GLaSS: Semi-supervised Graph Labelling with Markov Random Walks to Absorption

Max Glonek[1(✉)], Jonathan Tuke[1], Lewis Mitchell[1], and Nigel Bean[1,2]

[1] School of Mathematical Sciences, University of Adelaide, Adelaide, SA 5005, Australia
max.glonek@adelaide.edu.au

[2] ARC Centre of Excellence for Mathematical and Statistical Frontiers, University of Adelaide, Adelaide, SA 5005, Australia

Abstract. Graph labelling is a key activity of network science, with broad practical applications, and close relations to other network science tasks, such as community detection and clustering. While a large body of work exists on both unsupervised and supervised labelling algorithms, the class of random walk-based supervised algorithms requires further exploration, particularly given their relevance to social and political networks. This work proposes a new semi-supervised graph labelling method, the GLaSS method, that exactly calculates absorption probabilities for random walks on connected graphs, whereas previous methods rely on simulation and approximation. The proposed method models graphs exactly as a discrete time Markov chain, treating labelled nodes as absorbing states. The method is applied to a series of undirected graphs of roll call voting data from the United States House of Representatives. The GLaSS method is compared to existing supervised and unsupervised methods, demonstrating strong and consistent performance when estimating the labels of unlabelled nodes in graphs.

Keywords: Community detection · Graph labelling · Random walk Markov chain · Political networks

1 Introduction

Graph labelling is concerned with the problem of estimating the labels of one or more nodes within a graph, where an association between the graph's structure and the distribution of labels is assumed to exist. Many graph labelling algorithms exist, both supervised [2,7,13] and unsupervised [10,14]. In both cases, a graph comprises u unlabelled and ℓ labelled nodes, and the algorithms seek to estimate the labels of the unlabelled nodes. While a diverse range of graph labelling methods exist [4], this work focuses on the class of dynamical and statistical inference methods that use random walks.

L. M. Aiello et al. (Eds.): COMPLEX NETWORKS 2018, SCI 812, pp. 304–315, 2019.
https://doi.org/10.1007/978-3-030-05411-3_25

In unsupervised algorithms, the graph is organised into clusters, without consideration of the labelled nodes. Once clustered, labels for unlabelled nodes in the graph can be estimated based on the clusters to which labelled nodes belong. However, cases may arise where an identified cluster contains no labelled nodes, or where a cluster contains multiple nodes with different labels, creating uncertainty as to how labels should be estimated for nodes in such clusters.

The Walktrap algorithm is one commonly used random walk-based unsupervised graph labelling method [10]. Walktrap searches for densely connected subgraphs by simulating short random walks on a graph, reasoning that short walks are more likely to remain in the same cluster than to leave it. Walktrap quantifies the similarity between nodes using a distance metric, then recursively merges identified clusters based on short random walks, providing a hard classification for each node. Because Walktrap does not use information about labelled nodes, there is no generally accepted method for estimating the labels for unlabelled nodes based on the clusters it identifies.

Unlike unsupervised algorithms, supervised algorithms utilise the information contained in labelled nodes when estimating the labels of unlabelled nodes. A common approach is to treat labelled nodes as absorbing states and unlabelled nodes as transient states in a discrete time Markov chain (DTMC), and estimate the absorption probabilities or expected times to absorption for all transient states in the chain. Labels for each unlabelled state can then be estimated using the approximate probabilities or times. However, while supervised methods use both labelled nodes and the graph's structure to estimate labels, they only approximate absorption probabilities and times, rather than calculating them exactly.

The Rendezvous algorithm [2] labels nodes in a semi-supervised setting by constructing a simplified, "rendezvous" graph, where edges are drawn from an unlabelled node to only its M nearest neighbours. M is chosen to be as small as possible while ensuring that each unlabelled node in the rendezvous graph is connected to at least one labelled node. Once the renedezvous graph has been constructed, edge weights are calculated using a Euclidean distance metric, and absorption probabilities are calculated using the eigenvalues and eigenvectors of the rendezvous graph's transition matrix. Absorption probabilities for nodes in the rendezvous graph are then used to estimate the label of nodes in the full graph.

