Nonparametric Edge Density Estimation in Large Graphs

Note technique

Référence : **Autre référence :** **Version :** **Date d'édition :**	**FT/RD/TECH/11/02/13** 1.0 2 février 2011	*Vérifié par : Fabrice Clérot* *Affiliation : TECH/ASAP* Le : 2 février 2011
Auteurs :	Boullé Marc TECH/ASAP	*Approuvé par : Patrice Soyer* *Affiliation : TECH/ASAP* Le : 2 février 2011

Résumé :
The discovery and analysis of structures in graphs has been long studied in the past. With the recent availability of many network data on the web, such as social networks, there is a renewed interest for these research topics, especially for the automatic discovery of community structures in large networks.
In this paper, we present a novel way to summarize the structure of a large graph, based on non-parametric estimation of edge density.
Following the stochastic blockmodeling approach, we exploit a clustering of the vertices, with a piecewise constant estimation of the density of the edges across the clusters, and address the problem of automatically and reliably inferring the number of clusters, that is the granularity of the clustering.
We exploit a novel model selection technique based on a Bayesian approach with data dependent prior and obtain an exact evaluation criterion for the posterior probability of edge density estimation models.
We exploit combinatorial optimization algorithms to search the best model, with a super-linear algorithmic complexity with respect to the number of edges.
We demonstrate, both theoretically and empirically, that our data dependent modeling technique is consistent, resilient to noise, valid non asymptotically and asymptotically behaves as an universal approximator of the true edge density in directed multigraphs.
We evaluate our approach on numerous artificial and real graphs. The results show the validity of the approach, that automatically provides an insightful summary of large graphs.

Mots clés :

Thème : 7500 - Informatique théor., langages et program., algorithm., complexité

Le présent document contient des informations qui sont la propriété de la R&D de France Télécom. L'acceptation de ce document par son destinataire implique, de la part de ce dernier, la reconnaissance du caractère confidentiel de son contenu et l'engagement de n'en faire aucune reproduction, aucune transmission à des tiers, aucune divulgation et aucune utilisation commerciale sans l'accord préalable écrit de la R&D de France Télécom.

© Copyright France Télécom 2011

© Copyright France Télécom 2011

Nonparametric Edge Density Estimation in Large Graphs

Marc Boullé marc.boulle@orange-ftgroup.com
Orange Labs
2, avenue Pierre Marzin
22300 Lannion, France

Abstract

The discovery and analysis of structures in graphs has been long studied in the past. With the recent availability of many network data on the web, such as social networks, there is a renewed interest for these research topics, especially for the automatic discovery of community structures in large networks. In this paper, we present a novel way to summarize the structure of a large graph, based on non-parametric estimation of edge density. Following the stochastic blockmodeling approach, we exploit a clustering of the vertices, with a piecewise constant estimation of the density of the edges across the clusters, and address the problem of automatically and reliably inferring the number of clusters, that is the granularity of the clustering. We exploit a novel model selection technique based on a Bayesian approach with data dependent prior and obtain an exact evaluation criterion for the posterior probability of edge density estimation models. We exploit combinatorial optimization algorithms to search the best model, with a super-linear algorithmic complexity with respect to the number of edges. We demonstrate, both theoretically and empirically, that our data dependent modeling technique is consistent, resilient to noise, valid non asymptotically and asymptotically behaves as an universal approximator of the true edge density in directed multigraphs. We evaluate our approach on numerous artificial and real graphs. The results show the validity of the approach, that automatically provides an insightful summary of large graphs.

Keywords: Random graphs, Community detection, Clustering, Bayesianism, Model Selection, Density estimation

1. Introduction

Graph partitioning has long been studied in the operational research field. One of the oldest approaches is the minimum-cut method, where the graph is divided into a predetermined number of disjoint subsets, usually of approximately the same size, chosen such that the number of edges between the clusters of vertices is minimized. This combinatorial optimization problem arises in various practical applications like telecommunication network partitioning, VLSI (very large-scale integration) circuit placement or load balancing for parallel computing in order to minimize communication between processor nodes. Due to NP-hardness (Garey and Johnson, 1979), many heuristics have been proposed in the litterature. For example, algorithms such as (Kernighan and Lin, 1970; Fiduccia and Mattheyses, 1982) are frequently used to locally improve bisections. Many meta-heuristics have also been exploited such as simulated annealing (Kirkpatrick et al., 1983) evaluated by (Johnson et al., 1989), genetic algorithms used in (Bui and Moon, 1996) or tabu search

(Glover, 1987) enhanced and adapted to the bisection problem by (Battiti and Bertossi, 1999). The multilevel approach (Hendrickson and Leland, 1995; Karypis and Kumar, 1998) is specially fitted to very large graphs and constrained computation time. These families of heuristics represent a range of options for the trade-off between computation time and quality of the solution.

With the recent availability of many network data, such as world wide web, social networks, phone call networks, science collaboration graphs (Albert and Barabási, 2002), there is a renewed interest for the graph partitioning problem, especially for the automatic discovery of community structures in large networks. Whereas the classical graph balanced partitioning formulation works well in many of the applications for which it was originally intended, its is less appealing for the problem of finding "natural" cluster-based structures in general graphs since it will find clusters with constrained size and predefined number regardless of whether they are relevant in the graph. Many approaches have been studied for the problem of graph clustering, including hierarchical clustering, divisive clustering, spectral methods, random walk (see (Schaeffer, 2007) for a survey). To evaluate the quality of a clustering regardless of the cluster number, the modularity criterion proposed by (Newman and Girvan, 2003) is now widely accepted in the literature, and has even been treated as an objective function in clustering algorithms (Clauset et al., 2004; Danon et al., 2005; Blondel et al., 2008). This criterion aims at obtaining dense clusters where the within-cluster edge density is above the expected edge density in case of random edges following the same vertex degree distribution.

In this paper, we present a novel way of analyzing and summarizing the structure of large graphs, based on piecewise constant edge density estimation. The approach extends the stochastic blockmodeling approach (Wasserman et al., 2007; Copic et al., 2009; Bickel and Chen, 2009; Goldenberg et al., 2010) in that the modeling method is fully non-parametric with the number of clusters as a free parameter, and exploits a novel statistical model selection technique and scalable optimization algorithms. We apply data grid models (Boullé, 2008b) to graph data, where each edge is considered as a statistical unit with two variables, the source and target vertices. The objective is to find a correlation model between the two variables, owing to a data grid model, which in this case turns to be a coclustering of both the source and target vertices of the graph. The cells resulting from the cross-product of the two clusterings summarize the edge density in the graph. The best correlation model is selected using the MODL (Minimum Optimized Description Length) approach (Boullé, 2005, 2006), and optimized by the means of combinatorial heuristics with super-linear time complexity.

The rest of the paper is organized as follows. In Section 2, we present the MODL approach for data grid models and apply it to edge density estimation in graphs. We illustrate the distinctive features of the approach in Section 3 and evaluate it on benchmark graphs in Section 4. In Section 5, we relate our method to the literature on statistical blockmodeling, further analyze the problem of edge density estimation and demonstrate the consistency of the MODL approach as a universal density approximator. We present several research directions in Section 6 and finally give a summary in Section 7.

2. MODL Approach for Edge Density Estimation in Graphs

In this section, we first recall some basic notions of graph theory, next summarize the principles of data grid models introduced in (Boullé, 2010) in the data mining field for supervised and unsupervised data preparation. We then adapt the approach to the case of edge density estimation in a directed multigraph and relate it to information theory. We finally describe the optimization algorithm.

2.1 Basic Notions of Graph Theory

A *graph* $G = (V, E)$ consists of a set V of *vertices* and a set E of pairs of vertices called *edges*. A graph is *undirected* if the edges are unordered pairs of vertices, and is *directed* if the edges are ordered. A *loop* is an edge from one vertex to itself. A graph is *simple* in case of at most one edge per pair of vertices, and is *multiple* otherwise. Two vertices of an undirected graph are called *adjacent* if there is an edge connecting then. An edge is *incident* to its two vertices, called *extremities*. In case of directed graph, the extremities of an edges are called the *source* and *target* vertices of the edge. Figure 1 displays an example of directed simple graph, and Figure 2 an example of directed multigraph with loops.

A graph is called *bipartite* if the vertices can be partitioned into two disjoint sets, such that every edge connects one vertex of each vertex set. A simple graph is *complete* if it contains exactly one edge per pair of distinct vertices. A *subgraph* of a graph consists of a subset of the vertices and the edges of the graph. A complete subgraph is called a *clique*: it is a subset of pairwise adjacent vertices. A *coclique* is a subset of pairwise nonadjacent vertices.

The *degree* of a vertex is the number of edges incident to it. In a directed graph, the *in-degree* of a vertex v if the number of edges with target v, and the *out-degree* of v is the number of edges with source v. In an undirected graph, the sum of the degrees of the vertices is equal twice the number of the edges. In a directed graph, the sum of the in-degrees and the sum of the out-degrees of the vertices are equal to the number of edges.

Graphs can be represented by their *adjacency matrix*, where each cell of the matrix contains the number of edges per pair of vertices. The adjacency matrix of simple graphs contain only binary values, and that of undirected graphs is symmetrical. Figure 2 shows a directed multigraph and its adjacency matrix, as well as the in and out-degrees of each vertex.

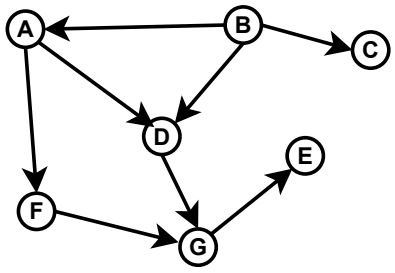
2.2 Data Grid Models for Data Preparation in Data Mining

Data mining is "the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data" (Fayyad et al., 1996). Most data mining techniques work on flat tabular data, with one instance per row and one variable, numerical or categorical, per column. Supervised data mining aims at predicting the value of one target variable given the other explanatory variables: the task is classification in case of a categorical target variable and regression in case of a target numerical variable. Unsupervised learning aims at discovering clusters in the data, association rules between the variables or at modeling correlations or joint density.

Data grid models (Boullé, 2008b, 2010) have been introduced for the data preparation phase of the data mining process (Chapman et al., 2000), which is a key phase, both time consuming and critical for the quality of the results. They allow to automatically, rapidly and reliably evaluate the class conditional probability of any subset of variables in supervised learning and the joint probability in unsupervised learning. Data grid models are based on a partitioning of each variable into intervals in the numerical case and into groups of values in the categorical case. The cross-product of the univariate partitions forms a multivariate partition of the representation space into a set of cells. This multivariate partition, called data grid, is a piecewise constant nonparametric estimator of the conditional or joint probability. The best data grid is searched using a Bayesian model selection approach and an efficient combinatorial algorithm.

2.3 Edge Density Estimation Models

We reformulate the data grid approach in the context of edge density estimation in directed multigraphs. As shown in Figure 1, a directed graph can be represented in a tabular format with two variables, source vertex and target vertex, and one line per edge described by its two vertices. We can then apply the data grid models in the unsupervised setting to estimate the joint density between these two variables, that is the density of edges in the graph.



Source	Target
A	D
A	F
B	A
B	C
B	D
D	G
F	G
G	E

Figure 1: Directed simple graph and its tabular representation.

Our objective is to provide a joint description of the source and target vertices, which amounts to a description of the edges in the graph. One simple way to describe the edges exploits the tabular format shown in Figure 1, with the count of edges per pair (Source, Target) of vertices. We can also summarize the location of edges at a coarser grain by introducing clusters of sources vertices and clusters of target vertices, and keeping the number of edges inside each cluster and across each cluster. Such clustering based model of the graph provides an estimator of the edge density, which is peacewise constant per pair of source and target cluster (cocluster). The coarsest summary is based on one single cluster of vertices with just the total number of edges, whereas the finest summary exploits one cluster per vertex. Coarse grained summaries tend to be reliable, whereas fine grained summaries are more informative. The issue is to find a trade-off between the informativeness of the edge density estimation and its reliability, on the basis of the granularity of the clustering.

We introduce a family of edge density estimation partitioning models, based on clusters of source and target vertices and on a multinomial distribution of the edges on the coclusters. This family of models is formalized in Definition 1.

Definition 1 *An edge density estimation model is defined by:*

- *a number of source and target clusters of vertices,*
- *the repartition of the source (resp. target) vertices into the source (resp. target) clusters of vertices,*
- *the distribution of the edges of the graph on the coclusters,*
- *for each source (resp. target) cluster of vertices, the distribution of the edges whose source (resp. target) belong to the cluster on the vertices of the cluster.*

Notation.

- $G = (V, E)$: graph with vertex set V and edge set E
- S, T: source and target vertex sets
- $n = |V|, n_S = |S|, n_T = |T|$: number of vertices, of source and target vertices
- $m = |E|$: number of edges
- k_S, k_T: number of clusters of source and target vertices
- $k_E = k_S k_T$: number of coclusters
- $k_S(i), k_T(j)$: index of the cluster containing source vertex i (resp. target vertex j)
- n_i^S, n_j^T: number of vertices in source cluster i (resp. target cluster j)
- $m_{i\cdot}, m_{\cdot j}$: number of edges for source vertex i (resp. target vertex j), i.e. out-degree of vertex i and in-degree of vertex j
- $m_{i\cdot}^S, m_{\cdot j}^T$: number of edges originating in source cluster i (resp. terminating in target cluster j)
- m_{ij}: number of edges for pair (i, j) of vertices
- m_{ij}^{ST}: number of edges for coclusters (i, j)

These notations are illustrated in Figure 2, where a directed multigraph is displayed with its adjacency matrix. A clustered version of this graph is presented in Figure 3, which results in a coclustering of its adjacency matrix.

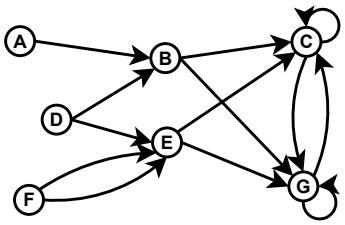
We assume that the numbers of edges m and of source and target vertices n_S and n_T are known in advance and we aim at modeling the joint distribution of the m edges on these two sets of vertices. This setting is general enough to account for directed graphs, bipartite graphs and undirected graph, where each edge comes twice with the two directions.

The family of models introduced in Definition 1 is completely defined by the parameters describing the partition of the vertices into clusters

$$k_S, k_T, \{k_S(i)\}_{1 \leq i \leq n_S}, \{k_T(j)\}_{1 \leq j \leq n_T},$$

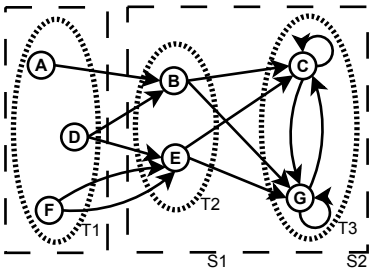
by the parameters of the multinomial distribution of the edges on the coclusters

$$\{m_{ij}^{ST}\}_{1 \leq i \leq k_S, 1 \leq j \leq k_T},$$



	A	B	C	D	E	F	G	$\sum$
A	0	1	0	0	0	0	0	1
B	0	0	1	0	0	0	1	2
C	0	0	1	0	0	0	1	2
D	0	1	0	0	1	0	0	2
E	0	0	1	0	0	0	1	2
F	0	0	0	0	2	0	0	2
G	0	0	1	0	0	0	1	2
$\sum$	0	2	4	0	3	0	4	13

Figure 2: Directed multigraph and its adjacency matrix. The numbers m_{ij} in the adjacency matrix are the numbers of edges for each pair of vertices (for example, two edges from F to E). The sums $m_{i.}$ on the right column are the out-degrees of the vertices, and the sums $m_{.j}$ on the bottom line are the in-degrees of the vertices. The total number of edges is on the bottom right corner of the adjacency matrix.



