# Structure-aware Distance Measures for Comparing Clusterings in Graphs

Jeffrey Chan*, Nguyen Xuan Vinh, Wei Liu, James Bailey, Christopher Leckie, Kotagiri Ramamohanarao, and Jian Pei

[1] Department of Computing and Information Systems, University of Melbourne, Australia
[2] School of Computing Science, Simon Fraser University, BC Canada

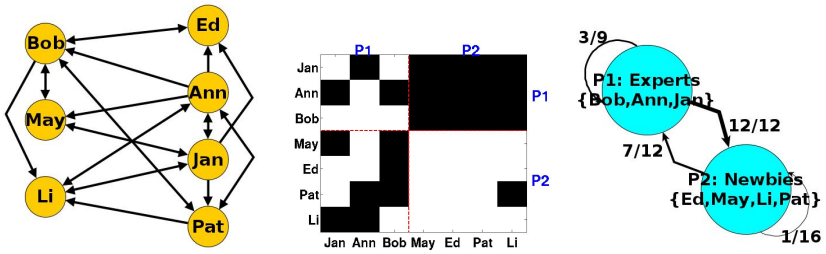
**Abstract.** Clustering in graphs aims to group vertices with similar patterns of connections. Applications include discovering communities and latent structures in graphs. Many algorithms have been proposed to find graph clusterings, but an open problem is the need for suitable comparison measures to quantitatively validate these algorithms, performing consensus clustering and to track evolving (graph) clusters across time. To date, most comparison measures have focused on comparing the vertex groupings, and completely ignore the difference in the structural approximations in the clusterings, which can lead to counter-intuitive comparisons. In this paper, we propose new measures that account for differences in the approximations. We focus on comparison measures for two important graph clustering approaches, community detection and blockmodelling, and propose comparison measures that work for weighted (and unweighted) graphs.

**Keywords:** blockmodelling; community; clustering; comparison; structural; weighted graphs

## 1 Introduction

Many data are relational, including friendship graphs, communication networks and protein-protein interaction networks. An important type of analysis that is graph clustering, which involves grouping the vertices based on the similarity of their connectivity. Graph clusterings are used to discover communities with similar interests (for marketing) and discovering the inherent structure of graphs. Despite the popularity of graph clustering, existing graph cluster comparison measures have focused on using membership based measures. To the best of the authors' knowledge, there are no work that evaluates if membership based comparison measures are appropriate for comparing graph clusterings. This analysis is important, as comparison measures are often used for comparing algorithms [1][2], form the basis of consensus clustering [3] and tracking algorithms [4], and knowing which properties measures possess allows us to evaluate if a measure is appropriate for a task, and if not, propose new ones that are. One of

* Corresponding author: jeffrey.chan@unimelb.edu.au




(a) Graph. (b) Adjacency matrix. (c) Image diagram.

Fig. 1: Example of an online Q&A forum. Each vertex corresponds to a forum user, and each edge represents replying between users (Figure 1a). For the corresponding adjacency matrix (Figure 1b), the positions induce a division of the adjacency matrix into blocks, delineated by the red dotted lines. Edge weights in Figure 1c are the expected edge probabilities between each pair of positions.

the important properties a measure should possess is the ability to be "spatially"/structurally aware. In traditional clustering of points, it was found that clusterings could be similar in memberships, but their points are distributed very differently. This could lead to counter-intuitive situations where such clusterings were considered similar [5][6]. We show the same scenario occurs when comparing graph clusterings. Therefore, in this paper, we address the open problems of analysing and showing why membership comparison measures can be inadequate for comparing graph clusterings and then proposing new measures that are aware of structural differences. We shortly illustrate why comparison measures should be structure aware, but we first introduce graph clustering.

*Clustering in Graphs – Community Detection and Blockmodelling:* Two popular approaches for graph clustering are community detection [7] and blockmodelling [8]. These approaches are used in many important clustering applications [9][1][2][4], and hence we focus on the comparison of communities and blockmodels in this paper. Community detection decomposes a graph into a community structure, where vertices from the same communities have many edges between themselves and vertices of different communities have few edges. Community structure has been found in many graphs, but it is only one of many alternatives for grouping vertices and possible graph structure (such as core-periphery). In contrast, an alternative approach is blockmodelling [8]. A blockmodel[3] partitions the set of vertices into groups (called *positions*), where for any pair of positions, there are either many edges, or few edges, between the positions.

As an example of the difference between the two approaches, consider a reply graph between a group of experts and questioners in a question and answer (Q&A) forum, illustrated in Figure 1. The vertices represent users, and the directed edges represent one user replying to another. If we use a popular community detection algorithm [7] to decompose the graph, all the vertices are

[3] We use the popular structural equivalence to assess similarity [8].

