Reformulations of the Map Equation for Community Finding and Blockmodelling

Neil Hurley
Insight Centre for Data Analytics
School of Computer Science
University College Dublin
Email: neil.hurley@insight-centre.org

Erika Duriakova
Insight Centre for Data Analytics
School of Computer Science
University College Dublin
Email: erika.duriakova@insight-centre.org

***Abstract*—Among the many community-finding algorithms that have been proposed in the last decade and more, the Infomap algorithm of Rosvall and Bergstrom has proven among the best. The algorithm finds good community structure in directed as well as undirected networks by abstracting information flow in the network as a random walk. In this paper, we reformulate the objective in terms of the Kullback-Leibler distance between the distribution of the random walk transitions and that of a *model* walk. The choice of model can be used to constrain the type of partition that the method extracts. This generalisation makes the method suitable for extracting other types of meso-structure from the network, enabling the analyst to explicitly control the type of extracted structure.**

I. Introduction

The Infomap method [1] is one of the best methods for finding non-overlapping community structure in complex networks [2]. Infomap finds communities by abstracting information flow in the network as a random walk over the nodes. It seeks a decomposition into modules, such that most of the transitions in this random walk occur within a module, rather than between two different modules. This is achieved by formulating the problem as one of finding a maximally compressed, two-level encoding of the random walk, where the two levels correspond to transitions over modules and nodes. The objective that the method optimises, the so called *Map Equation*, is a count of the average number of bits required to encode the random walk. In this paper, we generalise the Map Equation to allow other types of meso-scale structure to be extracted. Our reformulation focuses away from compression per se and instead identifies a *model* transition process of flow between the modules, which the algorithm seeks to identify. By inserting different models, we are then able to tune the type of structure found.

A. Notation

The methods discussed in this paper are concerned with a random walk over a graph, $G(V, E)$, with adjacency matrix A. The random walk is represented as an ergodic Markov Chain with transition matrix W $= \{w_{\alpha\beta}\}$, representing the probability of moving to node β, given that the walk is currently at α. The chain has a stationary distribution, denoted by p_α, the average proportion of time that the walk spends at node α. The nodes in the graph are partitioned into m *modules*, M=$\{1, \ldots, m\}$. We use roman letters $i, j, \ldots$, to represent module identifiers, and write $p_i \triangleq \sum_{\alpha \in i} p_\alpha$, as the *visit rate* to module i. We will use w_{ij} to represent a transition matrix of a random walk *between* modules, which has stationary distribution p_i.

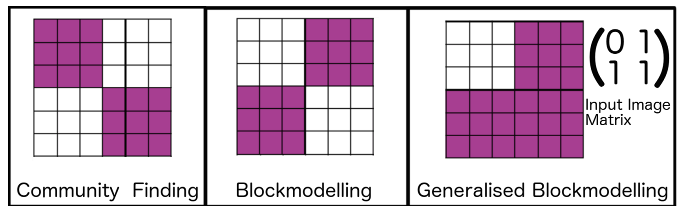


Fig. 1. Three problem formulations.

II. Community-Finding and Blockmodelling

There are three general problem formulations which are of interest, namely, community finding, blockmodelling and generalised blockmodelling. In the context of networks, we can define two nodes as similar to each other if they are connected to each other – this is community-finding. In terms of the adjacency matrix, a community structure has densely connected blocks along the diagonal with sparse connections between them (blocks off the diagonal), see figure 1. Another approach – *blockmodelling* – is to consider two nodes as similar if they have similar connection patterns to the clusters in the network. Blockmodelling seeks a partitioning where blocks are either sparse or dense, including the possibility of non-diagonal blocks being dense. In figure 1, the blockmodelling structure shows two modules that are very well connected to each other but with no internal connections. The nodes are clustered based on their common connections, rather than by being well connected to each other. In social science, *generalised blockmodelling* [3] describes an analysis technique that partitions the network into equivalence blocks. The analyst provides an *image matrix*, which describes the type of equivalence that is required for each block and the goal is to find a block pattern which conforms to this image. The generalised blockmodelling structure shown in figure 1 illustrates a core-periphery structure. Such structure identifies a group of well connected core nodes and a group of sparsely connected peripheral nodes. Unlike community structure, the core nodes are also well connected to the periphery nodes. In this work we extend the Map equation framework to extract not just the community structure, but also blockmodelling and generalised blockmodelling structure.

