Learning a proximity measure to complete a community

Maximilien Danisch, Jean-Loup Guillaume, Bénédicte Le Grand

Learning a Proximity Measure to Complete a Community

Maximilien Danisch*†, Jean-Loup Guillaume*† and Bénédicte Le Grand‡
* Sorbonne Universités, UPMC Univ Paris 06, UMR 7606, LIP6, F-75005 Paris, France.
† CNRS, UMR 7606, LIP6, F-75005 Paris, France.
‡ CRI, Université Paris 1 Panthéon-Sorbonne, 90 rue de Tolbiac, 75013 Paris, France.

***Abstract*—In large-scale online complex networks (Wikipedia, Facebook, Twitter, etc.) finding nodes related to a specific topic is a strategic research subject. This article focuses on two central notions in this context: *c*ommunities (groups of highly connected nodes) and *p*roximity measures (indicating whether nodes are topologically close). We propose a parameterized proximity measure which, given a set of nodes belonging to a community, learns the optimal parameters and identifies the other nodes of this community, called *m*ulti-ego-centered community as it is centered on a set of nodes. We validate our results on a large dataset of categorized Wikipedia pages and on benchmarks, we also show that our approach performs better than existing ones. Our main contributions are (i) a new ergonomic parametrized proximity measure, (ii) the automatic tuning of the proximity's parameters and (iii) the unsupervised detection of community boundaries.**

I. Introduction

The growth of online social networks has created many new opportunities to establish new contacts and to share information and knowledge. In addition, the huge size of these dynamic networks also raises new research challenges: which Facebook contacts should a specific message be sent to, considering that some members have hundreds of "friends"? Which Wikipedia pages should be read in priority to learn about a specific domain? These two applications refer to two strongly related topics in graph analysis and data mining: *community detection* and *proximity measures*.

Most large-scale community detection algorithms in the literature assume that nodes belong to a single community. However, if we could, for instance, detect the pages which are similar to the "social network" Wikipedia page, we would obtain a mix of pages from different subjects, including sociology, graph theory and many others. This result stems from the fact that in most real-life networks, nodes belong to several communities and not to a single one. This must be taken into account when designing community detection algorithms or ranking algorithms.

In this paper, we overcome this limit by proposing a solution to rank nodes with regard to their proximity to a given set of nodes. For instance, the Wikipedia pages close to "social network" and "epidemiology" define a more specific topic than the pages close to the sole "social network" page. A small set of reference nodes (typically two well-chosen ones) is generally enough to define a single community (as shown in previous work [1] and [2]) that we call *multi-ego-centered community* as it is based on multiple nodes and is a generalization of the notion of ego-centered community [3], [4]

In order to evaluate the proximity between nodes, we introduce a new parameterized proximity measure which extends existing ones with further insights of the expected overlapping structure of communities. In particular, since a community is generally defined as a strongly connected set of nodes, both a small topological distance and a high number of common neighbors are expected for two nodes within a community. In addition, hubs (very connected nodes) are very central in networks and are, therefore, expected to belong to several communities. The different parameters required to formalize these notions, can be learned in a semi-supervised way thanks to the reference nodes used as a learning set. Therefore, the proximity measure does not rely on manually tuned parameters: they can be computed automatically.

The next step is to compute communities from the ranking obtained through proximity values. This problem is similar to the one-class classification problem (also called unary classification or data domain description) in data-mining [5]. In the context of Wikipedia, one-class classification tries to identify pages from a specific category amongst all pages, given a selection of pages from the category. To the best of our knowledge there is no one-class classification method designed for graph-based data and the method proposed here is a step towards this goal.

This paper is organized as follows: we first present the state of the art on community detection and proximity measures and explain the limits of existing approaches in Section II. In Section III, we describe our proximity measure and its parameters and validate it on toy graphs. We then present some results on a dataset of categorized Wikipedia pages in Section IV, together with some results on the benchmark of [6]. We show that our approach provides a ranking of relevant nodes with regard to a set of reference nodes, and also identifies the corresponding multi-ego-centered communities. In addition, we show that similar frameworks without the learning step achieve much poorer results. We finally conclude and present some perspectives, both from the ranking and from the community detection points of view.

II. Related work

A. Ego-centered community detection

While it is known that the global community structure of a complex network is highly overlapping (for instance, in a social network each node belongs to many communities: family, colleagues, friends...), the community structure has often been considered as a partition, for simplicity reasons [7]. On the other hand, studying globally an overlapping community structure is very hard due to the lack of a proper definition of the notion itself and, also, because it can lead to a

very high number of communities, which causes problems of computation and validation [8], [9], [10]. Because of the too simplistic model of partition and the complexity of overlapping communities, and also because sometimes the full network is not accessible, some works have focused on local community: find the communit(y)ies of a given node (ego-centered) or a set of nodes (multi-ego-centered).

