Modelling Zeros in Blockmodelling

Laurence A. F. Park[1(✉)], Mohadeseh Ganji[2], Emir Demirovic[3], Jeffrey Chan[4], Peter Stuckey[5], James Bailey[6], Christopher Leckie[6], and Rao Kotagiri[6]

[1] Centre for Research in Mathematics and Data Science, Western Sydney University, Sydney, Australia
lapark@westernsydney.edu.au
[2] ANZ, Melbourne, Australia
[3] TU Delft, Delft, The Netherlands
[4] School of Computing Technologies, RMIT University, Melbourne, Australia
[5] Department of Data Science and AI, Monash University, Clayton, Australia
[6] School of Computing and Information Systems, The University of Melbourne, Parkville, Australia

Abstract. Blockmodelling is the process of determining community structure in a graph. Real graphs contain noise and so it is up to the blockmodelling method to allow for this noise and reconstruct the most likely role memberships and role relationships. Relationships are encoded in a graph using the absence and presence of edges. Two objects are considered similar if they each have edges to a third object. However, the information provided by missing edges is ambiguous and therefore can be measured in different ways. In this article, we examine the effect of the choice of block metric on blockmodelling accuracy and find that data relationships can be position based or set based. We hypothesise that this is due to the data containing either Hamming noise or Jaccard noise. Experiments performed on simulated data show that when no noise is present, the accuracy is independent of the choice of metric. But when noise is introduced, high accuracy results are obtained when the choice of metric matches the type of noise.

1 Introduction

Relationships between objects can be represented as a graph, where the graph vertices represent the objects and the edges represent the relationships between the objects. Many algorithms have been proposed for clustering/partitioning graph vertices based on their relationships (e.g. Spectral Clustering [9]). These algorithms allow us to identify clusters of objects that are closely related and are useful for tasks such as identifying a group of employees who work in the same department, a group of people who attended the same school, or a set of video games that are made by the same company.

Graphs also contain a deeper level of information that allows us to identify the roles of the objects. Roles are not identified by the similarity of objects, but they are identified by the relationships that the objects share with others. For example a set of employees within a department might have the role of Manager. Each manager is not likely to be connected to each other, but their relationships to others in the department are likely to be similar (each is likely to be acting as gateway between senior management and the other employees within the department).

J. Gama et al. (Eds.): PAKDD 2022, LNAI 13281, pp. 187–198, 2022.
https://doi.org/10.1007/978-3-031-05936-0_15

Blockmodelling allows us to discover clusters of objects that have the same or similar role in the graph. The name comes from its process of revealing blocks within the graph adjacency matrix, where a block is a set of objects that share the same links to other objects in the graph. It is not clear, however, if the similarity between two objects should be stronger if they have missing edges in common.

In this article, we examine the effect of the chosen block metric on blockmodelling accuracy. Experiments show that high accuracy is obtained using position based block metrics on some data and set based block metrics on other data. We hypothesise that these results are due to the noise within the data being either Hamming or Jaccard noise and run simulations examine this hypothesis.

The contributions of this article are:

- A presentation of a seriation based approach for blockbodelling, allowing block metric selection (Sect. 2).
- An analysis of the effect of Hamming and Jaccard noise when using each metric on specific block structures, and their interaction with the number of observations, number of roles and noise level (Sect. 3).

We also identify that block metrics from the same category behave similarly, and therefore, we conjecture that relational data can be either position based or set based, and that the block metric should be chosen to match the data noise type.

The article is organised as follows: Sect. 2 describes the initial investigation in to the effect of block metric on blockmodel accuracy. Section 3 continues the investigation by examining if the effect of the metrics are due to the type of noise. Section 4 examines the results.

2 Blockmodelling with a Chosen Block Metric

A graph G with vertices V and edges E, can be represented by its adjacency matrix A, where each element $a_{ij} \in A$ depicts the weight of the edge directed from vertex j to i. If many vertices have the same in or out edges, they form a block in A, where the block represents a role (a set of objects that have similar relationships to the remainder of the graph). If the rows and columns of A are ordered correctly, we are able to visualise the block, unfortunately identifying the correct permutation is difficult and so the existence of a particular block may not be obvious.