Another semi-supervised graph labelling method seeks to label nodes in a binary setting according to expected time to absorption, rather than absorption probability [7]. This "Censored Time" method simulates step-limited random walks over a graph, recording the number of steps taken for all walks that are absorbed before being terminated by the step limit. The censored times to absorption for absorbed walks are used to approximate the conditional expected time to absorption in each labelled node in the graph. A hard classification is used to estimate labels according to the lowest censored conditional time to absorption.

This work proposes a new semi-supervised graph labelling method, the Graph Labelling Semi-Supervised (GLaSS) method, using random walks to absorption. The method models a graph as a DTMC, where transient states correspond to unlabelled nodes, and absorbing states correspond to labelled nodes. The transition matrix P, for the DTMC, is formed from the graph's weighted adjacency matrix by normalising the weighted out-degree of each node in the network. From careful construction of P, the probability of absorption in each absorbing state can be calculated exactly, and these probabilities can then be used to estimate the label for every node corresponding to a transient state in the DTMC.

By calculating exact absorption probabilities and expected times to absorption, the GLaSS method provides better label estimates than contemporary supervised methods, which rely on approximations of these quantities. By utilising the information contained in labelled nodes in the graph, GLaSS also provides a clear method for estimating the label of unlabelled nodes using quantities that are meaningful and interpretable, unlike unsupervised random walk methods.

The GLaSS method is formally introduced in Sect. 2. Section 3 describes the data analysed, and a full description of all analyses performed is presented in Sect. 4. Conclusions and areas for further work are discussed in Sect. 5.

2 Method

Consider an undirected graph $G = (V, E)$ comprising n nodes, $V = \{v_1, \ldots, v_n\}$, connected by a set of positive real-weighted edges E. Define the weighed adjacency matrix $A = [a_{i,j}]$, where $a_{i,j} = a_{j,i}$ records the weight of the edge connecting v_i and v_j, and $a_{i,j} = 0$ if no edge connects v_i and v_j. Suppose the first u nodes in G are unlabelled, and the remaining ℓ nodes in G are labelled, where $n = u + \ell$, and construct the sets $U = \{1, \ldots, u\}$ and $L = \{u+1, \ldots, n\}$ to index the unlabelled and labelled nodes of G, respectively. Arrange A as

$$A = \begin{bmatrix} A_{U,U} & A_{U,L} \\ A_{L,U} & A_{L,L} \end{bmatrix}$$

where $A_{J,K}$ describes the weighted edges connecting nodes indexed by J to nodes indexed by K.

Consider a random walk on G, described by a discrete time Markov chain (DTMC) where all unlabelled nodes map to transient states and all labelled nodes map to absorbing states. Let X_t denote the state of the chain at time t. Calculate the transition probabilities for the DTMC using A, where

$$p_{i,j} = P(X_{t+1} = j \mid X_t = i) = \frac{a_{i,j}}{\sum_{k=1}^{n} a_{i,k}} \tag{1}$$

is the probability that the DTMC is in state j at the next time step, given that the DTMC is currently in state i. Construct the transition matrix

$$P = [p_{i,j}] = \begin{bmatrix} P_{U,U} & P_{U,L} \\ P_{L,U} & P_{L,L} \end{bmatrix} = \begin{bmatrix} R & S \\ 0 & I_\ell \end{bmatrix}. \tag{2}$$

The $u \times u$ matrix R governs transitions between transient states, the $u \times \ell$ matrix S governs transitions from transient states to absorbing states, 0 is an $\ell \times u$ zero matrix, and I_ℓ is the $\ell \times \ell$ identity matrix.

2.1 DTMC Absorption Probabilities

Let $h_{i,j}$ be the probability that the DTMC is eventually absorbed in state j, given that the chain starts in state i. Define the matrix of absorption probabilities $H = [h_{i,j}]$. H is restricted to have u rows and ℓ columns, corresponding to the u transient states and ℓ absorbing states of the DTMC, respectively. Then H can be calculated as

$$H = (I_u - R)^{-1} S \tag{3}$$

where I_u is the $u \times u$ identity matrix, and R and S are as above [6].

2.2 Semi-supervised Graph Labelling

Given a graph G and the matrix of absorption probabilities H, let Y_i be the label of an unlabelled node v_i, and let y_j be the label of a labelled node v_j. The distribution over Y_i can be directly derived from H, for all $i \in U$, as follows:

$$P(Y_i = k) = \sum_{j=u+1}^{n} h_{i,j} \mathbb{1}(y_j = k) \tag{4}$$

where $\mathbb{1}$ is an indicator function, taking value 1 if its argument is true, and 0 otherwise.