	A	D	F	B	E	C	G	$\sum$
A	0	0	0	1	0	0	0	1
D	0	0	0	1	1	0	0	2
F	0	0	0	0	2	0	0	2
B	0	0	0	0	0	1	1	2
E	0	0	0	0	0	1	1	2
C	0	0	0	0	0	1	1	2
G	0	0	0	0	0	1	1	2
$\sum$	0	0	0	2	3	4	4	13

	T1	T2	T3	$\sum$
S1	0	5	0	5
S2	0	0	8	8
$\sum$	0	5	8	13

Figure 3: Directed multigraph with two source and three target clusters. The adjacency matrix of the graph (reorganized by clusters) is presented in the middle, and that of the clustered graph on the right. The numbers m_{ij}^{ST} in the clustered adjacency matrix are the numbers of edges for each cocluster (for example, 5 edges from S1 to T2).

and by the parameters of the multinomial distribution of the edges originating in each source cluster (resp. terminating in each target cluster) on the vertices of the cluster

$$\{m_{i.}\}_{1 \leq i \leq n_S}, \{m_{.j}\}_{1 \leq j \leq n_T}.$$

The numbers of vertices per cluster n_i^S and n_j^T are derived from the specification of the partitions of vertices into clusters: they do not belong to the model parameters. Similarly, the number of edges originating or terminating in each cluster can be deduced by adding the frequencies of coclusters, according to $m_{i.}^S = \sum_{j=1}^{k_T} m_{ij}^{ST}$ and $m_{.j}^T = \sum_{i=1}^{k_S} m_{ij}^{ST}$.

In order to select the best model, we apply a Bayesian approach, using the prior distribution on the model parameters described in Definition 2.

Definition 2 *The prior for the parameters of an edge density estimation model are chosen hierarchically and uniformly at each level:*

- *the numbers of clusters k_S and k_T are independent from each other, and uniformly distributed between 1 and n_S for the source vertices, between 1 and n_T for the target vertices,*
- *for a given number k_S of source clusters, every partition of the n_S vertices into k_S clusters is equiprobable,*
- *for a given number k_T of target clusters, every partition of the n_T vertices into k_T clusters is equiprobable,*
- *for a model of size (k_S, k_T), every distribution of the m edges on the $k_E = k_S k_T$ coclusters is equiprobable,*
- *for a given cluster of sources (resp. target) vertices, every distribution of the edges originating (resp. terminating) in the cluster on the vertices of the cluster is equiprobable.*

Taking the negative log of the probabilities, this provides the evaluation criterion given in Theorem 3 (Boullé, 2010).

Theorem 3 *An edge density estimation model M distributed according to a uniform hierarchical prior is Bayes optimal if the value of the following criteria is minimal*

$$\begin{aligned} c(M) = & \log n_S + \log n_T + \log B(n_S, k_S) + \log B(n_T, k_T) \\ & + \log \binom{m + k_E - 1}{k_E - 1} + \sum_{i=1}^{k_S} \log \binom{m_{i.}^S + n_i^S - 1}{n_i^S - 1} + \sum_{j=1}^{k_T} \log \binom{m_{.j}^T + n_j^T - 1}{n_j^T - 1} \\ & + \log m! - \sum_{i=1}^{k_S} \sum_{j=1}^{k_T} \log m_{ij}^{ST}! \\ & + \sum_{i=1}^{k_S} \log m_{i.}^S! - \sum_{i=1}^{n_S} \log m_{i.}! + \sum_{j=1}^{k_T} \log m_{.j}^T! - \sum_{j=1}^{n_T} \log m_{.j}! \end{aligned} \tag{1}$$

$B(n, k)$ is the number of divisions of n elements into k subsets (with potentially empty subsets). When $n = k$, $B(n, k)$ is the Bell number. In the general case, $B(n, k)$ can be written as $B(n, k) = \sum_{i=1}^{k} S(n, i)$, where $S(n, i)$ is the Stirling number of the second kind (see Abramowitz and Stegun, 1970), which stands for the number of ways of partitioning a set of n elements into i nonempty subsets.

The first line in Formula 1 relates to the prior distribution of the cluster numbers k_S and k_T and to the specification the partition of the source (resp. target) vertices into clusters. These terms are the same as in the case of the MODL supervised univariate value grouping method (Boullé, 2005). The second line in Formula 1 represents the specification of the parameters of the multinomial distribution of the m edges on the k_E coclusters, followed by the specification of the multinomial distribution of the edges originating (resp. terminating) in each cluster on the vertices of the cluster. The third line stands for the likelihood of the distribution of the edges on the coclusters, by the mean of a multinomial term. The last line corresponds to the likelihood of the distribution of the edges originating (resp. terminating) in each cluster on the vertices of the cluster.

2.4 Relation with Information Theory

Null model. Let us first introduce the null model $M_\emptyset$, with one single cluster of source (resp. target) vertices and one cocluster, containing all the edges. Applying Formula 1, the cost $c(M_\emptyset)$ of the null model (its value according to evaluation criterion 1) reduces to

$$\begin{aligned} c(M_\emptyset) = & \log n_S + \log n_T + \log \binom{m + n_S - 1}{n_S - 1} + \log \binom{m + n_T - 1}{n_T - 1} \\ & + \log \frac{m!}{m_{1.}! m_{2.}! \ldots m_{n_S.}!} + \log \frac{m!}{m_{.1}! m_{.2}! \ldots m_{.n_T}!} \end{aligned} \tag{2}$$

which corresponds to the posterior probability of the multinomial model for the distribution of the edges on the source vertices and on the target vertices. This means that the source and target vertices are described independently.

Entropy of the null model. To get an asymptotic evaluation of the cost of the null model, we now introduce the Shannon entropy $H(X)$ (Shannon, 1948) of a discrete variable X, $H(X) = -\sum_{x \in X} p(x) \log p(x)$. Let us consider edges as statistical instances, with two vertex variables V_S and V_T having n_S and n_T values. As $m_{i.}$ stands for the out-degree of vertex i, the probability of originating from vertex i can be estimated by $\frac{m_{i.}}{m}$. We thus get $H(V_S) = -\sum_{i=1}^{n_S} \frac{m_{i.}}{m} \log \frac{m_{i.}}{m}$ and $H(V_T) = -\sum_{j=1}^{n_T} \frac{m_{.j}}{m} \log \frac{m_{.j}}{m}$.

Using the approximation $\log n! = n(\log n - 1) + O(\log n)$ based on Stirling's formula, the cost of the null model is asymptotically equivalent to m times the Shannon entropy of the source and target vertex variables V_S and V_T:

$$c(M_\emptyset) = mH(V_S) + mH(V_T) + O(\log m). \tag{3}$$

Coding length of edge density models. As the negative log of a probability can be interpreted as a coding length (Shannon, 1948), our model selection technique is closely related to the minimum description length (MDL) approach (Rissanen, 1978; Hansen and Yu, 2001; Grünwald et al., 2005), which aims to approximate the Kolmogorov complexity (Li and Vitanyi, 1997) for the coding length of the data (edges of the graph). The Kolmogorov complexity is the length of the shortest computer program that encodes the data. The prior terms in Formula 1 represent the coding length of the edge density model parameters whereas the likelihood terms represent the coding length of the data (the edges) given the model.

Robust edge density estimation in graphs. Overall, our prior approximates the Kolmogorov complexity of the edge density model given the vertices and our conditional likelihood encodes the edges given the model. In our approach, the choice of the null model corresponds to the lack of reliable structure in the graph. The coding length of the null model is asymptotically equivalent to the Shannon entropy of the distribution of the source and target vertex degrees (cf. Formula 3), which corresponds to a basic encoding of the edges, without any use of structure in the graph. This is close to the idea of Kolmogorov, who considers data to be random if its algorithmic complexity is high, that is if it cannot be compressed significantly. This makes our approach very robust, since detecting reliable structures using edge density models is necessarily related to a coding length better than

that of the null model, thus to non random patterns according Kolmogorov's definition of randomness. This robustness has been confirmed using extensive experiments in the case of univariate data preparation for supervised data mining (Boullé, 2006, 2005), and is evaluated in the case of graphs in Section 4.

2.5 Optimization Algorithm

Edge density estimation models are no other than data grid models (Boullé, 2010) applied to the case of joint density estimation of the source and target vertices of the edges. The space of data grid models is so large that straightforward algorithms almost surely fail to obtain good solutions within a practicable computational time. Given that criterion 1 is optimal, the design of sophisticated optimization algorithms is both necessary and meaningful. Such algorithms are described in (Boullé, 2008a). They finely exploit the sparseness of the adjacency matrix of the graph and the additivity of the criterion, and allow a deep search in the model space with $\mathrm{O}(m)$ memory complexity and $\mathrm{O}(m\sqrt{m}\log m)$ time complexity.

In this section, we give an overview of the optimization algorithms which are fully detailed in (Boullé, 2008a), and rephrase them using the graph terminology. The optimization of a data grid is a combinatorial problem. The number of possible partitions of n vertices is equal to the Bell number $B(n) = \frac{1}{e}\sum_{k=1}^{\infty}\frac{k^n}{k!}$. Even with very simple models having only two clusters of source and target vertices, the number of models involves $2^{n_S n_T}$ coclusterings of the vertices. An exhaustive search through the whole space of models is unrealistic. We describe in Algorithm 1 a greedy bottom up merge heuristic (GBUM) which optimizes the model criterion 1. The method starts with a fine grained model, with few vertices per source or target cluster, up to the maximum model M_{Max} with one vertex per source or target cluster. It considers all the merges between adjacent clusters (independently for the source and target sets of vertices), and performs the best merge if the criterion decreases after the merge. The process is reiterated until no further merge decreases the criterion.

Algorithm 1 Greedy Bottom Up Merge heuristic (GBUM)

Require: M {Initial solution}
Ensure: $M^*, c(M^*) \leq c(M)$ {Final solution with improved cost}
1: $M^* \leftarrow M$
2: **while** improved solution **do**
3: $M' \leftarrow M^*$
4: **for all** Merge m between two source or target clusters **do**
5: $M^+ \leftarrow M^* + m$ {Consider merge m for model M^*}
6: **if** $c(M^+) < c(M')$ **then**
7: $M' \leftarrow M^+$
8: **end if**
9: **end for**
10: **if** $c(M') < c(M^*)$ **then**
11: $M^* \leftarrow M'$ {Improved solution}
12: **end if**
13: **end while**

Each evaluation of the criterion for a model requires $O(n^2)$ time, since the initial model contains up to $n_S n_T$ coclusters (see formula (1)) in the case of the maximal model M_{Max}. Each step of the algorithm relies on $O(n^2)$ evaluations of merges of clusters of vertices, and there are at most $O(n)$ steps, since the model becomes equal to the null model $M_\emptyset$ once all the possible merges have been performed. Overall, the time complexity of the algorithm is $O(n^5)$ using a straightforward implementation of the algorithm. However, the method can be optimized in $O(m\sqrt{m}\log m)$ time, as demonstrated in (Boullé, 2008a). The optimized algorithm mainly exploits the sparseness of the data, the additivity of the criterion and starts from non-maximal models with pre and post-optimization heuristics.

- Large graph are often sparse, with far less edges than in complete graphs. Although a model may contain $O(n^2)$ coclusters, at most m clusters are non empty. Since the contribution of empty coclusters is null in the criterion 1, each evaluation of a data grid can be performed in $O(m)$ time owing to specific algorithmic data structures.

- The additivity of the criterion means that it can be decomposed on the hierarchy of the components of the models: extremity (sources vs target variable), cluster of vertices, cocluster. Using this additivity property, all the merges between adjacent clusters can be evaluated in $O(m)$ time. Furthermore, when the best merge is performed, the only impacted merges that need to be reevaluated for the next optimization step are the merges that share edges with the best merge. Since the graph is potentially sparse, the number of reevaluations of models may be small on average.

- Finally, the algorithm starts from initial fine grained solutions containing at most $O(\sqrt{m})$ clusters. Specific pre-processing and post-processing heuristics are exploited to locally improve the initial and final solutions of Algorithm 1 by moving vertices across clusters. The post-optimization algorithms are applied alternatively to the source and target vertex variables, for a frozen partition of the other variable. This allows to keep a $O(m)$ memory complexity and to bound the time complexity by $O(m\sqrt{m}\log m)$.

Sophisticated algorithmic data structures and algorithms are necessary to exploit these optimization principles and guarantee a time complexity of $O(m\sqrt{m}\log m)$ for initial solutions exploiting at most $O(\sqrt{m})$ clusters of vertices.

The optimized version of the greedy heuristic is time efficient, but it may fall into a local optimum. This problem is tackled using the variable neighborhood search (VNS) meta-heuristic (Hansen and Mladenovic, 2001), which mainly benefits from multiple runs of the algorithms with different random initial solutions. In practice, the main heuristic described in Algorithm 1, with its guaranteed time complexity, is used to find a good solution as quickly as possible. The VNS meta-heuristic is exploited to perform anytime optimization: the more you optimize, the better the solution.

The optimization algorithms summarized above have been extensively evaluated in (Boullé, 2008a), using a large variety of artificial datasets, where the true data distribution is known. Overall, the method is both resilient to noise and able to detect complex fine grained patterns. It is able to approximate any data distribution, provided that there are enough instances in the train data sample.

3. Illustration

This section, intended to be of a tutorial nature, points out the difference between stochastic edge density blockmodeling and deterministic clustering. This way, we illustrate the behavior of our approach using several small artificial datasets, and show how the discovered patterns encompass and significantly extend the community patterns discovered using modularity based clustering methods.

3.1 Artificial Graph Family

We introduce a family of artificial graphs consisting in four clusters of ten vertices, named A, B, C, D. For two-dimensional depiction purpose, we consider the case of undirected simple graphs, with at most one edge per pair of vertices and no loops, and control the proportion of potential edges per cocluster, that is per pair of clusters of vertices. This is illustrated in Figure 4, where the four cluster of vertices are drawn on circles for better readability. For example, choosing a proportion $p = 50\%$ for the edges of (A, B) means that 50% of the potential edges with one extremity in A and the other one in B (among $100 = 10 * 10$ edges) are in the graph. A random graph is produced by generating a random value $v \in [0, 1]$ for each edge of the complete graph and keeping the edge if $v \leq p$ in the related cocluster. In the rest of the section, we study several distributions of edges, with one randomly generated graph for each distribution.

