COMSIM: A bipartite community detection algorithm using cycle and node's similarity

Raphael Tackx, Fabien Tarissan, and Jean-Loup Guillaume

Abstract This study proposes COMSIM, a new algorithm to detect communities in bipartite networks. This approach generates a partition of $\top$ nodes by relying on similarity between the nodes in terms of links towards $\bot$ nodes. In order to show the relevance of this approach, we implemented and tested the algorithm on 2 small datasets equipped with a ground-truth partition of the nodes. It turns out that, compared to 3 baseline algorithms used in the context of bipartite graph, COMSIM proposes the best communities. In addition, we tested the algorithm on a large scale network. Results show that COMSIM has good performances, close in time to Louvain. Besides, a qualitative investigation of the communities detected by COMSIM reveals that it proposes more balanced communities.

Key words: Community detection; bipartite graph; social network

1 Introduction

Many complex networks lend themselves to the use of graphs for analyzing and modelling their structure. Usually, vertices of the graph stand for the nodes of the network and the edges between vertices stand for (possible) interactions between nodes of the network. This approach has proven to be useful to identify non trivial properties of the structure of networks in very different contexts, ranging from computer science (the Internet, peer-to-peer networks, the web), to biology (protein-

Raphael Tackx
Sorbonne Universités, CNRS, LIP6, UMR 7606, e-mail: raphael.tackx@lip6.fr

Fabien Tarissan
Universités Paris-Saclay, CNRS, ISP, cole Normale Suprieure de Paris-Saclay e-mail: fabien.tarissan@ens-paris-saclay.fr

Jean-Loup Guillaume
University of La Rochelle, L3I e-mail: jean-loup.guillaume@univ-lr.fr

protein interaction networks, gene regulation networks), social science (friendship networks, collaboration networks), linguistics, economy, etc. [1, 2, 3, 4, 5, 6, 7].

This abstraction into graphs allows in return to study formally different aspects of its structure. In this context, one question that has driven a lot of attention in the past decade is the identification of *communities*, that is sets of nodes that constitute cohesive groups inside the networks. Although no formal definition has led to a consensus in the scientific community, one usually assumes that members of a community should be more connected to each other than with the rest of the network. To identify such communities, one can rely on human expertise but, in the context of large-scale networks, the question of identifying *automatically* such communities has led to the proposition of several community detection algorithms [8].

It is striking to notice that most algorithms have been designed for graphs containing only one set of nodes. Although useful, such a simple representation is not particularly close to the real structure of most of real networks. If one considers for instance an actor network that links actors performing in the same movies [1, 9] or co-authoring network that links authors publishing together [9, 3], one would rather relate actors to the movies they performed in and authors to their papers. This observation led the community to use *bipartite graphs* instead, i.e. graphs in which nodes can be divided into two disjoint sets, $\top$ (e.g. movies) and $\bot$ (e.g. actors), such that every link connects a node in $\top$ to one in $\bot$.

In that regard, only few community detection methods have been proposed to take into account this inherent bipartite complexity of real networks [10, 11, 12]. The usual approach consists instead in projecting first the bipartite structure over one set of nodes and then applying standard community detection techniques. Although interesting, it has been shown that this approach suffers from limitations [13].

Our contribution in this paper is to propose a new community detection algorithm dedicated to bipartite networks, namely ComSim (Section 2). This algorithms relies on a measure of similarity between nodes exploiting the bipartite ties. Then the algorithm looks for cycles of connections maximizing the similarity between the nodes, thus defining the core of the communities.

In order to validate our approach, we rely on real dataset and compare the communities generated by our algorithm to baseline methods (Section 3). Results show that on dataset equipped with ground-truth communities, the communities inferred by ComSim are the closest to the real ones. We also show that ComSim obtains good results when applied on large-scale networks as it produces communities that are more homogeneous than the other approaches tested in this study.

