Multi-objective community detection for bipartite graphs

Anonymous Author(s)
Anonymous Institution
anonymous@e.mail

ABSTRACT

Bipartite graphs pose particular challenges for community detection. We explore a multi-objective approach to community detection that simultaneously considers different one-mode projections, treating them as different views of the same learning problem. Building on existing tools in the pymoo and igraph Python packages, we highlight how existing evolutionary approaches to clustering and community detection can be adapted to tackle bipartite community detection, and demonstrate the effectiveness of the approach on problems with a known ground truth. Our results show that multi-objective optimization is useful in this problem, both as a means of counter-balancing bias intrinsic to the modularity measure and as a means of combining the information obtained from individual one-mode projections.

CCS CONCEPTS

• **Theory of computation** → *Design and analysis of algorithms*; • **Computing methodologies** → Cluster analysis.

KEYWORDS

Clustering; community detection; bipartite graphs; multiobjective machine learning

ACM Reference Format:
Anonymous Author(s). 2022. Multi-objective community detection for bipartite graphs. In *Genetic and Evolutionary Computation Conference (GECCO '22), July 13–17, 2022, Boston, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3321707.3321761

1 INTRODUCTION

Bipartite graphs arise in a wide range of problems, most prominently recommender systems. Despite this practical relevance, specialised methodologies for their analysis remain underdeveloped relative to the much wider range of methodologies available for the analysis of univariate graph structures. How best to bridge this methodological gap has been an ongoing subject of discussion in the network community, in particular with respect to the important problem of community detection.

Building on previous work assessing the information loss associated with one-mode/unipartite projections, we reframe bipartite community detection as an issue of unsupervised multi-view learning, with incomplete views, over a single bipartite graph structure. This paves the way towards the implementation of an evolutionary algorithm for the problem that uses multiple criteria to simultaneously optimize and integrate all available one-mode projections, and to counterbalance the biases of individual objectives. Our experiments highlight the robust performance of the approach across a set of test instances presenting various levels of challenge to traditional methods.

We have striven for the implementation of the algorithm on top of existing, specialised Python packages for graph analysis (igraph) and multi-objective optimization (pymoo), to allow for the future seamless exchange of, and experimentation with, individual algorithm components, such as different criteria or optimizers. The code of our method is available at http://placeholder.

2 BACKGROUND

To help position and explain our work, this section introduces key concepts from network analysis and machine learning. We then proceed to highlight relevant prior work in community detection using statistical and evolutionary approaches.

2.1 Bipartite graphs

In a bipartite graph there exists a division of the vertices into two sets, labelled top and bottom, with the additional constraint that edges cannot exist between vertices from the same set. An example of such a graph can be seen in Figure 1. Such a structure can be found in many practical circumstances, such as manuscript authorship [3] and user interaction with content online [4]. Many of the metrics used to study unipartite graphs can be extended to the bipartite case, but often require careful consideration of how they are affected by the bipartite edge restriction.

2.2 Projections of bipartite graphs

One method available to reduce the computational complexity of studying bipartite graphs is via unipartite projection. Unipartite projection infers a new graph on the top or bottom vertices where edges exist between vertices that share at least one common neighbour in the bipartite graph, as illustrated in Figure 1. Some information is lost in the projection process (since many bipartite graphs can share the same projection) but it has been shown that by considering both the top and bottom projection together, it is possible to recover the original bipartite graph [8]. Another important consideration in unipartite projections is the choice of how to weight the resulting edges. Typically, edge weights in the unipartite projections correspond to the number of mutual neighbours of a given pair of nodes, but other schema have been developed (see e.g. [18]). The impact of projection weighting depends on the graph and metric in

GECCO '22, July 13–17, 2022, Boston, USA

ACM ISBN 978-1-4503-6111-8/19/07…$15.00
https://doi.org/10.1145/3321707.3321761

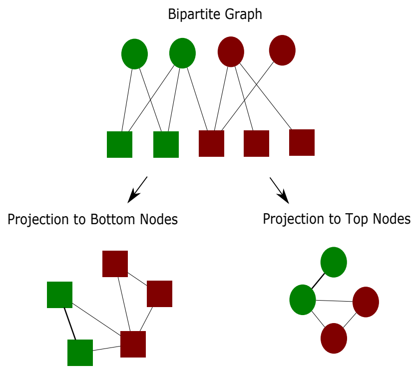

Figure 1: Example of a bipartite graph with 4 top and five bottom vertices (top) and two of its one-mode projections (bottom). Here, edge weights are determined as the number of mutual neighbours of a given pair of vertices, and visualized using line thickness.

question [16] and should be considered carefully in experimental contexts.

2.3 Largest connected component

Given a graph G and sub-graph $g \subset G$, the sub-graph can be said to be *connected* if it is possible to reach any node in g from any other node in g. A connected *component* is a connected sub-graph that includes all nodes in G that can be reached from any other node in the sub-graph. The connected component with the maximum number of nodes, is therefore, called *largest connected component*.