All the graph clusterings based on data grid models are produced using the Khiops tool[1]. Khiops is a general purpose data preparation and scoring tool which implements the method described in Section 2. For the graph clustering problem, it is applied on a graph dataset consisting of two variables, Source and Target, with one record per edge (two for undirected graphs), for the task of unsupervised bivariate analysis.

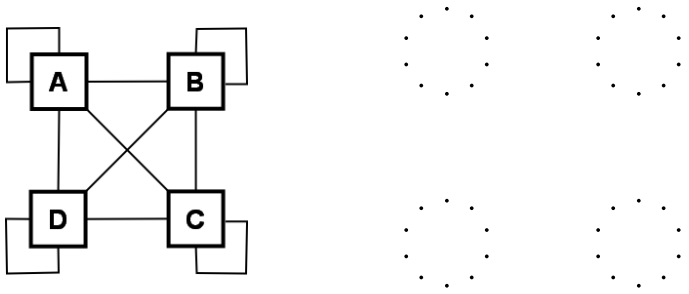


Figure 4: Artificial graph family.

3.2 Random Edge Distribution

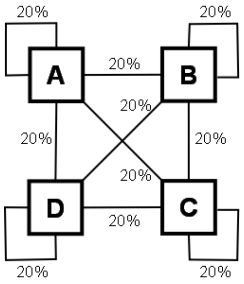
We study the case of a random graph, were the probability of edges is uniform; each potential edge has a probability 20% of being in the graph, which means that within each cocluster, each potential edge has a probability 20% of being in the graph. Figure 5 shows on the left the parameters of this distribution, on the right an example of a random graph generated

1. Khiops tool: available as a shareware on http://www.khiops.com

according to this distribution of edges, and in the middle a contingency table summarizing the graph by the numbers of edges per cocluster.

In order to check whether the source and target extremities of the edges are independent for the random graph generated in Figure 5, we use a chi-square test of independence with $9 = (4-1)(4-1)$ degrees of freedom. For two independent variables, the critical value at 5% is 16.919: this means that the probability of getting a chi-square value above 16.919 is 5%. The critical value is 21.666 at 1% and 27.877 at 0.1%. In the case of the random graph drawn in Figure 5, the chi-square computed from the contingency table is 7.198. Thus, using the chi-square test, the hypothesis of independence cannot be rejected. This sanity check will be used throughout the rest of the section.

Using the approach described in Section 2, our methods builds one single cluster of vertices. This confirms experimentally the analysis presented in Section 2.4, which claims that any random graph should be summarized using the null model, having one single cluster.



	A	B	C	D	$\sum$
A	18	26	17	19	80
B	26	14	20	23	83
C	17	20	22	21	80
D	19	23	21	22	85
$\sum$	80	83	80	85	328

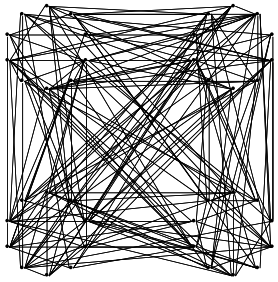


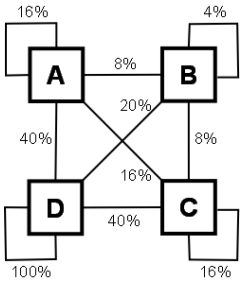
Figure 5: Artificial graph: random edges.

3.3 Random Edges with Unbalanced Distribution of Vertex Degrees

We now study the case of a graph where the extremities of the edges are chosen independently according to an unbalanced distribution of the vertex degrees. Each edge has a probability 50% of having an extremity in vertex cluster D, 20% in A, 20% in C and, 10% in B. Thus, the probability of having an edge in (D, D) is $25\% = 50\% * 50\%$, in (A, D) is $10\% = 50\% * 20\%$, in (B, D) is $5\% = 50\% * 10\%$... Generating around 400 edges, the proportions of edges used in the distribution on the left of Figure 6 follow this unbalanced distribution of the vertex degrees, with the source and target extremities independently drawn from the same distribution. The graph pictured on the right of Figure 6 shows an example of such a graph. The chi-square value computed from the contingency table is 3.321, far below the critical value, which confirms that the hypothesis of independence between the edges extremities cannot be rejected.

Our method builds one single cluster for this sample graph, which complies the analysis of Section 2.4. Although Figure 6 clearly exhibits a pattern, with one high density vertex cluster D, two medium density clusters A and C and one small density cluster B, the independence between the edges extremities provides the simplest explanation for this pattern. We have $p(edge) = p(source, target) = p(source)p(target)$. Thus $p(edge|source) = p(target)$. In other words, knowing the source vertex of an edge provides

no information about the target vertex of the edge. This is why there is no correlation pattern in this sample graph according to our approach.



	A	B	C	D	$\sum$
A	16	9	14	41	80
B	9	8	10	23	50
C	14	10	20	46	90
D	41	23	46	90	200
$\sum$	80	50	90	200	420

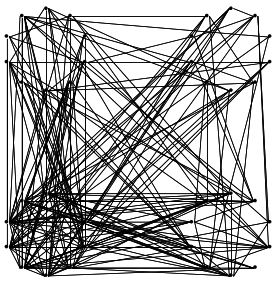


Figure 6: Artificial graph: random edges with unbalanced distribution of vertex degrees.

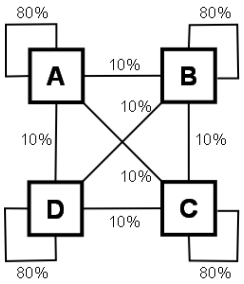
3.4 Quasi-cliques

Figure 7 provides a classical pattern consisting of four dense vertex clusters, with an intra edge density of 80% and an inter edge density of 10%. Our method recognizes this pattern, building four vertex clusters related to A, B, C, D. It is noteworthy that the number of partitions of 40 vertices is based on the Bell number $B(40) \approx 1.6\ 10^{35}$: Although the model space is huge, our algorithm manages to construct the correct number of clusters with their correct composition.

The chi-square value computed from the contingency table is 401.840, far beyond the critical value 27.877 at 0.1% rejection rate or even the critical value 44.811 at 0.0001%. Since a huge number of vertex partitions are considers, there could be a risk of overfitting the data. Even at a rejection rate $0.0001\% = 10^{-6}$, if millions of clustering models are evaluated using the chi-square test of independance, one of them could have a chi-square value beyond the rejection rate just by random. This problem is addressed in our approach using a Bayesian model selection technique with the prior for the model parameters introduced in Definition 2. The criterion 1 provides a balance between the model complexity, related to the number of considered models, and the likelihood of the observed edge distribution given the model. For example, the chi-square value 401.840 for the sample graph in Figure 7 corresponds to a rejection rate of around 10^{-80}. This demonstrates that our model selection approach clearly accounts for the huge number of considered models and that the selected model reliably fits the data.

3.5 Cocliques

Figure 8 exposes an unusual pattern with four vertex clusters, with a null intra edge density and an inter edge density of 50%. The four clusters are correctly identified by our method. The chi-square value is 217.714, still far beyond the critical value 27.877 at 0.1% rejection rate. This illustrates one main difference between our approach and most alternative graph clustering approaches. Whereas standard approaches are essentially parametric and aim at finding dense clusters in graphs, our approach behaves as a non-parametric edge density



	A	B	C	D	$\sum$
A	74	14	9	12	109
B	14	68	12	16	110
C	9	12	76	13	110
D	12	16	13	76	117
$\sum$	109	110	110	117	446

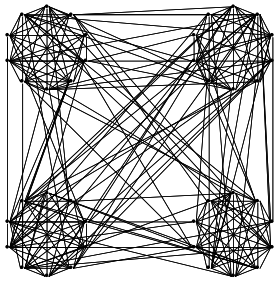
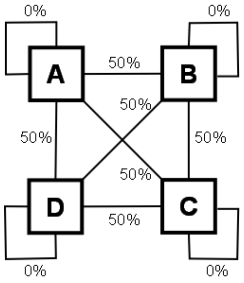


Figure 7: Artificial graph: quasi-cliques.

estimator whose objective is to summarize the edge density in the graph using a piecewise constant approximator.



	A	B	C	D	$\sum$
A	0	60	49	51	160
B	60	0	46	50	156
C	49	46	0	59	154
D	51	50	59	0	160
$\sum$	160	156	154	160	630

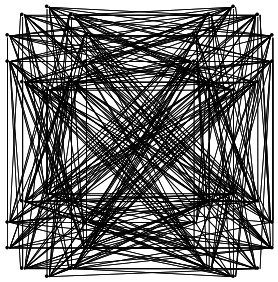


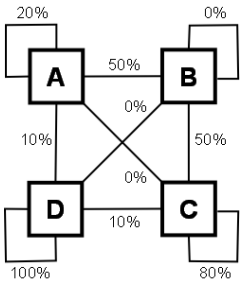
Figure 8: Artificial graph: cocliques.

3.6 Complex pattern

Figure 9 displays a complex pattern, with two dense vertex clusters, D which is a clique and C a quasi-clique, and two sparse clusters, B which is a coclique and A a quasi-coclique. The edge density in coclusters vary from 0% to 50%. Any method requiring a global threshold for the intra edge density or for the ratio between intra and inter edge densities would fail to recognize such patterns, since some clusters are denser and other sparser than on average. The four clusters are correctly identified by our method, with a chi-square value is 476.977, still far beyond the critical value 27.877 at 0.1% rejection rate. This example demonstrates the summarizing capacity of our method, which provides a simple model of the edge density in the graph, whatever be the underlying pattern.

3.7 Relation with Modularity Optimization Methods

The goal of community detection is to partition a network into clusters of vertices with high edge density, with the vertices belonging to different clusters being sparsely connected. To evaluate the quality of a partition, the modularity Q (Newman and Girvan, 2003) is a widely used criterion in recent community detection methods. The modularity measures the density of edges inside clusters as compared to the one expected in case of independence of the vertices.



	A	B	C	D	$\sum$
A	18	60	0	12	90
B	60	0	46	0	106
C	0	46	76	13	135
D	12	0	13	90	115
$\sum$	90	106	135	115	446

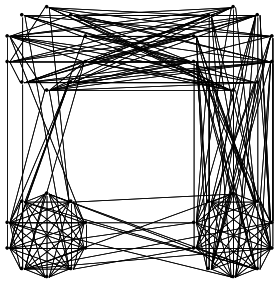


Figure 9: Artificial graph: complex edge density.

Given a graph $G = (V, E)$ with n vertices and m edges, let m_{ij} be an element of the adjacency matrix of the graph. $m_{ij} = 1$ if vertices i and j are connected by an edge, $m_{ij} = 0$ otherwise. The degree of a vertex i is defined by the number of edges incident upon it. In the case of undirected graphs, the ouput and input degree of a vertice are equal. Using the notation of Section 2.3, we have

$$m_{i.} = m_{.i} = \sum_j m_{ij} = \sum_j m_{ji}. \tag{4}$$

An undirected graph with m_U edges corresponds to a symmetrical directed graph with $m = 2m_U$ edges. Assuming that the vertex degrees are respected, the probability of a random edge between vertices i and j is $m_{i.}m_{.j}/m$. The modularity Q is defined as

$$Q = \frac{1}{m} \sum_{ij} \Big(m_{ij} - \frac{m_{i.}m_{.j}}{m} \Big) \delta(k_S(i), k_T(j)), \tag{5}$$

where $k_S(i) = k_T(i)$ is the index of the cluster to which vertex i is assigned, the δ-function $\delta(x, y)$ is 1 if $x = y$ and 0 otherwise and $m = \sum_{ij} m_{ij}$ is twice the number of (undirected) edges. The modularity takes its values between -1 and 1 and has positive values when the clusters have more internal edges that the expected edge number if connections where made at random, with the same vertex degrees. The value of this criterion is 0 in the two extreme cases of one single cluster and of as many clusters as vertices. The modularity criterion has two appealing properties: it is well founded for the discovery of clusters the density of which is higher than the expected density when the extremities of the edges are independent, and it does not require any parameter, such as the number of clusters.

Modularity has been used to evaluate the quality of partitions, but also as an objective function to optimize. In this paper, we compare our approach with two modularity based algorithms. The first one is a greedy bottom-up algorithm that comes from the seminal work of (Clauset et al., 2004) and the second one is the state of the art heuristic of (Blondel et al., 2008), which is very fast and builds high quality partitions (measured by the modularity criterion). For each sample graph introduced in this section, we report in Table 1 the number of clusters and the resulting modularity for these two algorithms and for our approach. The clusters discovered by the best modularity based algorithm (BGLL) and our approach are displayed in Figure 10, using a different color per cluster.

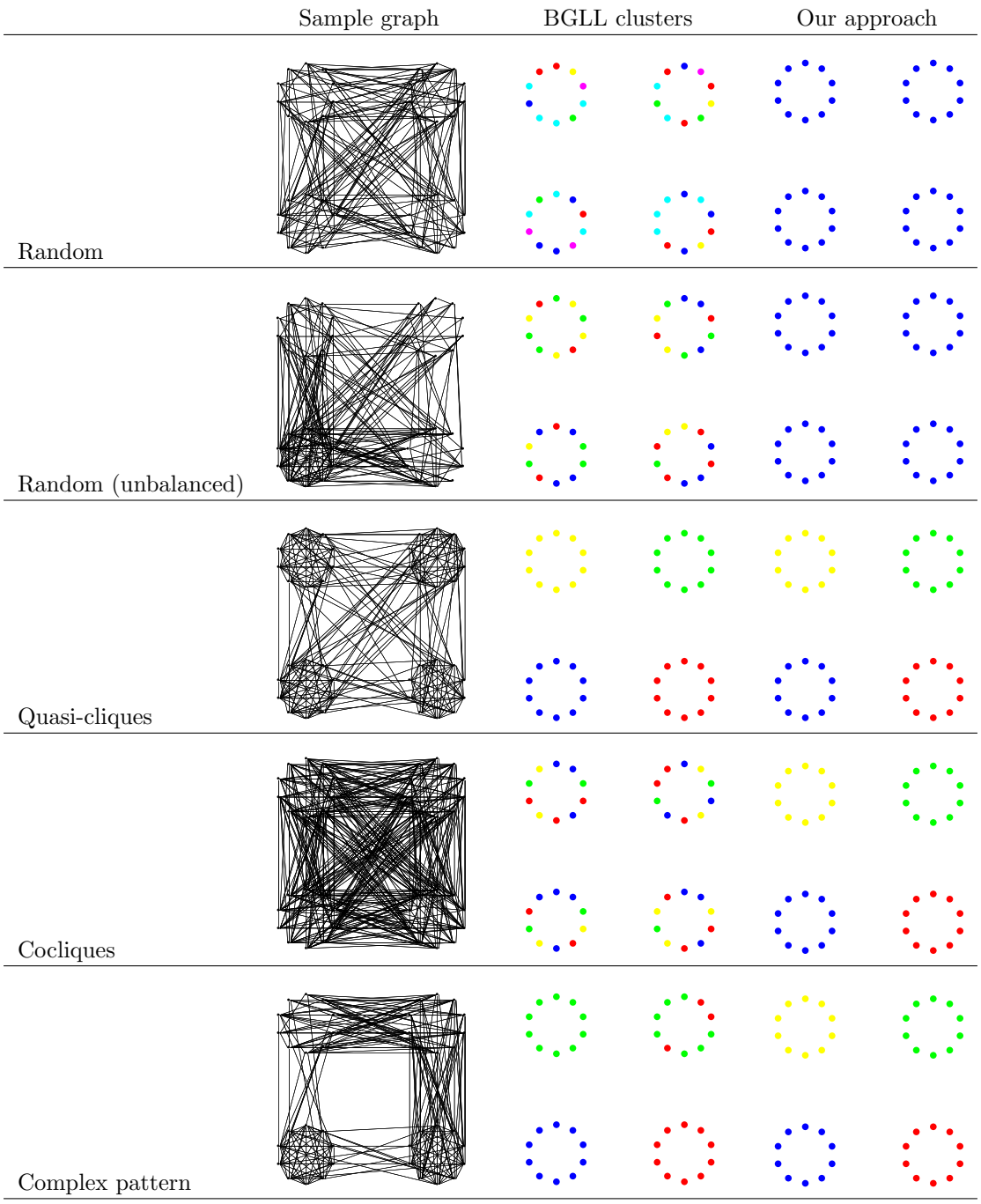


Figure 10: Clusters on sample graphs, for the modularity based algorithm (Blondel et al., 2008) and our approach. All patterns are consistently uncovered using our approach.

