Relaxations in Practical Clustering and Blockmodeling

[Extended Abstract]

Stefan Wiesberg
Institut für Informatik
Im Neuenheimer Feld 368
Universität Heidelberg, Germany
stefan.wiesberg@informatik.uni-heidelberg.de

Gerhard Reinelt
Institut für Informatik
Im Neuenheimer Feld 368
Universität Heidelberg, Germany
gerhard.reinelt@informatik.uni-heidelberg.de

ABSTRACT

Network analysts try to explain the structure of complex networks by the partitioning of their nodes into groups. These groups are either required to be dense (clustering) or to contain vertices of equivalent positions (blockmodeling). However, there is a variety of definitions and quality measures to achieve the groupings. In surveys, only few mathematical connections between the various definitions are mentioned.

In this paper, we show that most of the definitions practically used can be seen as Lagrangean-type relaxations of four basic graph theoretical definitions. Quality measures like *Newman's modularity* can be seen as Lagrangean functions of these relaxations. The theory holds for both clustering and blockmodeling. It can be used as the basis of a methodological analysis of different practical approaches.

Keywords

network analysis, clustering, blockmodeling

1. INTRODUCTION

To group the vertices of a complex network's graph $G = (V, E)$ is one established way to obtain information about its structure. *Clustering* is the best-known example of this kind. Here, every vertex group is required to induce a dense sub graph in G. Other examples are *structural* and *regular equivalence*. Here, vertices with equivalent structural positions are grouped; a notion to be defined below. In the latter cases, not only the groups themselves, but also the relations between them are of interest.

Formally, a vertex grouping is defined as a disjoint partition $\mathbf{V} = \{V_1, \dots, V_c\}$ of the vertex set V. The partition is often expressed in terms of a coloring. This is possible since every vertex coloring $\phi : V \to [c]$, where $[c] = \{1, 2, \dots, c\}$, naturally defines a partition into the color classes. W.l.o.g. we assume that ϕ is surjective, i.e., all c colors are used.

2. DEFINITIONS TO BE RELAXED

We will now discuss several ways to define the vertex groups in ways that turned out to be meaningful in network analysis. These ways are *clique coloring (CC), clique component coloring (CCC), structural coloring (SC)* and *regular coloring (RC)*. In the next sections we will see that these strict definitions are usually relaxed for practical computations. A vertex coloring ϕ with exactly k colors is called a *k-clique coloring* if every vertex $u \in V$ is adjacent to all other vertices of the same color $\phi(u)$. We call the coloring a *k-clique component coloring* if every vertex $u \in V$ is adjacent to all other vertices of the same color $\phi(u)$ and to no vertex of any other color. The latter colorings are used for clustering measures in which not only the density of the clusters, but also the sparsity of inter-cluster relations are taken into account. Another well-studied way is the grouping of vertices which have the very same neighbors. The corresponding mathematical definition of *k-structural colorings* is introduced by Lorrain and White [7]. Here, the same color of two vertices $u, v \in V$ implies that u and v have the same neighbors in G: $N(u) \setminus \{v\} = N(v) \setminus \{u\}$, where $N(u)$ denotes the set of vertices in V which are adjacent to u. The number k again represents the number of used colors. A fourth common way of grouping is based on *k-regular colorings*. A coloring $\phi : V \to [k]$ is *k-regular*, if for every two vertices $u, v \in V$ with $\phi(u) = \phi(v)$ holds:

$$\{\phi(w) \mid w \in V, uw \in E\} = \{\phi(w) \mid w \in V, vw \in E\}.$$

That is, if u and v have the same color, then their neighboring color sets are identical.

There is a well-known characterization of the four coloring types. It describes certain sub graphs of G. Given a coloring ϕ, there is one such sub graph $G_{\phi,c,d}$ for every pair (c, d) of colors. It is obtained from G by deleting all vertices but the ones colored with c or d and deleting all edges but those connecting an c-colored with a d-colored vertex. $G_{\phi,c,d}$ is hence bipartite for $c \neq d$. Note that all of these sub graphs are edge disjoint.

THEOREM 1. *Given a graph G, a k-coloring $\phi : V \to [k]$ of its vertex set is...*

(i) a k-clique coloring, if for all $c \in [k]$, the graph $G_{\phi,c,c}$ is complete.

(ii) a k-clique component coloring, if for all $c, d \in [k]$, the graph $G_{\phi,c,d}$ is complete if $c = d$ and empty if $c \neq d$.