In the state of the art of disjoint communities, it is classically assumed that a community is a densely connected group of nodes, poorly connected to the outside. Many work in the overlapping context are based on the optimization of a quality function and keep the same idea. However, in the context of overlapping communities, considering the outside is no longer relevant: a node x does not really belong "less" to a community A if it also belongs to a community B. Therefore, there is no reason to exclude x from A although there may be many outgoing links from node x to nodes in B.

We can thus simply state that a community is "a group of nodes highly connected together", which does not make any assumption on outgoing edges. A first approach in this direction is the *Cohesion* quality function, developed in [4], which evaluates the strength of a community by comparing the number of triangles within the community to the number of triangles pointing out of the community, i.e. with two nodes inside the community and one outside. This function allows nodes inside a community to have outgoing links, and only redundant outgoing links lower the value of Cohesion: if many nodes in a community A are connected to a node x outside of A, there will be many outgoing triangles (since nodes in A are highly connected) and node x might be included in A to increase Cohesion.

Another problem of quality functions is related to their optimization, which is often carried out in a greedy way, mainly for simplicity and time efficiency reasons: starting with a set of nodes (or a single one) in the community, neighbors are added one by one as long as the quality increases. However, the landscape of optimization has many local minima where greedy algorithms get stuck and such algorithms tend to favor small communities. An exception is detailed in [11]: the quality function is defined as the minimum degree of the nodes in the subgraph induced by the community, which therefore does not take the outside into account. For this specific quality function, an optimal, yet greedy, algorithm exists. But, conversely, this greedy algorithms often leads to very large communities.

Other solutions, from which our work is inspired, are [12] and [13] where a proximity measure approach is used rather than a quality function approach. They compute the proximity of every node in the graph to an input set of nodes, then they try to find the most relevant connected subgraph of size k (parameter), i.e., the set of connected nodes which are globally the closest to the input set.

B. Proximity measures

A full survey has been dedicated to the presentation and comparison of proximity measures [14]. Therefore, we focus in this paper on those that highlight some issues in community detection or are related to the proximity measure we will introduce below.

A commonly used proximity measure based on random walks is the personalized page-rank [15] (PPR). The problem with PPR is that it sometimes leads to nonintuitive results, as shown in [16]. Another problem is that hubs drain all random walkers and are thus close to any other node, while we may not wish them so. It is also possible to improve PPR by computing random walks using a reweighted transition matrix in order to handicap hubs as in [16] and [13]. While hubs are no longer a problem, the computation of these random walks for several sets of parameters is however slow. Other proximity measures based on random walks, like hitting time or commuting time[1] have similar counterintuitive behaviors [17].

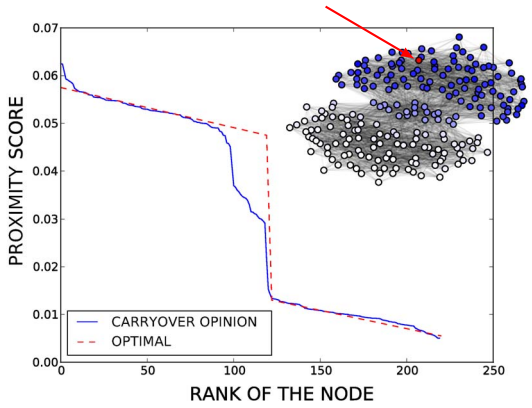


Fig. 1. Proximity score in decreasing order for the carryover opinion of the node pointed by an arrow. The graph is made of two overlapping communities of 110 nodes generated with a uniform edge probability of 0.3, overlapping on 20 nodes. The higher the proximity score, the darker the node.

Another proximity measure, the carryover opinion introduced in [1] acts like quality functions of the type $\frac{l_i}{l_i+l_o}$ used in community detection: if a node has links going out from the community of the node of interest, then its score is lower. This bias is illustrated in Figure 1 featuring two overlapping communities A and B: when computed for a node x (pointed by an arrow in Figure 1) in $A \setminus B$, the nodes belonging to $A \setminus B$ are closer to x than the nodes in the overlapping part $A \cap B$. If a community is only defined relatively to its inner density, then the proximity function should give something close to the optimal curve in Figure 1.

Various proximity measures have been introduced that address these issues, among which the Katz index [18]. This measure takes all paths into account and is defined as

$$S_{\text{Katz}}(i,j) = \sum_{l=1}^{\infty} \beta^l N_l^P(i,j) \ , \qquad (1)$$

where $N_l^P(i,j)$ is the number of paths of length l between i and j and β is a damping factor controlling the contribution of different lengths to the total proximity. A simpler version, the Local Path Index [19], restricts the computation to paths of length 2 and 3 only. It gives results close to the Katz index with a lower complexity for sparse graphs.

III. Parametrized proximity

A. Design of the proximity measure

Before introducing our proximity measure, we ask two questions to understand what needs to be taken into account.