Both Stochastic [5] and Spectral [8] forms of blockmodels exist, where the stochastic form allows us to identify the underlying sampling distribution, while the spectral form provides a hard or soft clustering of objects into roles. We focus on the spectral form. The adjacency matrix $A \in \{0,1\}^{n \times n}$ of a blockmodel with k roles, by definition, can be decomposed into $A = CMC'$ where $M \in \{0,1\}^{k \times k}$ contains the blockmodel structure and $C \in \{0,1\}^{n \times k}$ contains the membership of each of n objects to one of the k roles (such that the rows contain one 1 and the rest 0). Many methods of approximating this decomposition have been derived as gradient based optimisation problems that compute C and M by minimising a function of the error [10]. But the optimisation is difficult due to the binary nature of the problem [2]. Current methods in Spectral Blockmodelling [1] encourage sparsity by separately weighting errors on absent and present edges.

2.1 Blockmodelling Metric

Blockmodelling can be thought of as clustering based on secondary relationships; two objects are found in the same block if they have the same relationship to a third object. If the relationships between objects are binary, then the associated graph provides an edge for a weight of 1 and no edge for a weight of 0. If we find that objects x_1 and x_2 both have edges to x_3, then the similarity between x_1 and x_2 should increase. If both x_1 and x_2 don't have edges to x_3, it is not clear if that information should increase or decrease their similarity. For example if the edge to x_3 represents if an object likes x_3, then no edge from both x_1 and x_2 show that they both don't like x_3, which increases their similarity. On the other hand, if an edge to x_3 represents if an object knows x_3, then it is unclear how both x_1 and x_2 not knowing the person (having no edge to x_3) influences their similarity.

To examine this problem, we will investigate the effect of block metric choice on blockmodelling accuracy. This requires us to easily change the block metric without effecting the blockmodelling algorithm. Therefore we will perform blockmodelling using Seriation, taking inspiration from the cluster visualisation family xVAT [6], which permute the rows and columns of a relational matrix to visualise clustering and identify the number of clusters.

The structural matrix M shows association between roles. The membership matrix C simply replicates and permutes the structure in M to form the graph adjacency matrix A. If the rows of the membership matrix were ordered such that objects with the same role membership are placed together, we would see the shape of the structural matrix in A. Unfortunately, C is not likely to be ordered, and so the structure is difficult to observe in A. But this implies that if we apply the correct permutation π to the rows of C to obtain πC, or equivalently, the rows and columns of A to obtain $\pi A \pi^t$, then we can easily recover M and πC from the visible block structure and hence C. Discovery of this permutation π is a *seriation* problem [3].

Seriation is the process of computing a permutation for a set of objects, such that the similarity between each object and its neighbours is maximised. Given an appropriate measure of similarity, we are able to reveal the block structure and expose roles using seriation. Unfortunately, seriation only provides the permutation of the objects; further processing of the adjacency matrix is required to cluster the objects, but we know that the clustering can be performed by partitioning the ordered set of objects.

A common method for seriation is to use hierarchical clustering with optimal leaf ordering [4]. This is a two stage process, where 1) hierarchical clustering is applied to the dissimilarity matrix (based on a given metric), then 2) the permutation provided by the dendrogram is optimised by maximising the similarity between each pair of neighbouring objects. This reordering process is performed by swapping the children at nodes of the dendrogram, ensuring that all objects remain in the clusters they were assigned to. For example, the hierarchical clustering $\{\{\{1,2\},\{3\}\},\{4,5\}\}$ can be permuted to $\{\{4,5\},\{\{2,1\},\{3\}\}\}$, where the clustering has not changed, but the similarity between each point and its neighbour has changed. To obtain the set of block clusters, the process is:

1. Create a dissimilarity matrix of the objects, where the dissimilarity is measured in terms of the object connectivity,

2. Apply hierarchical clustering to the dissimilarity matrix to obtain a dendrogram,
3. Reorder the dendrogram leaves using optimal leaf ordering,
4. Partition the dendrogram leaves into clusters.

We must decide upon a metric for our data; candidates are presented in the following section.

2.2 Candidate Block Metrics

To perform blockmodelling, we cluster objects that have similar roles, implying similar in and out links in the graph. Therefore, a blockmodelling metric must compare the in and out links of a pair of objects. We define the blockmodelling distance between vertex v_x and v_y as

$$\Delta(v_x, v_y) = d(\vec{e}_{\cdot,x}, \vec{e}_{\cdot,y}) + d(\vec{e}_{x,\cdot}, \vec{e}_{y,\cdot})$$

where $\vec{e}_{\cdot,x} = [e_{1,x}\ e_{2,x}\ \ldots\ e_{N,x}]$ (the xth column of the adjacency matrix) is the vector of edge weights $e_{i,x}$ directed from vertex v_x to vertex v_i, and $\vec{e}_{x,\cdot} = [e_{x,1}\ e_{x,2}\ \ldots\ e_{x,N}]$ (the xth row of the adjacency matrix) is the vector of edge weights $e_{x,i}$ directed from vertex v_i to vertex v_x (if no edge exists, the weight is zero).

