Wmixnet: Software for Clustering the Nodes of Binary and Valued Graphs using the Stochastic Block Model

Leger, J.-B. *†

Abstract

Clustering the nodes of a graph allows the analysis of the topology of a network.

The *stochastic block model* is a clustering method based on a probabilistic model. Initially developed for binary networks it has recently been extended to valued networks possibly with covariates on the edges.

We present an implementation of a variational EM algorithm. It is written using `C++`, parallelized, available under a GNU General Public License (version 3), and can select the optimal number of clusters using the ICL criteria. It allows us to analyze networks with ten thousand nodes in a reasonable amount of time.

1 Introduction

Complex networks are being more and more studied in different domains such as social sciences and biology. The network representation of the data is graphically attractive, but there is clearly a need for a synthetic model, giving an enlightening representation of complex networks. Statistical methods have been developed for analyzing complex data such as networks in a way that could reveal underlying data patterns through some form of classification.

Unsupervised classification of the vertices of networks is a rapidly developing area with many applications in social and biological sciences. The underlying idea is that common connectivity behavior shared by several vertices leads to their grouping in one *meta-vertex*, without losing too much information. Thus, the initial complex network can be reduced to a simpler *meta-network*, with few *meta-vertices* connected by few *meta-edges*.

*INRA, UMR 518 MIA, 16 rue Claude Bernard, Paris
†AgroParisTech, UMR 518 MIA, 16 rue Claude Bernard, Paris

Picard et al. (2009) show applications of this idea to biological networks and Nowicki and Snijders (2001) and Handcock et al. (2007) to social networks.

Model-based clustering methods model the heterogeneity between nodes by grouping the nodes into classes. The model used in this paper is an extension of the *stochastic block model* (SBM) (Nowicki and Snijders, 2001). This model assumes that the nodes are distributed into groups, and connectivity between nodes is driven by node group memberships.

SBM for non-binary graphs, with or without covariates has been introduced in Mariadassou et al. (2010). In this paper, a *variational Expectation–Maximization* algorithm has been used to estimate parameters and to predict groups.

This article introduces `wmixnet`, an implementation of the *variational expectation–maximization* algorithm for this extension of the *stochastic block model* with or without covariates for three families of laws of probability: Bernoulli, Poisson, Gaussian.

This implementation allows us to estimate parameters and to predict node groups and covariate effects for graphs which are valued or binary, directed or not, and with or without covariates.

2 SBM model with covariates

We introduce here the *stochastic block model* with covariates and three probability distributions.

2.1 Notations

Graph. Consider a graph $G = (V, E, w)$, where

- V is the set of nodes, labelled in $\{1, \cdots, n\}$,
- E is the set of edges, which is a subset of V^2,
- $w \colon E \to \mathbf{R}$, is the function which gives edge weights.
- $Y \colon V^2 \to \mathbf{R}^p$, is the function which gives the covariate vector associated to each couple of nodes.

We assume *without loss of generality* that $E = V^2$, with the convention $w(i, j) = 0$ if there is no edge from vertex i to vertex j.

Groups. Consider Q classes of nodes. For a given partition $(\mathcal{C}_1, \cdots, \mathcal{C}_Q)$ of V, for a node i and a group q, let Z be defined as $Z_{iq} = 1 \Leftrightarrow i \in \mathcal{C}_q$. And let $Z_i = (Z_{i1}, \cdots Z_{iQ})$.

2.2 The model

Nodes. The class memberships of the nodes are driven by independent identically distributed multinomial distributions:

$$\forall i \in V \; Z_i \overset{\text{i.i.d.}}{\sim} \mathcal{M}(1, \alpha)$$

where $\alpha = (\alpha_1, \cdots, \alpha_Q)$ and $\sum_q \alpha_q = 1$.

