions fit into this framework Mariadassou and Robin (2007). For example, when the only available information is the presence or the absence of an edge, then $X_{ij}$ is assumed to follow a Bernoulli distribution :

$$X_{ij}|Z_{iq}Z_{jl} = 1 \sim \mathcal{B}(\pi_{ql}) \begin{cases} \eta_{ql} = \log \frac{\pi_{ql}}{1-\pi_{ql}}, \\ h(X_{ij}) = X_{ij}, \\ a(\eta_{ql}) = \log(1 - \pi_{ql}), \\ b(X_{ij}) = 0. \end{cases}$$

If additional information is available to describe the connections between vertices, it may be integrated into the model. For example, the Poisson distribution might describe the intensity of the traffic between nodes. A typical example in web access log mining is the number of users going from a page $i$ to a page $j$. Another example is provided by co-authorship networks, for which valuation may describe the number of articles commonly published by the authors of the network. In those cases, we have

$$X_{ij}|Z_{iq}Z_{jl} = 1 \sim \mathcal{P}(\lambda_{ql}) \begin{cases} \eta_{ql} = \log \lambda_{ql}, \\ h(X_{ij}) = X_{ij}, \\ a(\eta_{ql}) = -\lambda_{ql}, \\ b(X_{ij}) = X_{ij}! \end{cases}$$

**Joint distribution.** Since MixNet is defined by its conditional distribution, we first check that the joint distribution also belongs to the exponential family. Estimating the parameters in such a distribution family is a well-studied problem that can be tackled via EM-related strategies. In our context the joint distribution is expressed as

$$\log \Pr\{\mathbf{X}, \mathbf{Z}; \boldsymbol{\beta}\} = \boldsymbol{\beta}^t T(\mathbf{X}, \mathbf{Z}) - A(\boldsymbol{\beta}) + B(\mathbf{X}).$$

This factorization can be done thanks to the following notations:

$$\begin{cases} N_q & = \sum_i Z_{iq}, \\ H_{ql}(\mathbf{X}, \mathbf{Z}) & = \sum_{ij} Z_{iq} Z_{jl} h(X_{ij}), \\ G_{ql}(\mathbf{Z}) & = \sum_{ij} Z_{iq} Z_{jl} = N_q N_l, \\ \alpha_q & = \exp(\omega_q) / \sum_l \exp(\omega_l). \end{cases}$$

Then we have:

$$\begin{cases} T(\mathbf{X}, \mathbf{Z}) & = (\{N_q\}, \{H_{ql}(\mathbf{X}, \mathbf{Z})\}, \{G_{ql}(\mathbf{Z})\}), \\ \boldsymbol{\beta} & = (\{\omega_q\}, \{\eta_{ql}\}, \{a(\eta_{ql})\}), \\ A(\boldsymbol{\beta}) & = n \log \sum_l \exp \omega_l, \\ B(\mathbf{X}) & = \sum_{ij} b(X_{ij}). \end{cases}$$

The sufficient statistics of the complete-data model are : the number of nodes in the classes ($N_q$), the characteristics of the between-group links ($H_{ql}$ through function $h$) and $G_{ql}$ the product of frequencies between classes. In the following we aim at estimating $\boldsymbol{\beta}$.

### 2.2 Likelihoods and on-line inference

Existing estimation strategies are based on maximum likelihood, and algorithms related to EM are used for optimization purposes. In this context, the main difficulty is to obtain good estimations in a reasonnable time for datasets which can be made of tens of thousands of nodes. The technical difficulty underlying the optimization procedure lies in the complex dependency structure that exists in the model since $\Pr\{\mathbf{Z}|\mathbf{X}\}$ can not be factorized and needs to be approximated (Daudin et al., 2008).

A first strategy to simplify the problem is to consider a classification EM-based strategy, where only the prediction of $\mathbf{Z}$ is considered, leaving apart the problem of computing $\Pr\{\mathbf{Z}|\mathbf{X}\}$. This strategy has been the subject of a previous work (Zanghi et al., 2007). It is known to give biased estimates, but is very efficient from a computational time point of view. Another possibility relies on the Stochastic Approximation EM approach (Delyon et al., 1999) which approximates $\Pr\{\mathbf{Z}|\mathbf{X}\}$ using Monte Carlo simulations. A last strategy relies on the so-called variationnal approach, which consists in approximating $\Pr\{\mathbf{Z}|\mathbf{X}\}$ by $\mathcal{R}(\mathbf{Z})$, a newly and more tractable distribution on the hidden variables. In this article, we suppose that this distribution can be factorized

$$\mathcal{R}(\mathbf{Z}) = \prod_i \mathcal{R}(\mathbf{Z}_i).$$

This notation will be used for the SAEM algorithm as well.

In the following, every strategy will be based on the maximization of $\mathcal{Q}(\boldsymbol{\beta}|\boldsymbol{\beta}')$ the conditional expectation of the complete-data log-likelihood defined such that:

$$\begin{aligned}\mathcal{Q}(\boldsymbol{\beta}|\boldsymbol{\beta}') &= \mathbb{E}_{\mathbb{P}_{\beta'}}\left\{\log \mathcal{L}_C(\mathbf{X},\mathbf{Z};\boldsymbol{\beta})|\mathbf{X}\right\} \\ &\simeq \mathbb{E}_{\mathcal{R}}\left\{\log \mathcal{L}_C(\mathbf{X},\mathbf{Z};\boldsymbol{\beta})\right\},\end{aligned}$$

where $\mathbb{P}_{\beta'}$ stands for the conditional distribution $\Pr\{\mathbf{Z}|\mathbf{X};\beta'\}$, which is approximated by $\mathcal{R}(\mathbf{Z})$. The SAEM and the variational strategy are considered to solve the problem of maximum likelihood for models with complex dependency structure. We then need to assess the problem of computational efficiency, which is done using on-line strategies for the aforementionned algorithms.

**Principle of on-line methods** On-line algorithms are incremental algorithms which recursively update parameters, using current parameters and new observations. Their principle is to optimize $\mathcal{Q}(\boldsymbol{\beta}|\boldsymbol{\beta}')$ sequentially, while nodes are added. We introduce the following notations. Let us denote by $\mathbf{X}^{[n]} = \{X_{ij}\}_{i,j=1}^{n}$, the adjacency matrix of the data, when $n$ nodes are present, and by $\mathbf{Z}^{[n]}$ the associated labels. In the following, a convenient notation is also $\mathbf{X}_{i,\bullet} = \{X_{ij},\, j \in \mathcal{V}\}$, which denotes all the edges related to node $i$. Then we define: $\mathcal{Q}_n\left(\boldsymbol{\beta}|\boldsymbol{\beta}^{[n]}\right)$ the conditional expectation of the complete-data likelihood when $n$ nodes are present. It is worth mentionning that in the on-line context, there is an identity between the number of nodes being present, and the iteration increment of the algorithm.

Since on-line methods are based on sequential optimization, we first check that $\mathcal{Q}_{n+1}\left(\boldsymbol{\beta}|\boldsymbol{\beta}^{[n]}\right)$ linearly depends on $\mathcal{Q}_n\left(\boldsymbol{\beta}|\boldsymbol{\beta}^{[n]}\right)$. Since we suppose that the hidden variables are conditionally independent under the approximate conditional distribution $\mathcal{R}$, we need to study the joint distribution of $\mathbf{X},\mathbf{Z}$.