The graphs we will be examining are unweighted, therefore $\vec{e}_{\cdot,x}$ and $\vec{e}_{x,\cdot}$ will be binary vectors, or set membership vectors. We will examine the position based metrics Hamming and Euclidean, and the set based metrics Cosine, Jaccard and Dice, as candidates for $d(\cdot,\cdot)$.

Position based	Set based
$d_{\text{Ham}}(\vec{x},\vec{y}) = \frac{1}{N}\|\vec{x}-\vec{y}\|_1$	$d_{\text{Jac}}(\vec{x},\vec{y}) = \frac{\|\vec{x}-\vec{y}\|_1}{N-(\vec{1}-\vec{x})'(\vec{1}-\vec{y})}$
$d_{\text{Euc}}(\vec{x},\vec{y}) = \|\vec{x}-\vec{y}\|_2$	$d_{\text{Cos}}(\vec{x},\vec{y}) = 1-\frac{\vec{x}'\vec{y}}{\|\vec{x}\|_2\|\vec{y}\|_2}$
	$d_{\text{Dic}}(\vec{x},\vec{y}) = \frac{\|\vec{x}-\vec{y}\|_1}{N-(\vec{1}-\vec{x})'(\vec{1}-\vec{y})+\vec{x}'\vec{y}}$

where $\vec{x}$ and $\vec{y} \in \{0,1\}^N$ are binary vectors (containing either 0 or 1), N is the vector length, $\|\vec{x}\|_1$ is the l_1 norm of $\vec{x}$, and $\|\vec{x}\|_2$ is the l_2 norm of $\vec{x}$. Note that both Hamming and Euclidean metrics treat vectors as positions, where Cosine, Jaccard and Dice treat the vectors as representing sets. The major difference between these two categories is how they treat zeros.

2.3 Effect of Blockmodelling Metric on Real Data

We begin our investigation by examining how the choice of block metric effects the blockmodelling accuracy on real data commonly used in assessing community structure algorithms. The data used in this experiment (Sampson: $n = 18$, $k = 4$; Polbooks: $n = 105$, $k = 2$; Polblogs: $n = 1490$, $k = 2$; Karate: $n = 34$, $k = 2$; Football: $n = 115$,

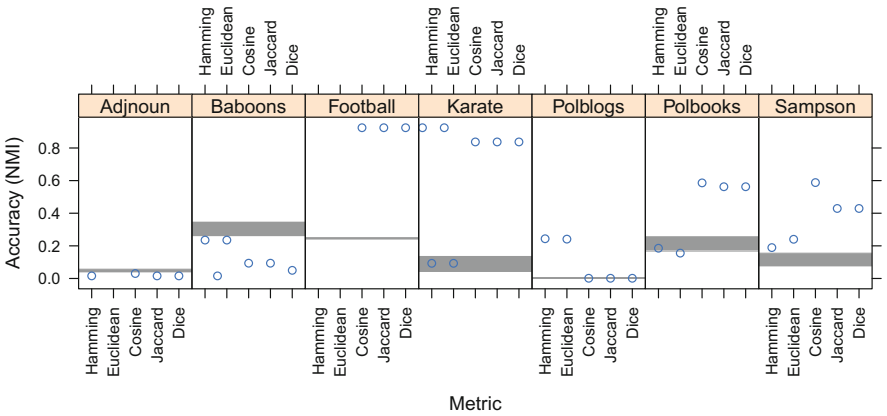

Fig. 1. NMI accuracy for blockmodel clustering using seriation with the given metric (x axis) and data set. The grey region is the 95% confidence interval for mean Coord NMI.

$k = 12$; Baboons: $n = 14$, $k = 2$; Adjnoun: $n = 112$, $k = 2$) are found on Mark Newman's homepage and the Pajek repository.[1] Initial experiments were performed to identify the effect of the hierarchical clustering merging method, and we found that Weighted merging consistently provided high accuracy results, so we focus on that merging method in this paper to reduce the number of variables in each experiment.