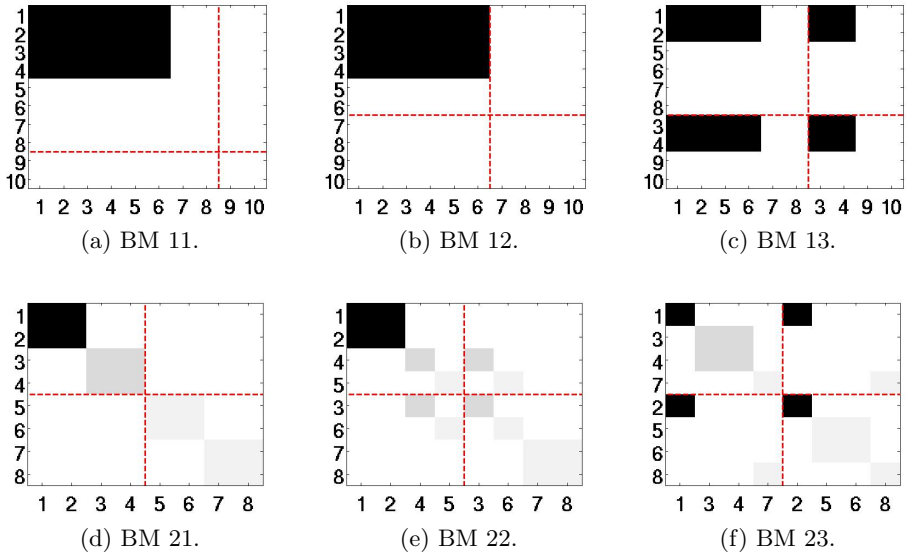
incorrectly assigned into a single group, as its underlying structure is not of a community type, and no structure is discerned. If we use a blockmodelling decomposition, we obtain the correct decomposition into two positions, the experts $C_1$ ({Jan, Ann, Bob}) and the questioners $C_2$ ({May, Ed, Pat, Li}) (see Figure 1b, which illustrates the adjacency matrix rearranged according to the positions). The experts $C_1$ reply to questions from the questioners ($C_1$ to $C_2$), the questioners $C_2$ have their questions answered by the experts ($C_1$ to $C_2$) and both groups seldom converse among themselves. The overall structure can be succinctly summarised by the image diagram in Figure 1c, which shows the positions as vertices, and the position-to-position aggregated interactions as edges. As can be seen, this graph has a two position bipartite structure, which the blockmodelling decomposition can find but a community decomposition fails to discover. The similarity definition of blockmodelling actually includes the community one, hence blockmodelling can be considered as a generalisation of community detection and comparison measures that work with blockmodels would work with comparing communities as well. Therefore we focus on measures that compare blockmodels, since they can also compare communities.

*Comparing Communities and Blockmodels:* As discussed earlier, almost all existing literature compares blockmodels (and communities) based on their positions and ignores any difference in the adjacency structure within positions and between positions. This can lead to unintuitive and undesirable behaviour. A blockmodel comparison measure should possess the three following properties, which we will introduce and motivate using two examples.

The first property is that the measure should be *sensitive to adjacency differences in the blocks*. Figures 2a (BM 11), 2b (BM 12) and 2c (BM 13) illustrate example 1. The vertices in each blockmodel are ordered according to their positions, and the boundary between the positions in the adjacency matrix is illustrated with red dotted lines. This effectively divides the adjacency matrix into *blocks*, which represent the position to position interactions. The positional differences from BM 12 and BM 13 to BM 11 are the same. However, the distribution of edges (frequency of edges and non-edges) are very different between BM 11 and BM 13 (there is one single dense block with all the edges in BM 11 (and BM 12), while the edges are distributed across the blocks more evenly in BM 13). Hence a measure should indicate BM 11 is more similar to BM 12 than to BM 13. Positional measures fail to achieve this.

The second property is that a measure should *account for differences in the edge distributions across all blocks*. In example 1, BM 11 and BM 12 have similar distributions across all blocks since their edges are concentrated in a block, while BM 13 is different because the edges are spread quite evenly across all blocks. We show that two existing comparison measures for blockmodels fail this property.

The third property is that a measure should be *sensitive to weighted edge distributions*. Many graphs have weights on the vertices and edges. It might be possible to binarise the weights, but this throws away important comparison information. For example, if the graph in the blockmodels of Figures 2d to 2f



(a) BM 11. (b) BM 12. (c) BM 13.

(d) BM 21. (e) BM 22. (f) BM 23.