**Proposition 1**

$$\begin{aligned}\log\Pr\{\mathbf{X}^{[n+1]},\mathbf{Z}^{[n+1]}\} &= \log\Pr\{\mathbf{X}^{[n]},\mathbf{Z}^{[n]}\} + \boldsymbol{\beta}^t T(\mathbf{X}_{n+1,\bullet},\mathbf{Z}^{[n+1]}) \\ &- A^{[1]}(\boldsymbol{\beta}) + B(\mathbf{X}_{n+1,\bullet}),\end{aligned}$$

*with* $T(\mathbf{X}_{n+1,\bullet},\mathbf{Z}^{[n+1]}) = \left(Z_{n+1,q}, \xi_{ql}^{[n+1]}, \zeta_{ql}^{[n+1]}\right)$ *the vector of additional information provided by the addition of a new node, such that:*

$$\begin{aligned}\xi_{ql}^{[n+1]} &= Z_{n+1,q}\sum_{j=1}^{n} Z_{jl}h(X_{n+1,j}) + Z_{n+1,l}\sum_{i=1}^{n} Z_{iq}h(X_{i,n+1}) \\ &+ Z_{n+1,q}h(X_{n+1,n+1}) \times \mathbb{I}\{Z_{n+1,q} = Z_{n+1,l}\}, \\ \zeta_{ql}^{[n+1]} &= Z_{n+1,q}N_l^{[n]} + Z_{n+1,l}N_q^{[n]} + Z_{n+1,q} \times \mathbb{I}\{Z_{n+1,q} = Z_{n+1,l}\}. \\ A^{[1]}(\beta) &= \log\sum_{l=1}^{Q}\exp\omega_l, \\ B(\mathbf{X}_{n+1,\bullet}) &= \sum_{j=1}^{n} b(X_{n+1,j}) + \sum_{i=1}^{n} b(X_{i,n+1}) + b(X_{n+1,n+1}).\end{aligned}$$

**Proof** The proof is straightforward when writing $T(\mathbf{X}^{[n+1]}, \mathbf{Z}^{[n+1]})$ as a function of $T(\mathbf{X}^{[n]}, \mathbf{Z}^{[n]})$. ■

Consequently, on-line methods can be applied to MixNet parameter inference, provided that the conditional distribution $\Pr\{\mathbf{Z}|\mathbf{X}\}$ is approximated by a factorizable distribution. Note that when considering on-line algorithms applied to networks the addition of one node leads to the addition of $n+1$ potential connections, explaining the terms depending on $\mathbf{X}_{n+1,\bullet}$. Thanks to Proposition 1, sufficient statistics can be written as follows:

$$\begin{aligned} N_q^{[n+1]} &= N_q^{[n]} + Z_{n+1,q}, \\ H_{ql}(\mathbf{X}^{[n+1]}, \mathbf{Z}^{[n+1]}) &= H_{ql}(\mathbf{X}^{[n]}, \mathbf{Z}^{[n]}) + \xi_{ql}^{[n+1]}, \\ G_{ql}(\mathbf{Z}^{[n+1]}) &= G_{ql}(\mathbf{Z}^{[n]}) + \zeta_{ql}^{[n+1]}. \end{aligned}$$

and those equations will be used for parameter updates in the on-line algorithms.

## 3. Stochastic Approximation EM for Network Mixture

### 3.1 Short presentation of SAEM

An original way to estimate the parameters of the MixNet model is to approximate the expectation of the complete data log-likelihood using Monte Carlo simulations associated to the Stochastic Approximation EM algorithm Delyon et al. (1999). In situations where the maximisation of

$$\mathcal{Q}(\boldsymbol{\beta}|\boldsymbol{\beta}') = \mathbb{E}_{\mathbb{P}_{\beta'}}\left\{\log \mathcal{L}_C(\mathbf{X}, \mathbf{Z}; \boldsymbol{\beta})|\mathbf{X}\right\} = \sum_{\mathbf{Z}} \Pr\{\mathbf{Z}|\mathbf{X}; \boldsymbol{\beta}'\} \log \mathcal{L}_C(\mathbf{X}, \mathbf{Z}; \boldsymbol{\beta})$$

is not in simple closed form, the SAEM algorithm proposes to maximize an approximation $\widehat{\mathcal{Q}}(\boldsymbol{\beta}|\boldsymbol{\beta}')$ of the expectation $\mathcal{Q}(\boldsymbol{\beta}|\boldsymbol{\beta}')$. The approximation at step $[k]$ is computed using standard stochastic approximation theory :

$$\widehat{\mathcal{Q}}(\boldsymbol{\beta}|\boldsymbol{\beta}')^{[k]} = \widehat{\mathcal{Q}}(\boldsymbol{\beta}|\boldsymbol{\beta}')^{[k-1]} + \gamma_k\Big(\widetilde{\mathcal{Q}}(\boldsymbol{\beta}|\boldsymbol{\beta}') - \widehat{\mathcal{Q}}(\boldsymbol{\beta}|\boldsymbol{\beta}')^{[k-1]}\Big), \tag{1}$$

where $\{\gamma_k\}_{k\geq 1}$ is a sequence of positive step size and $\widetilde{Q}(\boldsymbol{\beta}|\boldsymbol{\beta}')$ is obtained by Monte Carlo integration. It is a simulation of the expectation of the complete log-likelihood using the posterior $\Pr\{\mathbf{Z}|\mathbf{X}\}$. Each iteration $k$ of the algorithm is broken down into three steps :

**Simulation** of the missing data. This can be achieved using Gibbs Sampling of the posterior $\Pr\{\mathbf{Z}|\mathbf{X}\}$. The results at iteration number $k$ is $m(k)$ realizations of the latent class matrix $\mathbf{Z}$ : $(\mathbf{Z}(1), ..., \mathbf{Z}(m(k)))$.

**Stochastic Approximation** of $\mathcal{Q}(\boldsymbol{\beta}|\boldsymbol{\beta}')$ using Eq. 1, with

$$\widetilde{\mathcal{Q}}(\boldsymbol{\beta}|\boldsymbol{\beta}') = \frac{1}{m(k)} \sum_{s=1}^{m(k)} \log \mathcal{L}_C(\mathbf{X}, \mathbf{Z}(s); \boldsymbol{\beta}) \tag{2}$$

**Maximisation** of $\widehat{\mathcal{Q}}(\boldsymbol{\beta}|\boldsymbol{\beta}')^{[k]}$ according to $\boldsymbol{\beta}$.

When considering the on-line version of the algorithm, the number of iterations $k$ usually coincides with $n+1$, the number of nodes of the network.

### 3.2 Simulation of $\Pr\{\mathbf{Z}|\mathbf{X}\}$ in the on-line context

We used Gibbs sampling which is applicable when the joint distribution is not known explicitly, but the conditional distribution of each variable is known. Here we generate a sequence of $\Pr\{\mathbf{Z}|\mathbf{X}\}$ using $\Pr\{Z_{iq}=1|\mathbf{X},\mathbf{Z}_{\backslash i}\}$ where $\mathbf{Z}_{\backslash i}$ stands for the class of all nodes except node $i$. The sequence of samples comprises a Markov chain, and the stationary distribution of that Markov chain is the sought-after joint distribution. Each iteration is as follows:

1. pick a node $i \in \{1,...,n\}$ at random (according a uniform distribution);
2. pick a new class $Z_i$ according to $\mathcal{M}(1,\theta_{i1},...,\theta_{iQ})$, where

$$\begin{aligned}\theta_{iq} &= \Pr\{Z_{iq}=1|\mathbf{X},\mathbf{Z}_{\backslash i}\},\\ &= \frac{\Pr\{Z_{iq}=1,\mathbf{Z}_{\backslash i},\mathbf{X}\}}{\sum_{q=1}^{Q}\Pr\{Z_{iq}=1,\mathbf{Z}_{\backslash i},\mathbf{X}\}}.\end{aligned}$$

In the on-line context, when we simulate the class of the last incoming node using $\theta_{n+1,q}=\Pr\{Z_{n+1,q}|\mathbf{X}^{[n+1]},\mathbf{Z}^{[n]}\}$, we get the general expression

$$\theta_{n+1,q}=\frac{\alpha_q^{[n]}\exp\left\{\sum_{l=1}^{Q}\sum_{j=1}^{n}Z_{jl}^{[n]}\left(\eta_{ql}^{[n]}h(X_{n+1,j})+a(\eta_{ql}^{[n]})\right)\right\}}{\sum_q\alpha_q^{[n]}\exp\left\{\sum_{l=1}^{Q}\sum_{j=1}^{n}Z_{jl}^{[n]}\left(\eta_{ql}^{[n]}h(X_{n+1,j})+a(\eta_{ql}^{[n]})\right)\right\}},\forall q\in\{1,...,Q\}. \quad (3)$$