Baseline results were computed using Projected Gradient Descent (Grad) and Coordinate Descent (Coord) [1]. Note that the baseline methods are dependent on their initialisation, so we repeated the baseline clustering 10 times for each graph, using random initialisation. The clustering accuracy is presented in Fig. 1 containing the NMI when blockmodelling using each metric on each data set. The greyed out region in each plot shows the 95% confidence interval for the Coord mean. The 95% confidence interval for the Grad mean was also computed but it was lower than the interval for Coord, and so left off the plot.

The results in Fig. 1 show that the five metrics can be placed in two groups; the position based metrics (Hamming and Euclidean metrics) provide similar NMI for each data set, and the set based metrics (Jaccard, Dice and Cosine metrics) provide similar results for each data set. The data where set based metrics are preferred, show significant improvement over the state-of-the-art. The results for the position based metrics are generally equivalent in accuracy to the state-of-the-art. It is known that the Football data contains little noise, while the Adjnoun data contains large amounts of noise, and so we see that the accuracy of each are independent of the metric. It is likely that the difference in results is due to the different noise distributions in each network. This leads us to the definitions:

[1] www-personal.umich.edu/~mejn/ vlado.fmf.uni-lj.si/pub/networks/pajek/.

– We call data *position based data* when it obtains greater NMI when using the position based metrics (Euclidean and Hamming). We hypothesise that this is likely due to the data containing Hamming noise.
– We call data *set based data* when it obtains greater NMI when using the set based metrics (Cosine, Jaccard and Dice). We hypothesise that this is likely due to the data containing Jaccard noise.

We can see from the results that Karate, Polbooks and Sampson are set based data and that using a set based metric provides a huge increase over the baseline. It is interesting to see that the blockmodelling when using Hamming and Euclidean metrics are very similar to the confidence interval provided by the state-of-the-art, implying that the state-of-the-art is designed for position based data.

3 Simulated Data with Hamming and Jaccard Noise

The previous experiment revealed that there were two types of network data and stated that the difference is likely due to the noise distribution being different. In this section we will examine the validity of this assumption by simulating the noise and examining the effect of a set of parameters on the blockmodelling results. By simulating data, we are able to control the data parameters and hence examine the effect of seriation on the accuracy, given the block structure and noise type. To begin, we first describe the basic block structures, and then present an analysis using simulated data with Hamming and Jaccard noise.

3.1 Simulated Block Structures

When simulating data for this experiment, we use ring, star and tree block structures for the structure matrix M. A ring structure arranges the roles so that each is connected to exactly two other roles, forming a ring. A star structure assigns one role as a hub to which all remaining roles are connected. Finally, the tree structure requires that each role has a parent role, and at most two children roles, where one role (the root), has no parent role.

Simulated data was generated using the following parameters: *type* was chosen from *Ring*, *Star* or *Tree*; *the number of objects* was 50, 100, 200, 500, or 1000; and *the number of roles* (clusters) was 2, 4, 8, 16, or 32. We also generated three replicates of graphs using each parameter combination, providing 1,125 random graphs. An initial baseline experiment was run to examine the use of seriation on data with no noise. Experiments were then run to examine the effect of increasing Hamming and Jaccard noise in the data.

3.2 Generating Noise

Hamming noise is simple to generate, and so more likely to be used in blockmodel simulations. For a given binary vector of length n we can generate Hamming noise with expected Hamming distance of np by flipping each 0 or 1 value to a 1 or 0 value