[1] Hitting time H_{ij} is the expected time it takes for a random walk to go from i to j. Commute time is the expected round-trip time: $C_{ij} = H_{ij} + H_{ji}$.

1- Let (n_1, n_2) be a pair of connected nodes and (n_3, n_4) be a pair of non-connected nodes but sharing N neighbors, the rest of the network being similar for the two pairs. Which is the closest pair: (n_1, n_2) or (n_3, n_4)? This question can be generalized to paths of length 2 or more: are two non-connected nodes with a common neighbors closer to each other than two non-connected nodes with no common neighbors but with N paths of length 3 between them?
2- Let n_1 be a random node; n_2 a node highly connected to n_1 and its neighborhood but poorly connected to the rest of the graph; and n_3 a node highly connected to n_1, its neighborhood and also to the rest of the graph. n_3 is therefore a node with high degree (a hub). Which node is closer to n_1: n_2 or n_3?
The answers depend on N (first question) and node degrees (second question) and a knowledge of the network. This implies that proximity measures need to be parameterized.
In order to address these two questions, and the problems related to the various proximity measures reviewed in the previous section (the shape is also important as it needs to allow a fast learning of the parameters) we now propose a new proximity measure. It is based on the Katz index, but is more general and can be used on large graphs. Particularly it takes into account the point two concerning high degree nodes and is based on a combination of the number of *non-backtracking paths* (NBP) [2] of various lengths between nodes, which are easier to compute than the number of simple paths. The proximity of node j to node i is given by:

$$\text{Prox}_{\alpha,\beta,\lambda,\delta}(i,j) = \sum_{l=0}^{\lambda} \gamma_{l,d_j} N_l^{\text{NBP}}(i,j) \qquad (2)$$

where d_j is the degree of j, $N_l^{\text{NBP}}(i,j)$ the number of NBP of length l between node i and j, and $\gamma_{l,d_j} = \alpha^l / max(d_j, \delta)^\beta$.
The trade-off between topological distance and paths redundancy is controlled by the parameter α. The length of the various NBP between two nodes i and j must be taken into account and a coefficient is needed to account for the contribution of each of them. The shorter the length, the higher the impact. We therefore choose to multiply each NBP of length l by a coefficient α^l, where $\alpha \in]0, 1[$. The choice of this exponential handicap is related to the exponential growth of the number of NBP as a function of their length, as shown in Figure 2. To quicken the computation we added an upper cutoff for the length of NBP, reflected below in the λ parameter. This upper cutoff has little impact, as NBP longer than a given value are not relevant for community detection, see Figure 2. Indeed, whereas short NBP mostly end up at nodes of the community of the node of interest, long NBP may end up outside this community.
The need to take into account that high-degree nodes appear close to the others is controlled by the parameter β. As shown in Figure 2, the number of NBP grows, on average, linearly with the degree of the target node and this seems unrelated to the length of the NBP. Therefore, the two natural extreme choices would be to set β to 0 or to 1. The former option consists in giving no handicap, but this favors a lot high degree nodes which, therefore, will always be the closest nodes. Conversely, the latter will penalize all high degree nodes which will lose their natural centrality. We chose an intermediate solution: a polynomial handicap on the degree, represented by the β parameter.
Finally, we found empirically that nodes with a very small degree are particularly favored (this happens when a node of very low degree is adjacent to a hub which is adjacent to the node of interest). This problem is easily fixed using a lower cutoff δ: if a node's degree is lower than δ then it is penalized as if it had degree δ.
Note that this proximity measure is not symmetric ($\text{Prox}(i,j) \neq \text{Prox}(j,i)$) in general. However, this is not an issue since we use this proximity measure to compute the proximity of all nodes to a given node i and all nodes are treated the same way with regards to i.

B. Learning the parameters

When the input set of reference nodes is large, the four parameters of equation 2 can be easily learned and, on the contrary, when the input set of reference nodes is small, learning is much harder and a parameter-free proximity measure (such as the carryover opinion) is interesting.
We use the proximity $\text{Prox}_{\alpha,\beta,\lambda,\delta}$ defined above and learn the four parameters. This optimization could be conducted using a proper cost function and an efficient quasi-newton method like LBFGS [20] to optimize the two continuous parameters. However, since there are only two continuous parameters and two discrete parameters, we follow a brute force optimization to find the best 4-tuple $(\alpha, \beta, \lambda, \delta)$: given a set of potential 4-tuples and a node of interest i in the set of reference nodes, we compute the proximity of i to all nodes of the graph for each 4-tuple. Then, we select the 4-tuple maximizing the area under the ROC curve (AUC), i.e. which best ranks the other nodes of the set of reference nodes. We then repeat this procedure for all other nodes in the reference set and finally combine the different proximity scores for each node.
The complexity of the learning phase can be computed exactly. Given the number of NBP of any length up to l from a node of interest (computation taking time $\mathcal{O}(l(n+m))$, where n is the number of nodes and m the number of edges), it then takes only $\mathcal{O}(ln)$ to compute $\text{Prox}_{\alpha,\beta,\lambda,\delta}$ for all nodes in the graph for a given 4-tuple of parameters $(\alpha, \beta, \lambda, \delta)$. The algorithm is therefore essentially linear as long as the number of 4-tuples remains small, thus allowing a fast optimization of the parameters.