(iii) a k-structural coloring, if for all $c, d \in [k]$, the graph $G_{\phi,c,d}$ is either empty or a complete (bipartite for $c \neq d$) graph.

(iv) a k-regular coloring, if for all $c, d \in [k]$, the graph $G_{\phi,c,d}$ is either empty or contains no isolated vertices.

3. LAGRANGEAN TYPE RELAXATIONS

We will now show that in practical network analysis, relaxations of the four definitions above are usually used. We denote by $CP(k, G)$ the set of all k-clique colorings of the vertices of G. Analogously, we define $CPP(k, G)$, $SE(k, G)$ and $RE(k, G)$, or simply write $X(G)$ in a statement which holds for any fixed type and any fixed number k of colors. Practitioners, often implicitly, enlarge the set of feasible colorings $X(G)$ to a set $X_L(G) \supseteq X(G)$ and assign a penalty value $p(\phi) \geq 0$ to each member ϕ of the enlarged set $X_L(G)$. Afterwards, they solve the optimization problem of finding a coloring ϕ^* in $X_L(G)$ with the minimum penalty value $p(\phi^*)$. We now show that this is usually done in a Langrangean way. That is, the set $X(G)$ of feasible colorings is enlarged by dropping some of the requirements in the definition of X. Furthermore, the penalty function p is not arbitrary, but measures the degree of violation against the *dropped* requirements. The optimization problem to be solved is thus:

(MIN-P) Given the set $X_L(G)$ and the penalty function $p : X_L(G) \to \mathbb{R}_0^+$, find a $\phi^* \in X_L(G)$ which minimizes p.

That is, among the colorings satisfying the non-dropped requirements, find the one which violates the dropped requirements to the least possible extent. As a convention, a penalty value of 0 is assigned to the colorings in $X(G)$, as they do not violate any dropped requirements (*compatibility requirement*, see Doreian et. al. [5]). Hence, a coloring satisfying the original definition X is always an optimum solution to (MIN-P). We will now classify literature by the type of relaxation used. As we are considering colorings of graphs, two types of relaxations are suggesting: The relaxation of the coloring definition and the relaxation of the graph structure. Indeed, these possibilities are widely used. In Section 3.2, we will look at the cases where the general definition of coloring is relaxed. E.g., a vertex might be allowed be assigned several colors here. Section 3.3 subsequently addresses the case where the structure of the graph G is relaxed. We start, however, with a third possibility. There, the definition specific properties of CC, CCC, SC, and RC are being relaxed.

3.1 Problem Specific Relaxations

For problem specific relaxations, two forms of p are most commonly used: p is constant or decomposable over the set of all vertex pairs. In the first case, say $p \equiv 0$, some requirements are simply dropped, but not penalized.

Example (Sociometric Cliques.) Alba [1] finds the graph theoretical definition of *clique* to be not perfectly appropriate to describe friendship (or sociometric) cliques in social networks. He thus relaxes its definition to so-called n-cliques, where the distance of two members is not at most 1, but at most n. That is, the requirements "The distance is at most i" are dropped for $i = 1, \ldots, n-1$. If no penalties are introduced, the problem (MIN-P) merely consists in the search for *any* partition into n-cliques.

A second common type of problem specific relaxations is based on *vertex similarities*. In this special case of (MIN-P), the penalty function p can be decomposed over all vertex pairs, i.e., $p(\phi) = \sum_{u,v \in V} p_{uv}\delta(\phi(u), \phi(v))$. Here, $p_{uv} \geq 0$ are real numbers and δ denotes the Kronecker function. It is 1 if $\phi(u) = \phi(v)$ and 0 otherwise. In literature, the numbers p_{uv} are often called *(dis)similarity values*. The relaxation technique of using such a decomposable function is called *indirect blockmodeling approach* by Doreian et al. [5].

Example (Structural Equivalence.) Several functions p of the above form have been proposed for SE. This propositions were made indirectly by a specification of the values p_{uv}. They quantify how much a coloring violates this dropped requirement. That is, to quantify how similar two vertices are with respect to common neighbors. See Leicht, Holme, and Newman [6] for an overview on these functions.

