The Noise Cluster Model

A Greedy Solution to the Network Community Extraction Problem

Etienne Côme[*], Eustache Diemert[**]

[*] *INRETS, LTN, 2 Rue de la Butte Verte, 93166 Noisy-le-Grand Cedex - France*
etienne.come@inrets.fr

[**] *BestOfMedia Group, 485 avenue de l'Europe, F-38330 Montbonnot - France*
ediemert@bestofmedia.com

RÉSUMÉ. *Cet article présente un algorithme permettant d'extraire une communauté de nœuds densément connectés dans un graphe. La solution proposée à ce problème s'appuie sur une approche semi-supervisé au sens où un ensemble de graines (nœuds appartenant à la communauté à extraire) doit être fourni. En partant de ces nœuds l'algorithme explore le graphe en largeur et décide d'ajouter ou non les nouveaux nœuds rencontrés à la communauté en utilisant deux tests construit à l'aide d'un modèle génératif simple relié au modèle de mélange de graphe de type Erdös-Rényi [DAU 08]. Ce modèle simple sera appelé "noise cluster model". Une méthode d'estimation en ligne est utilisée pour mettre à jour les paramètres du modèle tout au long de la procédure d'extraction de la communauté. Cette approche est donc locale au sens ou elle ne nécessite pas une connaissance exhaustive du graphe ce qui permet d'utiliser celle-ci sur des graphes de tailles quelquonques. Finalement, des expériences sur des communautés réelles de blogs seront présentées pour juger de la pertinence de l'approche proposée.*

ABSTRACT. *This paper presents an algorithm designed to extract one community from a graph given some seeds inside the community. Starting from these nodes, new nodes will be added to the community by selecting them among the successors of the current community members. The selection process used to select the community members among the successors is based on a generative model closely related to Erdös-Rényi mixture [DAU 08] which we call the Noise Cluster Model. An on-line estimation procedure [ZAN 08] is used to update the model parameters during the community extraction process. This approach is therefore local in the sense that it did not require the graph to be completely known in advance and can therefore be used in huge graph. Finally, experiments on real blog communities will show the interest of such an approach.*

MOTS-CLÉS : *clustering de graphe, extraction de communautés, apprentisage semi-supervisé*

KEYWORDS: *graph clustering, community extraction, semi-supervised, noise cluster model*

1. Introduction

A community could be loosely described as a collection of vertices within a graph that are densely connected amongst themselves while being loosely connected to the rest of the graph. The main line of work on community analysis concerns graph clustering. In this setting the input graph must be partitioned into different sets or clusters which present communities characteristics. This task can be solved using different solutions such as the classical modularity optimization from [NEW 07]. Other algorithms for graph clustering can be found in the quite exhaustive state of the art of S. Fortunato [FOR 09]. The model we will use here to represent community structure is a constrained version of the Erdös-Rényi mixture model [DAU 08] which is related to block models [HOL 83, SNI 97]. Such models have already been used for clustering an entire graph, and we will demonstrate its use to address the related problem of community extraction.

In the context of community extraction, one is interested in extracting only one community. Furthermore, we assume in this paper that seed vertices that belong to the community of interest are provided as inputs. Such additional information provide an interesting front door which enables the use of local approaches, avoiding the examination of the entire graph to extract the seeds community.

The proposed solution is therefore a local algorithm built over the Erdös-Rényi mixture model that extracts the community which encloses the seeds. This algorithm does not require the graph to be completely known and has a complexity (in space and time) which is mainly influenced by the size of the extracted community and not by the size of the whole graph. This property is interesting since it enables the use of such a solution on very big graphs like the World Wide Web hyperlink graph. Experiments will highlight this fact by using this algorithm to extract blog communities.