2.4 Minimum spanning tree

The *minimum spanning tree* (MST) [27] of a graph contains all of the original vertices of a connected graph and a subset of its edges, so that all vertices remain connected, without any cycles and with the minimum possible total edge weight. Whilst a minimum spanning tree can be constructed for an unweighted graph, it would more likely to produce more possible solutions with the same total edge weight.

2.5 Betweenness centrality

Conceptually, the *betweenness centrality* of an edge describes the amount of traffic that flows through an edge [19], as illustrated in Figure 2. Mathematically, for an edge $e \in E$ in a graph G with node set V and edge set E, the betweenness centrality is given by

$$c_B(e) = \sum_{i,j \in V} \frac{\sigma(i,j|e)}{\sigma(i,j)}, \tag{1}$$

where $\sigma(i,j)$ is the number of shortest paths between vertices $i, j \in V$ and $\sigma(i,j|e)$ is the number of these paths that passes through edge e.

2.6 Community detection techniques

When studying graphs we are often interested in identifying groups of vertices which interact more frequently with their in-group than their out-group. Partitioning a graph into such groups is known as

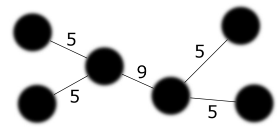

Figure 2: Example of betweenness centrality in a simple graph with six vertices and five edges. Each edge is annotated with its betweenness centrality $c_B(e)$.

community detection and a number of different approaches have been proposed by scholars (e.g. edge-betweenness [20], random walk [26], stochastic block models [22]). Among the most popular methods are the family of modularity-maximizing algorithms, named for the quality metric developed by Newman and Girvan [20]. The modularity of a graph under a partition C is defined by

$$Q = \frac{1}{2|E|} \sum_{i,j \in V} \left(A_{ij} - \frac{k_i k_j}{2|E|} \right) \delta(C(i), C(j)), \tag{2}$$

where A_{ij} is a binary (zero/one) indicator of the presence of an edge between vertices i and j, k_i is the degree of vertex i, $C(i)$ is the proposed community assignment of vertex i and $|E|$ is the number of edges in the graph with corresponding node set V and edge set E. $\delta(C(i), C(j))$ is the indicator function of vertex i and j being in the same community. To summarise this intuitively, modularity gives an indication of how many edges we observe within the assigned communities relative to a null model of random rewiring of the edges in the graph. Efficient calculation of modularity allows for optimization over large graphs, further increasing the popularity of widely available methods using modularity (e.g. [7]). Despite this popularity, modularity is not a perfect metric for community structure in graphs. It has been shown to have a resolution limit wherein merging small communities can lead to increases in modularity regardless of the ground truth structure [9].

Many of these community detection techniques can be extended to the case of bipartite networks. Methods such as the stochastic block model approach can be readily adapted to the two-mode requirements of bipartite networks [15], but modularity-based techniques require further work. The key difference is the formulation of an alternative null model to that seen in Equation 2, such as proposed by Barber [1]. These adaptations to modularity impact the efficiency of maximising algorithms, and as such a projection-based approach is often used to identify communities in bipartite networks. In the context of community detection, the projection must be done with care. The weighting of edges applied during the projection process can affect the accuracy of the detected communities [5]. Alternatively, communities found on the projections onto both the top and bottom vertex sets can be used to recover community structure in the bipartite graph [17].

2.7 Multi-view learning

As a consequence of the existence of different one-mode projections, we can think of the problem of community detection as a multi-view learning problem with incomplete views. In the machine learning literature, the term multi-view learning [29] is generally used to refer to machine learning problems in which entities are characterised by information from several distinct data sources, which may correspond to information captured at different levels of a system (e.g. in systems biology) or from different data sources. An integration of features into a single machine learning model may not be straightforward due to varying formats and reliability of the data, and multi-view techniques are devised to address these challenges. In an unsupervised learning context, data can alternatively present in the form of multiple feature sets [31] or multiple sets of relational information [14], and previous work has considered both multi-view clustering and multi-view community detection [12].

The majority of work on multi-view learning assumes that information for all entities is available for all views (i.e. complete), but certain scenarios may involve learning with incomplete views and algorithms for this case have been considered [28, 30].

Linking this back to bipartite graphs, projections to individual modes (or, potentially, with different weights) provide one specific, incomplete characterisation of a bipartite graph. Each projection is incomplete from the point of view of information loss [8] and provides explicit relational information about a subset of vertices (either bottom or top) only. It is clear that a better characterization of the graph should be obtained by an effective integration of these views, but the most effective approach for doing so is unclear.

2.8 Evolutionary approaches to community detection

Given the computational challenges of searching for communities in large graphs, meta-heuristic approaches for community detection have been proposed by a number of researchers, including a body of work in evolutionary computation [24]. A key design choice in these algorithms has been the choice of representation - for larger instances this has major implications for the efficiency of the search and the introduction of heuristic bias towards particular types of solutions. Following [23], the locus-based adjacency scheme has shown some success in this setting, consistently with its performance in evolutionary approaches to data clustering [11], and this is the representation adopted in our work. Furthermore, multi-objective optimization has been use to directly explore the trade-off between within-community connectivity and between-community separation [23].

