Artificial Intelligence Research and Development
A. Cortés et al. (Eds.)

doi:10.3233/FAIA220328

The Assessment of Clustering on Weighted Networks with R Package *clustAnalytics*

Argimiro ARRATIA [a,1] and Martí RENEDO-MIRAMBELL [a]
[a] *Soft Computing Research Group (SOCO)*
at Intelligent Data Science and Artificial Intelligence Research Center
Department of Computer Sciences,
Polytechnical University of Catalonia, Barcelona, Spain.
`argimiro@cs.upc.edu` `marti.renedo@gmail.com`

Abstract. We present *clustAnalytics*, an R package available now on CRAN, which provides methods to validate the results of clustering algorithms on unweighted and weighted networks, particularly for the cases where the existence of a community structure is unknown. *clustAnalytics* comprises a set of criteria for assessing the significance and stability of a clustering. To evaluate clusters' significance, *clustAnalytics* provides a set of community scoring functions, and systematically compares their values to those of a suitable null model. For this it employs a switching model to produce randomized graphs with weighted edges. To test for clusters' stability, a non parametric bootstrap method is used, together with similarity metrics derived from information theory and combinatorics. In order to assess the effectiveness of our clustering quality evaluation methods, we provide methods to synthetically generate networks (weighted or not) with a ground truth community structure based on the stochastic block model construction, as well as on a preferential attachment model, the latter producing networks with communities and scale-free degree distribution.

Keywords. clustering, networks, scoring functions, stochastic block model, non parametric bootstrap, R

1. Introduction

Clustering of networks is a popular research field, and a wide variety of algorithms have been proposed over the years. However, determining how meaningful the results are can often be difficult, as well as choosing which algorithm better suits a particular dataset. To help into this assessment and decision of clustering algorithms we have contributed to CRAN the new R package *clustAnalytics* [1], which contains a suite of novel methods to validate the partitions into communities of networks obtained by any given clustering algorithm. In particular, its clustering validation methods focus on two of the most important aspects of cluster assessment: the significance and the stability of the resulting clusters.

[1] Corresponding Author: Argimiro Arratia, e-mail: argimiro@cs.upc.edu

To assess the significance of communities structure, *clustAnalytics* has a collection of community scoring functions that measure some topological characteristics of the ground-truth communities, and whose values are compared against those obtained on null models with similar graph properties but without any expectations of a community structure. To evaluate stability, we designed and programmed in *clustAnalytics* a bootstrap technique with perturbations adapted to clustering on graphs. To compare how the clusters of the bootstrapped networks differ from the originals, three cluster similarity measures are provided: the adjusted Rand index, the Variation of Information, and the Reduced Mutual Information.

clustAnalytics handles weighted networks, as well as unweighted, and contains several other functionalities for producing different statistics on a network. It also contains methods for creating synthetic weighted networks based on the stochastic block model construction [2], and the preferential attachment model of Barabasi-Albert [3] that produce examples of ground-truth networks with community structure and degree distribution either binomial or scale-free. The mathematical and algorithmic aspects of *clustAnalytics* is explained in [4].

2. *clustAnalytics*: Examples of usage

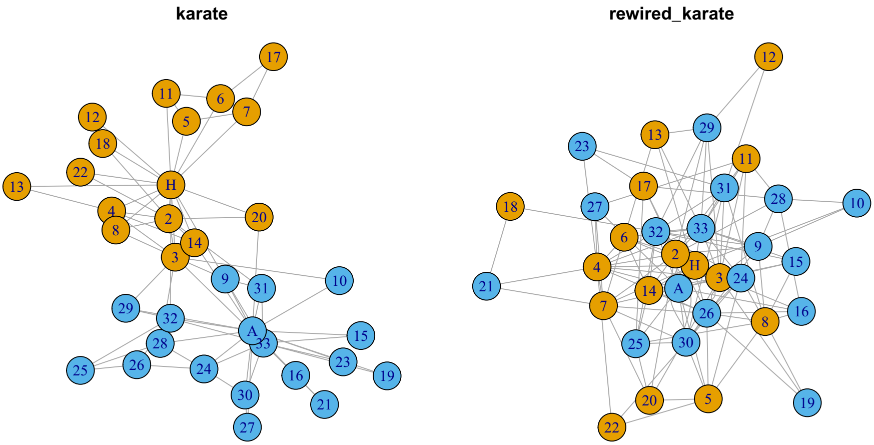
First to exhibit the graph randomization procedure programmed in *clustAnalytics*, we apply it to the Zachary's karate club graph, with the default settings (positive weights with no upper bound, which suits this graph):

```
> library(clustAnalytics)
> data(karate, package="igraphdata")
> rewired_karate <- rewireCpp(karate, weight_sel = "max_weight")
> par(mfrow=c(1,2), mai=c(0,0.1,0.3,0.1))
> plot(karate, main="karate")
> plot(rewired_karate, main="rewired_karate")
```

The function `rewireCpp` produces a random version of the original graph by rewiring the edges while keeping the degree distribution constant. The number of iterations is $Q \cdot \#edges = 100 \cdot 78$, where the parameter Q can be set by the user.