IV. Results and validation

Tests on toy graphs shows that our proximity measure addresses issues of classical ranking and community detection techniques (for instance with the right set of parameters, we can obtained the optimal curve shown on Figure 1). We now use it on a real dataset of more than 4 million Wikipedia pages together with more than 253 million hyperlinks between them[3] as of July, 2nd 2012. Furthermore, these pages are organized into user-annotated categories. However, a Wikipedia category is not always a community, i.e. a "group of nodes highly connected", as it may contain poorly connected nodes. Nonetheless, most categories are actual communities and can thus be discovered or completed using our framework. In the

[2] The number of paths forbidding cycles of lengths two (backward hops), but allowing longer cycles. The number of NBP between a node and each node of the graph is computable through linear algebra calculation in time $O(l(n+m))$, where n is the number of nodes and m the number of edges. This is detailed in the Appendix.

[3] We considered hyperlinks as undirected in this work.

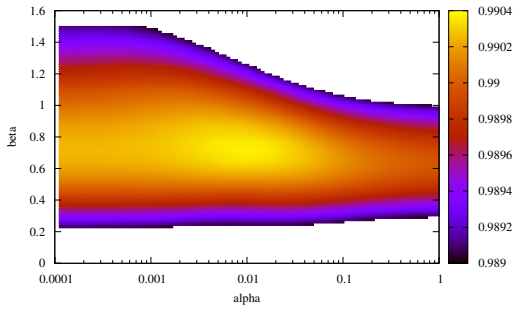


Fig. 3. Area under the ROC curve as a function of α and β for for $\lambda = 3$ and $\delta = 5$ for the node "Incidence list" in the category "Graph theory". A single maximum is obtained for $\alpha = 0.0105$ and $\beta = 0.705$, the area under the ROC curve is then 0.9904.

following we use the pages in the "Graph theory" category and in its direct subcategories. This leads to a set of 542 pages (or nodes) that we randomly divide into a training set and a test set of 271 nodes each.

A. Preliminary tests

In order to fasten the brute force optimization detailed previously, we carried out preliminary tests for various values of the parameters for each node in the training set. We recall the different parameters used in our proximity measure: α^l controls the contribution of paths of length l, λ is an upper limit on path length, d^β penalizes target nodes of degree d and δ penalizes target nodes of degree inferior to δ.
We tested a wide range of values: $\alpha \in [0, 1]$, $\beta \in [0, 1.5]$, $\delta \in [\![0, 100]\!]$ and $\lambda \in [\![2, 5]\!]$. For each set of parameters, we computed the AUC. In all cases the optimal λ value was equal to 3, the optimal δ value was around 5, the optimal value of α varied significantly between 0.002 and 1 and the one of β from 0.6 and 0.9. We thus fixed the values $\lambda = 3$ and $\delta = 5$ and optimized α over the set $\{0.001^{i/100} \mid i \in [\![0, 100]\!]\}$ and β over the set $\{0.5 + 0.005i \mid i \in [\![0, 100]\!]\}$.
The fact that paths of length longer than 3 are not very relevant can be further understood in Figure 2: there are much more short paths ending within the category than short paths ending outside the category. However, longer paths have equal chances to end up in the category or outside the category and, therefore, do not carry much information.

B. Learning individual scores

Using the previous results we can now learn the parameters of the proximity measure for all reference nodes, i.e. which belong to the training set of the "Graph Theory" category. For instance, the optimization computed for the node "Incidence list" (from the training set) gives the AUC heatmap as a function of α and β in Figure 3. As we can see, there is here only one maximum obtained for $\alpha = 0.0105$ and $\beta = 0.705$. Figure 4 shows the proximity values ranked in decreasing order for three nodes: one giving a bad AUC ("Global shipping network"), an average one ("Resistance distance") and a very good one ("Multiple edges"). The AUC and the fraction of nodes from the training set in the top 1000 nodes (called precision at top 1000) are presented in Table I.
The main observation is that if a node has a very good

Rank	Page	AUC	P1000
1	Multiple edges	0.9997	0.804
2	Graph (mathematics)	0.9996	0.646
3	Random regular graph	0.9995	0.520
144	Resistance distance	0.9971	0.4833
167	Four color theorem	0.9962	0.4944
198	Fan Chung	0.9939	0.3469
269	Agent Network Topology	0.9449	0.007
270	Ruth Aaronson Bari	0.9250	0.0258
271	Global shipping network	0.9073	0.0775

TABLE I. THREE NODES GIVING THE BEST, AVERAGE AND WORST RANKING, WITH AUC AND PRECISION AT TOP 1000.