III. A General Form of The Map Equation

Rosvall's original Map Equation favours community partitions. Here we generalise the equation to allow different types of meso-structure to be identified. For a given partition M, the actual sequence of modules that are visited by the random walk is a Markov Chain with transition matrix:

$$w_{ij} = \sum_{\alpha \in i} p_\alpha \sum_{\beta \in j} w_{\alpha\beta}/p_i$$

and stationary probability p_i. Consider another ergodic between-module *model flow*, defined as a random walk over the modules of the graph with transition matrix w_{ij}^{mod} and stationary distribution p_i^{mod}. Write $p(i,j) = p_i w_{ij}$ as the joint distribution of visiting module i followed by module j in the random walk. We seek a module partitioning M, such that $p(i,j) \approx p^{\mathrm{mod}}(i,j)$ and measure the closeness of the distributions using the Kullback-Leibler divergence. Furthermore, we compare the distance between $p(i,j)$ and $p^{\mathrm{mod}}(i,j)$, with the distance of $p(i,j)$ to a *null model*, $p^{\mathrm{null}}(i,j)$, specifically taken as the stationary distribution of the random walk. Thus, we define the generalised Map Equation objective as:

$$\boxed{M^* = \arg\min_M \left(D(p||p^{\mathrm{mod}}) - D(p||p^{\mathrm{null}}) \right)}$$

A. Blockmodelling Flow

One obvious choice for the model flow is $w_{ij}^{\mathrm{mod}} = w_{ij}$ which results in an objective of the form:

$$M^* = \arg\min_M - \sum_i p_i w_{ij} \log(w_{ij}) + \sum_i p_i \log(p_i)$$

While this objective attains a minimum when each node is placed in a separate module, it can be minimised for a given input number of m modules. It is minimised when $w_{ij} \approx 1$ or $w_{ij} \approx 0$, resulting in sparse and dense block structures in the adjacency matrix i.e. a *blockmodel*.

B. Generalised Blockmodelling

In generalised blockmodelling, we take an *image* matrix as input and seek structure that matches the input image. We construct a model flow transition matrix with high transition values in the non-zero blocks of the image matrix e.g. for the image matrix of figure 1, where a two-module partitioning is required in order to identify *core-periphery* structure, we may choose the model transition matrix as e.g.

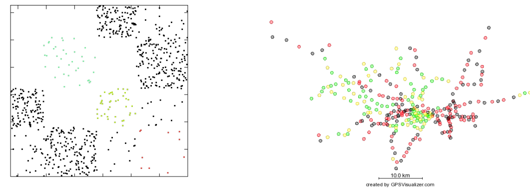
$$w_{ij}^{\mathrm{mod}} = \begin{pmatrix} 0.1 & 0.9 \\ 0.5 & 0.5 \end{pmatrix} \tag{1}$$

IV. Experiments

A. London Underground

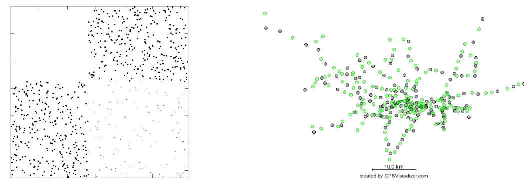
To demonstrate the use of the Generalised Map Equation to target non-community structure, we use the London underground network[1]. It consists of 312 London tube stations with a total of 724 unweighed, undirected connections between them.

[1]Available from http://www.tfl.gov.uk/info-for/open-data-users/our-feeds



(a) (b)

Fig. 2. Blockmodelling using the London Underground network.



(a) (b)

Fig. 3. Generalised blockmodelling using the London Underground network.

We firstly seek the best blockmodelling partition of the network for $m = 4$, we obtain the adjacency matrix shown in fig. 2(a). Now stations are clustered more so by their common connections than by the fact that they connect to each other. This is illustrated in figure 2(b), where we plot the stations based on their geographical location and colour them based on their module assignment. We also run the map equation for generalised blockmodelling using (1) and we expect to recover a core-periphery structure from this network. Figure 3(a) shows the resulting reorganised adjacency matrix of the London underground network based on the solution returned from our algorithm. Based on the matrix, we can conclude that our algorithm performed well since the matrix structure is very close to the input image matrix. However figure 3(b) shows that the sparse block is largely obtained by alternating the module assignment of stations along the different rail lines.

V. Conclusion

In this paper, we have reformulated and extended the Infomap method, so that the meso-scale structure extracted by the method can be controlled by the analyst. We have sketched how generalised blockmodelling can be carried out using this reformulation. We believe that a further examination of our method can provide a powerful means of querying the network for other types of structure.

Acknowledgment

The Insight Centre for Data Analytics is supported by Science Foundation Ireland under Grant Number SFI/12/RC/2289.

References

[1] M. Rosvall, D. Axelsson, and C. T. Bergstrom, "The map equation," *The European Physical Journal-Special Topics*, vol. 178, no. 1, pp. 13–23, 2009.

[2] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3-5, pp. 75 – 174, 2010.

[3] P. Doreian, V. Batagelj, and A. Ferligoj, *Generalized blockmodelling*. Cambridge University Press, 2005.