2.3 DTMC Expected Times to Absorption

Let t_i be the expected number of time steps before the DTMC is absorbed in any absorbing state, given that the chain starts in state i. Define the vector of expected times to absorption $\mathbf{t} = (t_1, \ldots, t_u)^T$, where the u elements of $\mathbf{t}$ correspond to the u transient states of the DTMC. Then $\mathbf{t}$ can be calculated as

$$\mathbf{t} = (I_u - R)^{-1} \mathbf{c} \tag{5}$$

where $\mathbf{c}$ is a column vector of length u whose entries are all 1, and I_u and R are as above [6].

2.4 The Graph Labelling Semi-supervised (GLaSS) Method

Consider a graph G, with u unlabelled nodes and ℓ labelled nodes, and suppose that all labelled nodes have one of two labels; either K_1 or K_2. From the weighted adjacency matrix A, construct the transition matrix P, as in (1). Using P, calculate the vector of expected times to absorption $\mathbf{t}$, as in (5). The expected times to absorption may, optionally, be used as a filtering criterion; nodes with

a large expected time to absorption, relative to the distribution of t_i over all nodes in the graph, may be excluded from further analysis.

Once nodes have been optionally filtered using $\mathbf{t}$, calculate the matrix of absorption probabilities H, by (3), and calculate $P(Y_i = K_1)$ and $P(Y_i = K_2)$ for all $i \in U$, as in (4). Because, by the Law of Total Probability, $P(Y_i = K_1) + P(Y_i = K_2) = 1$, only one probability is required to proceed. Consider $P(Y_i = K_1)$ for all i, and implement a binary classifier with some threshold α. If $P(Y_i = K_1) \geq \alpha$, estimate the label for node v_i as K_1; otherwise, if $P(Y_i = K_1) < \alpha$, estimate the label for node v_i as K_2. Choose α to maximise the binary classifier's discrimination between K_1 and K_2.

Using this method, it is possible to estimate the label for every unlabelled node in G. As a graph labelling method in a semi-supervised setting, the method is called the GLaSS method.

3 Data

Validating the GLaSS method requires graphs with a clear community structure and known labels for all nodes. To simulate a graph with few known labels, only a small subset of all known labels will be used by GLaSS, with remaining labels withheld to simulate "unlabelled" nodes in the graph. All labels estimated by GLaSS can be compared to actual, withheld labels, to assess performance. Therefore, United States roll call voting data is used to validate the GLaSS method.

In the United States House of Representatives (the House), parliamentary procedure occasionally gives rise to roll call votes. In a roll call vote, the vote of every member of the House is recorded, making it possible to see which members of the House voted the same way. Roll call voting data can be modelled as an undirected graph, where each node represents a member of the House, and a positive integer-weighted edge records the number of times respective members voted the same way.

The results of roll call votes in the House for the meetings of eight separate Congresses, between 1953 and 1997[1], have been collected for analysis [9], and modelled as eight separate undirected graphs. For simplicity, in each Congress, the following rules are applied:

1. Only "yea" and "nay" votes are considered.
2. Votes are disregarded if cast by the Speaker of the House[2,3].
3. Only members whose party affiliation is Democrat or Republican are considered.
4. In cases where a member's party affiliation changes during a meeting of Congress, their party affiliation at the time they were elected is used.

[1] Each meeting of Congress begins on January 3 and runs for a period of two years.
[2] Conventionally, the Speaker of the House participates in very few votes.
[3] The 101st Congress had two speakers, both of whose votes are disregarded in these analyses.

5. In rare cases, a member of the House does not sit for the entire meeting of Congress, and their seat is taken by a new member. In these cases, the voting records of both members are retained.[4]

Because the party affiliation of each member is known, all nodes in each graph are labelled. For random walks on each graph, only the labels of nodes corresponding to the Majority Leader and the Minority Leader are retained (one Democrat and one Republican), thus all other nodes in each graph are "unlabelled". Choice of Congresses is informed by recent work examining partisanship trends in the House [1], ensuring variation in partisanship and which party is in Majority. All graphs are either fully connected or nearly fully connected, and a detailed summary of each graph is contained in Table 1.