AUC then it generally allows establishing a proper distinction between nodes in the category and nodes outside. For instance, Figure 4(c) clearly shows a sharp decrease around rank 1000 which indicates that approximately 1000 nodes are very close to the node "Multiple edges" and the others are not. This is a good way to identify a community of "Multiple edges" which is also very similar to the "Graph theory" category. On the contrary, Figure 4(a) shows no such behavior and nodes from the "Graph theory" category are globally poorly ranked, which is confirmed by low AUC. Table I also shows the pages with rankings, some average rankings and the worst ones among the 271 nodes of the training set. As we can see, the ones with the best rankings are very related to "Graph theory": they are central within the community, while the worst ones are peripheral and also belong to other communities. The average ones are central, however, they are also linked to other communities. For instance, the "Resistance distance" page (ranked 144^{th}) deals with a measure of proximity between nodes using electrical circuits principle. Although this page is clearly related to "Graph theory", it is also linked to the "Resistance" and "Ohm" pages, so it also belongs to "Electricity" community. This AUC-based ranking also indicates if a node is representative of a given set of nodes of interest (or a community). In the example above, the page "Multiple edges" describes very well "Graph theory", while "Global shipping network" does not. More generally, we can identify nodes from the reference set which do not allow to properly describe the community (having a bad AUC). They may also belong to other communities or they may not be in the multi-ego-centered community. Conversely, we can identify nodes which are always badly ranked for other reference nodes and that, therefore, may not be in the community. We also examined by hand the best ranked nodes that are neither in the test set nor in the training set. We found that although these pages have not been classified by Wikipedia users in the "Graph theory" category or a direct subcategory, they are all strongly related to it. Indeed, in the top 25, one has been added to the "Graph theory" category in the most recent version of Wikipedia ("Walls and Lines" page, added on April, 19^{th} 2013), 23 are classified in a subsubcategory of "Graph theory" or even deeper (mostly "Graphs families" and "Regular graphs"). The last one, "Graphlets" belongs to the "Networks" category and, since a graphlet is a "small connected non-isomorphic induced subgraph", it would perfectly fit in the "Graph theory" category. We finally checked that the model shows no overfitting. This would be the case if it had good performances on the training set and poor predictive performance, i.e. bad results

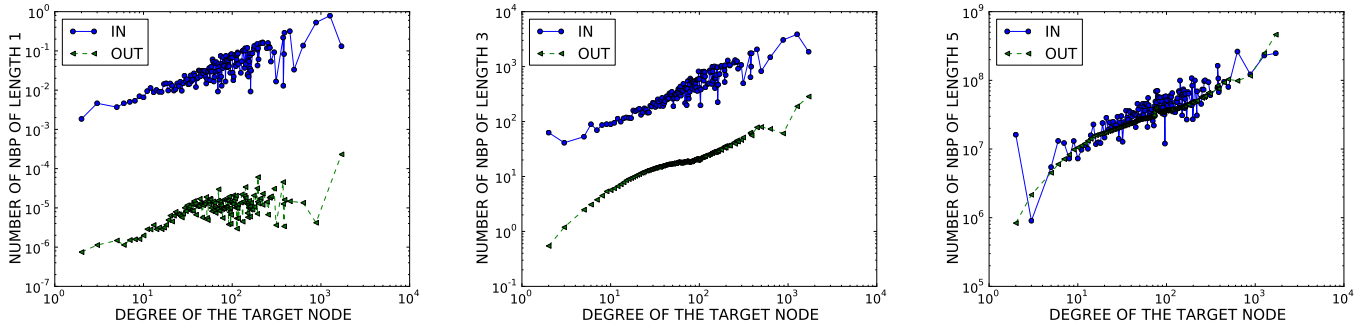
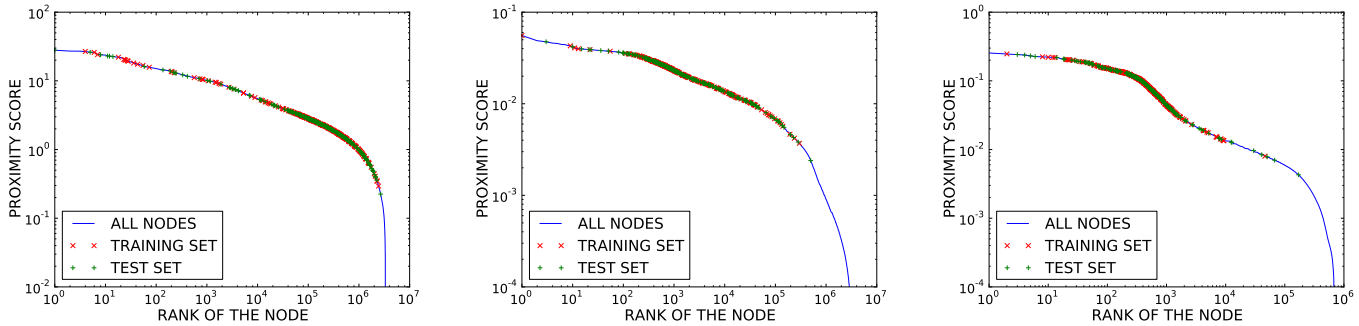


Fig. 2. Average number of NBP of length 1, 3 and 5 (from left to right) as a function of the degree of the target node between a source and a target node in the category "Graph theory" (in) and a source node within the category and target node outside the category (out).