For example, considering the simple Bernoulli model we get:

$$\theta_{n+1,q}=\frac{\alpha_q^{[n]}\exp\left\{\sum_{l=1}^{Q}\sum_{j=1}^{n}Z_{jl}^{[n]}\left(\log(\pi_{ql}^{X_{ij}}(1-\pi_{ql})^{1-X_{ij}})\right)\right\}}{\sum_q\alpha_q^{[n]}\exp\left\{\sum_{l=1}^{Q}\sum_{j=1}^{n}Z_{jl}^{[n]}\left(\log(\pi_{ql}^{X_{ij}}(1-\pi_{ql})^{1-X_{ij}})\right)\right\}},\forall q\in\{1,...,Q\}. \quad (4)$$

### 3.3 Computing $\widehat{\mathcal{Q}}(\beta|\beta')$ in the on-line context

Each realisation $\mathbf{Z}(s)$ can be used to compute a corresponding complete log-likelihood $\log\mathcal{L}_C(\mathbf{X},\mathbf{Z}(s);\boldsymbol{\beta})$. Those $m(k)$ complete log-likelihoods can be averaged to compute

$$\widetilde{\mathcal{Q}}(\boldsymbol{\beta}|\boldsymbol{\beta}')=\frac{1}{m(k)}\sum_{s=1}^{m(k)}\log\mathcal{L}_C(\mathbf{X},\mathbf{Z}(s);\boldsymbol{\beta}).$$

When considering the on-line version of the SAEM algorithm, it appears that the difference between the old and the new complete-data log-likelihood is expressed as:

$$\begin{aligned}\log\mathcal{L}_C(Z_{n+1,q}=1,\mathbf{X}^{[n+1]}) &= \log\mathcal{L}_C(\mathbf{X}^{[n+1]},\mathbf{Z}^{[n+1]},\boldsymbol{\beta})-\log\mathcal{L}_C(\mathbf{X}^{[n]},\mathbf{Z}^{[n]},\boldsymbol{\beta})\\ &= \log\alpha_q+\sum_l\sum_{i<n+1}Z_{il}\log\Pr\{X_{n+1,i}|Z_{n+1,q}Z_{il}\}.\end{aligned}$$

Recall that in the on-line framework, the label of the new node has been sampled from the Gibbs sampler described in Section 3.2. Consequently only one possible label is considered in this equation. Then a natural way to adapt Equation 1 to the on-line context is to consider that:

$$\widetilde{\mathcal{Q}}_{n+1}(\boldsymbol{\beta}|\boldsymbol{\beta}^{[n]}) - \widehat{\mathcal{Q}}_n(\boldsymbol{\beta}|\boldsymbol{\beta}^{[n]}) = \log \mathcal{L}_C(\mathbf{X}^{[n+1]}, \mathbf{Z}^{[n+1]}; \boldsymbol{\beta}).$$

Indeed this quantity correponds to the difference between the log-likelihood of the previous network and log-likelihood of the new one with the additional node. Notice that the larger the network, the larger its associated complete expected log-likelikelihood. Thus $\log \mathcal{L}_C(\mathbf{X}^{[n+1]}, \mathbf{Z}^{[n+1]}, \boldsymbol{\beta})$ becomes smaller and smaller compared to $Q(\boldsymbol{\beta}|\boldsymbol{\beta}')$ when $n$ increases. The decreasing step $\gamma_n$ is thus not necessary in this on-line context and we propose to consider the following update equation for stochastic on-line EM computation of MixNet conditionnal expectation:

$$\widehat{\mathcal{Q}}_{n+1}(\boldsymbol{\beta}|\boldsymbol{\beta}^{[n]}) = \widehat{\mathcal{Q}}_n(\boldsymbol{\beta}|\boldsymbol{\beta}^{[n]}) + \log \mathcal{L}_C(Z_{n+1,q} = 1, \mathbf{X}^{[n+1]}),$$

where $\mathbf{Z}_{n+1}$ is drawn from the Gibbs sampler.

### 3.4 Maximizing $\widehat{\mathcal{Q}}(\beta|\beta')$, and parameters update

The principle of on-line algorithms is to modify the current parameter estimation using the information brought by a new available $[n+1]$ node and its corresponding connections $\mathbf{X}_{n+1,\bullet}$ to the already existing network. Maximizing $\widehat{\mathcal{Q}}_{n+1}(\boldsymbol{\beta}|\boldsymbol{\beta}^{[n]})$ according to $\boldsymbol{\beta}$ is straightforward and produces the maximum likelihood estimates for iteration $[n+1]$. A simple version of the algorithm can be derived by chosing $m(k) = 1$ (one simulation of $\mathbf{Z}|\mathbf{X}$). When running the Gibbs sampler once, node $i$ may be changed to another class and the new partition scheme is $\mathbf{Z}(1)$. The difference between $\widehat{\mathcal{Q}}_n(\boldsymbol{\beta}|\boldsymbol{\beta}^{[n]})$ and $\widehat{\mathcal{Q}}_{n+1}(\boldsymbol{\beta}|\boldsymbol{\beta}^{[n]})$ implies only the terms of the complete log-likelihood which are function of node $n+1$. Using notation

$$\psi_{ql} = \frac{\partial a(\eta_{ql})}{\partial \eta_{ql}},$$

we get

$$\begin{aligned} \alpha_q^{[n+1]} &= \frac{N_q^{[n+1]}}{n+1}, \\ \psi_{ql}^{[n+1]} &= \frac{H_{ql}(\mathbf{X}^{[n+1]}, \mathbf{Z}^{[n+1]})}{G_{ql}(\mathbf{Z}^{[n+1]})}, \end{aligned}$$

with

$$\begin{aligned} N_q^{[n+1]} &= N_q^{[n]} + Z_{n+1,q}, \\ H_{ql}(\mathbf{X}^{[n+1]}, \mathbf{Z}^{[n+1]}) &= H_{ql}(\mathbf{X}^{[n]}, \mathbf{Z}^{[n]}) + \xi_{ql}^{[n+1]}, \\ G_{ql}(\mathbf{Z}^{[n+1]}) &= G_{ql}(\mathbf{Z}^{[n]}) + \zeta_{ql}^{[n+1]}. \end{aligned}$$

where $(\xi_{ql}, \zeta_{ql})$ have been defined in the previous Section, and where $Z_{n+1,q}$ is the simulated missing class at new incoming node. Considering the Bernoulli model the estimators become

$$\pi_{ql}^{[n+1]} = \gamma_{ql}^{[n+1]} \pi_{ql}^{[n]} + (1 - \gamma_{ql}^{[n+1]}) \frac{\xi_{ql}^{[n+1]}}{\zeta_{ql}^{[n+1]}},$$

where

$$\begin{aligned}
\gamma_{ql}^{[n+1]} &= \frac{N_q^{[n]}N_l^{[n]}}{Z_{n+1,q}N_l^{[n]} + Z_{n+1,l}N_q^{[n]}}, \\
\xi_{ql}^{[n+1]} &= Z_{n+1,q}\sum_{j=1}^{n} Z_{jl}^{[n]} X_{n+1,j} + Z_{n+1,l}\sum_{i=1}^{n} Z_{iq}^{[n]} X_{i,n+1} \\
&+ Z_{n+1,q}X_{n+1,n+1} \times \mathbb{I}\{Z_{n+1,q} = Z_{n+1,l}\}, \\
\zeta_{ql}^{[n+1]} &= Z_{n+1,q}N_l^{[n]} + Z_{n+1,l}N_q^{[n]} \\
&+ Z_{n+1,q} \times \mathbb{I}\{Z_{n+1,q} = Z_{n+1,l}\}.
\end{aligned}$$

Once all the nodes of the network are visited (or known), the parameters can be further improved and the complete log-likelihood better approximated by continuing with the previously described SAEM algorithm.