Cluster significance and stability. For gauging significance there is an ensemble of scoring functions in `evaluate_significance` which apply simultaneously to each of the clustering produced on a graph by a given list of algorithms. By default the clustering algorithms are Louvain, label propagation and Walktrap, but the function can take any list of clustering algorithms for *igraph* graphs. The function allows for comparison against ground-truth in case this is known. For the karate club graph this is known and we can include it in the analysis

```
#ground truth clusters for karate graph
> karate_gt_clustering <- c(1,1,1,1,1,1,1,1,1,2,1,1,1,1,2,2,1,1,
                          2,1,2,1,2,2,2,2,2,2,2,2,2,2,2,2)
> significance_table_karate <- evaluate_significance(karate,
                                ground_truth=TRUE,
                                gt_clustering=karate_gt_clustering)
> significance_table_karate
```


karate

rewired_karate

Figure 1. Karate club graph before and after the edge randomization process. Colors represent the faction of each participant, the *ground truth* clustering in this network.

This prints a table with all scores by the quality measures (Louvain, label propagation and Walktrap) including those for the ground-truth. Now we generate a graph from a stochastic block model in which we set very strong clusters: the elements in the diagonal of the matrix are much larger than the rest, so the probability of intra-cluster edges is much higher than that of inter-cluster edges.

```
> pm <- matrix (c(.3, .001, .001, .003,
                .001, .2, .005, .002,
                .001, .005, .2, .001,
                .003, .002, .001, .3), nrow=4, ncol=4)
> g_sbm <- sample_sbm(100, pref.matrix=pm,
                    block.sizes=c(25,25,25,25))
> E(g_sbm)$weight <- 1
> significance_table_sbm <- evaluate_significance(g_sbm)
> significance_table_sbm
```

We now assess for stability of clustering algorithms. Here we perform a nonparametric bootstrap to the karate club graph and the same selection of algorithms. For each instance, the set of vertices is resampled, the induced graph is obtained by taking the new set of vertices with the induced edges from the original graph, and the clustering algorithms are applied. These results are compared to the induced original clusterings using metrics: the variation of information (VI), normalized reduced mutual information (NRMI) and both adjusted and regular Rand index (Rand and adRand):

```
> b_karate <- boot_alg_list(g=karate, return_data=FALSE, R=99)
> b_karate
              Louvain label prop  walktrap
VI          0.2630337  0.2964607 0.2739508
NRMI        0.6957499  0.5447487 0.6698974
```

```
Rand       0.8558310  0.7849259 0.8460001
AdRand     0.6523059  0.5611139 0.6289277
n_clusters 5.6262626  4.9696970 5.8787879
```

And the same for the stochastic block model graph:

```
> b_sbm <- boot_alg_list(g=g_sbm, return_data=FALSE, R=99)
> b_sbm

             Louvain label prop  Walktrap
VI         0.1234341  0.1769217 0.1178832
NRMI       0.8536997  0.7841236 0.8656356
Rand       0.9411244  0.9230160 0.9472768
AdRand     0.8306925  0.7651778 0.8476909
n_clusters 6.9797980  7.7070707 7.4646465
```

We can clearly see that for all metrics, the results are much more stable, which makes sense because we created the sbm graph with very strong clusters.

Preferential attachment graphs with communities. The `barabasi_albert_blocks` function produces scale-free graphs using extended versions of the Barabsi-Albert model that include a community structure. The parameters that need to be set are m the number of new edges per step, the vector p of label probabilities, the fitness matrix B (with the same dimensions as the length of p), and t_max the final graph order. There are two variants of the model. If `type="Hajek"`, new edges are connected with preferential attachment to any existing vertex but using the appropriate values of B as weights. If ‘type=”block_first”‘, new edges are connected first to a community with probability proportional to the values of B, and then a vertex is chosen within that community with regular preferential attachment. In this case, the resulting degree distribution is scale-free (see [5] for a proof of this fact). This is a simple example with just two communities and a graph of order 100 and size 400:

```
> B <- matrix(c(1, 0.2, 0.2, 1), ncol=2)
> G <- barabasi_albert_blocks(m=4, p=c(0.5, 0.5), B=B, t_max=100,
                              type="Hajek",
                              sample_with_replacement = FALSE)
> plot(G, vertex.color=(V(G)$label),vertex.label=NA,vertex.size=10)
```

References

[1] Renedo-Mirambell M. clustAnalytics: Cluster Evaluation on Graphs; 2022. R package version 0.3.1. Available from: https://CRAN.R-project.org/package=clustAnalytics.

[2] Holland PW, Laskey KB, Leinhardt S. Stochastic blockmodels: First steps. Social networks. 1983;5(2):109-37.

[3] Barabási AL, Albert R. Emergence of Scaling in Random Networks. Science. 1999;286(5439):509-12. Available from: https://science.sciencemag.org/content/286/5439/509.

[4] Arratia A, Renedo-Mirambell M. Clustering assessment in weighted networks. PeerJ Computer Science. 2021;7:e600.

[5] Renedo-Mirambell M, Arratia A. Identifying bias in cluster quality metrics. arXiv:2112.06287; 2021. Available from: https://arxiv.org/abs/2112.06287.