Fig. 2: Example to illustrate the strength and weaknesses of different blockmodel comparison measures. Each row corresponds to 3 blockmodels of the same graph.

(BM 21 to BM 23, dubbed example 2) were binarised, then BM 22 and 23 have exactly the same edge distribution differences from BM 21. But when the actual weights are taken into account (darker pixels in the figures represent larger valued weights), BM 22 will be correctly considered as more similar to BM 21, as most of the high-value edge weights in the top left block match, while in BM 23 these edges are evenly distributed among the four blocks.

These properties have important applications. For example in algorithm evaluation, the blockmodel output of the algorithms are compared to a gold standard. Imagine that the gold standard is BM 11 and two algorithms produced BM 12 and 13. Measures without these properties will incorrectly rank the two algorithms to be equally accurate since they have the same position differences to BM 11. In contrast, a measure that possess the first two properties will correctly rank the algorithm producing BM 12 as more accurate. Another application is in consensus blockmodelling, which finds a blockmodel that is the average of a set of blockmodels. If a measure does not possess the 3rd property and only considered position similarity, then the consensus blockmodel might have very different weight distribution to the other blockmodels and hence should not be considered as a consensus (e.g., BM 23 as a consensus of BM 21 and 22).

In summary, our contributions are: a) we propose three structural-based properties that a blockmodel comparison measure should possess; b) we analyse existing measures and show that they do not possess these properties; c) we propose new measures that satisfy these properties; and d) perform experiments on synthetic and real data to study the monotonicity of the new measures.

## 2 Related Work

In this section, we describe related work in three key areas: set comparison, spatially aware clustering comparison and subspace clustering comparison.

***Set Comparison:*** There has been extensive work in using set comparison for cluster comparison. Hence we discuss a few selected measures, and refer interested readers to the excellent survey of [10]. The first class of measures in this area involves computing the agreement between two clusterings in terms of how many pairs of vertices are in the same and different clusters. Examples include the Rand and Jaccard indices [10]. The second class of measures are based on set matching and information theory, where the two clusterings are compared as two sets of sets. Popular examples include Normalised and Adjusted Mutual Information (NMI, AMI) [11]. As demonstrated in Section 1, these set-based measures do not take into account any adjacency differences between the blockmodels, resulting in some counter-intuitive comparison behaviour.
***Spatially Aware Clustering Comparison:*** In [12], Zhou et al. proposed a measure that compares clusters based on membership and the distances between their centroids. In [5], Bae et al. computed the density of clusters using a grid, and then used the cosine similarity measure to compare the cluster distributions. Coen et al. [6] took a similar approach but used a transportation distance to compute the distances between clusters and between clusterings. All these measures depend on a notion of a distance between points (between clusters of the two clusterings). In blockmodel and community comparison, there are positions of vertices and the edges between them, but no notion of a distance between vertices across two blockmodels and hence existing spatial-aware measures cannot be applied for blockmodel comparison.
***Subspace Clustering Comparison:*** In subspace clustering, the aim is to find a group of objects that are close in a subset of the feature space. In [13], Patrikainen and Meila proposed the first subspace validation measure that considered both the similarities in the object clusters and in the subspaces that the clusters occupied. They treated subspace clusters as sets of (object, feature) pairs, and then used set-based validation measures for comparing subspace clusters. When extended to compare the sets of (vertex,vertex) pairs between blockmodels, these measures are equivalent to comparing the positions (proof omitted due to lack of space), and hence have the same issues as the set comparison measures.

In summary, there has been much related work in cluster comparison, but none that address the unique problem of comparing blockmodels and communities. What is needed is a measure that is sensitive to differences in the adjacency structure as well as working in the relation spaces of graphs.

## 3 Blockmodelling (Graph Clustering) Background and Properties

In this section, we summarise the key ideas of blockmodelling (see [14][8] for more detail). A graph $G(V, E)$ consists of a set of vertices $V$ and a set of edges $E$,

where $E \in \{0,1\}^{|V|\times|V|}$ for unweighted graphs and $E \in \mathcal{N}^{|V|\times|V|}$ for weighted graphs[4]. The edge relation can be represented by an adjacency matrix $\mathbf{A}$ whose rows and columns are indexed by the vertices of $G$.