## 4. Application of *on-line* algorithm to variational methods

An alternative to the Stochastic Approximation EM algorithm, when dealing with a untractable conditionnal expectation of the complete log-likelihood (because $\Pr\{\mathbf{Z}|\mathbf{X}\}$ can not be factorized) is to be found in variational methods. Their principle is to optimize an approximation of the incomplete-data log-likelihood, $\log \Pr\{\mathbf{X};\boldsymbol{\beta}\}$, denoted by $\mathcal{J}(\mathbf{X}, \mathcal{R}(\mathbf{Z}); \boldsymbol{\beta})$ (Jordan et al., 1999). This approximation depends on $\mathcal{R}$ which is a newly introduced probability distribution on $\mathbf{Z}$. $\mathcal{J}(\mathbf{X}, \mathcal{R}(\mathbf{Z}); \boldsymbol{\beta})$ is defined such that:

$$\mathcal{J}(\mathbf{X}, \mathcal{R}(\mathbf{Z}); \boldsymbol{\beta}) = \log \Pr\{\mathbf{X};\boldsymbol{\beta}\} - KL(\mathcal{R}(\mathbf{Z}), \Pr\{\mathbf{Z}|\mathbf{X};\boldsymbol{\beta}\}),$$

with $KL(\bullet|\bullet)$ being the Kullback-Leibler divergence between probability distributions.

One must then choose the form of $\mathcal{R}$. One popular approximation is the mean-field approximation which consists in considering independency for the hidden variables such that $\log \mathcal{R}(\mathbf{Z}) = \sum_i \log \mathcal{R}(\mathbf{Z}_i; \tau_i)$, where $\tau_i$ is called the variational parameter. In the case of MixNet, a natural form for $\mathcal{R}(\bullet; \tau)$ is the multinomial distribution, such that $\log \mathcal{R}(\mathbf{Z}) = \sum_i \sum_q Z_{iq} \log \tau_{iq}$, with the constraint $\sum_q \tau_{iq} = 1$. In this case, the form of $\mathcal{J}(\mathbf{X}, \mathcal{R}(\mathbf{Z}); \boldsymbol{\beta})$ is:

$$\begin{aligned}
\mathcal{J}(\mathbf{X}, \mathcal{R}(\mathbf{Z}); \boldsymbol{\beta}) &= \mathcal{J}(\mathbf{X}; \boldsymbol{\tau}, \boldsymbol{\beta}) \\
&= \mathcal{Q}(\boldsymbol{\tau}, \boldsymbol{\beta}) + \mathcal{H}(\mathcal{R}(\mathbf{Z}; \boldsymbol{\tau})), \\
&= \sum_{\mathbf{Z}} \mathcal{R}(\mathbf{Z}; \boldsymbol{\tau}) \log \Pr\{\mathbf{X}, \mathbf{Z}; \boldsymbol{\beta}\} - \sum_{\mathbf{Z}} \mathcal{R}(\mathbf{Z}; \boldsymbol{\tau}) \log \mathcal{R}(\mathbf{Z}; \boldsymbol{\tau}),
\end{aligned}$$

with $\mathcal{Q}(\boldsymbol{\tau}, \boldsymbol{\beta})$ an approximation of the conditional expectation of the complete-data log-likelihood, and $\mathcal{H}(\mathcal{R}(\mathbf{Z}; \boldsymbol{\tau}))$ the entropy of the approximate *posterior* distribution of $\mathbf{Z}$. In the following we denote by $\mathcal{Q}_n(\boldsymbol{\tau}, \boldsymbol{\beta})$ the conditional expectation of the complete-data log-likelihood when $n$ nodes are present.

The implementation of variational methods in on-line algorithms relies on the additivity property of $\mathcal{J}(\mathbf{X}, \mathcal{R}(\mathbf{Z}); \boldsymbol{\beta})$ when nodes are added. This property is straightforward :

$\mathcal{Q}(\boldsymbol{\tau}, \boldsymbol{\beta})$ is additive thanks to Proposition 1 (because $\mathcal{R}(\mathbf{Z})$ is factorized), and $\mathcal{H}(\mathcal{R}(\mathbf{Z}; \boldsymbol{\tau}))$ is also additive, since the hidden variables are supposed independent under $\mathcal{R}$ and the entropy of independent variables is additive.

The variational algorithm is very similar to a EM algorithm, with the E-step being replaced by a variational step which aims at updating variational parameters. Then a standard M-step follows. In the following, we give the details of these two steps in the case of a variational on-line algorithm.

### 4.1 On-line variational step

When a new node is added it is necessary to compute its associated variational parameters $\{\tau_{n+1,q}\}_q$. If we consider all the other $\tau_{iq}$ for $i < n+1$ as known, the $\{\tau_{n+1,q}\}_q$ are obtained by deriving the criterion

$$\mathcal{J}\left(\mathbf{X}^{[n+1]}, \mathcal{R}(\mathbf{Z}^{[n+1]}); \boldsymbol{\beta}\right) + \sum_{i=1}^{n+1} \lambda_i \left(\sum_{q=1}^{Q} \tau_{iq} - 1\right),$$

where the $\lambda_i$ are the Lagrangian parameters. Since function $\mathcal{J}$ is additive according to the nodes, the calculation of its derivative according to $\tau_{n+1,q}$ gives:

$$\omega_q^{[n]} + \sum_{l=1}^{Q} \sum_{j=1}^{n} \tau_{jl}^{[n]} \left(\eta_{ql}^{[n]} h(X_{n+1,j}) + a(\eta_{ql}^{[n]})\right) - \log \tau_{n+1,q} + 1 + \lambda_{n+1} = 0$$

This leads to

$$\tau_{n+1,q} = \frac{\alpha_q^{[n]} \exp\left\{\sum_{l=1}^{Q} \sum_{j=1}^{n} \tau_{jl}^{[n]} \left(\eta_{ql}^{[n]} h(X_{n+1,j}) + a(\eta_{ql}^{[n]})\right)\right\}}{\sum_q \alpha_q^{[n]} \exp\left\{\sum_{l=1}^{Q} \sum_{j=1}^{n} \tau_{jl}^{[n]} \left(\eta_{ql}^{[n]} h(X_{n+1,j}) + a(\eta_{ql}^{[n]})\right)\right\}}, \forall q \in \{1, ..., Q\}. \qquad (5)$$

### 4.2 Maximization/Update step

To maximize the approximated expectation of the complete log-likelihood according to $\boldsymbol{\beta}$, we solve

$$\frac{\partial \mathcal{Q}_{n+1}(\boldsymbol{\tau}, \boldsymbol{\beta})}{\partial \boldsymbol{\beta}} \simeq \mathbb{E}_{\mathcal{R}^{[n]}} \left(\frac{\partial \log \Pr\{\mathbf{X}^{[n+1]}, \mathbf{Z}^{[n+1]}; \boldsymbol{\beta}\}}{\partial \boldsymbol{\beta}}\right) = 0.$$

This leads to two different kinds of equations:

$$\begin{cases} \mathbb{E}_{\mathcal{R}^{[n]}} \left(\frac{\partial \log \Pr\{\mathbf{X}^{[n+1]}, \mathbf{Z}^{[n+1]}; \boldsymbol{\beta}\}}{\partial \omega_q}\right) = 0, \\ \mathbb{E}_{\mathcal{R}^{[n]}} \left(\frac{\partial \log \Pr\{\mathbf{X}^{[n+1]}, \mathbf{Z}^{[n+1]}; \boldsymbol{\beta}\}}{\partial \eta_{ql}}\right) = 0. \end{cases}$$