Edges. For each couple of nodes (i, j) the probability law of the link is driven by their class memberships and the (i, j) covariate $Y(i, j)$:

$$(w(i, j)|(i, j) \in \mathcal{C}_q \times \mathcal{C}_l) \sim \mathcal{F}_{ql}(Y(i, j)).$$

2.3 Probability laws

Generally, various probability laws can be used. The probability distributions which are implemented in `wmixnet` are the following:

- *Bernoulli*:

  without covariates: $\mathcal{F}_{ql}(Y(i, j)) = \mathcal{B}(\pi_{ql})$. This model does not use covariates and can model only binary networks. This is the classical *stochastic block model* model.

  with covariates (with homogeneous effects): $\mathcal{F}_{ql}(Y(i, j)) = \mathcal{B}(\pi_{ql} \frac{1}{1+\exp(-\beta^T Y_{ij})})$. This model uses covariates and can model only binary networks. The effect of covariates is the same for all pairs of classes.

  with covariates (with heterogeneous effects): $\mathcal{F}_{ql}(Y(i, j)) = \mathcal{B}(\pi_{ql} \frac{1}{1+\exp(-\beta_{ql}^T Y_{ij})})$. This model uses covariates and can model only binary networks. The effect of covariates is *not* the same for all pairs of classes.

- *Poisson*:

  without covariates: $\mathcal{F}_{ql}(Y(i, j)) = \mathcal{P}(\lambda_{ql})$. This model does not use covariates and can model networks with non negative integer weights.

  with covariates (with homogeneous effects): $\mathcal{F}_{ql}(Y(i, j)) = \mathcal{P}(\lambda_{ql}(Y(i, j))$ where $\lambda_{ql}(Y(i, j) = \lambda_{ql} \exp(\beta^T Y(i, j))$. This model uses covariates and can model networks with non negative integer weight. The effect of covariates is the same for all pairs of classes.

  with covariates (with heterogeneous effects): $\mathcal{F}_{ql}(Y(i, j)) = \mathcal{P}(\lambda_{ql}(Y(i, j))$ where $\lambda_{ql}(Y(i, j) = \lambda_{ql} \exp(\beta_{ql}^T Y(i, j))$. This model uses covariates and can model networks with non negative integer weight. The effect of covariates is *not* the same for all pairs of classes.

- *Gaussian*:

  without covariates: $\mathcal{F}_{ql}(Y(i,j)) = \mathcal{N}(\mu_{ql}, \sigma^2)$. This model does not use covariates and can model networks with real weight.

  with covariates (with homogeneous effects): $\mathcal{F}_{ql}(Y(i,j)) = \mathcal{N}(\mu_{ql}(Y(i,j)), \sigma^2)$ where $\mu_{ql}(Y(i,j)) = \mu_{ql} + \beta^T Y(i,j)$. This model uses covariates and can model networks with real weight. The effect of covariates is the same for all pair of classes.

  with covariates (with heterogeneous effects): $\mathcal{F}_{ql}(Y(i,j)) = \mathcal{N}(\mu_{ql}(Y(i,j)), \sigma^2)$ where $\mu_{ql}(Y(i,j)) = \mu_{ql} + \beta_{ql}^T Y(i,j)$. This model uses covariates and can model networks with real weight. The effect of covariates is *not* the same for all pair of classes.

2.4 Analysis of groups when covariates are used

Without covariates, groups are sets of nodes which have the same connectivity behavior (in probability), and groups can be easily interpretable using the connectivity matrix ($[\pi_{ql}]$, $[\lambda_{ql}]$ or $[\mu_{ql}]$).

With covariates, groups are sets of nodes which have the same connectivity behavior (in probability) *conditionally to covariates*. Two nodes of the same group can have different connectivity behavior due to different values of covariates.

For a model with covariates, groups are covariate-residual groups. There are two points of view:

- the focus is on the effects of the covariates and groups model the (residual) connectivity which is *not* explained by covariates,
- the focus is on the groups which helps in suggesting some sources of heterogeneity after correcting the artefact due to covariates.

One can test the effect of covariates using a likelihood ratio test between models with and without covariates.

3 Estimation method

The estimation method is described in Mariadassou et al. (2010). The likelihood is not computable in a reasonable time, and a variational approximation is done and a *variational expectation–maximization* is used. The ICL criterion is used for choosing the number of groups, see Mariadassou et al. (2010).

Some estimation implementation details which differ from the framework introduced in Mariadassou et al. (2010) are explained here.

3.1 Initialization

As in the general case on *expectation–maximization* algorithm, the initialization plays a major role in the quality of the local maximum found.

In Mariadassou et al. (2010), the authors propose to use a hierarchical clustering to initialize the algorithm. In a real case of network analysis this initialization is often an extremal one (most of the initialized groups contain only one node) and the *expectation–maximization* algorithm converges to a local maximum which may be far from the global maximum.