Other local procedures have already been proposed to extract communities from graph starting from seeds nodes. Bagrow & al [BAG 05] propose a technique which relies upon growing a breadth first tree outward from one seed node until the rate of expansion (i.e. the proportion of edges found at the current level which lead to nodes which are yet unknown) falls below an arbitrary threshold. This simple solution is interesting. However, since all the nodes found at one level of the breadth first tree are added to the community (if the rate of expansion is bellow the threshold), it will succeed in extracting the community only if the source vertex is equidistant from all parts of its enclosing community boundary. Furthermore, the tuning of this threshold can be tricky and was to be defined a priori in the original algorithm. The threshold and seed must therefore be carefully chosen or multiple seeds used and the results combined (this second solution is advocated by the authors). Another solution proposed by [CLA 05] is based on greedy optimization of a quantity called local modularity. This quantity involves a specific set of nodes called the boundary. This set is defined as the set of nodes that have at least one neighbor in the set of yet unknown nodes. Local modularity is then defined as the number of edges between this set and the set of known nodes over the total number of edges with one extremity in this set. The

greedy optimization of this quantity simply adds the unknown node which gives the largest increase (or the smallest decrease) of the local modularity to the community until a predefined number of nodes is reached. As with the previous solution, only one node is used as seed, which makes a difference with our solution. Moreover, the optimized criterion is here derived from an ad-hoc definition and no solution to stop the extraction process automatically is supplied (the number of nodes to extract must be supplied by the user). Conversely, our method make use of online learning to estimate a stopping criterion based on the features of the discovered community.

Other solutions to the community extraction problem use conductance and random walks [AND 06] or combinatorial algorithms [SOZ 10] to define the extraction procedure. However, these solutions present complexities that scale linearly with the size of the graph, whereas our solution scales with the size of the community to extract.

The road map of the paper is the following : firstly some background on the Erdös-Rényi mixture model will be supplied in section 2. Then, the constrained version of this model used in the paper will be detailed in section 3. Finally, section 4 will present the proposed local algorithm and section 5 details preliminary experiments with the blog community extraction problem.

2. Background on Erdös-Rényi mixture model

Formally, the graph clustering problem is set-up in the Erdös-Rényi mixture model with the help of two sets of random variables with the following meaning :

– X_{ij} are binary variables indicating the presence or the absence of an edge from node i to node j :

$$x_{ij} = \begin{cases} 1, \text{ if there is an edge from } i \text{ to } j \\ 0, \text{ otherwise.} \end{cases} \tag{1}$$

– Z_{jk} are latent variables encoding cluster membership, such that :

$$z_{jk} = \begin{cases} 1, \text{ if } j \text{ belongs to cluster } k \\ 0, \text{ otherwise.} \end{cases} \tag{2}$$

Oriented graphs will be considered in this paper. Therefore we will consider that x_{ij} may differ from x_{ji}. These variables have the following distributions in this model :

$$Z_{jk} \overset{i.i.d}{\sim} \mathcal{M}(1, \gamma), \quad \forall i \in \{1, \ldots, N\} \tag{3}$$

$$X_{ij} | Z_{ik} \times Z_{jl} = 1 \overset{i.i.d}{\sim} \mathcal{B}(\pi_{kl}), \quad \forall i, j \in \{1, \ldots, N\}, \tag{4}$$

where $\mathcal{M}$ denotes the multinomial distribution and $\mathcal{B}$ the Bernoulli distribution. When $\pi_{kk} \gg \pi_{kl}, \forall k \neq l$ clusters correspond to dense components in the graph. We propose to use in the context of community extraction a simpler model with less parameters, which we present shortly in the sequel. This constrained version is proposed to

model the case were the graph contains one community and background noise with no specific structure. This is sufficient in our context were we only want to extract one community of interest.

3. The noise cluster model

We will consider only two mixture components, one for the community that the users want to extract and one for the nodes that do not belong to the community, that will be called the noise component. We will therefore use only one Bernoulli variable to deal with cluster membership Z_i, which is defined as :

$$z_i = \begin{cases} 1, \text{ if } i \text{ belongs to the community} \\ 0, \text{ if } i \text{ belongs to the noise component} \end{cases} \tag{5}$$

The model, which is a constrained version of the block model takes the following simple form :

$$Z_i \overset{i.i.d}{\sim} \mathcal{B}(\gamma), \quad \forall i \in \{1, \ldots, N\} \tag{6}$$

$$X_{ij}|Z_i \times Z_j = 1 \overset{i.i.d}{\sim} \mathcal{B}(\alpha), \quad \forall i, j \in \{1, \ldots, N\} \tag{7}$$

$$X_{ij}|Z_i \times Z_j = 0 \overset{i.i.d}{\sim} \mathcal{B}(\beta), \quad \forall i, j \in \{1, \ldots, N\} \tag{8}$$

We therefore have only three parameters $\theta = (\alpha, \beta, \gamma)$, γ is the prior probability of the community, α is the probability that two nodes from the community are linked and β is the probability that tune the noise cluster behavior. This simple model is sufficient to represent the community structure that we are interested in, provided $\alpha \gg \beta$. Let us introduce some notations and properties of this model that we will use in the sequel.