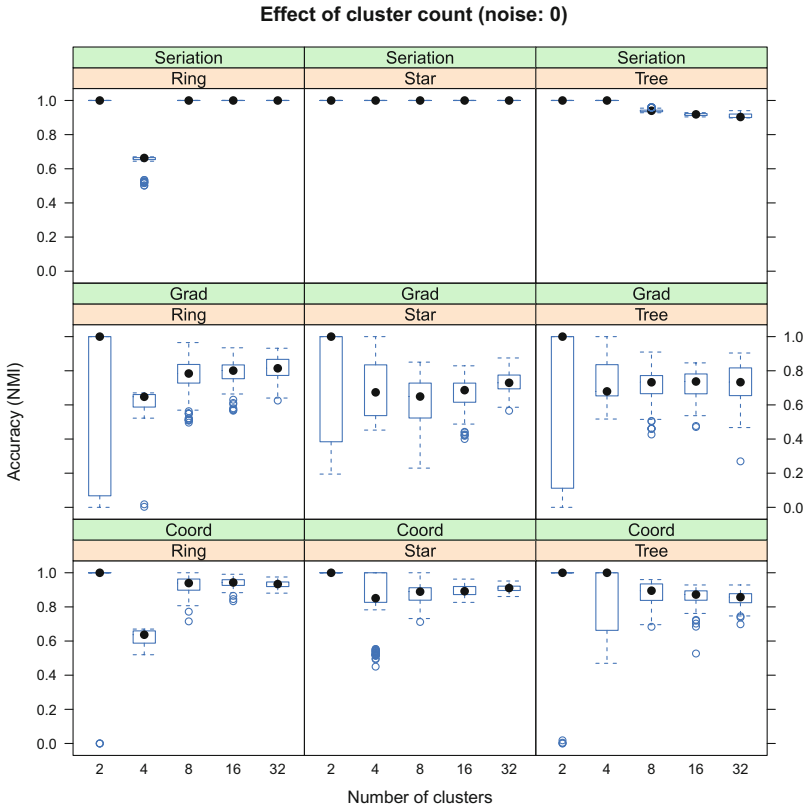

Fig. 2. Clustering accuracy versus cluster count (zero noise) for Ring, Star and Tree graphs. The results on the first row cover seriation using each metric, implying that the choice of metric has no effect on accuracy when no noise is present. The second and third rows contain baseline results.

with probability p. The probability of k flips is a Binomial distribution with n trials and probability of success p. Therefore, we can obtain the expected number and standard deviation number of flips from the Binomial distribution. For a graph, n is the number of vertices and so the noise level is controlled by the proportion p.

On the other hand, Jaccard noise applied to a vector of length n is dependent on the number of 1s in the vector. The Jaccard coefficient is the size of the intersection divided by the size of the union. When applying noise, the size of the intersection can only reduce (when a 1 is flipped to a 0). The size of the union can only increase (when a 0 is flipped to a 1). If we let q be the probability of a flip, we find that the change in intersection and union are independent of each other [7]. If we were to flip edges of $\vec{x}$ with probability q to obtain $\vec{x}^\star$, the expected Jaccard coefficient between binary vectors $\vec{x}$ and $\vec{x}^\star$ is:

$$\mathbb{E}[d_{\text{Jac}}(\vec{x}, \vec{x}^\star)] = \sum_{b=0}^{n-l} P(X_u = b) \frac{(1-q)l}{n-b}$$

where n is the length of the vector, l is the number of 1s in $\vec{x}$, and X_u is Binomial with probability of success $1-q$ and with $n-l$ trials. The $(1-q)l$ term is associated to the number of 1s that don't flip to 0 (the intersection) and the remainder is n minus the 0s that don't flip (the union).

To obtain a given expected Jaccard noise level, we must compute q for each vertex in the graph. We can see that if $q = 1$, then $\mathbb{E}[d_{\text{Jac}}(\vec{x}, \vec{x}^\star)] = 0$, and also if $q = 0$ then $\mathbb{E}[d_{\text{Jac}}(\vec{x}, \vec{x}^\star)] = 1$, therefore for any n and l, we can find a q that provides the desired Jaccard distance in expectation.

3.3 Analysis of Simulated Data

Using the simulated data, we examine if there is interaction between the seriation parameters and the number of objects, number of roles, structure of the graph and level of noise. Our hypothesis is that the choice of metric is dependent only on the noise type. Our first experiment examines the effect of each data parameter while holding the noise at zero (no noise) on the blockmodelling accuracy using each metric. The results are shown in Fig. 2 with a comparison to the existing Projected Gradient Descent (Grad) and Coordinate Descent (Coord) [1] methods. Note that the baseline methods are dependent on their initialisation, so we repeated the baseline clustering 10 times for each graph, using random initialisation. Cluster accuracy is measured using Normalised Mutual Information (NMI).

The box plots in Fig. 2 show the variation due to each of the experimental parameters, while holding the noise at 0. Individual results for each seriation metric are not shown because the variance between each method was minimal or zero. It is surprising to see that each of the seriation methods provides perfect results for each graph containing identifiable clusters, independent of the metric, number of objects and roles. The unidentifiable graphs lead to lower accuracy at 4 clusters for the Ring data, and 8, 16, and 32 clusters for the tree data. The baselines show higher variation and lower mean accuracy. This shows that there is no interaction between the seriation metric and the data parameters when there is no data noise, except for the slight interaction with the number of roles in the tree data due to the leaf unidentifiable roles.