The first equation resumes to :

$$\sum_{i=1}^{n} \tau_{iq}^{[n]} + \tau_{n+1,q} - (n+1)\alpha_q^{[n+1]} = 0,$$

which gives

$$\alpha_q^{[n+1]} = \frac{1}{n+1}\left(\sum_{i=1}^{n}\tau_{iq}^{[n]} + \tau_{n+1,q}\right).$$

Using the same notation

$$\psi_{ql} = \frac{\partial a(\eta_{ql})}{\partial \eta_{ql}},$$

we also get:

$$\psi_{ql}^{n+1} = \frac{\mathbb{E}_{\mathcal{R}^{[n]}}\left(H_{ql}(\mathbf{X}^{[n+1]}, \mathbf{Z}^{[n+1]})\right)}{\mathbb{E}_{\mathcal{R}^{[n]}}\left(G_{ql}(\mathbf{Z}^{[n+1]})\right)},$$

where $G_{ql}()$ and $H_{ql}()$ are the previously defined sufficient statistics. Thanks to proposition 1, which gives the relationships between sufficient statistics at two successive iterations, parameters can be computed recursively using the update of the expectation of the sufficient statistics, such that

$$\begin{aligned}
\mathbb{E}_{\mathcal{R}^{[n]}}\left(N_q^{[n+1]}\right) &= \mathbb{E}_{\mathcal{R}^{[n]}}\left(N_q^{[n]}\right) + \mathbb{E}_{\mathcal{R}^{[n]}}\left(Z_{n+1,q}\right), \\
\mathbb{E}_{\mathcal{R}^{[n]}}\left(H_{ql}(\mathbf{X}^{[n+1]}, \mathbf{Z}^{[n+1]})\right) &= \mathbb{E}_{\mathcal{R}^{[n]}}\left(H_{ql}(\mathbf{X}^{[n]}, \mathbf{Z}^{[n]})\right) + \mathbb{E}_{\mathcal{R}^{[n]}}\left(\xi_{ql}^{[n+1]}\right), \\
\mathbb{E}_{\mathcal{R}^{[n]}}\left(G_{ql}(\mathbf{Z}^{[n+1]})\right) &= \mathbb{E}_{\mathcal{R}^{[n]}}\left(G_{ql}(\mathbf{Z}^{[n]})\right) + \mathbb{E}_{\mathcal{R}^{[n]}}\left(\zeta_{ql}^{[n+1]}\right).
\end{aligned}$$

In the case of a Bernoulli distribution we get:

$$\pi_{ql}^{[n+1]} = \gamma_{ql}^{[n+1]}\pi_{ql}^{[n]} + (1-\gamma_{ql}^{[n+1]})\frac{\mathbb{E}_{\mathcal{R}^{[n]}}\left(\xi_{ql}^{[n+1]}\right)}{\mathbb{E}_{\mathcal{R}^{[n]}}\left(\zeta_{ql}^{[n+1]}\right)}, \tag{6}$$

where

$$\begin{aligned}
\gamma_{ql}^{[n+1]} &= \frac{\mathbb{E}_{\mathcal{R}^{[n]}}\left(N_q^{[n]}\right)\mathbb{E}_{\mathcal{R}^{[n]}}\left(N_l^{[n]}\right)}{\tau_{n+1,q}\mathbb{E}_{\mathcal{R}^{[n]}}\left(N_l^{[n]}\right) + \tau_{n+1,l}\mathbb{E}_{\mathcal{R}^{[n]}}\left(N_q^{[n]}\right)}, \\
\mathbb{E}_{\mathcal{R}^{[n]}}\left(\xi_{ql}^{[n+1]}\right) &= \tau_{n+1,q}\sum_{j=1}^{n}\tau_{jl}^{[n]}X_{n+1,j} + \tau_{n+1,l}\sum_{i=1}^{n}\tau_{iq}^{[n]}X_{i,n+1} \\
&+ \tau_{n+1,q}X_{n+1,n+1} \times \mathbb{I}\{Z_{n+1,q} = Z_{n+1,l}\}, \\
\mathbb{E}_{\mathcal{R}^{[n]}}\left(\zeta_{ql}^{[n+1]}\right) &= \tau_{n+1,q}N_l^{[n]} + \tau_{n+1,l}N_q^{[n]} \\
&+ \tau_{n+1,q} \times \mathbb{I}\{Z_{n+1,q} = Z_{n+1,l}\}.
\end{aligned}$$

Note the similarity of the formula compared with the SAEM strategy. Hidden variables $\mathbf{Z}$ are either simulated or replaced by their approximated conditional expectation (variational parameters).

## 5. Experiments

We carried out experiments to assess how well the proposed online clustering algorithms discover node clusters. We consider simulation experiments using synthetic data generated according to the assumed random graph model, as well as real data coming from the web. Using synthetic graphs allows us to evaluate the quality of the parameter estimation for each algorithm. The real data set consists of over 2,000 political U.S. websites, collected to analyze the existing communites created by political affinities. An ANSI `C++` implementation of the algorithms is available upon request, as well as an R package named `MixNet` .

### 5.1 Comparison of algorithms

In these experiments, we assume that edges are Bernoulli distributed. We consider a simple affiliation model where two types of egdes exist : egdes between nodes of the same class and egdes between nodes of different classes. Each type of edge has a given probability, respectively $\pi_{qq} = \lambda$ and $\pi_{ql} = \epsilon$. Five affiliation models were considered (see Table 1) with $\lambda = 1 - \epsilon$ to limit the number of varying parameters in the experiment.

The parameter $\lambda$ controls the complexity of the model. The difference among the five models is related to their modular structure, which varies from no structure (almost the Erdős-Rényi model) to strong modular structure (low inter-module connectivity and strong intra-module connectivity or strong inter-module connectivity and low intra-module connectivity).

[Figure 1 about here.]

[Table 1 about here.]

For each affiliation model, we generate graphs with $Q \in \{2, 5, 20\}$ groups mixed in the same proportions $\alpha_1 = ... = \alpha_Q = \frac{1}{Q}$ and with $n \in \{100, 250, 500\}$ nodes. We thus generated a total of 45 graph models.

We used the adjusted Rand index (Hubert and Arabie, 1985) to evaluate the agreement between the true and the estimated partitions. The computation of this index is based on a ratio between the number of node pairs belonging to the same and to different classes when considering the true partition and the estimated partition. It lies between 0 and 1, two identical partitions having an adjusted Rand index equal to 1.

In a previous work, we compared the variational MixNet approach with alternative clustering methods. We considered the following algorithm competitors : an online classification MixNet (Zanghi et al., 2007) which is called online CEM in this paper, a basic spectral clustering algorithm (Ng et al., 2002), and a kmeans-like algorithm (considering a dissimilarity matrix based on shortest paths as input). We concluded that MixNet models produce more accurate results than the two others methods and that online methods reduced significatively the computational cost. In the forthcoming experiments, we compare our two new online algorithms, online variational MixNet and online SAEM, with the online CEM and the variational (batch MixNet) as references.

We simulated 30 networks for each of the 45 models and run our algorithms to evaluate the parameter estimations and the estimated hidden partitions. Notice that as any local optimization algorithm, the proposed online MixNet estimation strategy strongly depends on the initialization. A common way to circumvent this problem consists in starting the algorithm with multiple initialization points and in selecting the best result in terms of likelihood. Thus, for each simulated network, the algorithm is run 10 times and the number of clusters is chosen using the Integrated Classification Likelihood criterion, as proposed in Daudin et al. (2008).

Online algorithms produce parameter estimates while discovering the graph nodes, but can further improve the parameter of the model if the nodes are visited many times (Zanghi et al., 2007). In these first experiments we use for the online algorithms the same stopping condition than the variational batch MixNet condition which is based on a stabilisation of the estimated parameters. The parameter comparison is done at the end of each epoch (one visit of all network nodes).