The Absolute Value Spectral Clustering algorithm is consistent for finding groups in SBM (with Bernoulli probability law without covariates), see Rohe et al. (2011). We use the absolute spectral clustering to initialize the *expectation–maximization* algorithm.

When there are covariates, the spectral clustering is done on the residual graph, after eliminating the effect of covariates by regression.

3.2 Smoothing

To determine if an estimation for Q groups has reached a bad local maximum, we use two findings:

- With an ascending number of groups, models are nested. A model with Q groups can be interpreted as a model with $Q + 1$ groups, so the likelihood must increase with Q.
- Empirical findings make us say that the ICL criterion is convex.

A reinitialization of the *expectation–maximization* can be done. The new initialization is obtained in two ways:

- merging two groups of the $Q + 1$ result (descend mode)
- splitting one group into two groups of the $Q - 1$ result (ascend mode), this split is done by a spectral clustering of the residual graph on $Q - 1$ groups.

There are two modes of reinitialization:

- the `minimal` one, reinitializations are done each time one of the two findings (see above) is not respected,
- the `exhaustive` one, all reinitializations are done; while it improves likelihood, this option is very time-consuming and cannot be used with non small graphs.

3.3 Parallelism

Many steps of the estimation can be done independently:

- The *expectation–maximization* algorithm for various Q
- Reinitialization in ascend and descend mode

Considering that computers and computing units have more than one logical processor, this implementation uses threads to parallelize the implementation as much as possible.

4 `wmixnet` program

This section introduces the `wmixnet` program and the program usage.

4.1 Sources availability and installation

`wmixnet` is provided on the GNU General Public Licence version 3, and `C++` sources are available on the `wmixnet` page:

- `http://www.agroparistech.fr/mia/productions:logiciels`
- `http://www.agroparistech.fr/mia/productions:logiciel:wmixnet`

`wmixnet` should be installable from sources on any Linux distribution, when dependencies are provided:

- `IT++` library, used for matrix calculation. This library uses `blas` and `lapack`, well-known algebra libraries.
- `boost` library, for many aspects including parallelism.

4.2 Input format

The input format is a plain text with the following specifications:

- each line describes a node
- for each line the first two columns describe the indexes of starting and ending nodes
- for each line the third column describes the weight of the edge
- for each line the fourth to end columns (if present) describe the covariates associated to the edge.

There are some constraints:

- node indexes must start from 1 to the number of nodes
- each edge must have the same number of covariates.

If an edge is not present, and if no covariates are used, the corresponding lines can be omitted; otherwise the line must be present with a weight of zero.

Functions are provided to write a file following these specifications, with adjacency matrices, and covariate matrices, for `GNU R`, and `MATLAB` or `GNU Octave`.

4.3 Output format

The output format contains the model parameters for all explored numbers of groups.

Model parameters are:

- α, the parameters of the multinomial distribution
- θ, the parameters of the probability law of the edge weight conditionally to groups of nodes,

  for the Bernoulli model, $\theta = (\pi)$,

  for the Poisson model, $\theta = (\lambda)$,

  for the Poisson model with covariates $\theta = (\lambda, \beta)$,

  for the Gaussian model $\theta = (\mu, \sigma^2)$,

  for the Gaussian model with covariates $\theta = (\mu, \sigma^2, \beta)$.

The output contains variational parameter estimates (τ) which give the nodes membership in groups.

The output also contains values of criteria such as pseudo-likelihood and the ICL criterion.

There are three output formats provided:

- Plain text output format (named text), which is a human readable file.
- `GNU R` file output format, which is an `GNU R` loadable file. Nevertheless this file can be easily read by a human.
- `MATLAB` or `GNU Octave` file output format, which is a `MATLAB` and `GNU Octave` loadable file. Nevertheless this file can be easily read by a human.