	CNM		BGLL		Our approach	
Sample graph	Clusters	Mod.	Clusters	Mod.	Clusters	Mod.
Random	4	0.203	6	0.217	1	0
Random (unbalanced degrees)	5	0.163	4	0.159	1	0
Quasi-cliques	4	0.409	4	0.409	4	0.409
Cocliques	3	0.120	4	0.132	4	-0.251
Complex pattern	3	0.318	3	0.340	4	0.157

Table 1: Number of clusters and modularity on sample graphs, for the algorithms of Clauset, Newman and Moore (Clauset et al., 2004), Blondel, Guillaume, Lambiotte and Lefebvre (Blondel et al., 2008) and our approach. The modularity is clearly not a good criterion for the evaluation of patterns discovered by our approach.

Whereas our method perfectly recognizes the underlying pattern in each case, the modularity based methods fail to do so, except for the quasi-clique sample. For the two random graphs, whether the distribution of the vertex degrees is balanced or not, the theoretical objective of modularity based methods is to build one single cluster with modularity zero. However, due to randomness, clusters with edge density slightly above the mean density may appear, and the modularity based method will build erroneous clusters. This is acknowledged in (Clauset et al., 2004):

> "Non zero values represent deviation from randomness, and in practice it is found that a value above about 0.3 is a good indicator of significant community structure in a network."

The results in Table 1 are compliant with this claim: both the CNM and BGLL algorithms find 4 to 6 irrelevant clusters in the two random graphs, but the related modularity value is about 0.2, below the 0.3 threshold.

	C1	C2	C3	C4	C5	C6	$\sum$
C1	56	14	8	5	13	8	104
C2	14	30	4	5	10	6	69
C3	8	4	10	2	5	3	32
C4	5	5	2	6	4	2	24
C5	13	10	5	4	28	5	65
C6	8	6	3	2	5	10	34
$\sum$	104	69	32	24	65	34	328

Table 2: Illustration of overfitting behavior on the random graph sample: contingency table for the six clusters produced using the BGLL approach.

Table 2 shows the contingency table related to the six erroneous clusters found using the BGLL approach for the random graph sample. Interestingly, the chi-square value 109.693 for this table with $25 = (6-1)(6-1)$ degrees of freedom is far above the 52.620 critical

value for a rejection rate at 0.1% of the hypothesis of independence between the source and target vertices. This value of 109.693 corresponds to a rejection rate of about 10^{-12}. This is a clear illustration of the overfitting phenomenon: although the best partition built using the modularity based approach looks informative, it does not account for the huge number of potential partitions. There are about 1.6 10^{35} different partitions of 40 vertices, and even around 1.8 10^{28} when restricting to partitions into six clusters, so that even a chi-square value with a rejection rate of 10^{-12} of the hypothesis of independence for one given partition does not allow to trust the discovered pattern. Our approach is regularized and not prone to overfitting: it has a strong theoretical foundation to build one single cluster in case of random graph, more precisely in case where the source and target vertices of the edges have independent distributions. This is empirically confirmed on the two random graphs.

The quasi-clique sample matches exactly the kind of patterns searched by community based methods: the algorithms retrieve the pattern with a modularity value of about 0.4, clearly above the 0.3 threshold of (Clauset et al., 2004). In this case, the modularity based algorithms and our approach retrieve the same pattern.

The coclique sample illustrates a case where the intra-edge density is far below the mean density. Actually, not all graphs have a structure consisting of natural clusters. Yet, all clustering algorithm output a partition into clusters for any input graph, and the modularity based algorithm build around 4 dubious clusters (with modularity below the 0.3 threshold). Our approach builds four empty clusters, with zero intra-cluster edge density and large inter-cluster density. In this case, the modularity is negative (-0.251), which reflects the fact that the ratio between observed and expected edge density is far below 1. The true pattern would likely be found by a minimization of the modularity, which is the exact opposite of the intention of modularity based algorithms. Let us give an example of potential real data exhibiting this kind of coclique patterns. In a distributed computing environment with hundred of computers (the vertices) coming from several universities (the clusters), if we collect the internet connexions between the computers (the edges), but are not able to collect the LAN (local area network) connexions within each university, our approach would be able to reconstruct the university pattern in the computing network. Given the collected data, the computers within each university are in cocliques, with potentially intense traffic across universities.

The last sample graph exhibits a complex pattern, with clusters of various densities, lower or higher than the mean density. Our approach, which is non-parametric in essence, is able to recognize any kind of pattern, provided that there is enough available data. Modularity based approaches clearly fail to uncover such patterns, whether the modularity optimization problem is turned into a maximization problem (to discover quasi-cliques) or a minimization problem (to discover quasi-cocliques).

To summarize, our approach retrieves the same patterns as the modularity based algorithms when the graph follows the assumption of being structured into dense clusters. It has the clear advantage of not overfitting the data, being able to build one single cluster in case of random graph with independently distributed source and target vertices for the edges. It is also non-parametric and able to approximate any edge distribution in graphs, without the strong assumption of cluster-based distribution of the edges.

4. Evaluation on Benchmark Graphs

In this section, we apply and analyze the results of our method on a variety of benchmark graphs introduced in the literature. The results are related to those of a modularity-based clustering algorithm, which makes sense insomuch as many real graphs have a cluster-based structure with dense clusters and few edges across clusters.

4.1 Artificial Graphs

The artificial graphs used for tests were introduced by (Johnson et al., 1989), with two kinds of randomly generated graphs. The first type of graph is the classical random graph of (Erdős and Rényi, 1976), with n vertices and every pair of vertices being connected with probability p. The expected average vertex degree is $p(n-1)$ and the expected number of edges is $pn(n-1)$. The random graphs are summarized in Table 3, with vertex numbers 124, 250, 500 and 1000, and parameter p chosen by (Johnson et al., 1989) so as to build sparse graphs with small expected average degrees. On all these graphs, our method builds one single cluster, which confirms its high resilience to noise.

Graph	Vertices	Edges
g124.02	124	149
g124.04	124	318
g124.08	124	620
g124.16	124	1271
g250.01	250	331
g250.02	250	612
g250.04	250	1283
g250.08	250	2421

Graph	Vertices	Edges
g500.005	500	625
g500.01	500	1223
g500.02	500	2355
g500.04	500	5120
g1000.0025	1000	1272
g1000.005	1000	2496
g1000.01	1000	5064
g1000.02	1000	10107

Table 3: Random graphs: our method produces one single cluster in each case.

The second type of random graphs may be closer from real applications, in that they have structure and clustering properties. For two-dimensionnal depiction purpose, they are based on n vertices with two dimensional coordinates independently and randomly generated on the interval $(0, 1)$. An edge is created between two vertices if and only if their Euclidian distance is d or less. The expected average degree is $n\pi d^2$ for points not too close from the boundary. These geometric random graphs are summarized in Table 4, with vertex numbers 500 and 1000 and expected vertex degree ranging from 5 to 40. The results of our method are reported in Table 4 as well, with the number of clusters, of non-empty coclusters (in the upper triangular part of the clustered adjacency matrix of graph), and the proportion of edges that are inside the clusters of vertices. The results show that our method builds dense clusters: most of the coclusters are empty and the clusters of vertices clearly have an edge density far above the expected density if clusters were made at random. For example, for graph U500.40, the methods builds 36 clusters of vertices, which contains 37% of the edges, whereas the fraction of edges falling into 36 random clusters of vertices is $\frac{1}{36} \approx 2.8\%$.

Graph	Vertices	Edges	Clusters	Coclusters	% edge intra
u500.05	500	1282	17	38	97%
u500.10	500	2355	23	62	87%
u500.20	500	4549	31	106	63%
u500.40	500	8793	36	161	37%
u1000.05	1000	2394	24	56	97%
u1000.10	1000	4696	33	98	89%
u1000.20	1000	9339	48	168	66%
u1000.40	1000	18015	54	236	45%

Table 4: Random geometric graphs and their clustering summary using our approach.

Figure 11 shows the clusters discovered by our method for the geometric graphs U500.05, U500.10 and U500.40. Remarkably, our method is able to reconstruct the geometric structure of each graph, with an increased precision as the number of edges grows.

As in Section 3.7, we compare our method with two modularity based algorithms and report the results in Table 5. Whereas our method always correctly builds one single cluster for classical random graphs, the modularity based algorithms falsely find cluster-based structures in these graphs, confirming the analysis of (Guimerà et al., 2004). In the case of random graphs, the modularity criterion ranges from 0.2 to 0.7, surprisingly far above the 0.3 threshold of (Clauset et al., 2004). Therefore, there is no simple way of deciding whether the structures found by the modularity based algorithms are spurious or not. In their paper, (Guimerà et al., 2004) suggest to always interpret the modularity in comparison with that obtained on randomized graphs.

For the highly structured geometric graphs, our method builds informative clusters with an increased precision as the number of edges grows. This results from our non-parametric statistical approach, which benefits from more data to select models with more parameters. On the opposite, modularity based methods build clusterings with fewer and fewer clusters as the number of edges grows. This behavior can be observed both for the classical random graphs and the geometric graphs. Although the modularity criterion allows to avoid any parameter for the choice of the cluster number, it is clearly sensitive to the sparsity of the graph, with a number of clusters increasing with the sparsity. Overall, the modularity criterion have the bad property of discovering clusters in random graphs and building too few clusters in some highly structured graphs.

4.2 Real Graphs

Table 6 presents a summary of 12 real graphs[2], all of them treated as unweighted undirected graphs. The three first graphs come from the Harwell-Boeing sparse matrix collection (Duff et al., 1989), the following five graphs are finite element meshes problems (Walshaw, 2000) and the last four are scientific co-authorship networks (Newman, 2001). The results of our method are reported in Table 6, with up to 570 clusters of vertices for the largest graph containing around 150.000 vertices and one million edges.

2. Isolated vertices have been discarded from the datasets.

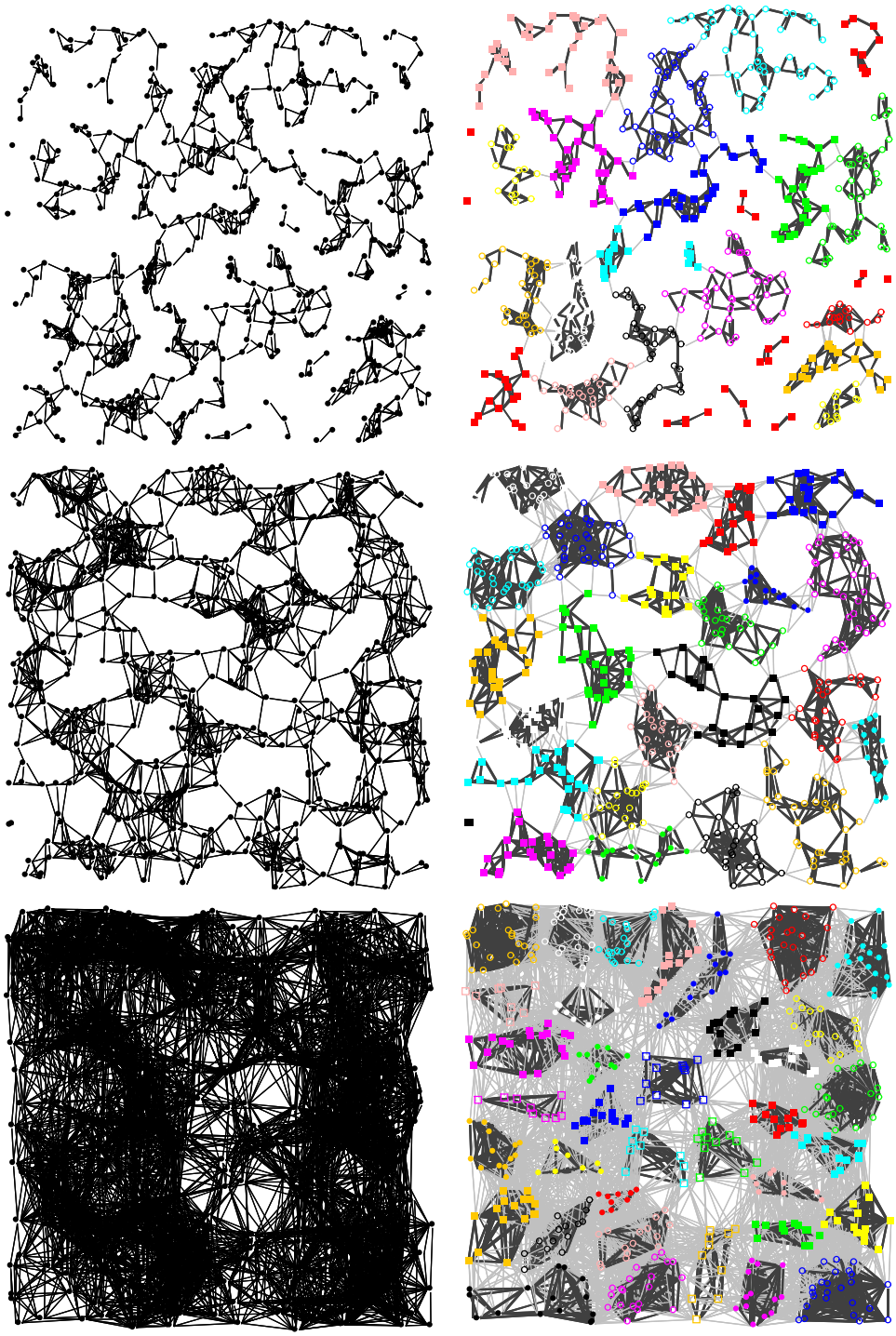


Figure 11: Clustering of geometric graphs: U500.05, U500.10 and U500.40.