Table 1. Years covered, total number of members (nodes), democrats, republicans, and votes for each congress. Congresses where the number of democrats is shown in **bold** had a democrat majority leader, and congresses where the number of republicans is shown in **bold** had a republican majority leader.

Congress	Years	Total members	Democrats	Republicans	Votes
83rd	1953–55	439	218	**221**	147
86th	1959–61	441	**285**	156	180
89th	1965–67	441	**299**	142	394
92nd	1971–73	442	**257**	185	649
95th	1977–79	440	**293**	147	1540
98th	1983–85	438	**271**	167	906
101st	1989–91	441	**262**	179	879
104th	1995–97	443	208	**235**	1321

4 Results

Each Congress is modelled as a graph, and each graph is analysed using the GLaSS method, as described in Sect. 2.4. Expected time to absorption is calculated for each "unlabelled" node in each graph; the mean and variance of t_i for each graph are given in Table 2. Based on the distribution of t_i for each graph, no filtering is required, and labels are estimated for all "unlabelled" nodes in each Congress.

As each graph contains only two labelled nodes (one Democrat, one Republican), only the probability of being absorbed in the Democrat state of the corresponding DTMC is considered. Histograms of absorption probabilities for the 83rd, 86th, 89th, and 92nd Congresses are shown in Fig. 1, and histograms for the 95th, 98th, 101st, and 104th Congresses are shown in Fig. 2. In all Congresses, Democrat and Republican members are clearly separated, though some overlap between clusters exists.

[4] Consequently, while the House has 435 seats, each graph has more than 435 nodes.

Using the binary classifier in GLaSS, a threshold α_k is chosen for the kth Congress. If $P(Y_i = Democrat) \geq \alpha_k$, then member i is labelled a Democrat; otherwise, member i is labelled a Republican. Estimated labels are compared to the true party affiliation for all "unlabelled" nodes. By varying α_k across the range of absorption probabilities calculated for each respective Congress, a ROC curve is derived. ROC curves for all eight Congresses are displayed in Fig. 3, and the AUC for each Congress is given in Table 2.