Definition Let d_j be node j degree with community members, d_j^{in} node j in-degree with community members and d_j^{out} node j out-degree with community members :

$$d_j^{in} = \sum_{i:z_i=1} x_{ij}, \quad d_j^{out} = \sum_{i:z_i=1} x_{ji}, \quad d_j = \sum_{i:z_i=1} (x_{ij} + x_{ji})$$

Definition Let p_j^i, p_j^{io} be the community membership posterior probabilities given cluster membership and respectively in-links or out-links and in-links for node j :

$$p_j^i = \mathbb{P}(Z_j = 1 | X_{ij} = x_{ij}, Z_i = z_i, \forall i \in \{1, \ldots, N\}), \tag{9}$$

$$p_j^{io} = \mathbb{P}(Z_j = 1 | X_{ij} = x_{ij}, X_{ji} = x_{ji}, Z_i = z_i, \forall i \in \{1, \ldots, N\}), \tag{10}$$

Proposition 3.1. *Community membership posterior probabilities p_j^i, and p_j^{io} depend only on parameters α, β, γ and $d_j^{in}, d_j^{in,out}$ respectively and are given by :*

$$p_j^i = \frac{\alpha^{d_j^{in}} \times (1-\alpha)^{(N_c - d_j^{in})} \times \gamma}{\alpha^{d_j^{in}} \times (1-\alpha)^{(N_c - d_j^{in})} \times \gamma + \beta^{d_j^{in}} \times (1-\beta)^{(N_c - d_j^{in})} \times (1-\gamma)} \tag{11}$$

$$p_j^{io} = \frac{\alpha^{d_j} \times (1-\alpha)^{(2\times N_c - d_j)} \times \gamma}{\alpha^{d_j} \times (1-\alpha)^{(2\times N_c - d_j)} \times \gamma + \beta^{d_j} \times (1-\beta)^{(2\times N_c - d_j)} \times (1-\gamma)}, \quad (12)$$

with $N_c = \sum_{i=1}^{N} z_i$ *the community size.*

The probabilities p_j^i and p_j^{io} depends uniquely on graph structure through d_j^{in} (the number of in-goings links from the community members) and $d_j^{in,out}$ (the total of links with community members) respectively. The interest of p_j^i comes from the fact that this probability can be computed without knowing the out-goings links of node j. This probability can thus be computed online, and therefore drive a greedy extraction procedure. In our experimental setting, this means that we can compute this probability if the graph is not entirely stored in memory as it is often the case when crawling HTML documents. Figure 1 (top) gives an example of this conditional law. As expected this quantity increases with d_j^{in} (with $\alpha \gg \beta$), more links from the community giving therefore a higher probability of belonging to the community. Starting from

Figure 1. *(top) values of* p_j^{in} *with respect to* d_j^{in} *with* $\alpha = 0.1$, $\beta = 0.001, \gamma = 0.05$ *and* $N_c = 200$*; (bottom)* d_{min} *evolution with respect to the community size* N_c *with* $\alpha = 0.1$, $\beta = 0.001, \gamma = 0.05$ *and* $s = 0.5$.

this simple model, we will describe an online, greedy algorithm that adds new nodes to the community from the community successors.

4. Local algorithm description

As explained in the introduction, the algorithm is supplied with seed nodes. These seeds are considered to belong to the community with certainty, and along its path the algorithm add new nodes to the community by looking at current community members out-going links.