			CNM		BGLL		Our approach	
Graph	Vertices	Edges	Clust.	Mod.	Clust.	Mod.	Clust.	Mod.
g124.02	112	149	10	0.622	9	0.644	1	0
g124.04	124	318	6	0.421	9	0.416	1	0
g124.08	124	620	7	0.255	9	0.259	1	0
g124.16	124	1271	4	0.159	7	0.163	1	0
g250.01	230	331	13	0.652	15	0.643	1	0
g250.02	248	612	11	0.434	11	0.443	1	0
g250.04	250	1283	6	0.274	10	0.284	1	0
g250.08	250	2421	5	0.182	9	0.182	1	0
g500.005	451	625	24	0.684	23	0.676	1	0
g500.01	493	1223	11	0.435	15	0.446	1	0
g500.02	500	2355	8	0.286	12	0.289	1	0
g500.04	500	5120	4	0.187	12	0.194	1	0
g1000.0025	933	1272	35	0.703	34	0.708	1	0
g1000.005	994	2496	15	0.445	19	0.446	1	0
g1000.01	1000	5064	7	0.274	15	0.279	1	0
g1000.02	1000	10107	5	0.184	9	0.202	1	0
u500.05	497	1282	29	0.909	29	0.912	17	0.905
u500.10	500	2355	8	0.770	15	0.84	23	0.818
u500.20	500	4549	4	0.631	11	0.754	31	0.596
u500.40	500	8793	4	0.592	6	0.682	36	0.340
u1000.05	993	2394	46	0.930	46	0.932	24	0.925
u1000.10	999	4696	14	0.778	24	0.877	33	0.854
u1000.20	1000	9339	5	0.641	12	0.794	48	0.639
u1000.40	1000	18015	4	0.579	9	0.740	54	0.426

Table 5: Number of clusters and modularity on two kinds of random graphs, classical random graphs (g*) and geometric random graphs (u*), for the two modularity based algorithms CNM and BGLL and our approach. The modularity based algorithms are prone to overfitting and not able to discover fine grained clusters in highly structured graphs.

As in the preceding section, we compare our method with two modularity based algorithms and report the results in Table 7. The BGLL algorithm gets significantly better modularity results than the CNM algorithm, which complies the results of (Blondel et al., 2008). In the following, we focus the analysis on comparing the results of the BGLL algorithm and our method.

Surprisingly, our method builds far more clusters than the modularity based algorithms in some graphs and far less clusters on other graphs. We now proceed with a more detailed analysis on three representative graphs: bcspwr09, netscience and wave.

On the bcspwr09 graph, the BGLL algorithm produces 26 clusters whereas our approach builds one single cluster. As it is hard to prove the non-existence of any partition that would get a better score, the result of our approach does not prove that there is no clustering information in the graph. A known behavior of our approach is that when there is not

Graph	Vertices	Edges	Clusters	Coclusters	% edge intra
Bcspwr09	1723	2394	1	1	100%
Bcsstk13	2003	40940	95	662	29%
Bcsstk15	3948	56934	145	669	52%
Airfoil1	4253	12229	45	136	90%
Nasa4704	4704	50027	134	628	63%
4elt	15606	45878	91	305	92%
Brack2	62631	366559	320	1728	83%
Wave	156317	1059333	570	4282	80%
Netscience	1461	2742	22	58	95%
Hep-th	7610	15751	48	727	80%
Cond-mat	16264	47594	127	2515	79%
Astro-ph	16046	121251	233	6692	59%

Table 6: Real graphs and their clustering summary using our approach.

			CNM		BGLL		Our approach	
Graph	Vertices	Edges	Clust.	Mod.	Clust.	Mod.	Clust.	Mod.
bcspwr09	1723	2394	26	0.897	26	0.900	1	0
bcsstk13	2003	40940	3	0.473	6	0.593	95	0.279
bcsstk15	3942	56934	3	0.561	7	0.719	145	0.508
airfoil1	4253	12289	7	0.81	19	0.859	45	0.881
nasa4704	4704	50026	8	0.677	14	0.772	134	0.622
4elt	15606	45878	7	0.814	34	0.923	91	0.911
brack2	62631	366559	5	0.692	31	0.905	320	0.824
wave	156317	1059331	8	0.655	31	0.876	570	0.794
netscience	1461	2742	276	0.956	278	0.960	22	0.898
hep-th	7610	15751	666	0.802	631	0.849	48	0.777
cond-mat	16264	47594	901	0.791	796	0.844	127	0.785
astro-ph	16046	121251	522	0.633	420	0.727	233	0.586

Table 7: Number of clusters and modularity on real graphs, for the two modularity based algorithms CNM and BGLL and our approach.

enough data (here: only 2394 edges for 1723 vertices), our approach is reluctant to conclude that there is a significant clustering structure in the graph. We also generated a random version of the bcspwr09 graph, with the same numbers of vertices and edges. Our approach still builds one single cluster, as expected, whereas the BGLL algorithm falsely builds 42 clusters (with modularity 0.674), more clusters than in the original graph. As the random version of the graph is not constrained by the vertex degrees of the original graph, there is still room for variability in the results, and it is hard to conclude whether the graph contains or not some cluster-based structure.

On the netscience graph, the BGLL algorithm builds 278 clusters versus 22 for our approach. We report in Figure 12 the observed versus expected edge number for each

cocluster. Our approach builds a small number of clusters, containing between 75 and 200 edges with reasonably balanced edge densities. This small number of clusters makes sense given the small number of edges in the graph. On the opposite, the BGLL algorithm builds many clusters of vertices containing few edges: less than 50 clusters contain at least 50 edges, and more than 100 clusters contain one single edge. While this might make sense in the context of community detection, most tiny connected components are grouped in one single cluster using our approach, since they are similar with respect to their sparse connectivity pattern. On a random version of the data set, the BGLL algorithm builds 25 clusters (with modularity 0.546), which confirms its tendency to build too many clusters.

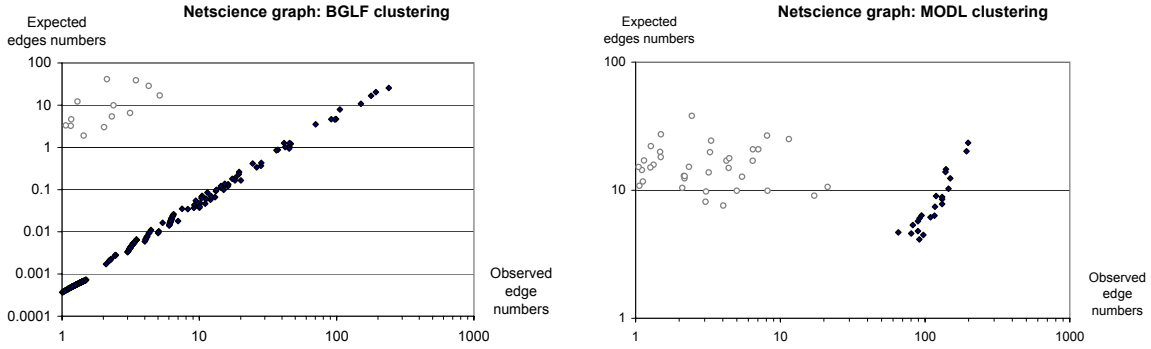


Figure 12: Clustering of netscience graph, with observed versus expected edge number for each cocluster, intra (black) and inter (gray).

On the wave graph, the BGLL algorithm builds 31 clusters versus 570 for our approach. Figure 13 displays the observed versus expected edge number for each cocluster. Our approach builds a large number of clusters, with balanced sizes (from 850 to 2250 edges per cluster) and high edge density (on average 450 times the expected density). Given the size of the graph, the BGLL algorithm builds very few clusters, with unbalanced sizes (from 8500 to 80000 edges per cluster) and low edge density (on average, 30 times the expected density).

Overall, the experiments on real graphs show that our approach is able to reliably uncover interesting structures on a variety of graphs. On the experiments, our method tends to build clusters with reasonably balanced edge frequency and high edge density, with an increasing number of clusters as the number of graph edges grows.

5. Consistency of the MODL Approach for Edge Density Estimation

In this section we present some of the research literature related to statistical graph models and point out why our edge density estimation method is interesting. We show how our approach behaves as a universal approximator and establish the consistency of the MODL model selection approach, which is data dependent and aims at modeling the finite data sample directly, but asymptotically converges towards the true edge density. We finally

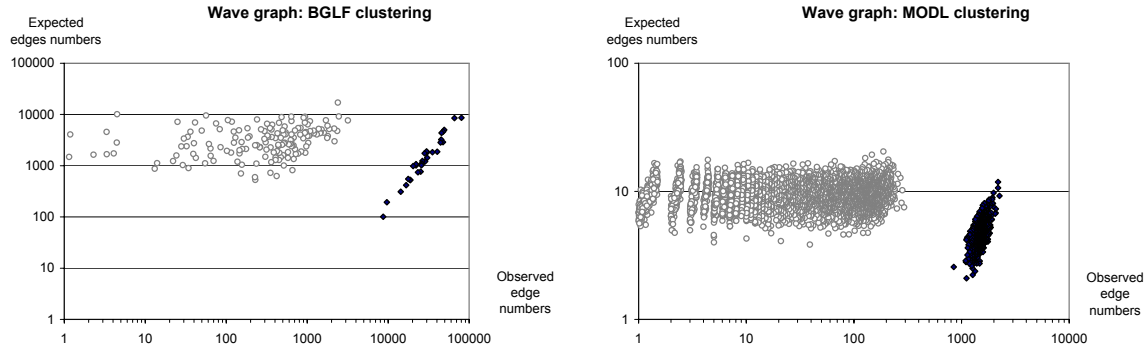


Figure 13: Clustering of wave graph, with observed versus expected edge number for each cocluster, intra (black) and inter (gray).

relate the MODL criterion to the modularity and chi-square criterions and show surprising similarities as well as significant differences.

5.1 Random Graphs, Stochastic Blockmodels and Edge Density Estimation

Several standard stochastic graph models have been studied in the literature (Chung and Lu, 2006). The classical random graph of (Erdős and Rényi, 1976) generates graphs with n vertices in which each of the $n(n-1)/2$ possible edges occurs with probability p. A variant of this model considers that all the graphs with n vertices and m edges are equiprobabble: this variant differs only on whether the number of edges is a hard or soft constraint. This model can be further constrained by a vector of vertex degrees, either by generating graphs with the expected degrees or with a hard constraint on the vertex degrees (Roberts, 2000; Blitzstein and Diaconis, 2006). These uniform graph models have been specialized to produce clusters. In the *planted l-partition model* (Condon and Karp, 2001), a graph is generated with l clusters of k vertices, with intra-cluster edges having a high probability p and inter-cluster edges having a low probability q.

More expressive graph models aim at searching a partition of the vertices into groups or blocks, with different types of interaction between blocks. In the applied mathematics field, the seminal work of (Hartigan, 1972) treats the similar problem of coclustering of a matrix, by looking at a partition of the rows and columns of the matrix. In the data mining field, in case of binary variables, this technique has been applied to the simultaneous partitioning of the instances into clusters and the variables into groups of variables (Bock, 1979; Dhillon et al., 2003; Govaert and Nadif, 2003), based on stochastic coclustering models and expectation maximisation (EM) algorithm. In the the sociometric literature, this modeling approach is called blockmodeling and has been thoroughly studied. Lorrain and White (1971) introduced the notion of structural equivalence, where the vertices are connected to the rest of the graph in similar ways. Early approaches (White et al., 1976; Arabie et al., 1978; White and Reitz, 1983; Doreian et al., 2004) consider non stochastic blocks, with a focus on predefined types of block patterns. The block model is searched either indirectly

using a (dis)similarity measure between pairs of vertices and then applying a standard clustering algorithm, or directly by optimizing an ad hoc function measuring the fit of real blocks to the corresponding predefined types of blocks. Using the framework of exponential family (Holland and S.Leinhardt, 1981), Holland et al. (1983) introduced stochastic block models, with blocks still specified a priori. In (Wasserman and Anderson, 1987; Wang and Wong, 1987), the approach is extended to the discovery of block structure and exploits a statistical criterion, e.g likelihood function, optimized using the EM algorithm. The method of (Snijders and Nowicki, 1997) considers block models where the edge probabilities depend only on the blocks to which the vertices belong. The considered models are limited to two blocks, and searched via maximum likelihood estimation using the EM algorithm for small graphs and via Bayesian Gibbs sampling for large graphs. In (Nowicki and Snijders, 2001; Gill and Swartz, 2004), the blockmodels are broaden to an arbitrary number of blocks, and optimized via Monte Carlo Markov Chain (MCMC) Bayesian inference. Recent work on stochastical blockmodeling via maximum likelihood methods include (Wasserman et al., 2007; Copic et al., 2009; Bickel and Chen, 2009), with a survey in (Goldenberg et al., 2010). Still, finding the optimal number K of clusters and optimizing the likelihood in the case of large K remain open problems. Airoldi et al. (2008) introduce a mixed membership stochastic blockmodel, where each vertex belongs to several blocks according to a mixture model. The likelihood cannot be evaluated analytically and the inference is approximated owing to variational methods. Following (Chakrabarti et al., 2004; Rosvall and Bergstrom, 2007), Lang (2009) considers undirected simple graphs and employ another model selection approach for the inference the whole set of blockmodel parameters, including the number of blocks. Using the MDL (Minimum Description Length) approach (Rissanen, 1978), several encoding schemes are explored. A fast multi-level algorithm is exploited to generate candidate partitions of the vertices for $K \in \{2, 4, 8, \ldots, 1024\}$, with a focus on the $K = 1$ versus $K > 1$ question. Lang (2009) shows that accounting for the vertex degrees brings a better resilience to randomness.

In our method, we consider the problem of edge density estimation in directed multiple graphs, with potential loops and multi-edges. Our approach differs from previous ones in several points:

1. we aim at modeling the edge density, that is all the edge probabilities for any pair of vertices; the inferred block structure is a by-product of the modeling approach(see Section 5.2),

2. we propose a fully non-parametric approach, where the whole set of block models is considered, from one single cluster of vertices to as many clusters as vertices, and the optimal number of clusters is found automatically (see Section 2.3),

3. we exploit a new model selection approach, where the finite data sample is modeled directly, with a data dependent prior distribution of the model parameters; we demonstrate new fundamental results that prove the consistency of the approach (see Section 5.3),

4. we exploit combinatorial algorithms with practical time complexity, that enable the processing of large real graphs (see Section 2.5).

Given these features, our method is able to reliably summarize large graphs without any assumption on their structure, and provides an approximation of the underlying edge density which asymptotically converges toward the true edge density.

5.2 Generative Models for Edge Density Estimation

In our method, we consider the graphs as generative models, where the statistical units are the edges with two variables per edge, source and target vertices of the edge. Whereas most blockmodeling approaches deal with simple graphs, focusing on their topology with at most one edge per pair of vertices, we regard graphs as statistical distributions of directed edges, with potential loops and multi-edges. A graph generative model for a set of n vertices V is entirely defined by a set of probability parameters $\{p_{ij}\}_{1\leq i\leq n, 1\leq j\leq n}$, where p_{ij} stands for the probability of each independent and identically distributed (i.i.d) edge of having source vertex i and target vertex j. Given these settings, a graph $G = (V, E)$ containing m edges is treated as a sample of size m drawn from the edge distribution. Therefore, large samples tend be produce complete graphs from a pure topological point of view, but with varying edge densities taking into account the generative model.