The small variance in the seriation method, with respect to each non-noise parameter, is ideal for examining the effects of noise. Therefore, we focus on the seriation blockmodelling method for the remainder of the article.

Our second experiment examines the robustness of blockmodelling using each metric, to *Hamming noise* (using all simulated 1,125 graphs). The results for seriation are shown in Fig. 3. To make the plots more visually appealing, we limited the data to graphs containing 16 clusters; results for the other cluster sizes have a similar trend. As expected, we find that increasing the Hamming noise reduces the accuracy of each blockmodelling method. It can be seen that results can be grouped in terms of position based and set based metrics; for each block structure, set based metrics are less tolerant of Hamming noise.

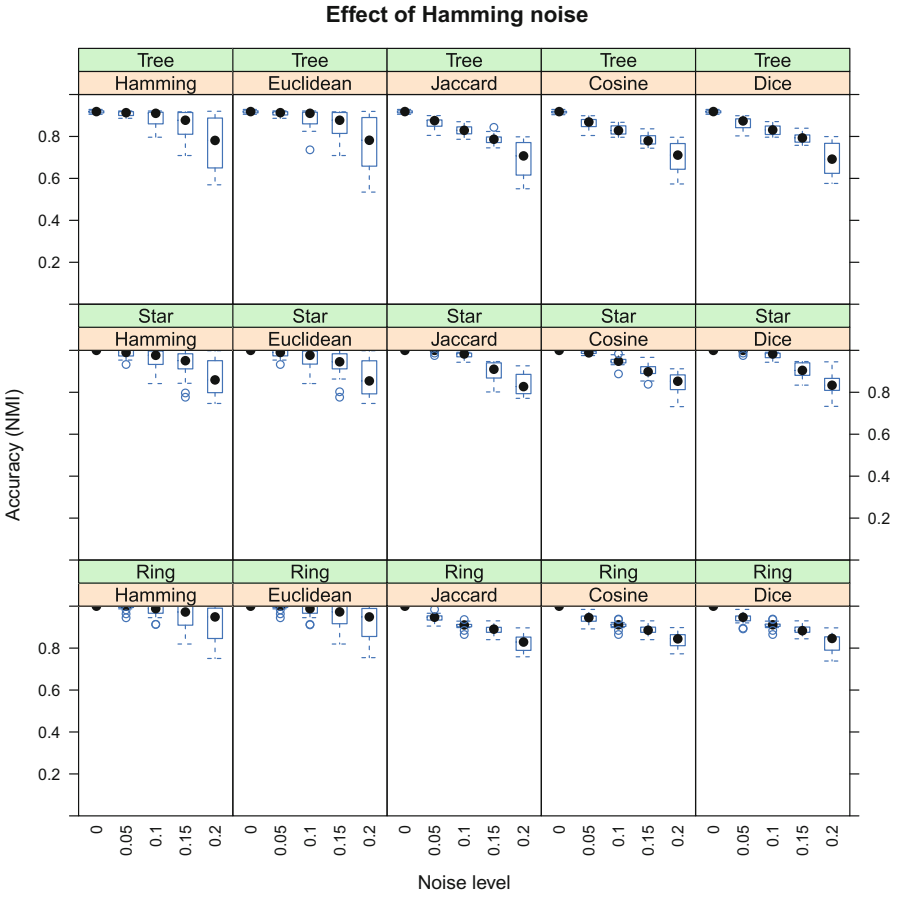

Fig. 3. Clustering accuracy versus Hamming noise level for seriation using each metric on tree, star and ring block structured data. Each data set contained 16 clusters.

Our third experiment examines the robustness of each blockmodelling method to *Jaccard noise* (using all simulated 1,125 graphs). The results for the seriation model are shown in Fig. 4. We find that the set based metrics provide a partitioning that is very robust to Jaccard noise (even when the expected Jaccard noise is 0.5), while the accuracy when using position based metrics drops as the noise level increases.

To identify the effect of Hamming and Jaccard noise on blockmodelling with each metric, we have computed the NMI decay rate (the expected drop in NMI when the data noise increases by 0.1). An NMI decay rate of 0 implies that noise has no effect on NMI, while a large NMI decay rate means that an increase in noise causes a large drop in NMI. The set of NMI decay rates are provided in Table 1. We find that the NMI noise decay rate is lowest for the position based metrics when the data contains Hamming noise, and for the set based metrics when the data contains Jaccard noise.