3.2 Coloring Relaxations

Beside the definition specific relaxations above there is the possibility to relax the definition of "coloring" itself. If we use the binary variables x_{vc} to express whether vertex v is colored with c ($x_{vc} = 1$) or not ($x_{vc} = 0$), the requirement "to be a k-coloring" can be decomposed into the following sub requirements:

$$\sum_{c=1}^{k} x_{vc} = 1 \quad \text{for all } v \in V, \tag{1}$$

$$\sum_{v \in V} x_{vc} \geq 1 \quad \text{for all } c \in [k], \tag{2}$$

$$0 \leq x_{vc} \leq 1 \quad \text{for all } v \in V, c \in [k], \tag{3}$$

$$x_{vc} \in \mathbb{Z} \quad \text{for all } v \in V, c \in [k]. \tag{4}$$

Example (Fuzzy Colorings.) In some applications, it is meaningful for a vertex to get several colors at the same time. E.g., a person might be a member of several clubs. In this case, requirement (1) is dropped. Alternatively, a vertex might be allowed to consist of color fractions that sum up to 1, such as 50% red, 30% green and 20% blue. In this case, requirement (4) is dropped. One speaks of *fuzzy colorings* or *partitionings* in both of these cases of relaxation. The penalty function p is usually constant with respect to these requirements.

Example (Number of Colors.) For many applications, a good choice for the number of colors is not a priori known and hence not fixed to a certain value k. That is, the requirement that k colors must be used is dropped. As small numbers are usually more suitable for interpretation, the penalty function p might be defined to assign each coloring ϕ the number of colors used by ϕ. The lower the number of colors, the less the amount of penalty. As an example, the algorithm CATREGE [3] solves (MIN-P) for such a p and X=RE. I.e., given a graph, it finds a k-regular coloring with the smallest possible k.

3.3 Graph Relaxations

Assume a practitioner is interested in 4-regular colorings on a given graph $G = (V, E)$. However, such a coloring does not

exist on G. It is then reasonable to consider a 4-coloring ϕ to be a good solution, if it is not regular on G, but turns regular if G is changed by a very small amount. Following this idea, the best 4-coloring is the one that requires the lowest amount of changes in G in order to become regular. Possible changes are usually the deletion and addition of edges. That is, requirements of the forms "$uv \in E$" and "$uv \notin E$" are dropped. If they are penalized by the function p, then the coloring ϕ^* which requires the lowest amount of edge changes in G will be the optimal solution to (MIN-P). In order to define a suitable penalty function p, we first need to define a function d to measure the amount of edge changes. More precisely, d measures the distance of two graphs $G = (V, E)$ and $H = (V, F)$ on the same vertex set V. A simple but common exemplary form of such a d is given by

$$d(G, H) = \sum_{u,v \in V, u \neq v} |A(G)_{u,v} - A(H)_{u,v}|, \qquad (5)$$

where A denotes the adjacency matrix of the graph. The function counts the number of different entries in the adjacency matrices of G and H. More complex distance functions are discussed below. d measures the distance of G to a single graph H. We can also measure its distance to a set of graphs $\mathbf{H}$, by defining the distance $d(G, \mathbf{H})$ as the distance of G to its closest element in $\mathbf{H}$. That is,

$$d(G, \mathbf{H}) := \min_{H \in \mathbf{H}} d(G, H).$$

To measure how much G has to be changed, it is compared to sets of ideal graphs $\mathcal{H}(\phi)$, on which ϕ perfectly satisfies the requirements. In our example, $\mathcal{H}(\phi)$ is defined such that ϕ is 4-regular on all $H \in \mathcal{H}(\phi)$. The penalty function for (MIN-P) is hence $p(\phi) = d(G, \mathcal{H}(\phi))$.

We now give more details on this procedure. First, we will see how ideal graphs $\mathcal{H}(\phi)$ can be defined. Then, we give an overview on the distance functions $d(G, H)$ which are practically used. Afterwards, a common variant of this procedure is discussed, which does not relax G, but several sub graphs of G simultaneously. We close by some examples on how graph relaxation is used in literature.

Ideal, Worst and Average Graphs

Given an ideal coloring definition X (for example CC, CCC, SC, RC), a graph $G = (V, E)$ and a coloring ϕ of its vertices, the set $\mathcal{H}(\phi)$ of ideal graphs can be naturally defined. It is the set of all graphs H with the same vertex set as G, such that ϕ is an X-coloring on H. Theorem 1 gives a characterization of these graphs. In the case of clustering (X = CC), the ideal graphs are those in which vertices of the same color induce cliques. Note that for every $\phi : V \to C$, the set $\mathcal{H}(\phi)$ is non-empty.

Alternatively, one can define $\mathcal{H}(\phi)$ to be the set of *worst* graphs instead of ideal ones. Worst graphs can be easily defined for CC, CCC, and SE. This is because their sub graph characterization in Proposition 1 use empty and complete graphs only. As "being empty" and "being complete" are opposite extremes, one can define worst graphs by interchanging the words "empty" and "complete" in the proposition. E. g., in a worst graph for clustering (CC) no cluster contains any edges. If worst graphs are used, the distance of the closest graph to G needs to be maximized instead of minimized.