Research on adapting evolutionary approaches to special graphs has been more limited, but has included some work on bipartite graphs [32]. The closest to our approach is the work reported in [25], which addresses the use of multi-objective optimization to optimize modularity across different layers of a multi-layer graph. Our work is significantly different from this approach, as our focus is on the specific requirements of bipartite graph structures, and exploits properties of the bipartite graph in more depth in the algorithm design, especially at the initialization stage.

3 OUR APPROACH

In the following, we describe a multi-objective approach to community detection that focuses on bipartite graph structures. The algorithm has been derived with two key objectives in mind:

- To support the seamless integration of different one-mode projections and, where relevant, facilitate the exploration of trade-offs between these.
- To support the effortless integration of additional objectives related to node rather than topological features.

The design of the approach draws on existing methodologies in the areas of network analysis and multi-objective optimization (as available in the igraph and pymoo packages), as well as ideas previously introduced in the context of multi-objective clustering [11].

3.1 Multi-objective optimizer

Our current implementation assumes use of a multi-objective optimizer as implemented in the pymoo package. Here, our evolutionary algorithm of choice is NSGA-II, with a population size of 50 and run for 1000 generations, but these choices have not been optimized and alternatives could readily be used, including the choice of a many-objective optimizer if > 3 objectives should be explored.

3.2 Objective Functions

The aim of our evolutionary algorithm is to evolve partitions that best reflect the community structure of the bipartite graph. For the algorithm instantiation examined in this paper, we use a three-objective formulation of the community detection problem. Specifically, our "3d" approach separately considers the modularity of two distinct one-mode projections (to the top and bottom set of vertices, respectively), and treats each as a separate optimization objective (to be maximized). Edge weights in the one-mode projections simply reflect the number of mutual neighbours, but other weighting schemes could be explored. The projections of the bipartite graph structure, and their respective modularity, are calculated using the implementations available in the igraph package.

Minimization of the number of communities is added as a third objective, in order to compensate for modularity's tendency to overestimate the number of communities, and ensures that the algorithm returns partitions for a range of possible choices of k.

The modularity of the original bipartite graph is not optimized directly in this approach, which allows us to assess any benefits or information loss associated with considering modularity of the one-mode projections only.

3.3 Representation

Each individual in our evolutionary algorithm represent a single partition of the original bipartite graph.

To encode this information, we use an indirect representation well-established in previous work on multi-objective clustering [11] and community detection [23]. For a graph containing N nodes, the linkage-based adjacency scheme results in a representation of length N, with the ith decision variable indicating an edge between i and one of its neighbours. Specifically, for a node i with degree D, the ith decision variable can take values in the range $[0, D]$, with 0 indicating a self-link (i.e. no link), and all other values indicating a link to one specific neighbour (i.e. the neighour's position in node i's ordered neighbour list from the adjacency list representation of the graph).

Consequently, each individual encodes a sub-graph that contains all vertices but only a sub-set of at most N of the edges of the original bipartite graph structure. This sub-graph can be interpreted as a partition by identifying all separate connected components within the sub-graph, and interpreting each as separate communities. In igraph, this decoding step is easily realised using existing functions.

3.4 Variation operators

In line with previous work, a simple graph-based mutation is used. Conceptually, mutation allows each node to be reconnected to one of its neighbours in the original bipartite graph structure or to form no connection, which translates to a reduction in the number of edges in the individual. From a practical point of view, the representation employed is an integer representation, and mutation is implemented using pymoo's polynomial mutation operator for integer representations *int_pm*. As all edges in the original graph are unweighted, the probability of mutation is kept at pymoo's default and equal across all possible edges, at this time. Variations based on betweenness centrality may be of promise.

Uniform crossover has been shown to be appropriate for the linkage-based adjacency scheme, and is applied with probability $p = 0.3$.

3.5 Initialization

The adjacency-based representation is initialised using the minimum spanning tree, an approach designed to reduce the redundancy of this encoding and to positively bias the search space. The approach is loosely inspired by previous work on cluster analysis [11], although several adjustments are necessary to reflect the change from clustering to an unweighted graph setting. The three key steps of our initialization routine are illustrated in Figure 3, and discussed in more detail below. The approach does not assume a bipartite nature of the underlying graph, and could be used in other unweighted graph settings.

As our original graphs are unweighted, we use betweenness centrality of edges as a proxy for edge weights. The rationale for this approach is to enrich the representation with additional bias. Betweenness centrality provides an indication of the importance of an edge in bridging communities within a graph, and therefore provides the basis for existing methods for community detection [10], including the proposed approach, by minimizing the use of cross-community edges and thus encouraging the easy decomposition of the MST into communities, based on the removal of only few cross-community links.

Once the MST representation is obtained, the resulting representation would decode to a one-cluster solution (one single connected component), so a final step is designed to create different partitions through the removal of a number of edges from individual solutions. Instead of a random approach, the Fast Greedy heuristic [6] is used to obtain fifty separate partitions with the number of communities k ranging from 1 to 50, in steps of 1. Each individual in the original population is then adjusted to be consistent with just one of these partitions: specifically, we identify any edges contained in the original MST but linking separate communities in the given Fast Greedy solution. These edges are removed from the solution, i.e. marked by a value of 0 in the representation. This results in a population that is diverse only with respect to those edges of the minimum spanning tree where Fast Greedy explored possible community boundaries.