2 New community detection algorithm: ComSim

In this section, we formally presents our detection algorithm devoted to bipartite graphs. We first recall the necessary definitions (Section 2.1) before describing ComSim algorithm (Section 2.2) and presenting baseline algorithms to which we compare our approach (Section 2.3).

2.1 Notations

A bipartite graph is defined by a triple $\mathbb{B} = (\top, \bot, E_B)$ (see Figure 1 for instance) where $\top$ is the set of *top* nodes (e.g movies), $\bot$ the set of *bottom* nodes (e.g. actors), and $E_b \subseteq \top \times \bot$ the set of links between $\top$ and $\bot$ (that relates for instance the actors to the movies they perform in).

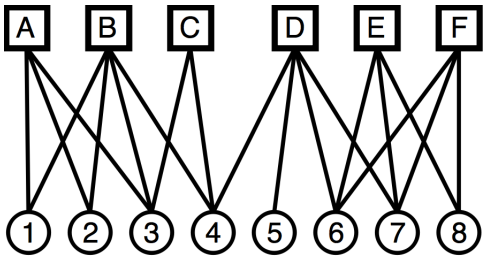


Fig. 1 Example of a bipartite graph $\mathbb{B} = (\top, \bot, E_B)$.

In addition, we define $N_\top(v) = \{x \in \bot | (v,x) \in E_b\}$ as the set of neighbors of a node $v \in \top$[1] and $N^2_\top(v) = N_\bot(N_\top(v))$ as the set of neighbors at distance 2 from v, that is the set of $\top$ nodes that share a $\bot$ node with v. Then we denote by $d_\top(v) = |N_\top(v)|$ the degree of a node $v \in \top$, $d_\top(v)$, and $d^2_\top(v) = |N^2_\top(v)|$ its number of neighbors at distance 2.

Compared to unipartite graphs, nodes in a bipartite graph are separated in two disjoint sets, and the links are always between a node in one set and a node in the other set. But it is natural to also investigate how nodes from the same set are in relation. This approach is usually captured by the notion of projection of a bipartite graph over one of its two sets.

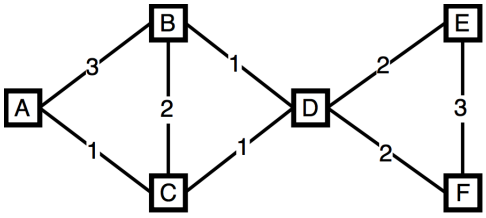


Fig. 2 Example of the weighted $\top$-projection of $\mathbb{B}$ using *common neighbors* as similarity function.

For instance, if one is interested in the $\top$-projection, one can study how $\top$ nodes connect according to their similarity measured by their links towards common $\bot$ nodes. Formally, such a similarity is captured by a *similarity function* θ. This allows

[1] We use a similar definition of $N_\bot(v)$ for $v \in \bot$.

to formally define the weighted projected graph $G_\top = (\top, \theta)$ where $\theta : \top \times \top \mapsto \mathbb{R}^+$. This graph thus indicates the strength of the relations between $\top$ nodes. The $\top$-projection of the bipartite graph in Figure 1 will therefore result in the graph depicted Figure 2.

Note that in the rest of the paper, we will use the standard *common neighbor* function $\theta(x,y) = |N_\top(x) \cap N_\top(y)|$. But this approach easily extends to other similarity functions such as *jaccard index* [14], *resource allocation* [15] or *adamic-adar coefficient* [16][2].

2.2 COMSIM *algorithm*

Given a bipartite graph $\mathbb{B} = (\top, \bot, E_B)$ and a similarity function θ such as *common neighbors*, COMSIM generate a partition of $\top$ in two steps. First, it identifies the *core communities*, that are groups of $\top$ nodes highly similar according to function θ.

As one can see in Algorithm 1 which details this first step, the algorithm generates a chain of nodes by following out-going links that have the highest weight according to θ. When this chain reaches a node already considered, it means that a cycle has been detected in the chain. This cycle then forms the core of a future community.