This generative model applies to many real graph data. In web log analysis, it seems natural to consider a bipartite graph, with users as source vertices, web pages as target vertices and edges representing web navigation. A sample graph corresponds to an extract of web log data, with the popular pages much more seen than the others. In a phone call network, each edge represents one phone call from a caller vertex to a called vertex, so that two vertices can be connected by multiple edges. Collecting the phone calls during a given time period corresponds to a sample of a directed multigraph, where the potential communities correspond to subgraphs with high multi-edge density. The case of undirected graphs can be treated with symmetrical edge probabilities and a pair of directed edges per undirected edge.

Given the random graph generative model, the problem is to estimate the edges densities in the graph from a finite data sample. Estimating the n^2 edge probability parameters p_{ij} from a sample of size m is not an easy task, especially in the case of sparse graphs.

In the following, we propose a new interpretation of our approach described in Section 2 and show how it reduces to a finite sample modeling, which asymptotically converges to an estimation of the edge density parameters.

Let us introduce a family of cluster-based random graphs, defined by the following parameters:

- k_S, k_T: number of clusters of source and target vertices
- $k_S(i), k_T(j)$: index of the cluster containing source vertex i (resp. target vertex j)
- $\{p_{ij}^{ST}\}_{1\leq i\leq k_S, 1\leq j\leq k_T}$: probability distribution of the edges falling in each cocluster (i, j)
- $\{p_{i,\mu.}\}_{k_S(\mu)=i}$: probability distribution of the out-degrees of the vertices μ of the source cluster i
- $\{p_{j,.\gamma}\}_{k_T(\gamma)=j}$: probability distribution of the in-degrees of the vertices γ of the target cluster j

For a sample graph of size m with edges counts m_{ij}, let us reuse the notations of Section 2.3, which correspond to different levels of aggregation of edge counts in the MODL approach.

$m_{i.}$	$= \sum_j m_{ij}$	out-degree of vertex i
$m_{.j}$	$= \sum_i m_{ij}$	in-degree of vertex j
m_{ij}^{ST}	$= \sum_{\mu,\gamma/k_S(\mu)=i,k_T(\gamma)=j} m_{\mu\gamma}$	edge count in cocluster (i,j)
$m_{i.}^{S}$	$= \sum_{\mu,\gamma/k_S(\mu)=i} m_{\mu\gamma}$	out-degree of cluster i
$m_{.j}^{T}$	$= \sum_{\mu,\gamma/k_T(\gamma)=j} m_{\mu\gamma}$	in-degree of cluster j

Using these notations, the probability parameters of the cluster-based random graphs can be empirically estimated according to:

$$p_{ij}^{ST} = \frac{m_{ij}^{ST}}{m}, \qquad p_{i,\mu.} = \frac{m_{\mu.}}{m_{i.}^{S}}, \qquad p_{j,.\gamma} = \frac{m_{.\gamma}}{m_{.j}^{T}}, \tag{6}$$

which shows that the MODL approach relates to an empirical estimation of the probabilities introduced in the family of cluster-based random graphs.

This is a piece-wise constant modeling of the edge density with respect to the coclusters, constrained by the distributions of the in and out-degrees of the vertices in each cluster. Assuming the independence between the source and target vertices of the edges inside each cocluster, we get the following estimation of the edge densities:

$$p_{\mu\gamma} = p_{ij}^{ST} p_{i,\mu.} p_{j,.\gamma} = \frac{m_{ij}^{ST}}{m} \frac{m_{\mu.}}{m_{i.}^{S}} \frac{m_{.\gamma}}{m_{.j}^{T}}, \tag{7}$$

where (i,j) is the cocluster containg the edges (μ,γ).

For the null model $M_\emptyset$ with one single cluster ($i=j=1$), we have

$$p_{11}^{ST} = 1, \qquad p_{1,\mu.} = \frac{m_{\mu.}}{m}, \qquad p_{1,.\gamma} = \frac{m_{.\gamma}}{m}, \qquad p_{\mu\gamma} = \frac{m_{\mu.}}{m} \frac{m_{.\gamma}}{m}, \tag{8}$$

which means that the joint probability distribution p_{ij} is the product of the two independent marginal distributions of the in and out-degrees of the vertices.

For the maximal model M_{Max} with one cluster per vertex, we have

$$p_{ij}^{ST} = \frac{m_{ij}}{m}, \qquad p_{i,i.} = 1, \qquad p_{j,.j} = 1, \qquad p_{\mu\gamma} = \frac{m_{\mu\gamma}}{m}, \tag{9}$$

which means that the joint probability distribution p_{ij} of the edges is directly estimated by the model parameters.

5.3 Asymptotic Convergence of the MODL Approach

The family of cluster-based random graphs is very expressive and can theoretically approximate any edge distribution provided that there is sufficient data. The problem is to select the best model given the data. Whereas classical Bayesian approaches rely on the delicate problem of choosing a prior for the model parameters and computing the likelihood of the

data given the model, and MDL approaches on the similar problem of choosing a compression scheme for model parameters and the data given the model, the MODL approach avoids these problems by modeling directly the finite data sample. Instead of modeling the real-valued edge probabilities, the data grid models consider all the potential discrete distributions of the m edges on the n_S, n_T source and target vertices. Working on a set of parameters of finite size allows to define "natural" hierarchical priors with uniform distribution at each level of the hierarchy, as in Definition 2. This data dependent modeling technique provides the criterion of Formula 1, which can be interpreted as the exact posterior probability of the sample graph given the model (Bayesian interpretation), or the exact coding length of the model parameters and edges given the model (MDL interpretation). Therefore, contrary to classical model selection approaches, the criterion does not rely on empirical estimation of continuous-valued parameters (such as probabilities or entropies), which are valid only asymptotically.

We now study whether for a given vertex number, this exact finite data sample modeling asymptotically converges towards the true edge density as the edge number goes to infinity.

Let us first recall some concepts from information theory. The Shannon entropy $H(X)$ (Shannon, 1948) of a discrete random variable X with probability distribution function p is defined as:

$$H(X) = -\sum_{x \in X} p(x) \log p(x). \tag{10}$$

The mutual information of two random variables is a quantity which measures the mutual dependence of the two variables (Cover and Thomas, 1991); it vanishes if and only if they are independent. For two discrete variables X and Y, the mutual information is defined as:

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}, \tag{11}$$

where $p(x,y)$ is the joint probability distribution function of X and Y, and $p(x)$ and $p(y)$ are the marginal probability distribution functions of X and Y respectively.

Let us consider edges as statistical instances, with two vertex variables V_S and V_T having n_S and n_T values, and two vertex cluster variables V_S^M and V_T^M having k_S and k_T for a given vertex coclustering model M.

We present in Theorem 4 a generalization of the asymptotic evaluation of the null model $M_\emptyset$ presented in Formula 3. As the criterion has to be minimized, this means that the method aims at selecting a coclustering model which maximizes the mutual information between the two vertex cluster variables. Since the mutual information of two variables is not other than the Kullback-Leibler divergence (Kullback, 1959) between the joint probability distribution of two variables and their independent joint distribution, this means that the best selected coclustering tends to highlight contrasts between the two variables, being as far as possible from their independent joint distribution.

Theorem 4 *The MODL evaluation criterion (Formula 1) for an edge density estimation model M is asymptotically equal to m times the entropy of the source and target vertex variables minus their mutual entropy.*

$$c(M) = m \left(H(V_S) + H(V_T) - I(V_S^M; V_T^M) \right) + O(\log m). \tag{12}$$

Proof See Appendix A. ■

We now present an important result in Theorem 5, which shows that the MODL approach asymptotically converges towards the estimation of the true edge distribution, that is the joint distribution of the source and target vertex variables. Although the modeling technique is data dependent (regarding the model space and the prior on the model parameters) and aims at modeling exactly the data sample with a discrete distribution of the sample edges on the vertices, not the true edge continuous valued-probability distribution, this theorem demonstrates the consistency of the approach, as it asymptotically estimates the true edge distribution.

Theorem 5 *The MODL approach for selecting an edge density estimation model M asymptotically converges towards the true edge distribution, and the criterion for the best model M_{Best} converges to m times the entropy of the edge variable, that is the joint entropy of the source and target vertices variables.*

$$\lim_{m \to \infty} \frac{c(M_{Best})}{m} = H(V_S, V_T). \tag{13}$$

Proof See Appendix A. ■

As a corollary of Theorem 5, Theorem 6 states that the MODL approach allows to estimate the mutual information between the source and target vertices variables.

Theorem 6 *The MODL approach for selecting an edge density estimation model M asymptotically converges towards the true edge distribution, and the criterion for the null model minus the best model M_{Best} converges to m times the mutual entropy of the source and target vertices variables.*

$$\lim_{m \to \infty} \frac{c(M_\emptyset) - c(M_{Best})}{m} = I(V_S; V_T). \tag{14}$$

5.4 Experimental Convergence Rate of the MODL Approach

We have shown that although the MODL approach aims at modeling the data sample directly, it asymptotically converges towards the true edge density. The assumption behind the non-asymptotic MODL approach is that the non-parametric edge density estimation will benefit from fine tuned finite data dependent model space and prior, so as to converge as fast and reliably as possible.

This convergence rate is hard to analyze theoretically in the non-parametric setting, without any assumption regarding the true edge density. For example, in the simple case of a cluster-based graphs, the adjacency matrix is block-diagonal and most of the edge probabilities are null. In this case, few parameters need to be estimated and the convergence is fast. In this section, we chose a more difficult sample graph where the distribution of the edge probabilities is rather smooth and never null (except for loops), and present an experimental study of the convergence rate of the approach.

Let us introduce *circular random graphs* as undirected graph, where the n vertices lie equidistant on the unit circle at positions $(x_i = \cos\frac{2\pi i}{n}, y_i = \sin\frac{2\pi i}{n})$. The euclidian distance between two vertices i and j being $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$, we define the probability of having an edge between two disjoint vertices in inverse proportion of their distance, according to:

$$\forall i,\ p_{ii} = 0, \qquad \forall i \neq j,\ p_{ij} = \frac{\frac{1}{d_{ij}}}{\sum_{\mu\neq\gamma}\frac{1}{d_{\mu\gamma}}}. \tag{15}$$

In such a circular random graph, the closest pairs of vertices (two consecutive points on the circle) are distant from about $\frac{2\pi}{n}$, whereas the farthest pairs (on the diameter of the circle) are distant from 2. Therefore, close vertices lead to edges that are about $\frac{n}{\pi}$ times more probable than for distant vertices, with a continuous decrease of edge probabilities in inverse proportion to their distance.

In our experiment, we chose a circular random graph with $n = 100$ vertices and randomly generate edges (by pairs, twice the number of undirected edges) from sample size varying from 100 to 10^7. For each sample size, we run the MODL algorithm and collect both the number of clusters and the difference between the estimated mutual information (see Theorem 6) and the true mutual information (known exactly for this artificial dataset). The results are presented in Figure 14, which shows tree phases in the convergence of the MODL algorithm. In the first phase (*stability phase*), the number of edges is not sufficient to reliably estimate the edge probabilities, and the approach evaluates the random graph with one single cluster as being the most probable. In the second phase ($non-parametric\ estimation\ phase$), the method reliably identifies structures in the graph by building clusters and approximating the true mutual information, with an increased precision as the sample size grows. In the third phase (*classical estimation phase*), the method has built one cluster per vertex and estimates all the n^2 edges probabilities simultaneously according to a classical empirical estimation of a set of multinomial parameters: the precision of the estimation "classically" increases with the sample size. In this sample, the non-parametric estimation phase starts when the number of available edges is about 1000, about $\frac{1}{10}$ of the number of edge probability parameters, and has converged at about 50000 edges, five times the number of edges probability parameters. It is noteworthy that in most large sparse real graphs such as those of Section 4, the method is in the non-parametric estimation phase (in the wave graph, the number of observed edges is about $\frac{1}{10000}$ of the number of potential edge probability parameters).

This shows that the method is reliable, quickly discovers true structures in the graph as soon as there is sufficient data, and is able to approximate any edge density distribution with a fast convergence rate.

5.5 Comparative Analysis of the MODL Criterion

We now compare three criterions, modularity, chi-square and criterion, by formulating them in terms of observed and expected edge counts per cocluster.

Using the notation of Section 2.3, m_{ij} is the number of edges between vertices i and j and m_{ij}^{ST} is the number of edges between clusters i and j. Let $e_{ij} = m_{i.}m_{.j}/m$ be the number of expected edges between vertices i and j in case of random edges constrained by

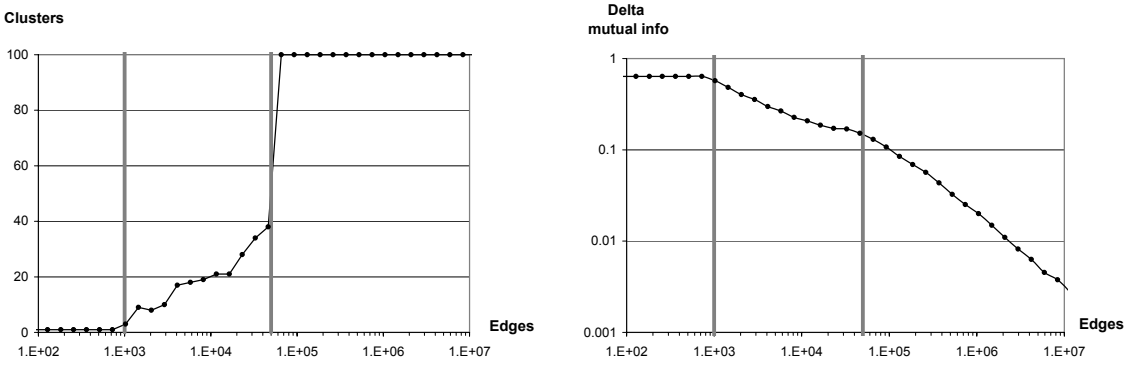


Figure 14: Convergence of the MODL approach on the random circular graph with 100 vertices: number of clusters and difference between the estimated and true mutual information $I(V_S;V_T)$ per edge number in the graph sample.

the vertex degrees, and $e_{ij}^{ST} = m_{i.}^S m_{.j}^T/m$ the number of expected edges between clusters i and j. The case of undirected graphs with m_U edges is treated as symmetrical directed graphs with $m = 2m_U$ edges.