4.4 Command line usage

`wmixnet` is usable with command line, and the following arguments must be provided:

- `--input` to specify the input file,
- `--symmetric` to indicate if the graph is an undirected graph if applicable,
- `--model` to specify the model in

  `bernoulli` for Bernoulli without covariate

  `BH` for Bernoulli with covariates (homogeneous effects)

  `BI` for Bernoulli with covariates (heterogeneous effects)

  `poisson` for Poisson without covariate

  `PRMH` for Poisson with covariates (homogeneous effects)

  `PRMI` for Poisson with covariates (heterogeneous effects)

  `gaussian` for Gaussian without covariate

  `GRMH` for Gaussian with covariates (homogeneous effects)

  `GRMI` for Gaussian with covariates (heterogeneous effects)
- `--Qmax` to specify the maximum number of groups, or `--Qauto` to let the program choose the maximum number of groups,
- `--smoothing` to specify the smoothing mode

  `none` no reinitialization is done (by default)

  `minimal` reinitializations are done for detected problems

  `exhaustive` all reinitialization are done (time-consuming option, only for small graphs)
- `--output` to specify the output file,
- `--output-format` to specify the output format

  `text` (by default)

  `R` for `GNU R` loadable file

  `matlab` or `octave` which are synonymous for `MATLAB` and `GNU Octave` loadable file.

4.5 Empirical complexity

Some simulations suggest the following estimation of complexity:

$$t = C_{\text{model}}\, n^{2.46}\, g^{2.1}\, 1.03^{p}$$

with

- t the total processor time (equivalent time on a mono-core computer, without parallelization, which executes only this job)
- C_{model} a constant which depends on the model. Since absolute values are not pertinent, ratios are given:
  $\frac{C_{\text{poisson}}}{C_{\text{bernoulli}}} = 3.9$
  $\frac{C_{\text{PRMH}}}{C_{\text{bernoulli}}} = 21$
  $\frac{C_{\text{gaussian}}}{C_{\text{bernoulli}}} = 840$
  $\frac{C_{\text{GRMH}}}{C_{\text{bernoulli}}} = 1350$
  This ratio is dependent on the way each model is implemented. Some models allow us to vectorize some steps, have explicit maxima, and thus are significantly faster
- n the number of nodes
- g the number of groups found
- p the number of covariates (the size of the covariate vector)

4.6 Capacity of extension

In the `wmixnet` program, the estimation procedure and other model-common parts are implemented once. Only model-specific functions are present for each model. Therefore it is relatively easy to add other models in the `wmixnet` program.

5 Example

Here we introduce the analysis of two ecological networks, already analyzed by Mariadassou et al. (2010).

5.1 The networks

The networks are two undirected, valued networks having parasitic fungal species ($n = 154$) and tree species ($n = 51$) as nodes, respectively. Edge strengths was defined as the number of shared host species and the number of shared parasitic species, respectively (see Mariadassou et al., 2010 for details).

5.2 Covariate data

For the tree species network, we know the taxonomic distance between tree species and the degree of geographic overlap between tree species distribution. For the fungal species network, we know the taxonomic distance between fungal species (see Mariadassou et al., 2010 for details).

5.3 Example of command line

For the analysis of the tree species network, for the Poisson model without covariates, the command line is:

```
wmixnet --input Trees.spm --symmetric \
  --model poisson \
  --Qauto --smoothing exhaustive \
  --output Trees.m --output-format octave
```

5.4 Results

5.4.1 On the tree species network

In Figure 1, we plot the ICL criterion for Poisson model without and with covariates (taxonomic distance, geographical distance or both). For the model without covariates the maximum is reached with 7 groups. With the geographical covariates, the maximum is reached for 6 groups, with a little improvement of the ICL criterion. For the model with the taxonomic covariates, the maximum is reached for 4 groups, with a larger improvement of the ICL criterion. Adding the geographical covariates to the taxonomic covariates does not improve the criterion.

Ecological interpretations are presented in Mariadassou et al. (2010).

5.4.2 On the fungi network

In Figure 2, we plot the ICL criterion for the Poisson model without and with covariates (taxonomic distance). For the model with and without taxonomic covariates the maximum is reached with 15 groups in both cases. There is no real improvement by adding the taxonomic covariates to the model.

Ecological interpretations are presented in Mariadassou et al. (2010).

Acknowledgments

We thank Dominique Piou, Corinne Vacher, and the Département Santé des Forêts of the French Ministère de l'Agriculture et de la Pêche for allowing us to use the network data.