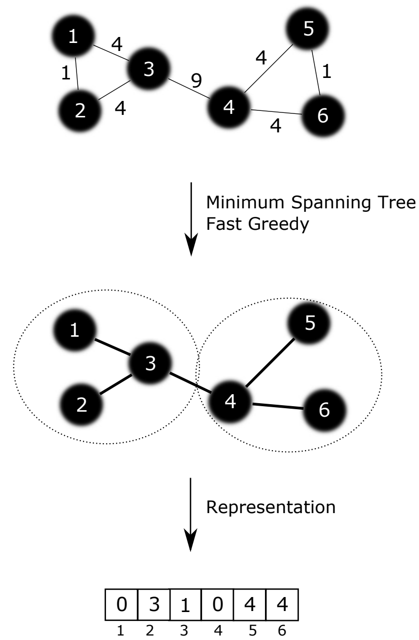

Figure 3: Key steps of initial solution generation in the evolutionary algorithm. The top shows the original graph with 6 vertices and 7 edges. Edges are annotated by a pseudo edge weight which is derived from the betweenness centrality of each edge. The middle displays the minimum spanning tree for this graph, which serves as the basis for the representation of all individuals. The ellipsoids further highlight a possible fast greedy solution (for $k = 2$). This is used to identify those edges that cross fast greedy communities (in this case the edge from vertex 4 to 3) and will need to be removed. Finally, the bottom shows a possible representation of the resulting individual, which consists of seven integer values (one for each vertex). For illustration purposes, this example assumes the use of absolute node indices, but note that the actual implementation would use instead the ordered neighbour position as given in the adjacency list representation of the graph. Thus, in the implementation the values of each decision variable fall into the range $[0, D]$ only.)

4 EXPERIMENTS

The experiments in this manuscript are designed as a first proof-of-principle to explore the potential of a multi-objective approach to bipartite community detection. As such our analysis focuses on synthetic benchmarks, and sensitivity to a range of potential data complications.

4.1 Benchmark data

A simple generating model, based on a stochastic block model [21, 22], is used:

- A bipartite graph structure with L vertices assigned to the lower part of the graph and U vertices assigned to the upper part of the graph.

- A two-class structure supported by the lower and upper vertices of the graph, with percentage M_L of lower nodes belonging to class 0, and $M_L - 1$ to class 1, and percentage M_U of upper nodes belonging to class 0, and $M_U - 1$ to class 1.
- N edges that obey the bipartite structure of the graph are randomly placed between any pair of nodes. Edges are placed between vertices from the same community with probability P and across communities with probability 1-P. The probabilities for edge placement are kept uniform at the class level, resulting in an interaction between class size and vertex degree. No duplicate, or self-loop edges are allowed and edges in the original bipartite graph are unweighted.

Our experiments focus on performance on the largest connected component of the graph only, as there is no meaningful way of assigning community structure across disconnected components. For each choice of parameters, 30 different instances are generated, and results for all algorithms are reported across these. Note that performance across instances varies significantly and drives significant amounts of the variation displayed in the following boxplots. Figure 4 shows some examples of networks generated with the synthetic model.

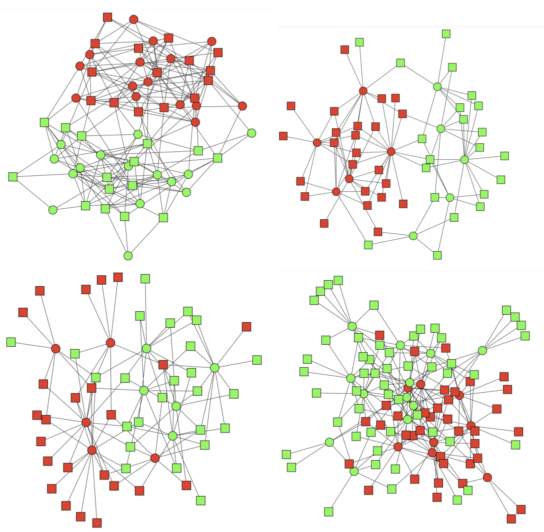

Figure 4: Example test instances with different ratios of top and bottom vertices, between-community connection and class imbalance. Top and bottom vertices are differentiated by shape and the ground truth is highlighted using color coding.

4.2 Contestant techniques

We compare to a range of established community detection techniques available in the igraph package. This includes greedy optimization of modularity (Fast Greedy FG, [6]), a random walk method (Walking Trap WT, [26]) and a hierarchical algorithm employing betweenness centrality (Edge Betweeness CE [20]). All three methods return hierarchies of communities and analysis focuses on the best solution contained in this set of partitions (with respect to the ground truth), considering a number of distinct communities $k \in [1, 15]$.