(a) (b) (c)

Fig. 4. Proximity score vs rank obtained for the proper $(\alpha, \beta, \delta, \lambda)$ for the nodes "Global shipping network", "Resistance distance" and "Multiple edges" (from left to right). The points corresponding to the test set and the training set are emphasized.

on the test set; this was not the case as we obtained similar predictive performance on both sets.

C. Combining individual scores

As explained before, ranking nodes with regards to a single node is sometimes not sufficient and, in general we need to combine proximity values to several reference nodes in order to obtain a good scoring. To do so, we must (i) for each reference node, learn independently the parameters that best rank the other reference nodes and, for each reference node, compute its proximity to all other nodes of the network with the optimal parameters, then (ii) for each node of the graph, combine the proximities obtained with regards to each reference node to obtain the proximity to the set. Combining the scoring obtained from the individual scores of two medium quality pages, namely "Resistance distance" (ranked 144 out of 271) and "Fan Chung" (ranked 198) leads to an AUC of 0.9996 and a top 1000 precision of 0.7933, which is comparable to the performance of the best individual scoring obtained. We further validated this idea by comparing the AUC obtained for all 271 nodes of the training set; the AUC obtained by the combination of the k best ranked nodes, for all values of k; and the AUC obtained by the combination of random sets of k nodes for many values of k. We found that considering more reference nodes leads to an improvement of the results compared to taking a single node. However, taking too many nodes induces a decrease of the result. We believe that a small number of nodes is sufficient to characterize a community, we therefore suggest to simply search for the best AUC among all pairs of scoring, i.e. with only two reference nodes. For our example, the best ranking is given by the product of the best and third best individual rankings, i.e. pages "Multiple edges" and "Random regular graph". These two pages are therefore the best description of the "Graph Theory" category.

D. From scoring to community

Once the proximity scores leading to the best AUC are obtained, a crucial question is to know how to cut and decide which nodes are in the community and which nodes are not. We propose to identify the first sharp decrease, if any (if there is no sharp decrease then it means that there is no community). Nodes way before (resp. way after) the decrease are clearly inside (resp. outside) the community. Nodes in between do not clearly belong to the community but they are not outside the community either. Our choice is to include them in the community. A good, yet simple, solution to achieve this is to cut at the first highest value of the second derivative, i.e. just after the sharp decrease. The operation, illustrated in Figure 5 for the best set of reference nodes (i.e. the one giving the best AUC) for the "Graph theory" category gives a group of 1708 nodes. This group contains 91% (resp. 92%) of the nodes in the test set (resp. training set).

We further validated our framework on the benchmark for overlapping community detection detailed in [6]. For this benchmark, the highest second derivative seems to best correspond to the delimitation between the inside and outside of

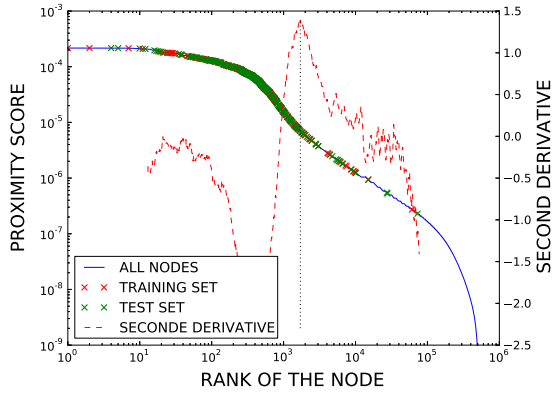


Fig. 5. Proximity score vs rank for the set of reference nodes giving the best AUC, and second derivative of the curve. Wikipedia dataset with the "Graph theory" category.

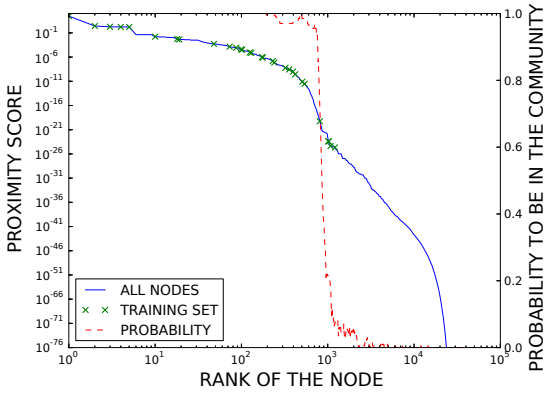


Fig. 6. Proximity score in decreasing order for a random graph with a known overlapping community structure generated with the benchmark of [6]. The graph has 100k nodes, an average degree of 15 and 10k nodes belong to 3 communities. Other parameters are set to the default values of the model. The considered community contains 961 nodes, 514 of which belong to 3 communities. The learning phase has been made of a random sample of 30 nodes of this community. The probability to be in the community corresponds, for a given rank k, to the proportion of nodes from rank k to $k + x$ in the community. To obtain a smooth curve we took $x = 100$.

the community, as shown in Figure 5. Cutting using the second derivative gives a group of 975 nodes. Amongst them, 870 are in the LF defined community of 961 nodes.