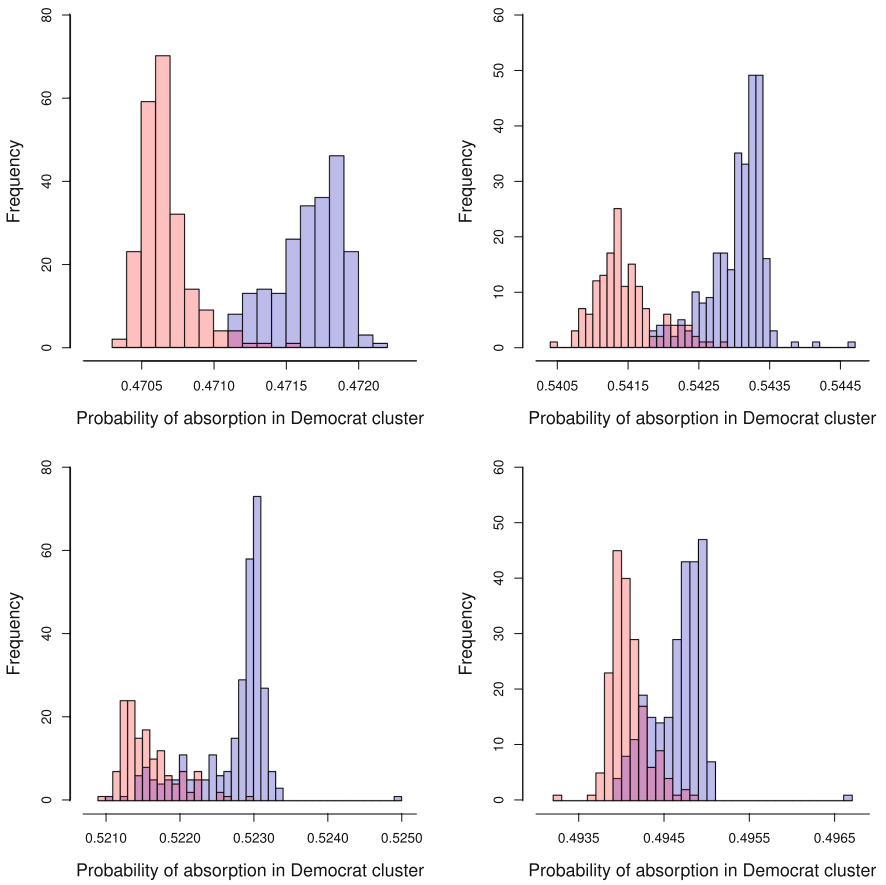

Fig. 1. Clockwise from top right: 86th, 92nd, 89th, and 83rd congresses. Histograms show the probability of absorption in the democrat cluster for each congress. Red bars show republican members, and blue bars show democrat members. The range of absorption probabilities for each congress is narrow, but clusters are clearly separated. Note there is more overlap between clusters in the 89th and 92nd congresses, corresponding to increased bipartisanship [1].

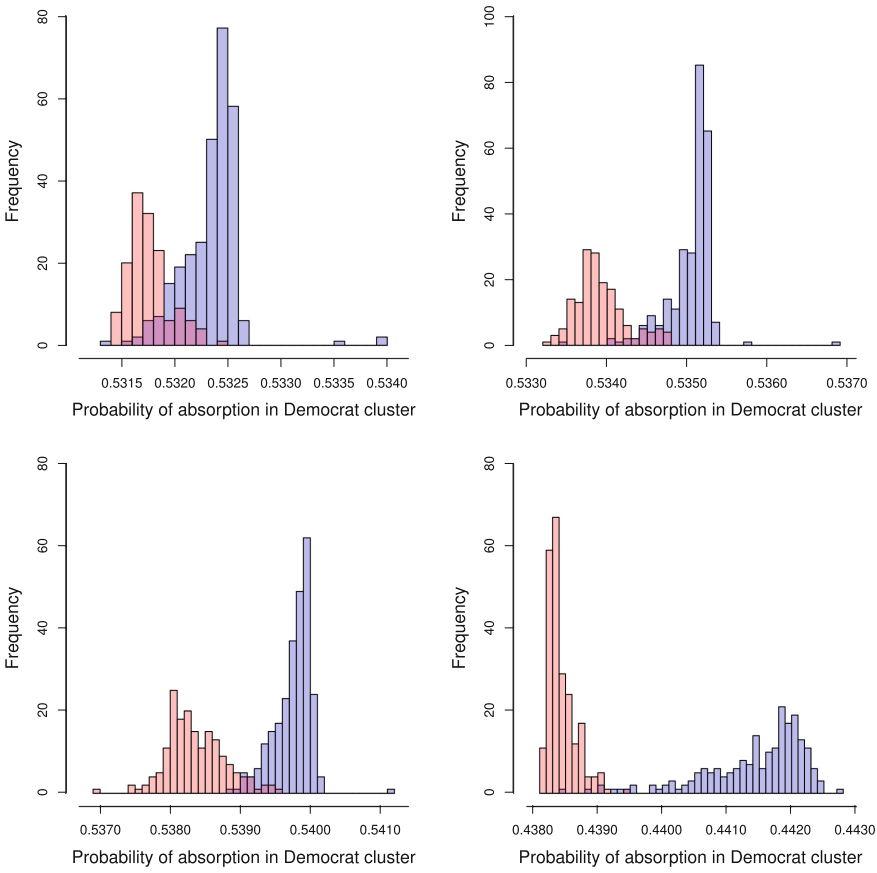

Fig. 2. Clockwise from top right: 98th, 104th, 101st, and 95th congresses. Histograms show the probability of absorption in the democrat cluster for each congress. Red bars show republican members, and blue bars show democrat members. The range of absorption probabilities for each congress is narrow, but clusters are clearly separated. Clusters become more separated over time, corresponding to an increase in partisanship within the house [1].

4.1 Comparison to Other Methods

The GLaSS method is compared to two alternative random walk-based graph labelling methods. The first method, the Walktrap algorithm [10], is an unsupervised method. Walktrap searches for densely connected subgraphs by simulating random walks on a graph, reasoning that short random walks are more likely to stay in the same cluster than to leave it. Because each Congress has two clearly defined clusters (Democrats and Republicans), the Walktrap algorithm is successful, in the first instance, if it places the Majority Leader and Minority Leader in different clusters, and if only two clusters are identified. If the Walktrap algorithm is successful in separating the Majority and Minority Leaders, the label for