Additionally, we compare against two methods specialised in handling bipartite graph structures: the first of these is the BRIM algorithm [1], as implemented in the condor package. This approach returns the optimal modularity partition only, so is at a disadvantage, compared to other approaches that are compared across a range of possible number of communities. The second approach, ML, is the multi-level method proposed by [2], which involves the separate application of the Louvain method on each projection, followed by the identification of a consensus clustering through hierarchical agglomerative clustering on the resulting partitions. The similarity between pairs of communities is measured as the number of edges that connect nodes in the two communities.

Finally, we compare against an alternative version of our multi-objective framework, 2d, optimizing modularity of the original bipartite graph and the number of clusters only (i.e. using just two rather than three objectives). This is to highlight any differences arising from the choice of objectives (i.e. the simultaneous optimization of multiple projections) rather than the representation-specific bias or the use of a meta-heuristic optimizer. The resulting algorithms, 2d and 3d, both return an approximation front covering a range of values of k. As for the other approaches, analysis focuses on the best solution contained in each set (with respect to the ground truth).

4.3 Performance assessment

Performance is assessed through comparison of the detected communities to the available ground truth. Similarity of the partitions is quantified using the Adjusted Rand Index, [13] suitable for the evaluation of performance across multiple values of k. It takes values in the range $[0, 1]$, with higher values indicating better results.

5 RESULTS

Our experiments incrementally analyse the robustness of the different techniques to a range of complicating factors, including edge sparsity, class imbalance, imbalance between bottom and top vertices, and percentage of between-community edges. Our results indicate distinct differences in the algorithms' ability to cope with these aspects, and a surprising robustness of the multi-objective approach.

Figure 5 illustrates the performance of all algorithms on a simple problem with an equal number of bottom and top vertices, equally-sized classes, and 10 per cent between-community edges. It can be seen that all algorithms perform well at recovering the ground truth, as may be expected. The BRIM algorithm shows the weakest performance, but it is the only algorithm that returns a single community structure (as compared to a list of candidate solutions with different numbers of communities), which may partially explain this result.

Figure 6 contrasts this to performance on a graph with reduced connectivity. As the number of edges is reduced from 200 to 100, we can observe a significant drop in the performance of all approaches, with the worst impact observed for the BRIM algorithm. The other methods continue to display reasonable (although increasingly variable) performance in retrieving the true class structure.

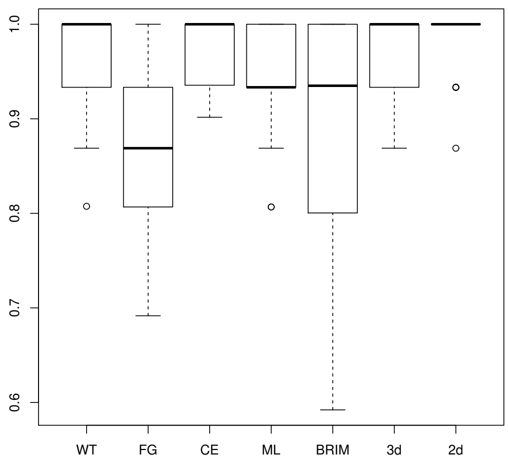

Figure 5: Adjusted Rand Index of best solution for 30 instances of a simple graph structure with a high node degree (200 edges, and 60 nodes).

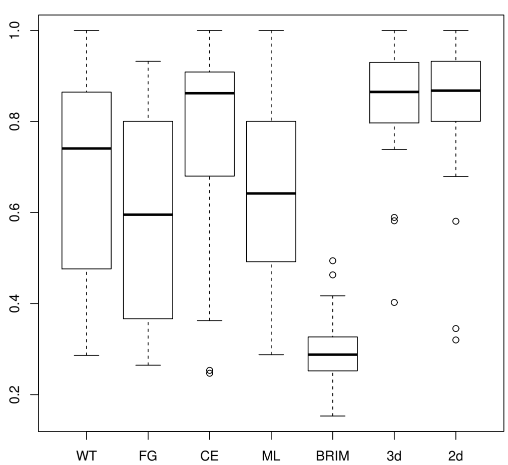

Figure 6: Adjusted Rand Index of best solution for 30 instances of a simple graph structure with a lower node degree (100 edges and 60 nodes).

Figure 7 considers the change in performance as the ratio between top and bottom vertices is varied from 1-1 to 5-1. The performance of the algorithms is not negatively affected by this adjustment. As a matter of fact, contrasting Figure 6 and 7, we can observe a general increase in performance as a result of this change. The likely reason is the associated increase in the average node degree

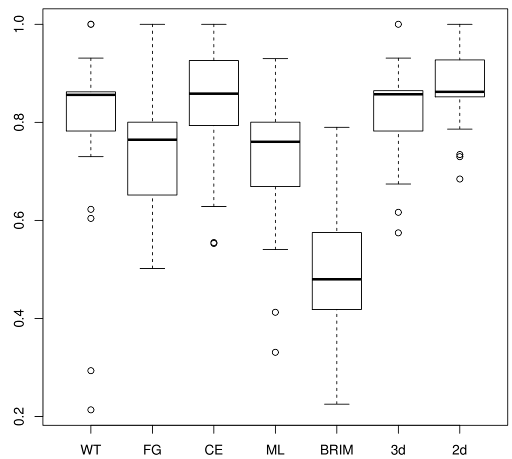

Figure 7: Adjusted Rand Index of best solution for 30 instances of a simple graph structure with 5-1 ratio of top to bottom nodes (other parameters unchanged relative to Figure 6).

in the bottom node set, which potentially aids the identification of the communities.