A third alternative has been used for CC and SE: G is compared to average graphs. For clustering, the sub graphs are hence neither empty nor complete, but have an average density. The distance of G to the average graphs $\mathcal{H}(\phi)$ can then be positive or negative, depending on whether G is worse (sparser) or better (denser) than average. The same holds hence for the penalty function. It is usually used as a reward function $\overline{p}$: The farther G is from average in the positive direction, the larger is $\overline{p}$, the better is ϕ.

Overview on Distance Functions

With (5), we already stated the most simple distance function to measure the distance between two graphs on the same vertex set. It counts the number of edges to be added or deleted (changed) in G to obtain the ideal graph H. The adjacency matrix of H is possibly weighted, especially when G is compared to an average graph. Practically, these functions are usually weighted. Clearly, the amount of change that is introduced by the change of an adjacency matrix entry is often not the same for every entry. There are two common reasons. Firstly, the amount of change can depend on the weight of the edge. That is, the deletion of a heavier weighted edge might be considered a greater change. Secondly, a variation in the amount of change can also evolve from a priori statistical computations. Adding an edge which is unlikely to exist can be considered more expensive than adding a likely edge. Hence, edge depended weights $c_{uv} \geq 0$ are commonly used in distance functions.

Another kind of weight is based on the following observation. Changing edges between yellow and red vertices can be modeled to be more expensive than changes between yellow and blue ones. To this end, scaling factors $w_{cd} \geq 0$ are introduced for all pairs of colors. The resulting weighted distance function can be stated as

$$d(G, H) = \sum_{u,v \in V, u \neq v} w_{\phi(u)\phi(v)} \cdot c_{uv} \cdot |A(G)_{u,v} - A(H)_{u,v}|. \quad (6)$$

The absolute value function is a problem for the comparison to average graphs. Here, we want to distinguish whether G is worse or better than average. Hence, the following function is more suitable in this case.

$$d(G, H) = \sum_{u,v \in V, u \neq v} w_{\phi(u)\phi(v)} \cdot c_{uv} \cdot (A(G)_{u,v} - A(H)_{u,v}). \quad (7)$$

Beside these linear functions, several non-polynomial functions have been proposed. Being derived from general statistical matrix correlation measures, they can be used to compare the adjacency matrices of G and H. See Wasserman and Faust [9] or Arabie et al. [2] for an overview.

Relaxing Sub Graphs Separately

We have seen in Theorem 1 that the coloring conditions can be equivalently formulated as requirements for the sub graphs $G_{\phi,c,d}$ of G. In the widely used *direct blockmodeling approach*, these sub graphs are relaxed separately. That is, there is a separate penalty value for each sub graph. However, the same distance function d us used for each sub graph. Whether the separate relaxations of the sub graphs is equivalent to the relaxation of G itself depends on the choice of d. In direct blockmodeling, we have single penalty values $p_{cd}(\phi) = d(G_{\phi,c,d}, \mathbf{H}_{\phi,c,d})$ for the sub graphs. They need to be combined to a total penalty value $p(\phi)$. In

most cases, the p_{cd} are simply summed up:

$$p(\phi) = \sum_{c,d \in C} p_{cd}(\phi). \quad (8)$$

For clustering (CC), the sum runs clearly only over those c, d with $c = d$. If scaling is used, the factor is usually $1/m_{cd}$, where m_{cd} is the number of possible edges in the sub graph $G_{\phi,c,d}$. More precisely, $m_{cd} = |c| \cdot |d|$ if $c \neq d$ and $m_{cc} = |c| \cdot (|c| - 1)$.

$$p(\phi) = \sum_{c,d \in C} \frac{1}{m_{cd}} \cdot p_{cd}(\phi). \quad (9)$$

In some statistically based approaches, the squares of the penalties are summed up instead. This mostly occurs so-called chi-squared approaches.

$$p(\phi) = \sum_{c,d \in C} (p_{cd}(\phi))^2. \quad (10)$$

Besides the above scaling factor, a second one can be used here. The distance of $G_{\phi,c,d}$ to $H_{\phi,c,d}$ can be seen in relation to the maximum distance $d^{\max}_{\phi,c,d}$ of any graph, on the same vertex set, to $H_{\phi,c,d}$.

$$p(\phi) = \sum_{c,d \in C} m_{cd} \cdot \left(\frac{p_{cd}(\phi)}{d^{\max}_{\phi,c,d}}\right)^2. \quad (11)$$

Examples

We now give some examples on how this kind of relaxation is used in literature, either for coloring type CC, CCC, SC, or RC. For each example, we need to specify the following modeling choices:

- Whether ideal, worst, or average graphs are used and how average is defined.
- Which distance function d is used.
- Whether G or its sub graphs $G_{\phi,c,d}$ are relaxed and how $p(\phi)$ is combined by the $p_{cd}(\phi)$.