In Table 2, we observe that for the three highly structured models (models 1, 2, 5) the estimation is, for each algorithm, very close or equal to the true parameters (with a two digits round approximation after the comma) and exhibits no or negligible variance. But in model 3, we notice that all algorithms do not behave identically. The Batch MixNet still estimates correctly the true parameters while other algorithms have more difficulty. Note that the online classification algorithm (CEM) performs best among the online versions, followed by the online variational algorithm. In the fourth model algorithms retrieve correct parameters with a small bias and variance for the online SAEM and CEM algorithm.

[Table 2 about here.]

When considering Table 3, we observe that the poor estimation of $\lambda$ reveals a small Rand index. This means that the poor estimation of $\lambda$ makes it impossible to retrieve the modular structure of the network. For example, model 3 has a poor Rand Index for the online algorithms which have produced poor estimation of parameters. Figure 2 displays the Rand Index evolution for $\lambda \in \{0.58, 0.59, \ldots, 0.68\}$. We observe that the online CEM and online variational algorithms perform always better than the online SAEM and the online CEM algorithm is better than the variational algorithm until $\lambda = 0.61$.

[Table 3 about here.]

[Figure 2 about here.]

Let us observe also that good estimates do not always lead to a correctly estimated partition. For example, in model 4 althougth our algorithms produce good estimates, they do not find the correct partition because of the non modular structure of the network.

In Table 4, we observe that the larger the number of nodes, the better the algorithm's retrieval of the modular structure. We observe that the phenomenon reverses when the number of classes increases.

[Table 4 about here.]

In the previous experiments, let us remind the reader that the same stop condition is used for both batch and online algorithms. But in numerous applications, online algorithms are prefered because they are faster than batch algorithms and better adapted when the data set grows over time. They allow the reduction of the latent period between the appearance of new data and its treatment (its classification). Thus, if one does not reconsider nodes in the network, the stopping condition corresponds to a single epoch. We obtain Table 5 which shows means of the Rand index and of the execution time for different network sizes with a fixed $q$ and $\lambda$.

[Table 5 about here.]

The important speed difference between the batch variational algorithm with our online algorithms (Figure 3) with the low loss of quality in the estimated partition make online algorithms attractive and suitable for large graphs.

[Figure 3 about here.]

### 5.2 The 2008 U.S. Presidential WebSphere

As mentionned in Adamic and Glance (2005), the 2004 U.S. Presidential Election was the first one in the United States in which the web and more precisely blogging played an important part. Although a minority of Americans actually used these Weblogs, their influence was extended beyond their readership through their interactions with national mainstream media. With the impact of new social network websites like Myspace or Facebook, the web should have a stronger influence during the U.S.political campaign in 2008.

[Figure 4 about here.]

In this real community extraction experiment, we used a real data set gathered in November the 7th 2007 by RTGI with a specific methodology similar to Fouetillou (2007). This data set consists of a single day snapshot of over two thousand websites of which one thousand comes from two online directories : `http://wonkosphere.com` and `http://www.politicaltrends.info`. The first site provides a manual classification and the second an automatic classification based on text analysis. From this seed (the thousand previoulsy mentionned sites) a web crawler (Drugeon, 2005) collects a maximum of 100 pages per hostname. External links are examined to check the connectivity with visited or unvisited websites. If websites are still unvisited and if there exists a minimal path of distance less than two between a hostname which belongs to the seed and these websites then the web crawler collects them.

According to that seed extension, 200,000 websites have been gathered and a network of websites was created where nodes represent hostnames (a hostname contains a set of pages) and edges represent hyperlinks between different hostnames. If several links exist between two different hostnames, we collapse them into a single one. Note that intra domain links can be considered if hostnames are not similar. On this web network, we computed an auhority score (Kleinberg, 1999) and keyword score TF/IDF (Salton et al., 1975) on focused words (political entities) in order to identify respectively nodes with high-quality website (high authority scores) and centered on those topic (on a political corpus). 870 new websites came out of these two criteria. They have been checked by experts and the validity of the seed has been confirmed.

In the end, there were 130,520 links and 1,870 sites : 676 liberal, 1,026 conservative and 168 independent. Considering that website's authors tend to link according to political affinities, this network presents a priori an interesting community organization.

In our experiment, we benefit from a three groups manual partition as reference which allows us to analyze the agreement between both real and estimated partitions. A first interesting experiment consists in comparing this manual partition with an estimated one of three groups. From here, we will use the online variational algorithm to predict the partitions. Table 6 shows a contingency table of the counts of given and estimated website groups. We observe a relative coherence between these two partitions confirmed by an acceptable $randIndex = 0.25$ and a $modularity = 0.20$.

[Table 6 about here.]

However, reducing the US polical communities to three groups may be considered as an oversimplification and it appears relevant to find a more realistic number of groups. As the algorithm relies on a statistical model, it is possible to use the Integrated Classification Likelihood (ICL) to choose the optimal number of classes. This choice is made by running our online algorithm concurrently for models from 2 to $Q$ classes and selecting the solution which maximizes the ICL criterion. Using ICL, we obtain an optimal number of groups $Q = 20$ which reflects more effectively the diversity of communities based on political opinions. Then, considering this new number of classes, we produce a new contingency table of counts of given and estimated website groups. We observe very interesting subdivisions and some of them are described here.

The first example deals with four liberal clusters (C6, C7, C13 and C8) the size of which is decreasing and where the mean of node degrees increases. In addition, when analyzing the probability of connectivity between clusters : $\pi_{ql}$ for $q, l \in \{1, .., 20\}$, we notice that the sum of these probabilities : $\sum_l \pi_{ql}$, with $q$ fixed, increases. This behaviour is due to the increasing degree of nodes in these clusters and the presence of significant connectivities with other clusters. As far as political communities are concerned, this phenomenon reveals their degree of opening to others. By opening we mean that a cluster of one class will be connected to many cluster of different classes. Indeed, the most liberal cluster C6 which can be represented by the famous `feministe.us`, links mainly to liberal clusters what can be explained by its the radical political positions.

In the conservative part of the network, our algorithm produces clusters C10, C9, C19, and C16 with a behaviour similar to liberal clusters. Again, we notice the increasing mean degree of nodes and the increasing opening to other communities. Moreover, it seems important to notice that clusters C19 and C16 which own the best conservative autorities (nodes with high degree) have also a greater intra-cluster connectivity than their comparable liberal clusters (C13 and C8). This tendency was already observed by Adamic and Glance (2005) during the 2004 U.S. Presidential Election.

Furthermore, we remark a media cluster C17 made up of the four main US online portals (`nytimes.com`, `washingtonpost.com`, `cnn.com`, `msn.com`). These well known websites own the greatest node degree in the network and their cluster C17 has consequently meaningful connectivities with all clusters. It is interesting to note that the probability to have an edge from C17 to C17 is the weakest transition probability of this cluster. It means that cluster C17 is an interface cluster linked with everybody and where it is hard to stay in. Considering the need for bloggers to base their arguments on famous medias and the competition bewteen medias which disallows links among them, the comprehension of this phenomenon is obvious.

Besides, largest clusters C11, C1, C2 and C5 possess websites with small node degrees. They are constituted of liberal and conservative websites which are not influential (few incoming links) and are not hubs (important outgoing links). In these clusters, we can observe very weak connectivities with others.Then, their greatest connectivity is for the media cluster C17 (close with all the clusters).

Finally cluster C5 has the weakest connectivity transitions of all the estimated clusters. Then C5 has nodes with very poor degrees. Given that our algorithm requires the knowledge of node neighbourhoods to enable its classification, we can predict important mistakes in this cluster. Generally this kind of cluster is our trash cluster and in this experiments it aggregates nodes of the three manual groups.

[Table 7 about here.]