The algorithm proceed one vertex at a time in a breath first fashion, but use the previous generative model to decide which found node to add to the community. A first test which use only in-links information is performed to find new nodes that may belong to the community. Such nodes are then added to the queue of nodes which require further investigation. When a node succeeds in this first test, another test (which takes into account the in and out-goings links of the node) is performed to decide whether to add it permanently to the community. This process is repeated until no more nodes are accepted by the first test. During all the community extraction process the three model parameters are updated using an on-line estimation strategy [ZAN 08]. The core of the algorithm is the two tests used to decide to add one node to the community or not and the on-line parameters estimation procedure. The two tests are derived directly from equations (11) and (12). We describe them shortly and give some insights into the on-line estimation procedure.

4.1. *Community Membership tests*

When only in-links are known it is natural to decide that node j belongs to the community when $p_j^i > s$. Starting form equation (11) we may rewrite the test in terms of d_j^{in}, $d_j^{in} > d_{min}^{in}$, with d_{min}^{in} equals to :

$$d_{min}^{in} = \left\lfloor \frac{\log\left(s \times (1-\beta)^{N_c} \times (1-\gamma)\right) - \log\left((1-s) \times (1-\alpha)^{N_c} \times \gamma\right)}{\log\left(\alpha \times (1-\beta)\right) - \log\left((1-\alpha) \times \beta\right)} \right\rfloor \quad (13)$$

Figure 1 (bottom) presents the evolution of d_{min}^{in} with respect to the community size N_c which has a simple step profile. Similar expressions can be obtained for the test which used in and out links using equation (12) which is performed in a second step when the node out-links have been retrieved.

4.2. *Parameters estimation*

This section describes how the incremental Classification version of the EM algorithm, proposed by [ZAN 08] can be adapted to estimate the previous model parameters during the crawling process. We first present the criterion used to estimate the parameters, known as classification likelihood, and then the estimation procedure itself. In the case of a full adjacency matrix, the classification log-likelihood is defined as :

$$\begin{aligned} L_c(\mathbf{X}, \mathbf{Z}, \theta) = & \sum_i z_i \log(\gamma) + \sum_i (1 - z_i) \log(1 - \gamma) \\ & + \sum_{i,j:i \neq j} z_i \times z_j \times x_{ij} \log(\alpha) + \sum_{i,j:i \neq j} z_i \times z_j (1 - x_{ij}) \log(1 - \alpha) \\ & + \sum_{i,j:i \neq j} (1 - z_i \times z_j) \times x_{ij} \log(\beta) + \sum_{i,j:i \neq j} (1 - z_i \times z_j) \times (1 - x_{ij}) \log(1 - \beta) \end{aligned}$$

with $\mathbf{Z} = \{z_1, \ldots, z_N\}$, $\mathbf{X} = \{x_{ij} : i \neq j, i, j \in \{1, \ldots, N\}\}$, and $\theta = (\gamma, \alpha, \beta)$ the parameters vector.

If the partition $\mathbf{Z} = \{z_1, \ldots, z_N\}$ is known and with a square adjacency matrix of size $N \times N$, the parameter vector maximizing the Classification likelihood is given by :

$$\hat{\gamma} = \frac{N_c}{N}, \quad (14)$$

$$\hat{\alpha} = \frac{1}{N_c^2} \sum_{i,j=1,\, i \neq j}^{N} (z_i \times z_j) x_{ij}, \quad (15)$$

$$\hat{\beta} = \frac{1}{N_{\bar{c}} \times (N + N_c)} \sum_{i,j=1,\, i \neq j}^{N} (1 - z_i \times z_j) x_{ij}, \quad (16)$$

with $N_{\bar{c}}$ the number of nodes that do not belong to the community $N_{\bar{c}} = \sum_{i=1}^{N}(1 - z_i)$ and N the total number of nodes. However, the partition $\mathbf{Z} = \{z_1, \ldots, z_N\}$ is unknown and must also be estimated, an on-line alternating optimization solution can be used to solve this problem. For this purpose the two previous tests are used to estimate the partition for every new nodes and equations (14, 15, 16) are used to update the parameters after each test. Such solution is sub-optimal but works well in practice and is really fast. Finally, it's important to note that equations (14, 15 and 16) can be computed incrementally to avoid unnecessary calculus.

We now present some results on several blog communities extraction tasks.