On the toy example of Figure 1, it would result in detecting that nodes A and B form the core of a community, as well as E and F. This is in accordance to Figure 2 which shows that A and B, as well as E and F have the highest weighted links. The other nodes (C and D) are left in the remaining set K.

The second phase of the algorithm then tries to position the remaining nodes of K in the existing communities by maximizing the similarity between these nodes and all the nodes of the core communities.

As described in Algorithm 2, the second step considers all remaining nodes that are not part of the partition after the first step. For each node x, it identifies the communities that have at least one link with x. The algorithm then chooses the community that maximizes the sum of similarities between x and all the nodes of the community.

On the toy example of Figure 1, and independently of the order in which nodes C and D are considered during step 2, it would result in affecting node C to the community $A-B$ (the sum of similarities is 3) and node D to community $E-F$ (the sum of similarities is 4).

It is worth noticing that because several links can have a similar weight, the two steps might face several equal options. In that case, the algorithm selects one option uniformly at random among all possible ones. For this reason, the algorithm is undeterministic and several runs might end up with different partitions.

[2] Depending on the similarity function used, the projection might result in a directed weighted graph if θ is not symmetric.

Algorithm 1: COMSIM- first step

Data: a bipartite graph $\mathbb{B} = (\top, \bot, E_B)$, a similarity function θ
Result: return a partition P of $\top$ nodes and a set K of remaining nodes (for the **second step**)
$P := \emptyset$ // the partition set
$T := \top$ // the set of nodes to be considered
$x := rand\_and\_remove(T)$ // random node
$V := \emptyset$ // set of nodes currently considered
$K := \emptyset$ // set of remaining nodes
while $T \neq \emptyset$ **do**
  /* finds a neighbor $y \in N^2_\top(x)$ of x maximizing $\theta(x,y)$ */
  $y := \text{argmax}_{y \in N^2_\top(x)} \theta(x,y)$
  if $y \in V$ **then**
    $C := cycle(V,y,x)$ // extract the detected cycle from y to x in V
    $P.add(C)$
    $K := K \cup (V - C)$ // stores nodes not in the cycle C
    $V := \emptyset$
    $x := rand\_and\_remove(T)$
  else
    if $y \in T$ **then**
      $V := V \cup \{y\}$
      $x := y$
      $T := T - \{y\}$
    else
      /* y is already part of an element of P, visited nodes are stored */
      $K := K \cup V$
      $V := \emptyset$
      $x := rand\_and\_remove(T)$
return P and K

Algorithm 2: COMSIM- second step

Data: a bipartite graph $\mathbb{B} = (\top, \bot, E_B)$; a partition P; a set K of remaining nodes (from **first step**), a similarity function θ.
Result: return a partition P' of $\top$ nodes and unsatisfied nodes R
$R := \emptyset$ // Remaining nodes
$P' := P$
foreach $x \in K$ **do**
  $P_x := com\_neigh(x,P)$ // Find all neighbor communities of x
  if $P_x = \emptyset$ **then**
    $R := R \cup \{x\}$
  else
    $C := \text{argmax}_{C_x \in P_x} \sum_{y \in C_x} \theta(x,y)$
    Add x into the partition C of P'
return P' and R

2.3 Standard approaches

In order to evaluate the relevance of COMSIM, we will compare the detected communities with the ones of the three baseline detection algorithms described below.

Louvain: Louvain algorithm [17] is a greedy algorithm that optimizes a quality function in order to extract communities from large unipartite networks. It is commonly used with modularity [18] which measures the density of the communities compared to their expected density if the links were randomly distributed over the network.
In order to evaluate Louvain's performance, and for fair comparison, we first project the bipartite graph over the $\top$ nodes, generating a weighted graph according to the similarity function θ (common neighbor in our case). Then we apply Louvain on the weighted graph.