A last remark is that a popular method to cluster nodes of a graph is to use the community algorithm proposed by Newman (2006). However Newman aims at finding modules which are defined by high intra connectivity and low inter connectivity, whereas MixNet aims at merging nodes which share the same connectivity profile without any constraint on the between clusters connections. The quality of a partition in terms of Newman's modules can be measured by the so-called modularity of the partition. The value of this modularity is a scalar bewteen -1 and 1 and measures the density of links inside communities as compared to links between communities (Newman, 2006). As MixNet classes are not necessarily in the form of modules, it is expected to get modularity index which are not "optimal" using our approach. For instance when choosing the optimal number of clusters $Q = 20$, we obtain a $modularity = 0.04$ which is smaller than for the run with $Q = 3$. This last observation emphasises the fact that MixNet groups together nodes with the same behaviour in terms of connectivity and does not aim to maximize the modularity.

## 6. Conclusion

In this paper we proposed online versions of estimation algorithms for random graphs which are based on mixture of distributions. These strategies allow the estimation of the model parameters in a reasonnable computational time for datasets which can be made of thousands of nodes. These methods constitute a trade-off between the amount of data, which one can deal with and the quality of estimation : even if online methods are not as precise as "batch" methods for estimation, they provide an answer when the size of the network is too large for any existing estimation strategy. Furthermore, our simulation study shows that the quality of the remaining partition is goods when using online methods. In the network of the 2008 US political websites we have found coherent clusters composed by nodes with close degree and with a significative common edges. It appears to provide a very interesting overview of networks which differs from other community detection algorithms. Overall, the MixNet model seems promising for the investigation of the structure of the political sphere, and the application of online algorithm in this context should allow the investigation of structure in larger networks.

## Acknowlegements

We would like to thanks Guilhem Fouetillou for the 2008 U.S. Presidential corpus and its manual classification, and Mathieu Jacomy for its software GraphFiltre, which allow to produce nice network layouts.

## References

L.A. Adamic and N. Glance. The political blogosphere and the 2004 US election: divided they blog. *Proceedings of the 3rd international workshop on Link discovery*, pages 36–43, 2005.

J.J. Daudin, F. Picard, and S. Robin. A mixture model for random graph. *Statistics and computing*, 18(2):1–36, 2008.

B. Delyon, M. Lavielle, and E. Moulines. Convergence of a stochastic approximation version of the em algorithm. *The Annals of Statistics*, 27:94–128, 1999.

T. Drugeon. A technical approach for the french web legal deposit. *5th International Web Archiving Workshop (IWAW05)*, 2005.

G. Fouetillou. Le web et le traité constitutionnel européen, écologie d'une localité thématique, 2007.

Ove Frank and Frank Harary. Cluster inference by using transitivity indices in empirical graphs. *Journal of the American Statistical Association*, 77(380):835–840, 1982. ISSN 0162-1459.

G. Govaert. Algorithme de classification d'un tableau de contingence. In *First international symposium on data analysis and informatics*, pages 487–500, Versailles, 1977. INRIA.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2: 193–218, 1985.

M.I. Jordan, Z. Ghahramani, T.S. Jaakkola, and L.K. Saul. An introduction to variational methods for graphical models. *Mach. Learn.*, 37(2):183–233, 1999.

J.M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.

Z. Liu, J. Almhana, V. Choulakian, and R. McGorman. Online em algorithm for mixture with application to internet traffic modeling. *Computational Statistics & Data Analysis*, 50(4):1052–1071, February 2006. available at http://ideas.repec.org/a/eee/csdana/v50y2006i4p1052-1071.html.

J. MacQueen. Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematics, Statistics and Probability*, 1:281–296, 1967.

M. Mariadassou and S. Robin. Uncovering latent structure in valued graphs: a variational approach. Technical Report 10, SSB, october 2007.

ME. Newman and EA. Leicht. Mixture models and exploratory analysis in networks. *PNAS*, 104(23):9564–9569, 2007.

MEJ Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

A.Y. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS 14*, 2002.

G. Salton, A. Wong, and CS Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

T. A. B. Snijders and K. Nowicki. Estimation and prediction for stochastic block-structures for graphs with latent block structure. *Journal of Classification*, 14:75–100, 1997.

D. M. Titterington. Recursive parameter estimation using incomplete data. *JRSS-B*, 46: 257–267, 1984.

S. Wang and Y. Zhao. Almost sure convergence of titterington's recursive estimator for finite mixture models. *IEEE International Symposium on Information Theory IST*, 2002.

Hugo Zanghi, Christophe Ambroise, and Vincent Miele. Fast online graph clustering via Erdös-Rényi mixture. Technical report, INRA, SSB, 2007.

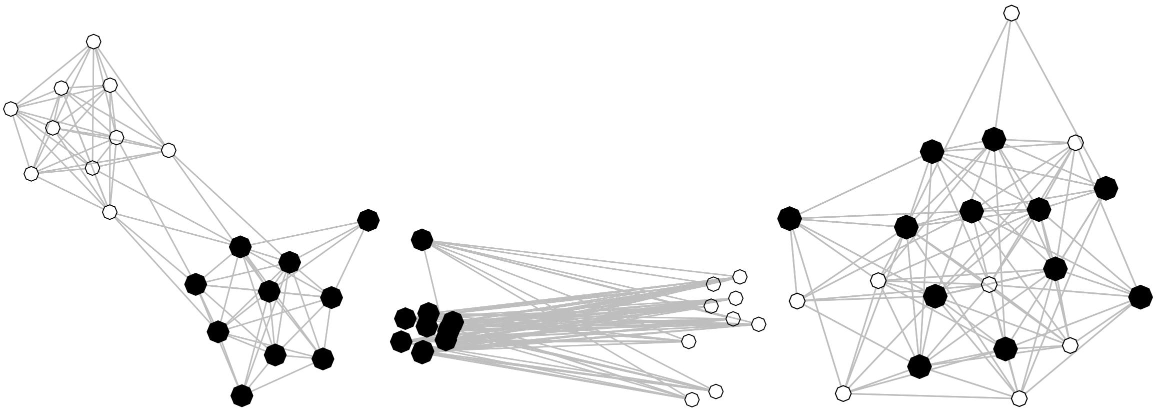

Figure 1: Top left : low inter-module connectivity and strong intra-module connectivity (model 1), Top right : strong inter-module connectivity and low intra-module connectivity (model 5), Bottom center : Erdős-Rényi model (model 4).

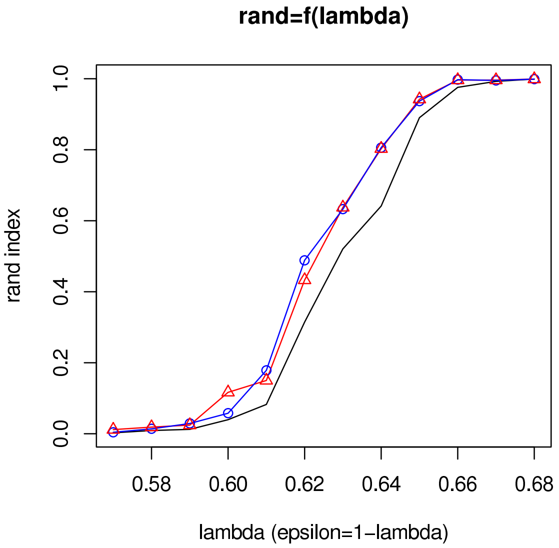

Figure 2: Rand Index evolution for $\lambda \in \{0.58, 0.59, \ldots, 0.68\}$. The plain line represents the online SAEM algorithm, line with $\triangle$ represents the online CEM algorithm and line with $\circ$ represents the online variational algorithm.