5. Experiments : blog community crawler

An experimental version of the algorithm was developed to deal with HTML documents and used to extract blog communities. This experimental tool is basically a multi-threaded web crawler coupled with the community extraction procedure described above. The seeds URLs supplied to the algorithm were taken from a blog portal called Wikio (http ://www.wikio.com) which proposes several rankings of blogs for several topics. Theses ranking were used to provide 100 or 50 seeds to the algorithm for 4 test communities. Table 1 presents the model parameters estimated by the algorithm and several global statistics of the retrieved communities.

	Comics (Fr)	Scrapbooking (Fr)	Food (U.S.)	Politics (U.S.)
Nb seed	100	100	50	50
N_c	1 263	1 130	1 681	1 884
Nb edges	20 434	24 248	100 597	74 219
α	0.01821	0.01899	0.03560	0.02091
β	0.00093	0.00147	0.00091	0.00065
γ	0.03048	0.05579	0.03060	0.01808
Biggest S.C.C.	1 251	1 129	1 667	1 877
Max Level	3	2	5	4
Diameter	6	7	7	8
Radius	4	4	4	3
Clustering Coeff.	0.287	0.265	0.381	0.320
Transitivity	0.198	0.2	0.290	0.223

Tableau 1. *Global statistics and model parameters for 4 communities.*

The communities extracted have all more than 1000 nodes and are very dense in terms of links between members. They all have the property $\alpha \gg \beta$ with α around 10^{-2} and β around $10^{-3}, 10^{-4}$. The biggest Strongly Connected Component is almost equal to the community size for all the communities which is an interesting finding. Indeed, this means that the extracted nodes are not randomly connected, but have enough internal linkage to produce a community structure.

Other statistics give also some clues on the community structures which seem relevant : high transitivity and clustering coefficient, small diameter and radius, all of which are typical of graph community structures [FOR 09] .

To validate the results, we performed a quick manual analysis of the blogs. All the blogs were not investigated but top blogs according to PageRank [PAG 98] computed on the community graphs were manually visited and a very good match with community topics was found for all the inspected blogs. Table 2 gives the top ten blogs according to local PageRank for the Comics (Fr) community, which are all dealing with sketches or comics, but are not all in French. Therefore the extracted community seems coherent in terms of topic but a drift from French to Spanish and U.S. English can be observed for this community.

	names	level
1	www.bouletcorp.com	0
2	louromano.blogspot.com	2
3	www.cartoonbrew.com	2
4	yacinfields.blogspot.com	1
5	polyminthe.blogspot.com	1
6	marnette.canalblog.com	1
7	blackwingdiaries.blogspot.com	2
8	bastienvives.blogspot.com	1
9	donshank.blogspot.com	2
10	john-nevarez.blogspot.com	2

Tableau 2. *Best blogs according to local PageRank for the Comics (Fr) community*

One manual analysis of a community was also performed on a smaller dataset (704 nodes) concerning embroidery. The blogs were all dealing with embroidery or related hand-made activity such as patchwork, knitting, and painting (only 2 blogs) therefore the precision of the method is very high. However, the recall can't be evaluated on such a task. Further studies will be of interest to evaluate this point. Figure 2 presents the graph of this community, and show that as for the Comics (Fr) community, blogs from different countries and written in different languages have been retrieved by the methods.

Figure 2. *Visualization of one community graph (embroidery blogs), nodes colors represent nationality.*

Finally, the text content of the retrieved communities was also analyzed. Texts were stemmed [LOV 68] and word stem frequencies in documents (fraction of documents where the stem appears at least once) were computed for each stem. Then the Kullback-Leibler divergence between this word document frequency and the document frequency of the same word in a negative class of random blogs (pre-processed

in the same way) were calculated. By sorting the words according to their divergence and keeping the best ones, the core vocabulary of each community was extracted. Figures 3 presents word clouds of this core vocabulary for two communities. It appears that the words are in adequacy with the communities topics, which reinforce the fact that extracted blogs are topic relevant.

Politics (U.S.)

Food (U.S.)