We use the Q&A example of Figure 1 to illustrate the notation. A blockmodel partitions a set of vertices into a set of positions $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$. We denote the number of positions in $\mathcal{C}$ by $k$. $\mathcal{C}$ can be alternatively specified by $\phi(.)$, a mapping function from vertices to the set of positions (i.e., $\phi : V \rightarrow \mathcal{C}$). A block $\mathbf{A}_{r,c}$ is a submatrix of $\mathbf{A}$, with the rows and columns drawn from $C_r$ and $C_c$ respectively, $C_r, C_c \in \mathcal{C}$, e.g., $\mathbf{A}_{1,1}$ defines the upper left submatrix in Figure 1b. Let $\Psi_{r,c}$ denote the random variable representing the probability mass function of the edge weights in block $\mathbf{A}_{r,c}$, e.g., $p(\Psi_{1,1} = 0) = \frac{7}{9}$, $p(\Psi_{1,1} = 1) = \frac{2}{9}$. This representation allow us to model both weighted and unweighted graphs. For simplicity, we introduce $\Upsilon_{r,c} = p(\Psi_{r,c} = 1)$ for unweighed graphs. We define $\mathbf{M}$ as the blockmodel image matrix that models inter-position densities, $\mathbf{M} : \mathcal{C} \times \mathcal{C} \rightarrow [0,1]$, $\mathbf{M}_{r,c} = \Upsilon_{r,c}$. For the rest of the paper, we use a superscript notation to distinguish two different instances of a variable (e.g., $G^{(1)}$ and $G^{(2)}$).

A blockmodel $\mathfrak{B}^{(l)}(\mathbf{C}^{(l)}, \mathbf{M}^{(l)})$ is defined by its set of positions $\mathcal{C}^{(l)}$ (the matrix version is $\mathbf{C}^{(l)}$) and its image matrix $\mathbf{M}^{(l)}$. The (unweighted) blockmodelling problem can be considered as finding a blockmodel approximation of $\mathbf{A}^{(l)}$ as $\mathbf{C}^{(l)}\mathbf{M}^{(l)}(\mathbf{C}^{(l)})^T$ that minimises a sum of squared errors[5][15], $||\mathbf{A}^{(l)} - \mathbf{C}^{(l)}\mathbf{M}^{(l)}(\mathbf{C}^{(l)})^T||^2$. We denote $\mathbf{C}^{(l)}\mathbf{M}^{(l)}(\mathbf{C}^{(l)})^T$ as $\hat{\mathbf{A}}^{(l)}$. Given two blockmodels , the distance between two blockmodels $\mathfrak{B}^{(1)}$ and $\mathfrak{B}^{(2)}$ is defined as $d(\mathfrak{B}^{(1)}, \mathfrak{B}^{(2)})$.

### 3.1 Desired Properties of Comparison Measures

In this section, we formalise the properties that a blockmodel comparison measure should possess (as discussed in Section 1). These properties allow us to formally evaluate the measures in the next section.

The following properties assume that there are three blockmodels with approximations $\hat{\mathbf{A}}^{(1)}$, $\hat{\mathbf{A}}^{(2)}$ and $\hat{\mathbf{A}}^{(3)}$ of the same graph, $\hat{\mathbf{A}}^{(1)} \neq \hat{\mathbf{A}}^{(2)} \neq \hat{\mathbf{A}}^{(3)}$ and they are not co-linear, i.e., $|\hat{\mathbf{A}}^{(1)} - \hat{\mathbf{A}}^{(3)}| \neq |\hat{\mathbf{A}}^{(1)} - \hat{\mathbf{A}}^{(2)}| + |\hat{\mathbf{A}}^{(2)} - \hat{\mathbf{A}}^{(3)}|$. They can be considered as measuring the sensitivity to differences in the structure (approximation) of the blockmodels.

P1: Approximation sensitivity: Given an **unweighted** graph and three blockmodels, a measure is approximation sensitive if $d(\hat{\mathbf{A}}^{(1)}, \hat{\mathbf{A}}^{(2)}) \neq d(\hat{\mathbf{A}}^{(2)}, \hat{\mathbf{A}}^{(3)})$.

P2: Block edge distribution sensitivity: Given $d_{mKL}(\hat{\mathbf{A}}^{(2)}, \hat{\mathbf{A}}^{(1)}) < d_{mKL}(\hat{\mathbf{A}}^{(3)}, \hat{\mathbf{A}}^{(1)})$, then a measure is block edge distribution sensitive if $d(\hat{\mathbf{A}}^{(2)}, \hat{\mathbf{A}}^{(1)}) < d(\hat{\mathbf{A}}^{(3)}, \hat{\mathbf{A}}^{(1)})$. $d_{mKL}(\hat{\mathbf{A}}^{(1)}, \hat{\mathbf{A}}^{(2)})$ is the KL divergence [16] for matrices, and is defined as $d_{mKL}(\hat{\mathbf{A}}^{(1)}, \hat{\mathbf{A}}^{(2)}) = \sum_{i,j}^{|V^1|} \hat{\mathbf{A}}^{(1)}_{i,j} \log\left(\frac{\hat{\mathbf{A}}^{(1)}_{i,j}}{\hat{\mathbf{A}}^{(2)}_{i,j}}\right) - \hat{\mathbf{A}}^{(1)}_{i,j} + \hat{\mathbf{A}}^{(2)}_{i,j}$. It measures the difference in the edge (weight) distribution across all the blocks.