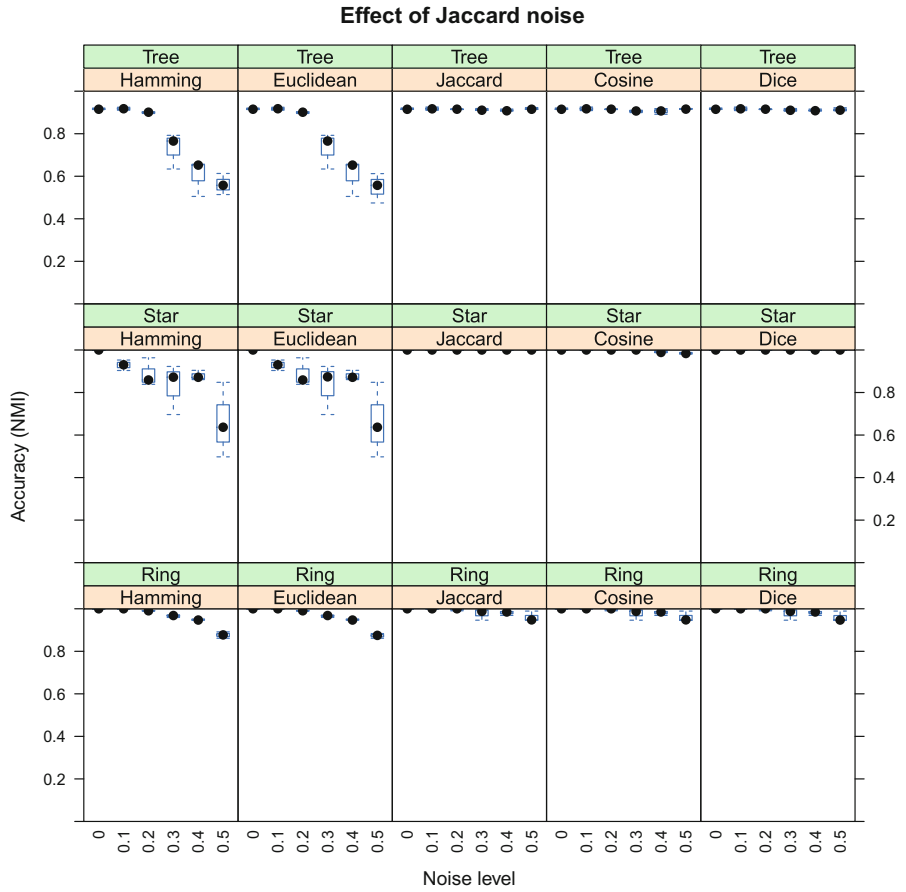

Fig. 4. Clustering accuracy versus Jaccard noise level for seriation using each metric on tree, star and ring block structured data. Each data set contained 16 clusters.

A paired difference permutation significance test was performed to compare the NMI noise decay rate between set based and position based metrics for each type of data. The results showed a significant difference in each case.

4 Discussion

The major difference between position and set based metrics, and the associated Hamming and Jaccard noise, are their treatment of True Negatives (missing edges remaining missing edges). Position based metrics use true negatives as evidence of similarity between the two items; if two vertices are both not connected to a third vertex, it means that the two are similar since they both have a poor relationship with the third. While set based metrics ignore true negatives. This distinction is important when missing edges have different meanings. For social networks, a missing edge might represent

Table 1. The NMI noise decay rate (expected drop in NMI when the noise increases by 0.1) using data with Hamming and Jaccard noise for blockmodelling with each metric. A value of zero implies that the noise level has no effect on the accuracy. The bolded set provide a statistically significant difference to the unbolded set for each network type.

	Euclid	Hamming	Jaccard	Cosine	Dice
Hamming noise					
Tree	**0.0534**	**0.0532**	0.0913	0.1007	0.0902
Ring	**0.0298**	**0.0295**	0.0616	0.0561	0.0624
Star	**0.0339**	**0.0342**	0.0589	0.0570	0.0564
Jaccard noise					
Tree	0.8427	0.8235	**0.0057**	**0.0129**	**0.0111**
Ring	0.2321	0.2271	**0.0777**	**0.0782**	**0.0786**
Star	0.5436	0.5437	**0.0000**	**0.0292**	**0.0000**

a poor relationship between the two items, or it might mean that the relationship was not measured (missing information). For these cases, the missing edges should be treated differently.