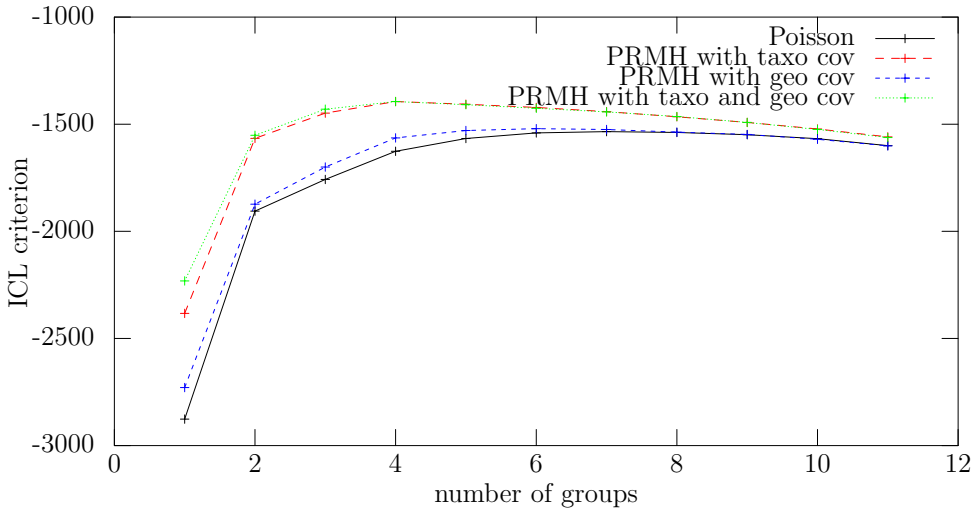

Figure 1: ICL criterion values obtained for Poisson and Poisson with covariates on the trees network

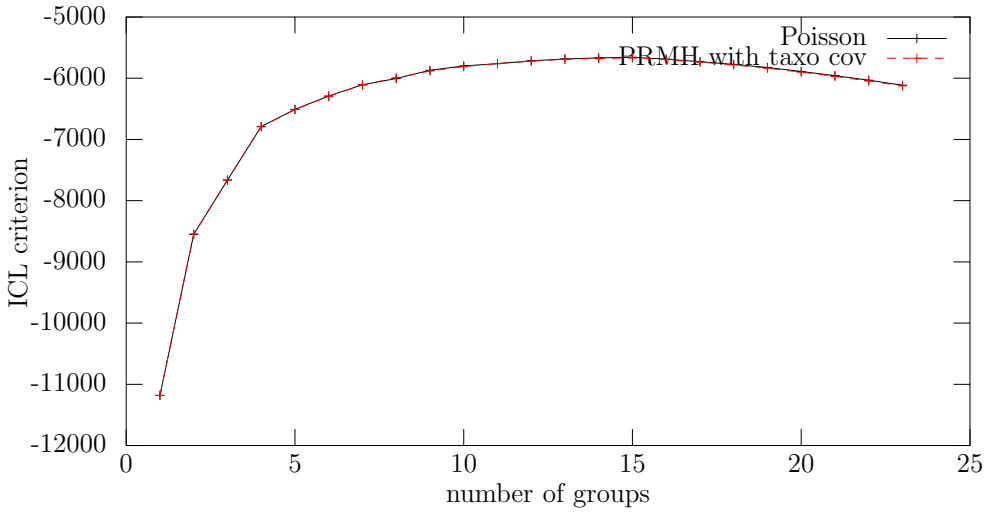

Figure 2: ICL criterion values obtained for the Poisson model and the Poisson with covariates model on the fungus species network.

We thank Stephane Robin for providing explanation and advice on the estimation procedure and Pierre Barbillon for the Bernoulli model with covariates (BH and BI) and for helping find some bugs in the estimation

procedure.

References

M. Handcock, A. Raftery, and J. Tantrum. Model-based clustering for social networks. *JRSSA*, 54:301–354, 2007.

M. Mariadassou, S. Robin, and C. Vacher. Uncovering latent structure in valued graphs: a variational approach. *Ann Appl Stat*, 4:715–742, 2010.

K. Nowicki and T. Snijders. Estimation and prediction for stochastic block-structures. *J. Am. Stat. Assoc.*, 96:1077–1087, 2001.

F. Picard, V. Miele, J.-J. Daudin, L. Cottret, and S. Robin. Deciphering the connectivity structure of biological networks using MixNet. *BMC Bioinformatics*, 10, 2009.

K. Rohe, S. Chatterjee, and B. Yu. Spectral clustering and the high-dimensional stochastic blockmodel. *The Annals of Statistics*, 39(4):1878–1915, 2011.