E. Comparison to baselines

Related approaches are not designed to complete a set of nodes into a community. For instance, in [12] and [13], authors use a proximity driven approach to find the k (parameter) most relevant connected nodes. In [2], the goal is to find all communities of a given node. Using these methods to complete a community would disadvantage them too much, we thus modified them to have baselines to which we can compare our framework. More precisely, we used four other proximity measures: (i) the distance, next referred to as "DISTANCE", (ii) the proximity of [13] "T and F" with the parameters recommended by the authors ($c = 0.5$ for the restart probability and $\alpha = 0.5$ for the normalization), (iii) the parameter-free carryover opinion of [2] "CAROP", and (iv) the Local Path

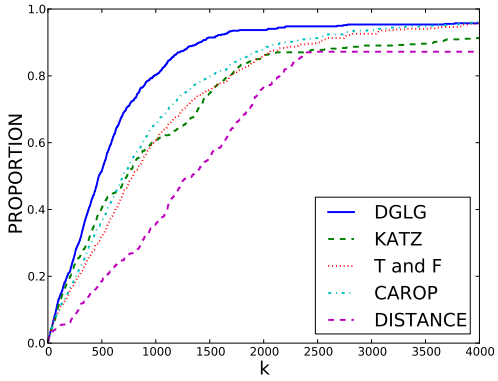


Fig. 7. Proportion of nodes in the test set among the first k ranked nodes as a function of k for the best ranking obtained for the four baselines and our framework "DGLG".

Index [19] learning the β parameter to handicap long paths "KATZ".

We then: (i) computed the proximity of all nodes to each node in the input set, (ii) combined the scoring to obtain the proximity to the input set, (iii) cut at the highest second derivative to obtain the multi-ego-centered community.

We used the four measures on the training set of the pages in the Wikipedia "Graph theory" category and evaluated their performance on the test set. In terms of precision, our method performs better than the other ones at every step. Figure 7 compares the number of nodes in the test set among the first k nodes as a function of k for the best individual rankings obtained for each measure.Our proximity measure and the learning step are essential contributions for the task of community completion; indeed we obtained 80% of the nodes in the test set within the top 1000 ranked nodes, while other measures achieved only 60%; we obtained 95% of the test set within the top 2000 nodes, while other methods needed the top 3000 or more.

Note that using the Local Path Index learning β lead to very small values of β, around 10^{-5}. This is because higher values of β gives a too high value to hubs and these (poorly related) hubs are then ranked before very related nodes with smaller degree, this is why giving a handicap to hubs is so important. Note also that changing the parameters of "T and F" is slow and does not change much the final ranking obtained contrarily to our proximity measure.

V. Conclusion

We have presented in this paper an efficient proximity measure between nodes of a graph that requires several parameters which (i) control the trade-off between the number of links between the nodes and the number of different paths between them, and (ii) limits the bias favoring high degree nodes. This proximity can be evaluated quickly for a large number of sets of parameters. The ergonomy of our proximity allows for the parameters to be learned automatically and efficiently when the proximity measure is used to rank nodes according to a set of input nodes (called reference set). This leads to a finely tuned scoring of all nodes of the graph with regard to their proximity to the nodes in the reference set. Individual scorings can be combined to obtain a scoring of all nodes of the graph

with regard to their proximity to the reference set. When the nodes in the reference set belong to a same community, we showed that our proximity measure could efficiently identify the underlying so-called multi-ego-centered community: if the curve of proximity values in decreasing order exhibits a plateau followed by a strong decrease, a community can be identified. Indeed it means that all nodes before the decrease are close to the the nodes in the reference set while nodes after the decrease (the rest of the nodes) is far and thus do not belong to this community. The label of the nodes that best characterizes (i.e. that have the best individual scorings) the reference set can be used to label the community itself. We illustrated this approach through conclusive examples carried out on toy graphs, benchmarks and a real-world dataset of Wikipedia user-annotated pages. We finally shown that our approach performs much better than existing ones.