Figure 3. *Word Clouds for the Politics (U.S.) and Food (U.S.) community. For each cloud, the first 50 words in descending order of their Kullback-Leibler divergence are extracted (between word document frequency in the community and in a negative class of 2000 random blogs, texts have been first preprocessed using stop lists and stemming). Word sizes are proportional to the word document frequencies in the community.*

6. Conclusion and future works

The experimental solution to the community extraction problem proposed in this paper seems relevant. It is quite important to note that a simple, greedy approach is able to extract communities with high precision. Such simplicity and scalability is of great importance when dealing with multi-billion nodes graphs as is the case with some real world examples like web or online social graphs.

From an experimental point of view, blog community extraction was performed using such a tool with success. However, more work is needed to better understand and evaluate the model.

Firstly, we could find other application domains were different community structures exist with different characteristics [FOR 09]. Applying the method to biological systems like protein interaction networks or online social networks and the like may provide clues about the robustness of the approach with respect to the different graphs structures one may find in such different contexts. We could also try to find a generic method to set the initial value of the parameters given these various application domains. Experimenting with such different structures may lead to generalize the algorithm so as to make it able to decide if there is only one or several community structures in the explored network. The only drawback of such an approach is the need to have annotated corpora with ground-truth communities.

Secondly, robustness of the methods to perturbations of the seeds set must be investigated. Comparing the communities extracted by the methods starting from different random samples may help to evaluated this point.

Finally, we could make use of the related field of graph generation algorithms. The purpose of those algorithms is to be able to generate realistic graphs with predefined output parameters like radius or clustering coefficient for instance. A quite comprehensive overview of this field may be found in [CHA 06]. In our case this kind of algorithm may be used to produce synthetic datasets for which we have by construction the ground truth communities. This may greatly help to experiment with our detection algorithm with a broad range of graph structures (by changing the generator algorithm) and variations (by changing the output parameters values).

7. Bibliographie

[AND 06] ANDERSEN R., LANG K., « Communities from seed sets », *Proceedings of the 15th International Conference on World Wide Web*, 2006.

[BAG 05] BAGROW J., BOLLT E., « A Local Method for Detecting Communities », *Phys Rev E Stat Nonlin Soft Matter Phys*, vol. 72, n° 4, 2005, page 046108.

[CHA 06] CHAKRABARTI D., FALOUTSOS C., « Graph mining : Laws, generators, and algorithms », *ACM Comput. Surv.*, vol. 38, n° 1, 2006, ACM Press.

[CLA 05] CLAUSET A., « Finding local community structure in networks. », , 2005.

[DAU 08] DAUDIN J., PICARD F., S. R., « A mixture model for random graph », *Statistics and computing*, vol. 18, 2008, p. 1–36.

[FOR 09] FORTUNATO S., « Community detection in graphs », rapport, 2009, Complex Networks Lagrange Laboratory, ISI Foundation, Viale S. Severo 65, 10133, Torino, ITALY.

[HOL 83] HOLLAND J., LASKEY K., LEINHARD S., « Stochastic block models : First steps », *Social Networks*, vol. 5, 1983, p. 109-137.

[LOV 68] LOVINS J., « Development of a stemming algorithm », *Mechanical Translation and Computational Linguistics*, vol. 11, 1968, p. 22–31.

[NEW 07] NEWMAN M., , LEICHT E., « Mixture models and exploratory analysis in networks », *PNAS*, vol. 104, 2007, p. 9564–9569.

[PAG 98] PAGE L., BRIN S., MOTWANI R., WINOGRAD T., « The PageRank Citation Ranking : Bringing Order to the Web », rapport, 1998, Stanford Digital Library Technologies Project.

[SNI 97] SNIJDERS T., NOWICKI K., « Estimation and prediction for stochastic blockstructures for graphs with latent block structure », *Journal of Classification*, vol. 14, 1997, p. 75–100.

[SOZ 10] SOZIO M., GIONIS A., « The community-search problem and how to plan a successful cocktail party », *Proceedings of the 16th ACM SIGKDD Conference On Knowledge Discovery and Data Mining (KDD)*, 2010, p. -.

[ZAN 08] ZANGHI H., AMBROISE C., MIELE V., « Fast online graph clustering via Erdos-Renyi mixture », *Pattern Recognition*, vol. 41, n° 12, 2008, p. 3592–3599.