Modularity. Using these notations, the modularity criterion (5) can be rewritten as a weighted sum over of the clusters of a difference between observed and expected edge numbers:

$$Q = \frac{1}{m} \sum_{ij} e_{ij}^{ST} \left(\frac{m_{ij}^{ST}}{e_{ij}^{ST}} - 1 \right) \delta(i,j). \tag{16}$$

Chi-square. The Pearson's chi-square test is a widespread statistical test for deciding whether two variables (in our case, the clusters) are independent. Using the same terms as above, the chi-square value can be written as:

$$\chi^2 = \sum_{ij} e_{ij}^{ST} \left(\frac{m_{ij}^{ST}}{e_{ij}^{ST}} - 1 \right)^2. \tag{17}$$

MODL. Using Formula (23) from the proof of Theorem 4, we have:

$$\begin{aligned} c(M) = & - m \sum_{i=1}^{k_S} \sum_{j=1}^{k_T} \frac{m_{ij}^{ST}}{m} \log \frac{\frac{m_{ij}^{ST}}{m}}{\frac{m_{i.}^S}{m} \frac{m_{.j}^T}{m}} \\ & - m \sum_{i=1}^{n_S} \frac{m_{i.}}{m} \log \frac{m_{i.}}{m} \quad - \quad m \sum_{j=1}^{n_T} \frac{m_{.j}}{m} \log \frac{m_{.j}}{m} \quad + \quad O(\log m). \end{aligned} \tag{18}$$

Since the terms of the second line are constant given a sample graph, the minimization of the MODL criterion reduces to the maximization of the following criterion:

$$c'(M) = \sum_{ij} m_{ij}^{ST} \log \frac{m_{ij}^{ST}}{e_{ij}^{ST}} + O(\log m). \tag{19}$$

Comparison of the Criterions. Interestingly, the three criterions, modularity, chi-square and MODL can be written using the same observed and expected edges counts per cocluster. The modularity criterion focusses on the diagonal of the clustered graph adjacency matrix only, ignoring the inter-cluster distribution of the edges. Maximizing the modularity comes down to identifying clusters where the observed edge count is as much as possible above the expected count. The chi-square and MODL criterions account for the whole edge distribution, looking for a clustered adjacency matrix as contrasted as possible compared to the "grayed" independent-based related adjacency matrix. The chi-square criterion comes from statistical test theory, is valid asymptotically with conditions related to the minimum expected edge counts per cell of the clustered adjacency matrix (Cochran, 1954). The MODL criterion comes from information theory, which is usually valid asymptotically. In the MODL approach, the criterion relates to an exact encoding of the finite graph sample and benefits from a non-asymptotic validity. Finally, whereas both the modularity and chi-square criterions ignore the model selection problem and are consequently prone to overfitting (see Section 3), the MODL criterion accounts for the complexity of the cluster-based model of the graph and is theoretically and experimentally resilient to overfitting.

6. Future Research Direction

We propose several extensions of our approach, regarding the classes of graphs and the optimization algorithms.

6.1 Classes of Graphs Addressed by our Approach

Although all the experiments in Section 4 deal with undirected graphs with at most one edge per pair of vertices, our approach is meant to handle directed multigraphs. We show how it can be specialized or extended to different classes of graphs.

Multigraph. A multigraph can have loops and multiple edges between each pair of vertices. Our approach treats the edges as statistical units, which naturally encompasses the case of multiple edges.

Weighted graph. A weighted graph has values assigned to each edge, such as for example cost, length, capacity, edge creation or deletion date. Some approaches, such as modularity based approached, treat weighted graphs naturally by replacing the unit values in the adjacency matrix by the edge weights, and the vertex degrees by the sum of weights of the edges adjacent to each vertex. Our approach cannot deal with weights this way, since edges, considered as statistical units, only have integer counts. These edge counts are not weights: for example, doubling the count of each edge has no impact in modularity based

approaches, whereas in our approach, this reenforces the statistics of the edges and allows a better edge density approximation with potentially more vertex clusters.

Our approach can be extended to weighted graphs by representing them in a tabular format with one edge per line and three variables: source vertex, target vertex and edge weight. The related 3-dimensional data grid will produce clusters of source and target vertices, and intervals of weights, so as to estimate the joint density of the three variables.

Directed graph. In directed graphs, the pair of vertices describing each edge is ordered. Our approach is meant for directed graph, by building clusters of source vertices and clusters of target vertices, which may differ. For example, it has been exploited to analyze the web graph, where the vertices are web pages and the edges are links between web pages.

Bipartite graph. In bipartite graphs, the vertices can be divided into two sets, so that every edge has one vertex in each of the two sets. This can be seen a a special case of directed graphs, where each vertex is either source or target. We have applied our approach to bipartite graphs, for the coclustering of texts versus words in text mining, of customers versus products in marketing or of cookies versus pages in web log analysis.

Undirected graph. In undirected graphs, edges have no orientation. Throughout this paper, we have applied our approach to undirected graph by analyzing the related symmetrical directed graph with twice the number of edges. Our approach could be specialized, by considering one single clustering of the vertices, distributing the edges on half of the contingency matrix and exploiting specialized clustering algorithms, potentially more efficient than coclustering algorithms.

Graph with edge data. Weighted graphs can be extended to graphs with several variables per edge. For example, in a telecommunication network, call detail records (edges) between the calling and called parties (vertices) can be characterized by the date and time of the call, its length, cost, type (voice, SMS...). As for weighted graphs, our data grid approach can be applied to edges with multiple variables, using a table of edges with these variables in addition to the source and target vertex variables. Each categorical variable is clustered into groups of values and each numerical variable is partitioned into intervals. The cross-product of these univariate partitions forms a multivariate partition consisting of cells, which behave as a estimator of the joint-probability distribution of all the variables. Experiments are needed to evaluate the practical interest and the limits of the approach.

Hypergraph. A hypergraph is a generalization of a graph, where an edge (called hyperedge) can connect any number of vertices. A hypergraph can be described by its incidence matrix $A = (a_{ij})$, where $a_{ij} = 1$ if vertex j belongs to hyperedge i. By applying the bivariate MODL coclustering approach to this incidence matrix, we obtain clusters of hyperedges which are similar w.r.t. their vertex incidence and clusters of vertices which are similar w.r.t. the connected hyperedges. We have applied this method to VPN (virtual private network) data, where each VPN can be considered as a hyperedge related to a subsets of vertices that need to be connected in a telecommunication network. Coclustering scientific papers (hyperedges) versus authors (vertices) is another application of hypergraph analysis.

K-partite hypergraph. A hypergraph is k-uniform if all its hyperedges have the same size k and k-partite if the vertex set can be divided into k sets in such a way that each

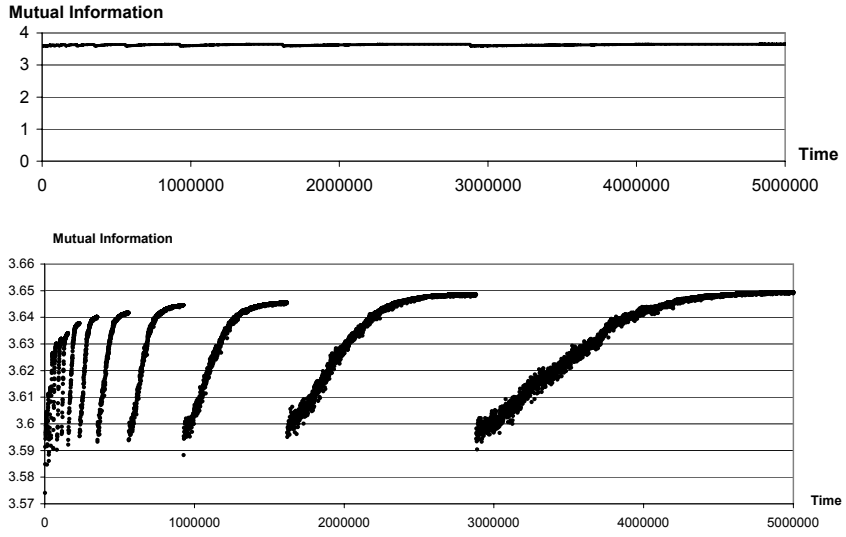


Figure 15: Anytime optimization of the MODL criterion (with reported estimated mutual information $I(V_S; V_T)$ (see Theorem 6) for the sample graph *brack*2). The bottom figure is a zoom of the top figure. The first result comes after 2000 seconds, and is improved by 2% after five million seconds.

hyperedge has one vertex per vertex set. By representing the hyperedges in a single table with k variables relative to its extremities in the k vertex sets, our approach can be extended to build k-clustering of the vertices, with one clustering per vertex set and the hyperedges distributed in the k-clusters resulting from the cross-product the vertex clusterings.

6.2 Algorithms

The CNM algorithm (Clauset et al., 2004) has a practical complexity of $O(m \log^2 n)$ in case of sparse graphs when the dendrogram is balanced, but may degenerate in $O(mn \log n)$. The BGLL algorithm (Blondel et al., 2008) is essentially linear in the number of edges, but its phase of swapping vertices across clusters may have a quadratic linear complexity in case of dense graphs.

Our algorithm has a guaranteed time complexity of $O(m\sqrt{m} \log m)$ for the main greedy heuristic and behaves as an anytime algorithm when the meta-heuristic is exploited to further explore the search space. It is noteworthy that modularity based algorithm only consider the diagonal of the block-clustered adjacency matrix (see Section 5.5), whereas our approach exploits all the adjacency matrix (the size of which is quadratic w.r.t. its diagonal). Therefore, our algorithm can hardly compete with the most efficient modularity based algorithms.

On the sample graph *brack*2 (around 60000 vertices and 370000 edges), the CNM algorithms outputs its clustering in about 500 seconds, and the BGLL in only 4 seconds (on a PC Windows with Pentium IV 3.2 Ghz, 2 Go RAM). The greedy heuristic used in our

algorithm requires about 2000 second to output its first solution. Figure 15 displays the results of the anytime algorithm running during five millions seconds. The meta-heuristic manages to improve the solution, with a fast rate at the beginning. However, the solution is improved by only 2% after five millions seconds. The first solution contains 316 clusters, not far from the 305 clusters of the best obtained solution. The first solution looks very good, but waiting 2000 seconds is too long in some application contexts. Our algorithm is also limited by the size of the graph, which is assumed to fit entirely into central memory. Some applications need far more scalable algorithms to analyze very large graphs, potentially several orders of magnitude above the standard available central memory. Different trade-offs between computational efficiency and quality of the clustering can be considered, and there is room for faster and more scalable algorithms, inspired from the state of the art multi-level clustering algorithms.

7. Conclusion

In this paper, we have presented a novel way of discovering structures in graph, by considering graphs as generative models whose statistical units are the edges, with unknown joint density of the source and target vertices. Our approach applies the MODL approach based on data grid models introduced in (Boullé, 2010) to the case of graphs. By clustering both the source and target vertices of a graph, the method behaves as a non-parametric estimator of the unknown edge density.

Our approach is compared experimentally with state of the art modularity based algorithms. It consistently builds better graph clusterings, being both resilient to randomness and accurate with fine grained clusters in case of informative graphs. Whereas alternative graph clustering algorithms assume cluster-based patterns, our approach does not make any assumption about the distribution of the edges in the graph. It is able to discover any kind of density-based patterns, such as cocliques where the intra-cluster edge density is null.

The main originality of the modeling approach is that it is data dependent and non-asymptotic in essence: it aims at modeling the finite graph sample directly. The modeling task is then easier, with finite modeling space and model priors which essentially reduce to counting. This modeling approach is both non-asymptotic and consistent, with an asymptotic convergence towards the true edge density, without any assumption regarding this density.

The approach can easily be applied or extended to many classes of graphs, such as multigraphs, directed or undirected graphs, bipartite graphs or hypergraphs. Still, further work is necessary to explore the practical interest, the benefits and the limitation of the approach in such extended settings.

The algorithm exploited in our approach has a $O(m\sqrt{m}\log m)$ time complexity, where m is the number of edges. Although this is acceptable for many applications, this does not scale enough in the case of very large graphs. We do believe that the MODL criterion for edge density estimation in graphs is very relevant, and working on faster and more scalable optimization algorithms is a potentially valuable research direction.

Appendix A.

In this appendix we prove theorems 4 and 5 from Section 5.3.

Theorem 4 *The MODL evaluation criterion (Formula 1) for an edge density estimation model M is asymptotically equal to m times the entropy of the source and target vertex variables minus their mutual entropy.*

$$c(M) = m\left(H(V_S) + H(V_T) - I(V_S^M; V_T^M)\right) + O(\log m).$$

Proof According to Formula 1, we have:

$$\begin{aligned} c(M) =& \log n_S + \log n_T + \log B(n_S, k_S) + \log B(n_T, k_T) \\ &+ \log \binom{m + k_E - 1}{k_E - 1} + \sum_{i=1}^{k_S} \log \binom{m_{i.}^S + n_i^S - 1}{n_i^S - 1} + \sum_{j=1}^{k_T} \log \binom{m_{.j}^T + n_j^T - 1}{n_j^T - 1} \\ &+ \log m! - \sum_{i=1}^{k_S}\sum_{j=1}^{k_T} \log m_{ij}^{ST}! + \sum_{i=1}^{k_S} \log m_{i.}^S! - \sum_{i=1}^{n_S} \log m_{i.}! + \sum_{j=1}^{k_T} \log m_{.j}^T! - \sum_{j=1}^{n_T} \log m_{.j}!. \end{aligned} \tag{20}$$

The first two lines of the criterion, corresponding to the encoding of the model prior parameters, can be bounded by $O(\log m)$. Using the approximation $\log n! = n(\log n - 1) + O(\log n)$ based on Stirling's formula and rearranging the terms with new $m \log m$ terms, we get:

$$\begin{aligned} c(M) =& m \log m - \sum_{i=1}^{k_S}\sum_{j=1}^{k_T} m_{ij}^{ST} \log m_{ij}^{ST} \\ &- \left(m \log m - \sum_{i=1}^{k_S} m_{i.}^S \log m_{i.}^S\right) - \left(m \log m - \sum_{j=1}^{k_T} m_{.j}^T \log m_{.j}^T\right) \\ &+ \left(m \log m - \sum_{i=1}^{n_S} m_{i.} \log m_{i.}\right) + \left(m \log m - \sum_{j=1}^{n_T} m_{.j} \log m_{.j}\right) + O(\log m). \end{aligned} \tag{21}$$

Using the fact that the sum of the edge counts in each partition (per cocluster, per cluster in and out-degree and per vertex in and out-degree) is always equal to m, we obtain:

$$\begin{aligned} c(M) =& -m\left(\sum_{i=1}^{k_S}\sum_{j=1}^{k_T} \frac{m_{ij}^{ST}}{m} \log \frac{m_{ij}^{ST}}{m} - \sum_{i=1}^{k_S} \frac{m_{i.}^S}{m} \log \frac{m_{i.}^S}{m} - \sum_{j=1}^{k_T} \frac{m_{.j}^T}{m} \log \frac{m_{.j}^T}{m}\right) \\ &- m\sum_{i=1}^{n_S} \frac{m_{i.}}{m} \log \frac{m_{i.}}{m} - m\sum_{j=1}^{n_T} \frac{m_{.j}}{m} \log \frac{m_{.j}}{m} + O(\log m). \end{aligned} \tag{22}$$

As the marginal distributions $m_{i.}^S$ and $m_{.j}^T$ can be decomposed by summation on the joint distribution m_{ij}^{ST}, we have:

$$\begin{aligned} c(M) = & - m \sum_{i=1}^{k_S} \sum_{j=1}^{k_T} \frac{m_{ij}^{ST}}{m} \log \frac{\frac{m_{ij}^{ST}}{m}}{\frac{m_{i.}^S}{m} \frac{m_{.j}^T}{m}} \\ & - m \sum_{i=1}^{n_S} \frac{m_{i.}}{m} \log \frac{m_{i.}}{m} \quad - \quad m \sum_{j=1}^{n_T} \frac{m_{.j}}{m} \log \frac{m_{.j}}{m} \quad + \quad O(\log m). \end{aligned} \tag{23}$$