[4] For simplicity, we assume a discrete set of weights. But this can easily be extended to continuous weight values.

[5] This is one of several popular blockmodelling objective formulations. See [1] for an objective for weighted graph.

P3: Weight sensitivity: Given a **weighted** graph and three blockmodels[6], a measure is weight sensitive if $d(\hat{\mathbf{A}}^{(1)}, \hat{\mathbf{A}}^{(2)}) \neq d(\hat{\mathbf{A}}^{(2)}, \hat{\mathbf{A}}^{(3)})$.

In addition, we evaluate the important monotonicity property of the measures. Basically, we desire measures that increase in value as the compared blockmodels become more different. We measure "more different" by the minimum number of position changes to transform one blockmodel to another.

## 4 Blockmodel (Graph Clustering) Comparison Approaches

In this section, we describe existing measures, propose new blockmodel comparison approaches, and analyse their properties. Existing work for comparing blockmodels falls into two categories: *positional* and *reconstruction* measures. Positional measures compare the sets of positions associated with each blockmodel [10]. Reconstruction measures compare the blockmodel approximation of the graphs and can be expressed as $d(\mathfrak{B}^{(1)}, \mathfrak{B}^{(2)}) = d(\hat{\mathbf{A}}^{(1)}, \hat{\mathbf{A}}^{(2)})$. We next provide details on two existing reconstruction blockmodel measures.

### 4.1 Edge and Block Reconstruction Distances

The edge and block reconstruction distances were proposed in [4] and [8] respectively. The edge reconstruction distance [8] measures the difference in the expected edge probabilities across all edges (recall that $V^1 = V^2$):

$$d_{RE}(\mathfrak{B}^{(1)}, \mathfrak{B}^{(2)}) = \sum_{i}^{|V^1|} \sum_{j}^{|V^2|} |\Upsilon^{(1)}_{\phi(i),\phi(j)} - \Upsilon^{(2)}_{\phi(i),\phi(j)}| \tag{1}$$

The block reconstruction distance [4] measures the difference in block densities over all pairs of blocks, weighted by the overlap of the positions:

$$d_{RB}(\mathfrak{B}^{(1)}, \mathfrak{B}^{(2)}) = \sum_{r1,c1}^{k1} \sum_{r2,c2}^{k2} \frac{|C^{(1)}_{r1} \cap C^{(2)}_{r2}|}{n} \frac{|C^{(1)}_{c1} \cap C^{(2)}_{c2}|}{n} \cdot |\Upsilon^{(1)}_{r1,c1} - \Upsilon^{(2)}_{r2,c2}| \tag{2}$$

The two measures in fact differ only by a factor of $\frac{1}{n^2}$ (we prove this new result in Theorem 1). It is easier to understand and propose new measures based on the edge than the block reconstruction distance. But in terms of computational complexity, it takes $O(k^2)$ to compute the block reconstruction distance while $O(n^2)$ for the edge distance, hence it is more efficient to compute the block distance. Therefore Theorem 1 permits us to use whichever measure that is more convenient for the task at hand.

**Theorem 1.** $d_{RB}(\mathfrak{B}^{(1)}, \mathfrak{B}^{(2)}) = \frac{1}{n^2} d_{RE}(\mathfrak{B}^{(1)}, \mathfrak{B}^{(2)})$

[6] $\hat{\mathbf{A}}$ consists of $n^2$ PMFs of a random variable with the event space of edge weights.

*Proof.* The proof involves arithmetric manipulation. We detail the proof in a supplementary[7] paper due to space limitations.

Both these reconstruction distances possess the approximation sensitivity property (P1) but fail the block edge distribution property (P2). Reconsider the example of Figures 2a to 2c. $d_{RB}(BM11, BM12) = 0.21$ and $d_{RB}(BM11, BM13) = 0.18$. This means that the reconstruction distances incorrectly ranks BM 13 to be closer to BM 11 than BM 12. To explain why, we first state that the Earth Movers Distance (EMD) can be considered as a generalisation of the reconstruction measures (see Section 4.3). The EMD finds the minimum amount of mass to move from one PMF (of a block) to another. What this means is that the reconstruction distance considers the cost to move a unit of mass the same, whether the source PMF is a unimodal distribution or uniformly distributed one. Hence the reconstruction distances only consider the number of total units of mass moved and do not consider the differences in distribution of edge densities across all the blocks, leading them to fail property P2.