Infomap: Infomap is a recursive algorithm, similar to Louvain, where each node is moved to a neighboring community if this modification minimizes the length of the map equation [19]. Infomap can account for the bipartite structure and we use this feature to generate a partition of $\top$ nodes only.

LPBRIM: LPBRIM [10] is a community detection algorithm that optimizes the bimodularity [20] which is an extension of the modularity for bipartite graphs. It relies on BRIM algorithm (Bipartite, Recursively Induced Modules) and uses a label propagation procedure.
Because LPBRIM provides a partition of the complete bipartite networks – communities are composed of $\top$ and $\bot$ nodes –, we adapt the algorithm and define a community by keeping only the $\top$ nodes of the partitions. This allows a fair comparison in the evaluation process.

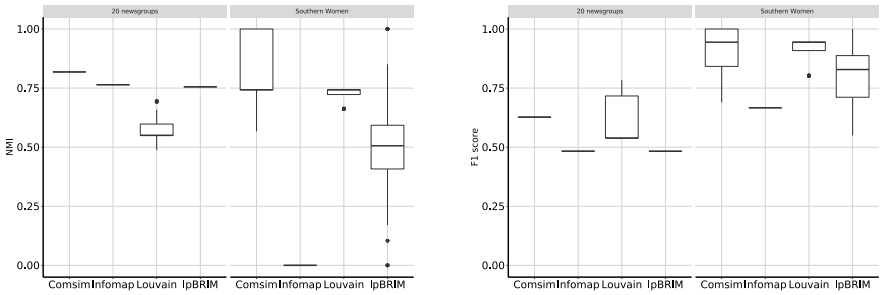
3 Evaluation of COMSIM

This section is devoted to assess the relevance of the proposed method. We start by investigating how the different algorithms behave on two small networks equipped with existing communities (Section 3.1) before showing how COMSIM scales up when dealing with large-scale networks (Section 3.2).

3.1 On dataset with ground-truth communities

We first apply our algorithm to two networks which are small but are provided with a notion of ground-truth communities that we use as a reference to compare the four algorithms.

Southern women [21] is a network depicting the participation of 18 women to 14 events in the United States observed during a nine-months period in 1930. Although small, this dataset is very interesting since it has been extensively studied by social scientists to understand how social groups form and evolve (see [22, 23] for instance). In this study, we use the partition found in the literature as the ground-truth communities to which we compare the four algorithms.



(a) NMI (b) F1-score

Fig. 3 Evaluation of the quality of the partitions detected by the algorithms on *20 newsgroups* and *Southern Women*.

20 newsgroups [24] is a record of approximately 50 000 posts submitted by 30 000 users (*bot*) over 20 groups of discussion ($\top$).

Figure 3 presents the results of the comparison between ComSim and the three baseline algorithms for bipartite community detection described in Section 2.3. All algorithms were applied on *Southern Women* and *20 newsgroups* dataset 100 times. The box plots provides the maximal, minimal and average values.

Since we have a ground-truth partition for the dataset, we use first the usual Normalized Mutual Information (NMI, see [25] for instance) to compute how far the detected partitions are from the ground-truth ones. Figure 3a reveals that for both datasets ComSim is the algorithm that proposes the best partition in average. One can also notice that Infomap and lpBRIM generate good partitions for *20 newsgroups* and Louvain good partitions for *Southern Women*.

It is interesting to notice that Infomap completely fails to detect the expected communities for *Southern Women*. Manual investigation revealed that all women of *Southern Women* are actually gathered in a single community, which is well captured by NMI (NMI$=0$). On the opposite, each node of *20 newsgroups* is positioned in a different community, which is completely overestimated by the NMI (NMI$=0.7643$).