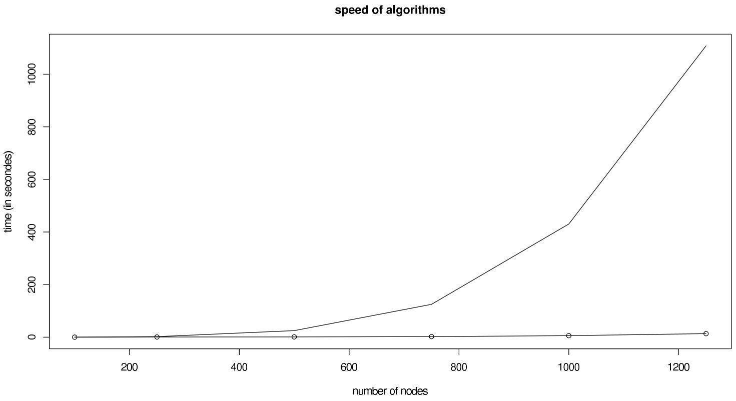

Figure 3: Algorithm speed in seconds for different network sizes. The plain line represents the batch variational algorithm and the line with $\circ$ represents the online variational algorithm.

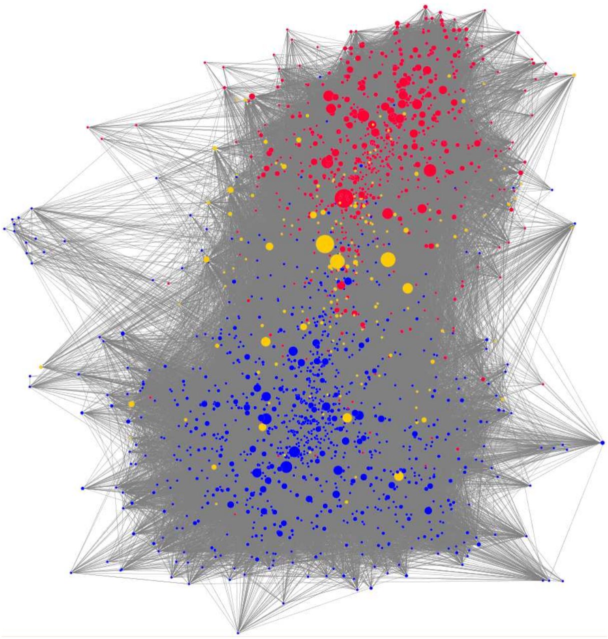

Figure 4: Network of the US political websites.

| Model | $\epsilon$ | $\lambda$ |
|---|---|---|
| 1 | 0.3 | 0.7 |
| 2 | 0.35 | 0.65 |
| 3 | 0.4 | 0.6 |
| 4 | 0.5 | 0.5 |
| 5 | 0.9 | 0.1 |

Table 1: Parameters of the five affililation models considered in the experimental setting.

| | online SAEM | | online Variational | | batch MixNet | | online CEM | |
|---|---|---|---|---|---|---|---|---|
| Model | $\bar{\epsilon}$ | $\bar{\lambda}$ | $\bar{\epsilon}$ | $\bar{\lambda}$ | $\bar{\epsilon}$ | $\bar{\lambda}$ | $\bar{\epsilon}$ | $\bar{\lambda}$ |
| model 1 | 0.30 | 0.69 | 0.30 | 0.70 | 0.30 | 0.70 | 0.30 | 0.70 |
| model 2 | 0.36 | 0.60 | 0.35 | 0.64 | 0.35 | 0.65 | 0.35 | 0.64 |
| model 3 | 0.44 | 0.44 | 0.44 | 0.45 | 0.40 | 0.60 | 0.43 | 0.47 |
| model 4 | 0.51 | 0.48 | 0.50 | 0.50 | 0.50 | 0.50 | 0.51 | 0.48 |
| model 5 | 0.10 | 0.90 | 0.10 | 0.90 | 0.10 | 0.90 | 0.10 | 0.90 |

Table 2: Parameters of the four affililation models of the experiment. The $Q$ modules are mixed in the same proportion. Each model considers $n = 500$ nodes and $Q = 5$ groups.

| | online SAEM | | online Variational | | batch MixNet | | online CEM | |
|---|---|---|---|---|---|---|---|---|
| Model | $\overline{rand}$ | $\sigma_{rand}$ | $\overline{rand}$ | $\sigma_{rand}$ | $\overline{rand}$ | $\sigma_{rand}$ | $\overline{rand}$ | $\sigma_{rand}$ |
| model 1 | 0.98 | 0.02 | 0.98 | 0.02 | 0.99 | 0.02 | 0.98 | 0.02 |
| model 2 | 0.96 | 0.07 | 0.97 | 0.07 | 0.98 | 0.01 | 0.97 | 0.07 |
| model 3 | 0.13 | 0.13 | 0.10 | 0.15 | 0.85 | 0.14 | 0.25 | 0.16 |
| model 4 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| model 5 | 1 | 0.00 | 1 | 0.01 | 1 | 0.01 | 1 | 0.01 |

Table 3: Means and standard deviations of the Rand index for all models with $q$ and $n$ fixed.

| | online SAEM | | online Variational | | batch MixNet | | online CEM | |
|---|---|---|---|---|---|---|---|---|
| N | $\overline{rand}$ | $\sigma_{rand}$ | $\overline{rand}$ | $\sigma_{rand}$ | $\overline{rand}$ | $\sigma_{rand}$ | $\overline{rand}$ | $\sigma_{rand}$ |
| $n = 100$ | 0.16 | 0.09 | 0.26 | 0.14 | 0.26 | 0.14 | 0.23 | 0.13 |
| $n = 250$ | 0.94 | 0.08 | 0.97 | 0.06 | 0.99 | 0.01 | 0.97 | 0.06 |
| $n = 500$ | 0.97 | 0.07 | 0.97 | 0.07 | 1 | 0.00 | 0.97 | 0.07 |

Table 4: Means and standard deviations of the Rand index for all models with $n$ fixed. $q = 5$, model 2

| | online SAEM | | online Variational | | batch MixNet | | online CEM | |
|---|---|---|---|---|---|---|---|---|
| N | $\overline{rand}$ | $\overline{time}$ | $\overline{rand}$ | $\overline{time}$ | $\overline{rand}$ | $\overline{time}$ | $\overline{rand}$ | $\overline{time}$ |
| $n = 100$ | 0.14 | 0.07 | 0.14 | 0.11 | 0.26 | 0.21 | 0.14 | 0.07 |
| $n = 250$ | 0.47 | 0.76 | 0.48 | 0.77 | 0.99 | 2.08 | 0.48 | 0.74 |
| $n = 500$ | 0.64 | 0.97 | 0.67 | 1.02 | 1 | 25.00 | 0.66 | 0.95 |
| $n = 750$ | 0.82 | 2.20 | 0.83 | 2.36 | 1 | 125.3 | 0.83 | 2.14 |

Table 5: Means of the Rand index with speed of the algorithms. $q = 5$, model 2

| | | True | | |
|---|---|---|---|---|
| | | Conservative | Independent | Liberal |
| | cluster 1 | 734 | 135 | 238 |
| Estimated | cluster 2 | 290 | 26 | 8 |
| | cluster 3 | 2 | 7 | 430 |

Table 6: Contingency table comparing true and estimated partitions.

| | | True | | |
|---|---|---|---|---|
| | | Conservative | Independent | Liberal |
| Estimated | C1 | 210 | 2 | 2 |
| | C2 | 393 | 11 | 3 |
| | C3 | 13 | 37 | 31 |
| | C4 | 0 | 1 | 25 |
| | C5 | 192 | 97 | 219 |
| | C6 | 0 | 0 | 58 |
| | C7 | 0 | 0 | 51 |
| | C8 | 0 | 0 | 20 |
| | C9 | 66 | 0 | 0 |
| | C10 | 55 | 0 | 1 |
| | C11 | 3 | 5 | 199 |
| | C12 | 0 | 0 | 37 |
| | C13 | 0 | 0 | 23 |
| | C14 | 1 | 0 | 0 |
| | C15 | 10 | 10 | 3 |
| | C16 | 22 | 2 | 0 |
| | C17 | 0 | 3 | 1 |
| | C18 | 19 | 0 | 0 |
| | C19 | 36 | 0 | 0 |
| | C20 | 0 | 0 | 3 |

Table 7: Contingency table comparing true and estimated partitions