Considering that the empirical estimation asymptotically converges towards the related probabilities, the claim follows. ■

Theorem 5 *The MODL approach for selecting an edge density estimation model M asymptotically converges towards the true edge distribution, and the criterion for the best model M_{Best} converges to m times the entropy of the edge variable, that is the joint entropy of the source and target vertices variables.*

$$\lim_{m \to \infty} \frac{c(M_{Best})}{m} = H(V_S, V_T).$$

Proof Using Theorem 4, we have

$$c(M) = -mI(V_S^M; V_T^M) + mH(V_S) + mH(V_T) + O(\log m). \tag{24}$$

We apply the Data Processing Inequality (DPI) (Cover and Thomas, 1991), that states that post-processing cannot increase information. More precisely, the DPI applies for three random variables X, Y, Z that form a Markov chain $X \to Y \to Z$. It means that the conditional distribution of Z depends only on Y and is conditionally independent of X. More specifically, for three random variables such that $p(Z|X,Y) = P(Z|Y)$, the DPI states that $I(X;Y) \geq I(X;Z)$

We apply the DPI to the variables V_S, V_T, V_T^M. As the vertex cluster variable V_T^M can be computed according to a partition of the vertex variable V_T ($V_T^M = f(V_T)$), we have $p(V_T^M|V_S, V_T) = p(V_T^M|V_T)$ and thus obtain:

$$I(V_S; V_T) \geq I(V_S; V_T^M). \tag{25}$$

We apply again the DPI to the variables V_T^M, V_S, V_S^M. As the vertex cluster variable V_S^M is a function of V_S, we have $p(V_S^M|V_S, V_T^M) = p(V_S^M|V_S)$ and get:

$$I(V_T^M; V_S) \geq I(V_T^M; V_S^M). \tag{26}$$

By transitivity and since the mutual information is symmetrical, we get:

$$I(V_S; V_T) \geq I(V_S^M; V_T^M). \tag{27}$$

It is noteworthy that this result applies to compare any pair of coclustering models, one of the models being a sub-partition of the other: the finer model brings a higher mutual information.

The model selection approach corresponds to a minimization of the MODL criterion. Let us now show that the best selected model asymptotically tends to be finer and finer, until reaching the finest possible model with one cluster per vertex, which is the maximal model M_{Max} that enables a direct estimation of the edge probabilities p_{ij} (see Formula 9):

$$I(V_S;V_T) = I(V_S^{M_{Max}};V_T^{M_{Max}}) \geq I(V_T^M;V_S^M). \tag{28}$$

If $\forall\, M, I(V_S^{M_{Max}};V_T^{M_{Max}}) = I(V_T^M;V_S^M)$, then using Theorem 4, the MODL approach asymptotically converges towards the true edge distribution.

If $\exists\, M_f, M_c, I(V_S^{M_{Max}};V_T^{M_{Max}}) = I(V_T^{M_f};V_S^{M_f}) > I(V_T^{M_c};V_S^{M_c})$, with M_f a fine-grained model having the same mutual information as the maximal model and M_c a coarse-grained model, then let us show that the approach asymptotically selects the fine-grained model M_f rather than the coarser model M_c.

Let $\epsilon = \frac{I(V_S^{M_f};V_T^{M_f}) - I(V_T^{M_c};V_S^{M_c})}{2}$.

Using Theorem 4 for the convergence of the criterion for model M_c,

$$\exists\, m_1, \forall\, m \geq m_1, \quad \left| \frac{c(M_c)}{m} - \Big(H(V_S) + H(V_T) - I(V_S^{M_c};V_T^{M_c}) \Big) \right| < \frac{\epsilon}{2}.$$

Similarly, for model M_f,

$$\exists\, m_2, \forall\, m \geq m_2, \quad \left| \frac{c(M_f)}{m} - \Big(H(V_S) + H(V_T) - I(V_S^{M_f};V_T^{M_f}) \Big) \right| < \frac{\epsilon}{2}.$$

Thus,

$$\begin{aligned} &\forall\, m \geq \max(m_1, m_2), \\ &\qquad \frac{c(M_c)}{m} > H(V_S) + H(V_T) - I(V_S^{M_c};V_T^{M_c}) - \frac{\epsilon}{2}, \\ &\qquad \textit{and} \quad \frac{c(M_f)}{m} < H(V_S) + H(V_T) - I(V_S^{M_f};V_T^{M_f}) + \frac{\epsilon}{2}. \end{aligned}$$

$$\begin{aligned} \forall\, m \geq \max(m_1, m_2), \quad & \frac{c(M_f)}{m} - \frac{c(M_c)}{m} < I(V_S^{M_c};V_T^{M_c}) - I(V_S^{M_f};V_T^{M_f}) + \epsilon, \\ & \frac{c(M_f)}{m} < \frac{c(M_c)}{m} - \epsilon. \end{aligned}$$

Since the model selection approach corresponds to a minimization of the MODL criterion, this means that the best selected model M_{Best} asymptotically tends to be a fine-grained model M_f having the same mutual information as the maximal model, which allows the estimation of the true edge distribution. Using Theorem 4 with the best selected model, we have:

$$c(M_{Best}) \approx -mI(V_S;V_T) + mH(V_S) + mH(V_T) + O(\log m). \tag{29}$$

As $I(X;Y) = H(X) + H(Y) - H(X,Y)$, the claim follows. ■

References

M. Abramowitz and I. Stegun. *Handbook of mathematical functions*. Dover Publications Inc., New York, 1970.

E.M. Airoldi, d S.E. Fienberg D.M. Blei a, and E.P. Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9:1981–2014, 2008.

R. Albert and A-L. Barabási. Statistical mechanics of complex networks. *Rev. Mod. Phys.*, 74:47–97, 2002.

P. Arabie, S.A. Boorman, and P.R. Levitt. Constructing blockmodels: How and why. *Journal of Mathematical Psychology*, 17(1):21–36, 1978.

R. Battiti and A. Bertossi. Greedy, prohibition, and reactive heuristics for graph partitioning. *IEEE Transactions on Computers*, 48(4):361–385, 1999.

P.J. Bickel and A. Chen. A nonparametric view of network models and newman-girvan and other modularities. *Proceedings of the National Academy of Sciences*, 106(50):21068–21073, 2009.

J. Blitzstein and P. Diaconis. A sequential importance sampling algorithm for generating random graphs with prescribed degrees. Technical report, Stanford University, 2006.

V.D. Blondel, J-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10): P10008, 2008.

H. Bock. Simultaneous clustering of objects and variables. In E. Diday, editor, *Analyse des Données et Informatique*, pages 187–203. INRIA, 1979.

M. Boullé. Data grid models for preparation and modeling in supervised learning. In I. Guyon, G. Cawley, G. Dror, and A. Saffari, editors, *Hands on pattern recognition*. Microtome, 2010. in press.

M. Boullé. A Bayes optimal approach for partitioning the values of categorical attributes. *Journal of Machine Learning Research*, 6:1431–1452, 2005.

M. Boullé. MODL: a Bayes optimal discretization method for continuous attributes. *Machine Learning*, 65(1):131–165, 2006.

M. Boullé. Bivariate data grid models for supervised learning. Technical Report NSM/R&D/TECH/EASY/TSI/4/MB, France Telecom R&D, 2008a. http://perso.rd.francetelecom.fr/boulle/publications/BoulleNTTSI4MB08.pdf.

M. Boullé. Multivariate data grid models for supervised and unsupervised learning. Technical Report NSM/R&D/TECH/EASY/TSI/5/MB, France Telecom R&D, 2008b. http://perso.rd.francetelecom.fr/boulle/publications/BoulleNTTSI5MB08.pdf.

T.N. Bui and B.R. Moon. Genetic algorithm and graph partitioning. *IEEE Transactions on Computers*, 45(7):841–855, 1996.

D. Chakrabarti, S. Papadimitriou, D.S. Modha, and C. Faloutsos. Fully automatic cross-associations. In *In KDD*, pages 79–88. ACM Press, 2004.

P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth. *CRISP-DM 1.0 : step-by-step data mining guide*, 2000.

F. Chung and L. Lu. *Complex Graphs and Networks (Cbms Regional Conference Series in Mathematics)*. American Mathematical Society, Boston, MA, USA, 2006.

A. Clauset, M.E.J. Newman, and C. Moore. Finding community structure in very large networks. *Physical Review E*, 70(6), 2004. 066111.

W.G. Cochran. Some methods for strengthening the common chi-squared tests. *Biometrics*, 10(4):417–451, 1954.

A. Condon and R.M. Karp. Algorithms for graph partitioning on the planted partition model. *Random Structures & Algorithms*, 18(2):116–140, 2001.

J. Copic, M. O. Jackson, and A. Kirman. Identifying community structures from network data via maximum likelihood methods. *The B.E. Journal of Theoretical Economics*, 9 (1), 2009.

T.M. Cover and J.A. Thomas. *Elements of information theory*. Wiley-Interscience, New York, NY, USA, 1991.

L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, page P09008, 2005.

I. S. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic co-clustering. In *Proceedings of The Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining(KDD-2003)*, pages 89–98, 2003.

P. Doreian, V. Batagelj, and A. Ferligoj. Generalized blockmodeling of two-mode network data. *Social Networks*, 26:29–53, 2004.

I.S. Duff, R.G. Grimes, and J.G. Lewis. Sparse matrix test problems. *ACM Transactions on Mathematical Software*, 15(1):1–14, 1989.

P. Erdős and A. Rényi. On random graphs I. *Selected Papers of Alfréd Rényi*, 2:308–315, 1976. First publication in Publ. Math. Debrecen 1959.

U.M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. Knowledge discovery and data mining: towards a unifying framework. In *KDD*, pages 82–88, 1996.

C.M. Fiduccia and R.M. Mattheyses. A linear-time heuristic for improving network partitions. In *DAC '82: Proceedings of the 19th Design Automation Conference*, pages 175–181, Piscataway, NJ, USA, 1982. IEEE Press.

M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

P.S. Gill and T.B. Swartz. Bayesian analysis of directed graphs data with applications to social networks. *Journal Of The Royal Statistical Society Series C*, 53(2):249–260, 2004.

F. Glover. Tabu search methods in artificial intelligence and operations research. *ORSA Artificial Intelligence Newsletter*, 1(2), 1987.

A. Goldenberg, A.X. Zheng, S.E. Fienberg, and E.M. Airoldi. A survey of statistical network models. *Source Foundations and Trends in Machine Learning*, 2(2):129–233, 2010.

G. Govaert and M. Nadif. Clustering with block mixture models. *Pattern Recognition*, 36 (2):463–473, 2003.

P.D. Grünwald, I.J. Myung, and M.A. Pitt. *Advances in minimum description length : theory and applications*. MIT Press, 2005.

R. Guimerà, M. Sales-Pardo, and L.A.N. Amaral. Modularity from fluctuations in random graphs and complex networks. *Physical Review E*, 70, 2004. 02501.

M.H. Hansen and B. Yu. Model selection and the principle of minimum description length. *J. American Statistical Association*, 96:746–774, 2001.

P. Hansen and N. Mladenovic. Variable neighborhood search: principles and applications. *European Journal of Operational Research*, 130:449–467, 2001.

J.A. Hartigan. Direct clustering of a data matrix. *Journal of the American Statistical Association*, 67(337):123–129, 1972.

B. Hendrickson and R. Leland. A multilevel algorithm for partitioning graphs. In *Conference on High Performance Networking and Computing, Proceedings of the 1995 ACM/IEEE conference on Supercomputing*, 1995. Article No.: 28.

P.W. Holland and S.Leinhardt. An exponential family of probability distributions for directed graphs. *Journal of the American Statistical Association*, 76(373):33–50, 1981.

P.W. Holland, K.B. Laskey, and S. Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137, 1983.

D.S. Johnson, C. Aragon, L. McGeoch, and C. Schevon. Optimization by simulated annealing: An experimental evaluation, part 1, graph partitioning. *Operations Research*, 37: 865–892, 1989.

G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1):359–392, 1998.

B. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell Systems Technical Journal*, 49:291–317, 1970.

S. Kirkpatrick, C.D. Gellat Jr., and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

S. Kullback. *Information theory and statistics*. John Wiley and Sons., New York, 1959. republished by Dover, 1968.

K.J. Lang. Information theoretic comparison of stochastic graph models: Some experiments. In *WAW '09: Proceedings of the 6th International Workshop on Algorithms and Models for the Web-Graph*, pages 1–12, Berlin, Heidelberg, 2009. Springer-Verlag.

M. Li and P.M.B. Vitanyi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer-Verlag, Berlin, 1997.

F. Lorrain and H.C. White. Structural equivalence of individuals in social networks. *Journal of Mathematical Sociology*, 1:49–80, 1971.

M.E.J. Newman. The structure of scientific collaboration networks. *Proceedings of the National Academy of Science of the USA*, 98:404–409, 2001.

M.E.J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69, 2003. 026113.

K. Nowicki and T.A.B. Snijders. Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, 96(455):1077–1097, 2001.

J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

J.M. Roberts. Simple methods for simulating sociomatrices with given marginal totals. *Social Networks*, 22(3):273–283, 2000.

M. Rosvall and C.T. Bergstrom. An information-theoretic framework for resolving community structure in complex networks. *Proc Natl Acad Sci U S A*, 104(18):7327–7331, 2007.

S.E. Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, 2007.

C.E. Shannon. A mathematical theory of communication. Technical Report 27, Bell systems technical journal, 1948.

T.A.B. Snijders and K. Nowicki. Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *Journal of Classification*, 14(1):75–100, 1997.

C. Walshaw. The graph partitioning archive. University of Greenwich, UK, 2000. URL `http://staffweb.cms.gre.ac.uk/~c.walshaw/partition/`.

Y.Y. Wang and G.Y. Wong. Stochastic blockmodels for directed graphs. *Journal of the American Statistical Association*, 82(397):8–19, 1987.

S. Wasserman, G. Robins, and D. Steinley. Statistical models for networks: A brief review of some recent research. In E. Airoldi, D.M. Blei, S.E. Fienberg, A. Goldenberg, E.P. Xing, and A.X. Zheng, editors, *Statistical Network Analysis: Models, Issues, and New Directions*, volume 4503 of *Lecture Notes in Computer Science*, chapter 4, pages 45–56. Springer Berlin Heidelberg, 2007.

S.S. Wasserman and C. Anderson. Stochastic a posteriori blockmodels: Construction and assessment. *Social Networks*, 9(1):1–36, 1987.

D.R. White and K.P. Reitz. Graph and semigroup homomorphisms on networks of relations. *Social Networks*, 5(2):193–234, 1983.

H.C. White, S.A. Boorman, and R.L. Breiger. Social structure from multiple networks. I. blockmodels of roles and positions. *American Journal of Sociology*, 81:730–780, 1976.