In order to provide a second point of view, we also use the F1-score, a classical metric to evaluate the performance of prediction algorithms (see [26] for an example of F1-score used in the context of community detection issues). Figure 3b shows again that ComSim is the best community detection algorithm for both datasets in average. Interestingly, for this metric, Louvain seems to propose good partitions in average and for both dataset.

All in all, it seems that ComSim proposes coherent communities when compared to ground-truth partitions of bipartite networks. The next section intends to investigate how the algorithm behaves on a large scale network.

3.2 On large-scale networks

In order to test the performance of our algorithm both in terms of efficiency and quality, we rely here on a large dataset extracted from the Internet Movie Database (*IMDb*). This dataset [27] presents a bipartite network composed of 118 258 actors ($\bot$) who played in 122 131 movies ($\top$) between 1980 and 2010[3].

	Southern women	20 newsgroups	IMDb
$\|\top\|/\|\bot\|$/links	18/14/89	20/30K/42K	122K/118K/531K
ComSim	**1.7 ms** / 11.5 MB	1.1 s / 30 MB	33.5 s / 591.6 MB
Infomap	13 ms / 10.7 MB	951 ms / **6.3 MB**	100s / 374 MB
Louvain	11 ms / 6.5 MB	**86 ms** / 10.1 MB	**21 s / 43 MB**
lpBRIM	6.7 s / **6.1 MB**	74.2 s / 59.7MB	-/-

Table 1 Performances in terms of execution time and memory peak for the four algorithms.

Table 1 presents the performances in terms of execution time and memory peak for the four algorithms on the three datasets. This shows that Louvain remains the most efficient algorithm in terms both of time and memory, revealing to be slower only on the smallest dataset.

However, it should be highlighted here that the performances of Louvain shown in Table 1 have been recorded *after* the $\top$-projection. This means that part of the computation load related to the θ function has been avoided, which is not the case for the other algorithms. It thus mechanically favour the Louvain approach.

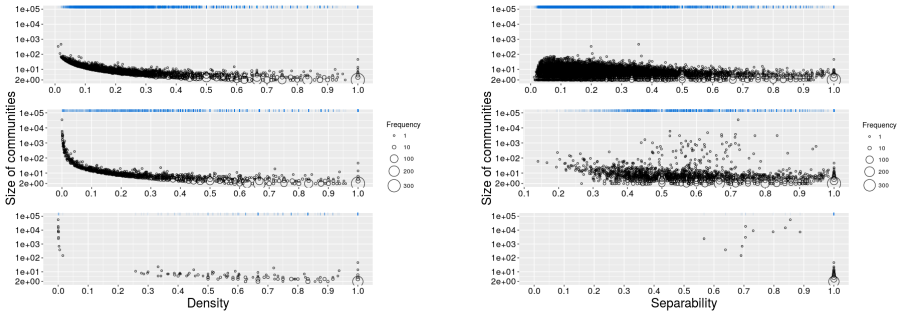
To that regard, it is worth noticing that our algorithm presents good results. On *IMDb* in particular, ComSim is only slightly slower than Louvain and three times faster than Infomap.

The results above show that our algorithm can scale up to large networks but that it provides no insight on the quality of the detected communities. In contrast to the previous section where we had ground-truth knowledge of the good partitions, no study conducted on the *IMDb* dataset proposes an objective and external partition of the nodes. It is thus impossible to use here either NMI or F1-score to compare the three remaining algorithms [4].

In order to assess the quality of the proposed communities, we follow instead the proposition made in [28] where the authors introduce two goodness functions in an attempt to quantify how relevant a community is regarding two properties that we adapted for the case of bipartite graphs: the *Density* (or *Internal Density*) and *Separability*.

[3] For an homogeneous analysis, we removed all TV shows and documentaries and kept only the 7 first actors listed in the casting.

[4] Since lpBRIM does not scale up to the size of *IMDb*, we avoid mentioning this approach in the rest of the study.



(a) Internal Density (b) Separability

Fig. 4 Scatter plot displaying the relation between properties of the communities and their size for COMSIM (top), Louvain (middle) and Infomap (bottom) on *IMDb*.