### 4.2 KL Reconstruction Measure

We propose to use the KL divergence [17] to compare the block densities (PMFs). Although it is similar in form to the P2 property definition, the proposed measure is in the form of the reconstruction distance and the KL divergence is a natural approach to measure distribution differences. It is defined as:

**Definition 1.** ***KL Reconstruction:***

$$d_{RKL}(\mathfrak{B}^{(1)}, \mathfrak{B}^{(2)}) = \sum_{r1,c1}^{\mathcal{C}^{(1)}} \sum_{r2,c2}^{\mathcal{C}^{(2)}} \frac{|C_{r1} \cap C_{r2}|}{n} \frac{|C_{c1} \cap C_{c2}|}{n} \cdot d_{KL}(\mathbf{A}^{(1)}_{r1,c1}, \mathbf{A}^{(2)}_{r2,c2}) \quad (3)$$

where $d_{KL}(\Psi^{(1)}_{r1,c1}, \Psi^{(2)}_{r2,c2}) = \sum_x p(\Psi^{(1)}_{r1,c1} = x) \log(\frac{p(\Psi^{(1)}_{r1,c1}=x)}{p(\Psi^{(2)}_{r2,c2}=x)}$.

The KL reconstruction measure can be considered as using the edge distributions (across the blocks) in $\mathfrak{B}^{(2)}$ to encode the edge distributions in $\mathfrak{B}^{(1)}$. This means it is sensitive to differences in the distribution of the block densities, and it gives the correct ranking for example 1 (see Section 5).

The KL divergence is asymmetric, hence $d_{RKL}(.)$ is also asymmetric, and can handle weighted graphs. The Jeffrey and Jenson-Shannon divergences [17] are symmetric versions of the KL divergence, and are included for comparison. Unfortunately they fail the block edge distribution property (see Section 5).

### 4.3 Weighted Block and Edge Reconstruction

In this section, we show how the edge/block reconstruction measures can be generalised to weighted graphs. We compare the blockmodels of weighted graphs

[7] Available at `http://people.eng.unimelb.edu.au/jeffreyc/`

by comparing the PMFs of the blocks. We desire a measure that can compare multi-valued PMFs and consider the difference in the actual weight values. One such measure is the Earth Mover's distance (EMD) [18], which also reduces to the reconstruction measures for unweighted graphs (see Theorem 2).

**Definition 2.** ***EMD Reconstruction Measure:***

$$d_{REMD}(\mathfrak{B}^{(1)}, \mathfrak{B}^{(2)}) = \sum_{r1,c1}^{\mathcal{C}^{(1)}} \sum_{r2,c2}^{\mathcal{C}^{(2)}} \frac{|C_{r1} \cap C_{r2}|}{n} \frac{|C_{c1} \cap C_{c2}|}{n} \cdot d_{ED}(\Psi^{(1)}_{r1,c1}, \Psi^{(2)}_{r2,c2}) \quad (4)$$

where $d_{ED}(\Psi^{(1)}_{x,y}, \Psi^{(2)}_{a,b}) = \min_{\mathbf{M}} \sum_{u=0}^{L} \sum_{w=0}^{L} m_{u,w} \cdot d(u,w)$, subject to:
**a)** $m_{u,e} \geq 0$; **b)** $\sum_w m_{u,w} = p(\Psi^{(1)}_{x,y} = u)$; **c)** $\sum_u m_{u,w} = p(\Psi^{(2)}_{a,b} = w)$ and **d)** $d(u,w) = |u - w|$.

We now show that the EMD reconstruction measure is in fact a generalisation of the reconstruction measures.

**Theorem 2.** *When we are comparing unweighted graphs,*

$$d_{ED}(\mathbf{A}^{(1)}_{x,y}, \mathbf{A}^{(2)}_{a,b}) = |\Upsilon^{(1)}_{x,y} - \Upsilon^{(2)}_{a,b}|$$

*and* $d_{REMD}(\mathfrak{B}^{(1)}, \mathfrak{B}^{(2)}) = d_{RB}(\mathfrak{B}^{(1)}, \mathfrak{B}^{(2)})$.

*Proof.* The proof involves arithmetic manipulation and reasoning on the constraints. Due to space limitations, please refer to supplementary for details.

Theorem 2 is a useful result, as it helps to explain why the block reconstruction distance fail property P2. In addition, it means EMD reconstruction distance can be used in place of block distance, since Theorem 2 tells us that the EMD distance is a generalisation of block one for unweighted graphs. At the same time, the EMD distance satisfies property P3 while the block one does not.