Next, we consider an increase in the proportion of between-community edges, raising this from 10 to 20 per cent of edges. Figure 8 highlights that this has the expected negative impact on performance, for all algorithms. However, the impact is weaker for the evolutionary algorithms, and paired testing starts to show systematic performance differences between these and all contestant methods.

Finally, we consider performance changes due to the relative size of the communities. Representative results are shown for generating models involving a mix of complex features, including a larger number of nodes (120 / 600), medium connectivity (200 / 1000 edges), medium level noise (20 per cent between community edges), and various levels of class imbalance (2-3 in both layers). The results shown in Figure 9 indicate a clear advantage of the evolutionary approach in this framework.

To understand the behaviour of the algorithms in more detail, Figure 10 provides a representative example contrasting the solution sets generated by the different algorithms with respect to one-mode modularity (top), optimization of overall modularity (center), and recovery of the ground truth (as measured by the Adjusted Rand Index. - bottom), on an instance from the experiments in Figure 9.

Unsurprisingly, the results confirm appropriate optimization performance of the multi-objective optimizers, with the 3d algorithm outperforming other algorithms in respect of one-mode modularity (top figure), and the 2d algorithm achieving the best modularity results in the original bipartite graph (central figure). More interestingly, these improvements in optimization performance translate into more accurate recovery of the ground truth, as shown in the

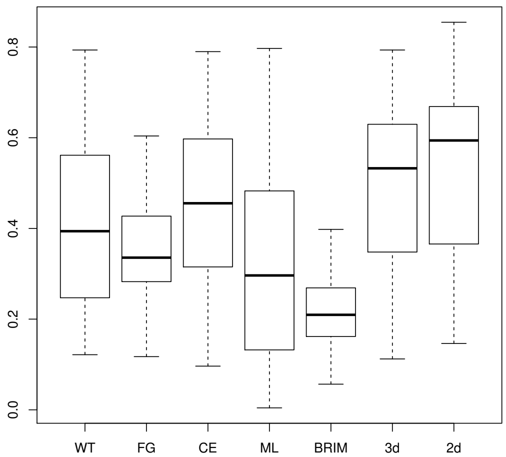

Figure 8: Adjusted Rand Index of best solution for 30 instances of a simple graph structure with increased between-community connectivity (other parameters unchanged relative to Figure 7).

bottom figure (and across our wider experiments). These improvements are due to a combination of factors, including the problem formulation, the representational bias (in particular, the use of betweenness centrality to bias the MST), and the use of a global optimizer.

Figure 9 (centre) further highlight that, on our current set of test instances, modularity tends to overestimate the number of communities, impacting on the results of the BRIM algorithm in particular. This highlights the importance of running algorithms across a range of choices of k and consider alternative ways of model selection. Interestingly, our multi-objective algorithms produce approximation sets that are small compared to other settings. This suggests a good correlation between the objectives, at least for graphs with a pronounced community structure.

Finally, our results show that optimization of the modularity of the bipartite structure does not translate into the full optimization of the modularity of each one-mode projection, and vice versa. Nevertheless, there is distinct evidence of the correlation of the objectives, with the 3d optimizer achieving reasonable levels of modularity on the bipartite structure, outperforming most contestants in that respect. This suggests that the most significant signal has been recovered by the simultaneous consideration of both projections, consistent with previous findings [17].

6 CONCLUSION AND FUTURE WORK

This work has focused on community detection in bipartite graph structures, framing it as a multi-view learning problem. The projection of a bipartite graph into its top or bottom layer provides two readily available, incomplete views of the system that are amenable to objective functions devised for the optimization of univariate graph structures. Our experiments indicate distinct promise of

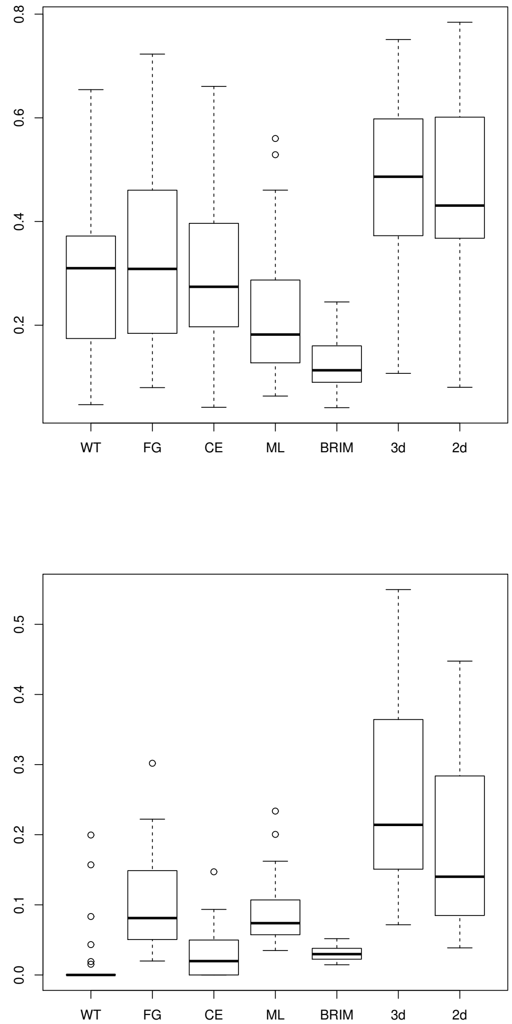

Figure 9: Adjusted Rand Index of best solution for 30 instances of two generating models incorporating a mix of challenges (class imbalance and noise).