More formally, let P be a partition of $\top$ nodes. Given a community $C_i \in P$, we denote by $\overline{C_i} = \bigcup_{x_i \in C_i} N(x_i)$ the set of $\bot$ nodes induced by the neighborhood of C_i. Let m_{C_i} be the number of edges between C_i and $\overline{C_i}$ (*internal* number of edges) and $m_{\overline{C_i}}$ the number of edges between $\overline{C_i}$ and C_j (*external* number of edges), where $j \neq i$. Then we define the internal density and the separability as follow:

- Internal Density of community C_i: $\frac{m_{C_i}}{|C_i|*|\overline{C_i}|}$
- Separability of community C_i: $\frac{m_{C_i}}{m_{C_i}+m_{\overline{C_i}}}$

Those two indicators allow to evaluate how coherent a community is regarding internal and external edges. Figure 4 presents the distribution of the Internal Density (Figure 4a) and the Separability (Figure 4b) of communities according to their size and for the three algorithms (COMSIM top, Louvain middle and Infomap bottom).

This shows that Infomap mostly fails to detect coherent communities. Indeed, although most of the communities have very high values for both properties, it concerns mostly very little communities composed of few nodes. But for large communities, the indicators drop. This is particularly obvious on Figure 4a. To that regard, it is striking to notice that the largest community detected by Infomap gathers more than 46% of the nodes and the 7 largest communities involve more than 96% of the nodes. The same observation can be made for Louvain, although to a lesser extent. The largest community involves 29% (48% for the 7 largest communities).

Compared to Louvain and Infomap, COMSIM proposes more balanced communities in terms of size. The largest community is rather small (only 1% of the nodes) while keeping a profile of density close to the ones of Louvain (see Figure 4a). Regarding the separability however, there is a slight shift towards low values compared to Louvain, which indicates that the quality of the partitions could be improved for this property.

All in all, an although more experiments should be made in order to complete the comparison, we claim that this study is a first step establishing the relevance

of the proposed approach, both in terms of efficiency and quality of the detected communities.

4 Conclusions

In this study we proposed ComSim, a new algorithm to detect community in bipartite networks. This approach generates a partition of the $\top$ nodes by relying on similarity between the nodes in terms of connections towards $\bot$ nodes. To do so, it tries to find and maximize cycles of relations between $\top$ nodes. This defines the core of the communities which are enriched with new nodes during a second phase of the algorithm.

We implemented and applied this algorithm on 3 datasets and compared the generated partitions with the ones proposed by three baseline algorithms used on bipartite graphs. The empirical results showed that, on small networks for which we had a ground-truth knowledge of the good partition, ComSim is the algorithm that generates the best communities.

In addition, ComSim proved to scale up with a time complexity close to Louvain. Investigating qualitatively the partitions, we showed that the communities generated by ComSim are more balanced in terms of size, while keeping quality indicators reasonable and comparable to the ones proposed by Louvain for instance.

It is worth noticing that other algorithms could have been used for the comparison. For instance, biSBM [11] or SCD [12] are relevant, although not completely adapted to this context. The former requires to provide the number of expected communities, while the latter proposes overlapping communities.

We claim that this study establishes the relevance of the approach and we let more in-depth study for future work.

Acknowledgements

This work is funded in part by the European Commission H2020 FETPROACT 2016-2017 program under grant 732942 (ODYCCEUS), by the ANR (French National Agency of Research) under grants ANR-15-CE38-0001 (AlgoDiv) and ANR-13-CORD-0017-01 (CODDDE), by the French program ”PIA - Usages, services et contenus innovants” under grant O18062-44430 (REQUEST), and by the Ile-de-France program FUI21 under grant 16010629 (iTRAC).