This work opens many perspectives in the area of community detection. Indeed, it shifts from the traditional partition paradigm while computing communities efficiently and precisely. The obtained communities may be overlapping, which is more realistic than disjoint communities. The proposed method is already very efficient, accurate and easy to use. However, possible improvements could consist in using the proximity values or the NBP and the degree as features for one class-classification algorithms. The brute-force optimization can also be refined. Moreover, this multi-ego-centered approach is very promising for the study of community evolution over time. Indeed, we expect the ranking to be quite stable to changes in the topology, which could simplify this issue.

ACKNOWLEDGMENT

The authors thank D. F. Bernardes, R. Fournier, S. Kirgizov and D. Obradovic for useful discussions. This work is supported in part by the French National Research Agency contract CODDDE ANR-13-CORD-0017-01 .

REFERENCES

[1] M. Danisch, J. Guillaume, and B. Le Grand, "Towards multi-ego-centered communities: a node similarity approach," *Int. J. of Web Based Communities*, 2012.

[2] ——, "Unfolding ego-centered community structures with a similarity approach," in *4th Workshop on Complex Networks (CompleNet)*, 2013.

[3] A. Clauset, "Finding local community structure in networks," *Physical Review E*, vol. 72, no. 2, p. 026132, 2005.

[4] A. Friggeri, G. Chelius, and E. Fleury, "Triangles to capture social cohesion," in *Privacy, security, risk and trust (passat), International Conference on Social Computing (socialcom)*. IEEE, 2011, pp. 258–265.

[5] S. Khan and M. Madden, "A survey of recent trends in one class classification," in *Artificial Intelligence and Cognitive Science*. Springer, 2010, pp. 188–197.

[6] A. Lancichinetti and S. Fortunato, "Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities," *Physical Review E*, vol. 80, no. 1, p. 016118, 2009.

[7] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3, pp. 75–174, 2010.

[8] T. Evans and R. Lambiotte, "Line graphs, link partitions, and overlapping communities," *Physical Review E*, vol. 80, no. 1, p. 016105, 2009.

[9] A. Lancichinetti, S. Fortunato, and J. Kertész, "Detecting the overlapping and hierarchical community structure in complex networks," *New Journal of Physics*, vol. 11, no. 3, p. 033015, 2009.

[10] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, 2005.

[11] M. Sozio and A. Gionis, "The community-search problem and how to plan a successful cocktail party," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2010, pp. 939–948.

[12] Y. Koren, S. C. North, and C. Volinsky, "Measuring and extracting proximity in networks," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006, pp. 245–255.

[13] H. Tong and C. Faloutsos, "Center-piece subgraphs: problem definition and fast solutions," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006, pp. 404–413.

[14] S. Cohen, B. Kimelfeld, and G. Koutrika, "A survey on proximity measures for social networks," in *Search Computing*. Springer, 2012, pp. 191–206.

[15] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: bringing order to the web." 1999.

[16] L. Backstrom and J. Leskovec, "Supervised random walks: predicting and recommending links in social networks," in *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, 2011, pp. 635–644.

[17] U. von Luxburg, A. Radl, and M. Hein, "Hitting and commute times in large graphs are often misleading," *arXiv preprint arXiv:1003.1266*, 2010.

[18] L. Katz, "A new status index derived from sociometric analysis," *Psychometrika*, vol. 18, no. 1, pp. 39–43, 1953.

[19] L. Lv, C. Jin, and T. Zhou, "Effective and efficient similarity index for link prediction of complex networks," *arXiv preprint arXiv:0905.3558*, 2009.

[20] D. C. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization," *Mathematical programming*, vol. 45, no. 1-3, pp. 503–528, 1989.

[21] L. G. Valiant, "The complexity of enumeration and reliability problems," *SIAM Journal on Computing*, vol. 8, no. 3, pp. 410–421, 1979.

APPENDIX

The computation of the number of simple paths is not practical for paths of length four and more in large real world graphs such as the Wikipedia network [21], even though approximations exist. Instead of using this quantity, we thus use the number of *non-backtracking paths* (NBP), i.e. the number of paths forbidding cycles of lengths two (backward hops) but allowing longer cycles.

This quantity is much easier to compute for a starting node i using Equations 3. Let X_l be the vector containing the number of NBP of length l, i.e. the j^{th} coordinate corresponds to the number of NBP between i and j. X_l is straightforwardly given for any l by

$$\begin{aligned} X_0 &= \delta_i \\ X_1 &= AX_0 \\ X_2 &= AX_1 - DX_0 \\ \forall l \geq 3, X_l &= AX_{l-1} - (D-I)X_{l-2} \ , \end{aligned} \tag{3}$$

where δ_i is set to the null vector except for the coordinate of node i which is set to one, A is the graph adjacency matrix, D the diagonal matrix of the degrees and I is the identity matrix. The term $(D-I)X_{l-2}$ eliminates the walks which backtrack in the l^{th} step.

The complexity to compute the number of NBP of all lengths smaller than l from one node to all nodes of the graph is thus linear with the size of the graph: in $O(l(n+m))$, where n is the number of nodes and m the number of edges.