This leads us to investigate the meaning of the edges in the network data where set based metrics provided high accuracy. We found that of the seven network data sets from Fig. 1, three provided greater NMI when using set based metrics. Further investigation of these three data sets showed: **Sampson**: only the top three and bottom three relationships between monks were provided, so a missing edge represents an unknown relationship. **Polbooks**: an edge represents if the associated books were bought together. A missing edge does not mean that the books are not related. **Karate**: an edge represents interaction of the members outside of the club. A missing edge does not imply a poor relationship. So for each of these network data sets, a missing edge represents unknown information, not a poor relationship, hence set based metrics are ideal for these particular network data.

Simulations using Hamming noise showed that the position based metrics were more robust to the noise, while simulations using Jaccard noise showed that set based metrics were more robust to the noise. These results reinforced our belief that the network data contained either Hamming or Jaccard noise. We also found that the number of objects, number of roles, and basic structure of the data has little effect on NMI when using seriation for blockmodelling. Therefore there is a direct link to the robustness for a given metric and the noise type.

Based on our results and observations, we conjecture that if network data missing edges represent a poor relationship, then it is likely that it contains Hamming noise, and so position based metrics should be used. If missing edges represent missing information, then the noise is likely to be Jaccard and so set based metrics should be used.

Finally, the experiments showed that when there is little to no noise in the data (regardless of the type), that all metrics performed equally well when using seriation. When a sufficient noise level was reached, there was a difference in accuracy when using the different metrics. The Football data from Fig. 1 exhibits that same behaviour

(high accuracy, with no difference in metrics). An edge in that data represents that a game was held between two teams and since most games in the season are held within a conference, there are only few inter conference labels, and hence little noise. This supports our simulation result.

5 Conclusion

Blockmodelling is the process of clustering similar roles within a graph, which are visualised as blocks in the graph adjacency matrix. Edges in the graph increase the strength of block relationships, but it is unclear if missing edges in common should increase or decrease the strength of a relationship. In this article, we examined the effect of the choice of block metric on blockmodelling accuracy. We found that block metrics can be categorised into position based and set based metrics. Experiments on simulated data showed that the blockmodelling accuracy is independent of the block metric when no noise is present, but when noise was introduced, high accuracy results are obtained when the choice of block metric matched the noise type.

References

1. 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: CIKM, pp. 811–816. ACM (2013)
2. Fiala, J., Paulusma, D.: The computational complexity of the role assignment problem. In: Baeten, J.C.M., Lenstra, J.K., Parrow, J., Woeginger, G.J. (eds.) ICALP 2003. LNCS, vol. 2719, pp. 817–828. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-45061-0_64
3. Hahsler, M.: An experimental comparison of seriation methods for one-mode two-way data. Eur. J. Oper. Res. **257**(1), 133–143 (2017)
4. Hurley, C.B.: Clustering visualizations of multidimensional data. J. Comput. Graph. Stat. **13**(4), 788–806 (2004)
5. Karrer, B., Newman, M.E.: Stochastic blockmodels and community structure in networks. Phys. Rev. E **83**(1), 016107 (2011)
6. Park, L.A.F., Bezdek, J.C., Leckie, C., Kotagiri, R., Bailey, J., Palaniswami, M.: Visual assessment of clustering tendency for incomplete data. IEEE TKDE **28**(12), 3409–3422 (2016)
7. Park, L.A.F., Read, J.: A blended metric for multi-label optimisation and evaluation. In: Berlingerio, M., Bonchi, F., Gärtner, T., Hurley, N., Ifrim, G. (eds.) ECML PKDD 2018. LNCS (LNAI), vol. 11051, pp. 719–734. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-10925-7_44
8. Reichardt, J., White, D.R.: Role models for complex networks. The Eur. Phys. J. B **60**(2), 217–224 (2007)
9. Von Luxburg, U.: A tutorial on spectral clustering. Stat. Comput. **17**(4), 395–416 (2007)
10. Zhang, Y., Yeung, D.Y.: Overlapping community detection via bounded nonnegative matrix tri-factorization. In: Proceedings of the 18th ACM SIGKDD, pp. 606–614. ACM (2012)