## 5 Evaluation of the Measures

In this section, we evaluate the measures using the proposed properties and empirically demonstrate their monotonicity (since it is difficult to prove this analytically). We also show how each of the measures perform in the examples from Section 1. In the experiments, we compare NMI, a popular and representative example of the positional measures, against the block (RB), EMD (REMD), KL (RKL), Jeffrey (RsKL) and JS (RJS) reconstruction distances.

### 5.1 Evaluation of the Examples

Table 1 shows the comparison results between the original blockmodel (BM *1) against the two other blockmodels (BM *2 and *3) of the examples in Figure 2. As can be seen, NMI cannot distinguish between the three blockmodels across all the datasets. RB fails to correctly rank the blockmodels of example 1, and fails

| | Example 2 | | Example 3 | |
|---|---|---|---|---|
| | **BM 12** | **BM 13** | **BM 22** | **BM 23** |
| NMI | 0.3845 | 0.3845 | 0.1887 | 0.1887 |
| RB | 0.2100 | 0.1800 | 0.2188 | 0.2188 |
| REMD | 0.2100 | 0.1800 | 7.2266 | 10.5469 |
| RKL | 0.2803 | 4.4432 | 7.2407 | 7.2407 |
| RsKL | 5.5650 | 4.6501 | 9.0060 | 9.0060 |
| RJS | 0.1676 | 0.1259 | 0.2633 | 0.2633 |

Table 1: Measure values when comparing the original blockmodel (BM*1) against the other blockmodels in the Karate club and other examples from Section 1. In each case, ideally $d(\text{BM } *2, \text{BM } *1)$ should be less than $d(\text{BM } *3, \text{BM } *1)$.

to distinguish the weighted blockmodels of example 2. As expected, REMD has the same values as RB for example 1, but correctly classifies the relative ordering of example 2. RsKL and RJS fail example 1. RKL correctly distinguishes the ordered example 1, but like the other distributional measures (RsKL and RJS) cannot distinguish the blockmodels of example 2.

### 5.2 Monotonicity Analysis

To evaluate monotonicity, we vary the membership of the positions for several real datasets. Each position change corresponds to a change in the membership of a vertex, and we ensure a vertex can only change membership once in each simulation. From a starting blockmodel, we generated 100 different runs, where up to 50% of the vertices change position. Table 2 shows the statistics of the three real datasets[8] on which we evaluate monotonicity.

| **Dataset** | **Vert #** | **Edge #** | **Weighted?** | **Pos. #** |
|---|---|---|---|---|
| Karate | 34 | 78 | N | 2 |
| Football | 115 | 613 | N | 12 |
| C.Elegans | 297 | 2359 | Y | 5 |

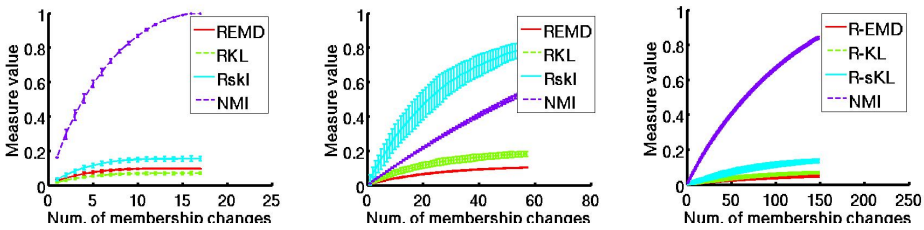
Table 2: Statistics of the real datasets.

Figure 3 shows the results for REMD, RKL, RsKL and NMI (we plotted 1-NMI). As can be seen, all measures are monotonically increasing as the positions change. However, in the rare case where the adjacency approximation remains the same after a position splits into two, then the reconstruction distances can fail to distinguish the pre-split and post-split blockmodels.

### 5.3 Properties of the Measures

Table 3 shows the measures and the properties they have. For each measure, we prove (see supplementary) whether it possesses each of the properties.

[8] Available at `http://www.personal-umich.edu/~mejn/netdata`.




(a) Karate club. (b) Football. (c) C.Elegans.

Fig. 3: Evaluation of the monotonicity of the distance measures as the difference in the positions of the starting blockmodel and modified blockmodel increases.

| Property | NMI | RB | REMD | RKL | RsKL | RJS |
|---|---|---|---|---|---|---|
| P1: Edge. Dist. Sensitivity | - | ✓ | ✓ | ✓ | ✓ | ✓ |
| P2: Block Dist. Sensitivity | - | - | - | ✓ | - | - |
| P3: Weight Sensitivity | - | - | ✓ | -* | -* | -* |
| Monotonicity | ✓ | ✓# | ✓# | ✓# | ✓# | ✓# |

Table 3: List of blockmodel measures and their properties. *: The KL-based measures can fail when the distributions have the same shape but have different weight values. #: not strictly monotonic (can fail the co-incidence axiom).