References

1. Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world' networks. *nature*, 393(6684):440–442, 1998.
2. Ramon Ferrer i Cancho and Richard V Solé. The small world of human language. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 268(1482):2261–2265, 2001.
3. Mark EJ Newman, Duncan J Watts, and Steven H Strogatz. Random graph models of social networks. *Proceedings of the National Academy of Sciences of the United States of America*, 99(Suppl 1):2566–2572, 2002.
4. Stefano Battiston and Michele Catanzaro. Statistical properties of corporate board and director networks. *The European Physical Journal B-Condensed Matter and Complex Systems*, 38(2):345–352, 2004.
5. Fabrice Le Fessant, Sidath Handurukande, A-M Kermarrec, and Laurent Massoulié. Clustering in peer-to-peer file sharing workloads. In *Peer-to-Peer Systems III*, pages 217–226. Springer, 2005.
6. Christophe Prieur, Dominique Cardon, Jean-Samuel Beuscart, Nicolas Pissard, and Pascal Pons. The stength of weak cooperation: A case study on flickr. *arXiv preprint arXiv:0802.2317*, 2008.
7. Yong-Yeol Ahn, Sebastian E Ahnert, James P Bagrow, and Albert-László Barabási. Flavor network and the principles of food pairing. *Scientific reports*, 1, 2011.
8. Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3):75–174, 2010.
9. Mark EJ Newman, Steven H Strogatz, and Duncan J Watts. Random graphs with arbitrary degree distributions. *Physics Reviews E*, 64, 2001.
10. Xin Liu and Tsuyoshi Murata. Community detection in large-scale bipartite networks. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, WI-IAT '09, pages 50–57, Washington, DC, USA, 2009. IEEE Computer Society.
11. Daniel B Larremore, Aaron Clauset, and Abigail Z Jacobs. Efficiently inferring community structure in bipartite networks. *Physical Review E*, 90(1):012805, 2014.
12. Arnau Prat-Pérez, David Dominguez-Sal, and Josep-Lluis Larriba-Pey. High quality, scalable and parallel community detection for large real graphs. In *Proceedings of the 23rd international conference on World wide web*, pages 225–236. ACM, 2014.
13. Sune Lehmann, Martin Schwartz, and Lars Kai Hansen. Biclique communities. *Physical Review E*, 78(1):016108, 2008.
14. Paul Jaccard. *Le coefficient generique et le coefficient de communaute dans la flore marocaine*. Impr. Commerciale, 1926.
15. Tao Zhou, Linyuan Lü, and Yi-Cheng Zhang. Predicting missing links via local information. *The European Physical Journal B-Condensed Matter and Complex Systems*, 71(4):623–630, 2009.
16. Lada A Adamic and Eytan Adar. Friends and neighbors on the web. *Social networks*, 25(3):211–230, 2003.
17. Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.
18. Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582, 2006.
19. Martin Rosvall, Daniel Axelsson, and Carl T Bergstrom. The map equation. *The European Physical Journal-Special Topics*, 178(1):13–23, 2009.
20. M. J. Barber. Modularity and community detection in bipartite networks. *Physical Review E*, 76(6):066102, December 2007.
21. Allison Davis, Burleigh B. Gardner, and Mary R. Gardner. *Deep South; a Social Anthropological Study of Caste and Class*. The University of Chicago Press, Chicago, 1941.
22. Elna C Green. *Southern strategies: Southern women and the woman suffrage question*. Univ of North Carolina Press, 1997.

23. Linton C Freeman. *Finding social groups: A meta-analysis of the southern women data*. na, 2003.
24. Ken Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 331–339, 1995.
25. Andrea Lancichinetti, Santo Fortunato, and János Kertész. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3):033015, 2009.
26. Jaewon Yang and Jure Leskovec. Overlapping community detection at scale: a nonnegative matrix factorization approach. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 587–596. ACM, 2013.
27. Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.
28. Jaewon Yang and Jure Leskovec. Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems*, 42(1):181–213, 2015.