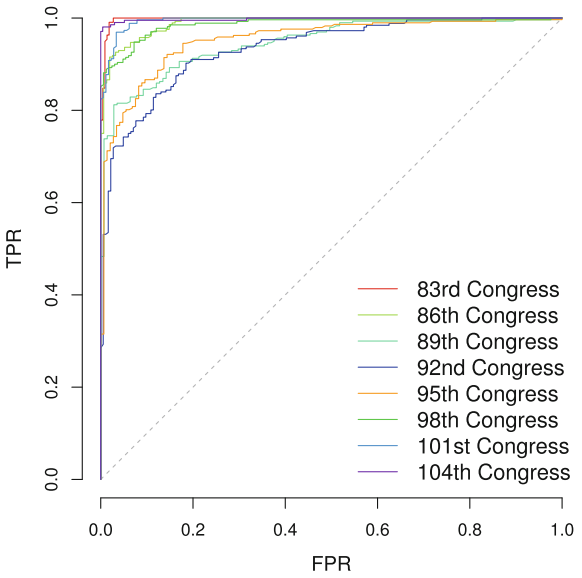

Fig. 3. ROC curves showing the performance of the binary classifier in GLaSS for each Congress, as α_k is varied. As expected, the three most partisan congresses analysed (89th, 92nd, and 95th) [1] show the weakest performance.

each member is estimated to be the same as the label of the Leader in that member's cluster. All analysis is conducted using a popular default implementation of the Walktrap algorithm [8].

The second method (Censored Time) is semi-supervised, and estimates the expected time to absorption, conditional on being absorbed in each labelled state [7]. Censored Time simulates step-limited random walks on a graph, where a walk is terminated if it is not absorbed before reaching the step limit. For walks that are absorbed, the censored conditional time to absorption is recorded, and these are used to estimate the conditional expected time to absorption for each labelled state. For a graph with two labels, Censored Time labels nodes according to the state with the smaller estimated conditional expected time to absorption. For each graph, the exact expected time to absorption is calculated for all nodes, as specified in Sect. 2.3, and the ceiling of the mean expected time to absorption is adopted as the step limit for Censored Time. For each "unlabelled" state in each graph, 1000 step-limited random walks are simulated, to estimate the conditional expected time to absorption for each labelled state.

To compare the performance of Walktrap, Censored Time, and GLaSS, an F1 score is calculated for each method and each Congress. For each Congress, the value of α_k is chosen to maximise GLaSS's F1 score. F1 scores for all three methods and all eight Congresses are given in Table 2. From the F1 scores, it is clear that GLaSS outperforms Censored Time for all Congresses, and equals or surpasses Walktrap in mose cases. Walktrap provides comparable performance to

GLaSS for the most partisan Congresses (101st and 104th), but its performance decreases with decreasing partisanship, and it fails for two Congresses (83rd and 89th), by identifying more than two clusters. The GLaSS method exceeds, or effectively matches, the performance of Walktrap and Censored Time for all Congresses, while also showing greater resilience to decreasing separation of clusters caused by decreases in partisanship [1].

Table 2. F1 scores for walktrap and censored time, and the maximal F1 score for GLaSS (highest scores shown in **bold**). Additionally, α_k gives the range of cutoffs that yield the maximal F1 score using GLaSS. AUC gives the area under the curve for the ROC curves for GLaSS. The mean and variance of the expected time to absorption (see Sect. 2.3) for each Congress are given in μ_t and σ_t, respectively.

Congress	F1 score			GLaSS			
	Walktrap	Censored time	GLaSS	α_k	AUC	μ_t	σ_t
83rd	-[a]	0.5068	**0.9864**	(0.471117, 0.471149)	0.9978	191.77	0.0037
86th	0.8479	0.5779	**0.9561**	(0.542170, 0.542193)	0.9899	203.47	0.0062
89th	-[a]	0.5957	**0.9122**	(0.521823, 0.521827)	0.9464	214.97	0.0014
92nd	0.7975	0.5259	**0.8876**	(0.494203, 0.494204)	0.9338	208.33	0.0046
95th	0.8333	0.5558	**0.9293**	(0.531896, 0.531899)	0.9528	218.68	0.0014
98th	0.8907	0.5602	**0.9531**	(0.534291, 0.534311)	0.9857	216.22	0.0044
101st	**0.9738**	0.5980	0.9736	(0.539083, 0.539084)	0.9949	223.42	0.0007
104th	**0.9878**	0.5667	**0.9878**	(0.439130, 0.439215)	0.9979	223.20	0.0030