Table 3 confirms the empirical evaluation of Section 5.1. The positional measures (e.g., NMI) do not consider blockmodel approximations and hence fail the structural sensitivity properties. The existing RB and RE distances cannot be applied to weighted graphs and are not block edge distribution sensitive. The proposed REMD generalises the reconstruction distances to weighted graphs, but still possesses the same assumptions as those distances, and hence fails the block edge distribution sensitivity property. The proposed KL-based distances are block edge distribution sensitive, but not weight sensitive as they measure distribution differences but ignore difference in weight values.

From these analyses, we recommend to use the KL reconstruction distance when comparing unweighted blockmodels, as it possess the first two properties. When comparing weighted graphs, the EMD distance might be preferred as it satisfies the weight sensitivity property. A further option is to combine several measures together via ideas from multi-objective optimisation, e.g., as a weighted linear sum, which we leave to future work.

## 6 Conclusion

Blockmodel comparison measures are used for validating blockmodelling and community detection algorithms, finding consensus blockmodels and other tasks that require the comparison of blockmodels or communities. In this paper, we have shown that popular positional measures cannot distinguish important differences in blockmodel structure and approximations because they do not re-

flect a number of key structural properties. We have also proposed two new measures, one based on the EMD and the other based on KL divergence. We formally proved these new measures possess a number of the desired structural properties and used empirical experiments to show they are monotonic.

Future work includes introducing new measures for evaluating mixed membership blockmodels [1] and evaluating multi-objective optimisation approaches for combining measures.

## References


1. Airoldi, E.M., Blei, D.M., Fienberg, S.E., Xing, E.P.: Mixed membership stochastic blockmodels. J. of Machine Learning Research **9** (June 2008) 1981–2014
2. Pinkert, S., Schultz, J., Reichardt, J.: Protein interaction networks–more than mere modules. PLoS computational biology **6**(1) (2010) e1000659
3. Lancichinetti, A., Fortunato, S.: Consensus clustering in complex networks. Nature **2**(336) (2012)
4. Chan, J., Liu, W., Leckie, C., Bailey, J., Kotagiri, R.: SeqiBloc: Mining Multi-time Spanning Blockmodels in Dynamic Graphs. In: Proceedings of KDD. (2012)
5. Bae, E., Bailey, J., Dong, G.: A clustering comparison measure using density profiles and its application to the discovery of alternate clusterings. Data Mining and Knowledge Discovery **21**(3) (2010) 427–471
6. Coen, M.H., Ansari, H.M., Filllmore, N.: Comparing Clusterings in Space. In: Proceedings of ICML. (2010) 231–238
7. Rosvall, M., Bergstrom, C.T.: Maps of random walks on complex networks reveal community structure. Proceedings of PNAS **105** (2008) 1118–1123
8. Wasserman, S., Faust, K.: Social Network Analysis: Methods and Applications. Cambridge Univ. Press (1994)
9. Chan, J., Lam, S., Hayes, C.: Increasing the Scalability of the Fitting of Generalised Block Models for Social Networks. In: Proceedings of IJCAI. (2011)
10. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: On Clustering Validation Techniques. J. of Intelligent Information Systems **17**(2/3) (2001) 107–145
11. Vinh, N., Epps, J., Bailey, J.: Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. The J. of Machine Learning Research **11** (2010) 2837–2854
12. Zhou, D., Li, J., Zha, H.: A new Mallows distance based metric for comparing clusterings. In: Proceedings of the ICDM. (2005) 1028–1035
13. Patrikainen, A., Meila, M.: Comparing Subspace Clusterings. IEEE Trans. on Know. Eng. **18**(7) (2006) 902–916
14. Doreian, P., Batagelj, V., Ferligoj, A.: Generalized blockmodeling. Cambridge Univ. Press (2005)
15. Chan, J., Liu, W., Kan, A., Leckie, C., Bailey, J., Kotagiri, R.: Discovering latent blockmodels in sparse and noisy graphs using non-negative matrix factorisation. In: Proceedings of CIKM. (2013) To appear.
16. Lee, D., Seung, H.: Algorithms for non-negative matrix factorization. In: Proceedings of NIPS. (2000) 556–562
17. Cover, T., Thomas, J.: Elements of Information Theory. Wiley-Interscience (2006)
18. Rubner, Y., Tomasi, C., Guibas, L.: The earth mover's distance as a metric for image retrieval. International Journal of Computer Vision **40**(2) (2000) 99–121