multi-objective evolutionary algorithms in this scenario, although more comprehensive benchmarking is needed to establish the full potential of the approach.

It is worth highlighting that the possible uses of our implementation is not limited to bipartite graph structures or projection-based objectives. The key assumption made in our method is the availability of a single graph structure that remains relevant across all objectives, and can inform the adjacency-based representation of individual solutions, biasing the search. Beyond this, the method makes no assumptions about the nature of the objectives and additional criteria could easily be added e.g. to integrate node-specific features or prior knowledge of class membership for specific nodes.

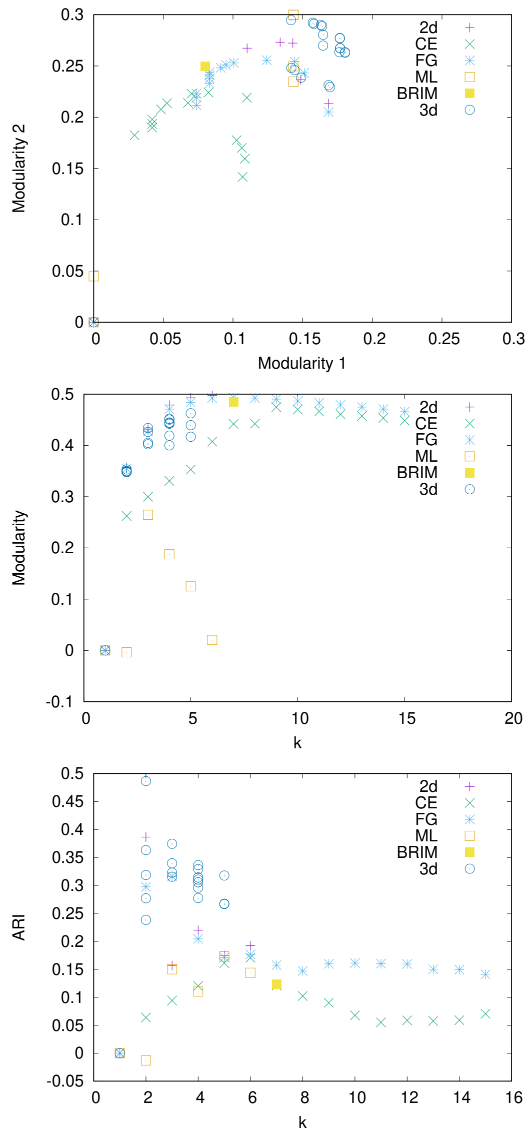

Figure 10: Indicative behaviour of different algorithms when viewed in different spaces. Partitions identified by the different algorithms (top) when scored with respect to modularity in each mode; (centre) when scored by modularity on the bipartite graph (and plotted as a function of k); (bottom) when assessed relative to the ground truth (and plotted as a function of k)

REFERENCES

[1] Michael J. Barber. 2007. Modularity and community detection in bipartite networks. *Phys. Rev. E* 76 (Dec 2007), 066102. Issue 6. https://doi.org/10.1103/PhysRevE.76.066102

[2] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* 2008, 10 (oct 2008), P10008.

[3] Christian Bongiorno, András London, Salvatore Miccichè, and Rosario N. Mantegna. 2017. Core of communities in bipartite networks. *Phys. Rev. E* 96 (Aug 2017), 022321. Issue 2. https://doi.org/10.1103/PhysRevE.96.022321

[4] Javier Borge-Holthoefer, Raquel A. Baños, Carlos Gracia-Lázaro, and Yamir Moreno. 2017. Emergence of consensus as a modular-to-nested transition in communication dynamics. *Scientific Reports* 7, 1 (30 Jan 2017), 41673. https://doi.org/10.1038/srep41673

[5] Tristan J. B. Cann, Iain S. Weaver, and Hywel T. P. Williams. 2020. Is it correct to project and detect? How weighting unipartite projections influences community detection. *Network Science* 8, S1 (2020), S145–S163. https://doi.org/10.1017/nws.2020.11

[6] Aaron Clauset, Mark EJ Newman, and Cristopher Moore. 2004. Finding community structure in very large networks. *Physical review E* 70, 6 (2004), 066111.

[7] Aaron Clauset, M. E. J. Newman, and Cristopher Moore. 2004. Finding community structure in very large networks. *Phys. Rev. E* 70 (Dec 2004), 066111. Issue 6. https://doi.org/10.1103/PhysRevE.70.066111

[8] M.G. Everett and S.P. Borgatti. 2013. The dual-projection approach for two-mode networks. *Social Networks* 35, 2 (2013), 204–210. https://doi.org/10.1016/j.socnet.2012.05.004 Special Issue on Advances in Two-mode Social Networks.