Example (Maximal Cluster Density.) A basic measure for the quality of a clustering (X=CC) on $G = (V, E)$ is the sum over all *intra-cluster densities* $\delta_{int}(V_i)$. They give the proportion of actual edges to theoretically possible edges within the i-th cluster:

$$\delta_{int}(V_i) = \frac{\#\text{ internal edges of } V_i}{|V_i|(|V_i| - 1)/2}.$$

The search for a coloring ϕ^* with maximum total intra-cluster density can be expressed as (MIN-P). To see this, compare G to the standard ideal graphs $H(\phi)$ with the most basic distance function (5). Apply it separately on each sub graph $G_{\phi,c,c}$ and combine the penalty values linearly by formula (9).

Example (Maximal Structural Density.) Wasserman and Faust explain a simple measure for structural colorings in their survey [9]. It is a generalization of the preceding example from clique to structural colorings. For each pair c, d of colors, they sum up the values $|I_{cd} - \Delta_{cd}|$. Here, I denotes the image matrix and Δ_{cd} denotes the density. The density is defined as the number of edges from c-colored to d-colored vertices, divided by the maximum possible number m_{cd} of such edges. They hence compare the sub graphs $G_{\phi,c,d}$ to ideal graphs with the most basic distance function (5). Again, the separate penalties are combined linearly by formula (9).

Example (Newman-Girvan-Modularity.) Newman and Girvan [8] present a well-known relaxation for clustering. They choose $H(\phi)$ to contain average graphs. More precisely, $H(\phi)$ consists of exactly one graph $H = (V, F)$. The edge weight of $uv \in F$ is $deg(u)deg(v)/2|E|$. This is precisely the probability of the edge to exist in a random graph with the same degree distribution as G. For this reason, H can be interpreted as the average graph w.r.t. to the degree distribution of G. The distance function is the unweighted version of (7). It is applied separately to each $G_{\phi,c,c}$. These values are summed-up by the again unweighted formula (8) to obtain the total penalty $p(\phi)$. Note that $p(\phi)/2|E|$ is called the *modularity* of ϕ. The factor $1/2|E|$ is however constant and can thus be ignored in the solution of (MIN-P). Other so-called *Newman-like modularities* can be modeled analogously.

Example (Berkowitz-Carrington-Heil Index.) The index [4] is designed for structural colorings (X=SC). It compares G to an average graph H. The user is asked to specify an average density α from the interval between 0 and 1. H is then the complete graph with edge weights all α, letting its density equal α. The distance function d is (5), hence the most simple one. It is applied on sub graphs. Since the index is a chi-squared approach, the function $p(\phi)$ is composed as in (11).

4. REFERENCES

[1] R. D. Alba. A graph-theoretic definition of a sociometric clique. *Journal of Mathematical Sociology*, 3(1):113–126, 1973.

[2] P. Arabie, S. A. Boorman, and P. R. Levitt. Constructing blockmodels: How and why. *Journal of mathematical psychology*, 17(1):21–63, 1978.

[3] S. P. Borgatti and M. G. Everett. Two algorithms for computing regular equivalence. *Social Networks*, 15(4):361–376, 1993.

[4] P. J. Carrington, G. H. Heil, and S. D. Berkowitz. A goodness-of-fit index for blockmodels. *Social Networks*, 2(3):219–234, 1980.

[5] P. Doreian, V. Batagelj, and A. Ferligoj. *Generalized blockmodeling*, volume 25. Cambridge University Press, 2005.

[6] E. Leicht, P. Holme, and M. Newman. Vertex similarity in networks. *Physical Review E*, 73(2):026120, 2006.

[7] F. Lorrain and H. C. White. Structural equivalence of individuals in social networks. *The Journal of mathematical sociology*, 1(1):49–80, 1971.

[8] M. E. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

[9] S. Wasserman and K. Faust. *Social network analysis: Methods and applications*, volume 8. Cambridge university press, 1994.