[a] More than two clusters identified

5 Discussion

Graph labelling is a fundamental task within network science, with diverse applications. This work proposes a new semi-supervised graph labelling method, the GLaSS method, using random walks to absorption. The GLaSS method has been used to analyse a series of undirected graphs, showing very strong performance when estimating the labels of unlabelled nodes. The GLaSS method represents a compelling alternative to existing supervised and unsupervised random walk methods. The key features of the GLaSS method are that, unlike other supervised methods, it calculates exact absorption probabilities and expected times to absorption, and, unlike unsupervised methods, it provides a clear method for the labelling of unlabelled nodes based on identified clusters.

Results show the GLaSS method meets or exceeds the performance of the supervised and unsupervised methods to which it is compared, as measured using F1 score. ROC curves and AUC for each graph analysed also show that the GLaSS method shows consistently very strong performance. Future work will

extend this work to examine the performance of the GLaSS method for graphs of varying size, connectedness, density, and with different numbers of known labels. Extending the GLaSS method can be generalised to label graphs with more than two clusters, and graphs with fewer labelled nodes than clusters, is of particular interest. Future work will also explore the use of expected time to absorption as a filtering criterion for nodes, particularly in cases where the number of clusters exceeds the number of known labels.

In an applied setting, future work will also use GLaSS to further explore social, political, and other networks. Online and social-media networks are of particular interest, with a growing body of work examining the structure, dynamics, and polarisation of online social networks [3,5,11,12]. Future applied work with GLaSS will examine these characteristics for new and existing graphs.

Acknowledgements. The authors thank Data to Decisions CRC and the ARC Centre of Excellence for Mathematical and Statistical Frontiers for their financial support.

References

1. Andris, C., Lee, D., Hamilton, M.J., Martino, M., Gunning, C.E., Selden, J.A.: The rise of partisanship and super-cooperators in the US house of representatives. PLoS One **10**(4), e0123507 (2015)
2. Azran, A.: The rendezvous algorithm: multiclass semi-supervised learning with markov random walks. In: Proceedings of the 24th International Conference on Machine Learning (ICML), pp. 49–56 (2007)
3. Fish, B., Huang, Y., Reyzin, L.: Recovering social networks by observing votes. In: Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems, pp. 376–384 (2016)
4. Fortunato, S.: Community detection in graphs. Phys. Rep. **486**(3–5), 75–174 (2010)
5. Garimella. K., Weber, I.: A long-term analysis of polarization on Twitter. arXiv:1703.02769 (2017)
6. Grinstead, C.M., Snell, J.L.: Introduction to probability. Amer. Math. Soc. (2012)
7. Hassan, A., Radev, D.: Identifying text polarity using random walks. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 395–403 (2010)
8. Csardi, G., Nepusz, T.: The igraph software package for complex network research. InterJ. Complex Syst. 1695 (2006). https://igraph.org. Accessed 28 Aug 2018
9. Lewis, J.B., Poole, K., Rosenthal, H., Boche, A., Rudkin, A., Sonnet, L.: Voteview: congressional roll-call votes database (2018). https://voteview.com/data. Accessed 21 Aug 2018
10. Pons, P., Latapy, M.: Computing communities in large networks using random walks. In: International Symposium on Computer and Information Sciences, pp. 284–293 (2005)
11. Rizoiu, M.A., Graham, T., Zhang, R., Zhang, Y., Ackland, R., Xie, L.: #debatenight: the role and influence of socialbots on twitter during the 1st us presidential debate. arXiv:1802.09808 (2018)
12. Shai, S., Stanley, N., Granell, C., Taylor, D., Mucha, P.J.: Case studies in network community detection. arXiv:1705.02305 (2017)

13. Talukdar, P.P., Reisinger, J., Paşca, M., Ravichandran, D., Bhagat, R., Pereira, F.: Weakly-supervised acquisition of labelled class instances using graph random walks. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 582–590 (2008)
14. Zhou, H., Lipkowsky, R.: Network brownian motion: A new method to measure vertex-vertex proximity and to identify communities and subcommunities. In: International Conference on Computational Science (ICCS), pp. 1062–1069 (2004)