[9] Santo Fortunato and Marc Barthélemy. 2007. Resolution limit in community detection. *Proceedings of the National Academy of Sciences* 104, 1 (2007), 36–41. https://doi.org/10.1073/pnas.0605965104 arXiv:https://www.pnas.org/content/104/1/36.full.pdf

[10] Michelle Girvan and Mark EJ Newman. 2002. Community structure in social and biological networks. *Proceedings of the national academy of sciences* 99, 12 (2002), 7821–7826.

[11] Julia Handl and Joshua Knowles. 2007. An evolutionary approach to multiobjective clustering. *IEEE Transactions on Evolutionary Computation* 11, 1 (2007), 56–76.

[12] Chaobo He, Yong Tang, Hai Liu, Xiang Fei, Hanchao Li, and Shuangyin Liu. 2019. A robust multi-view clustering method for community detection combining link and content information. *Physica A: Statistical Mechanics and its Applications* 514 (2019), 396–411.

[13] Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of classification* 2, 1 (1985), 193–218.

[14] Adán José-García, Julia Handl, Wilfrido Gómez-Flores, and Mario Garza-Fabre. 2021. An evolutionary many-objective approach to multiview clustering using feature and relational data. *Applied Soft Computing* 108 (2021), 107425.

[15] Daniel B. Larremore, Aaron Clauset, and Abigail Z. Jacobs. 2014. Efficiently inferring community structure in bipartite networks. *Phys. Rev. E* 90 (Jul 2014), 012805. Issue 1. https://doi.org/10.1103/PhysRevE.90.012805

[16] Yongjun Li and Chun You. 2013. What is the difference of research collaboration network under different projections: Topological measurement and analysis. *Physica A: Statistical Mechanics and its Applications* 392, 15 (2013), 3248–3259. https://doi.org/10.1016/j.physa.2013.03.021

[17] David Melamed. 2014. Community Structures in Bipartite Networks: A Dual-Projection Approach. *PLOS ONE* 9, 5 (05 2014), 1–5. https://doi.org/10.1371/journal.pone.0097823

[18] M. E. J. Newman. 2001. Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality. *Phys. Rev. E* 64 (Jun 2001), 016132. Issue 1. https://doi.org/10.1103/PhysRevE.64.016132

[19] M. E. J. Newman. 2018. *Networks: an introduction* (second ed.). Oxford University Press, Oxford, UK.

[20] M. E. J. Newman and M. Girvan. 2004. Finding and evaluating community structure in networks. *Phys. Rev. E* 69 (Feb 2004), 026113. Issue 2. https://doi.org/10.1103/PhysRevE.69.026113

[21] Krzysztof Nowicki and Tom A B Snijders. 2001. Estimation and prediction for stochastic blockstructures. *Journal of the American statistical association* 96, 455 (2001), 1077–1087.

[22] Tiago P. Peixoto. 2014. Hierarchical Block Structures and High-Resolution Model Selection in Large Networks. *Phys. Rev. X* 4 (Mar 2014), 011047. Issue 1. https://doi.org/10.1103/PhysRevX.4.011047

[23] Clara Pizzuti. 2008. Ga-net: A genetic algorithm for community detection in social networks. In *International conference on parallel problem solving from nature.* Springer, 1081–1090.

[24] Clara Pizzuti. 2017. Evolutionary computation for community detection in networks: A review. *IEEE Transactions on Evolutionary Computation* 22, 3 (2017), 464–483.

[25] Clara Pizzuti and Annalisa Socievole. 2017. Many-objective optimization for community detection in multi-layer networks. In *2017 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 411–418.

[26] Pascal Pons and Matthieu Latapy. 2006. Computing communities in large networks using random walks. In *J. Graph Algorithms Appl.* Citeseer.

[27] Robert Clay Prim. 1957. Shortest connection networks and some generalizations. *The Bell System Technical Journal* 36, 6 (1957), 1389–1401.

[28] Weixiang Shao, Lifang He, Chun-ta Lu, and S Yu Philip. 2016. Online multi-view clustering with incomplete views. In *2016 IEEE International Conference on Big Data (Big Data)*. IEEE, 1012–1017.

[29] Chang Xu, Dacheng Tao, and Chao Xu. 2013. A survey on multi-view learning. *arXiv preprint arXiv:1304.5634* (2013).

[30] Chang Xu, Dacheng Tao, and Chao Xu. 2015. Multi-view learning with incomplete views. *IEEE Transactions on Image Processing* 24, 12 (2015), 5812–5825.
[31] Yan Yang and Hao Wang. 2018. Multi-view clustering: A survey. *Big Data Mining and Analytics* 1, 2 (2018), 83–107.
[32] Weihua Zhan, Zhongzhi Zhang, Jihong Guan, and Shuigeng Zhou. 2011. Evolutionary method for finding communities in bipartite networks. *Physical Review E* 83, 6 (2011), 066120.