Applying Power Graph Analysis to Weighted Graphs

Niels Bloom

Pagelink Interactives
Sherwood Rangers 29,
7551 KW Hengelo, The Netherlands
n.bloom@pagelink.nl

Abstract. We expanded Power Graph Analysis for use with weighted graphs, applying the technique to document categorisation with promising results. With the additional weight information we were able to create more accurate representations of the underlying data while maintaining a high level of edge reduction and improving visualisation of the graph.

Keywords: power graph analysis, graph theory, power graphs, weighted graphs, document categorisation

1 Introduction

Power Graphs are abstractions of unweighted undirected graphs that mark star, clique and bi-clique motifs in the graph (see Figure 1). These patterns are represented using power nodes, which are sets of nodes grouped together, and power edges, which signify relations of these sets with individual nodes and with other power nodes.

Power graph analysis has been used to help analyse and understand protein networks, specifically to gain insight into the biological relationships between proteins. Royer et al. [5] found power graphs to reveal aspects of the underlying biology by simplifying the representation of the data without loss of information. Their results are in line with other motif finding algorithms that perform similar functions [1, 2].

In this paper we extend power graph analysis to weighted graphs and look at its application in document categorisation; specifically we look at weighted graphs modelling the relationships between documents. We establish that power graphs can indeed be used to reveal aspects of the underlying structure in networks of related documents, in the same way that it did for biological relationships in protein networks.

2 Extending Power Graph Analysis to Weighted Graphs

Because edges in weighted graphs contain additional information, weighted graphs are not immediately obvious candidates for power graph analysis: edges can only

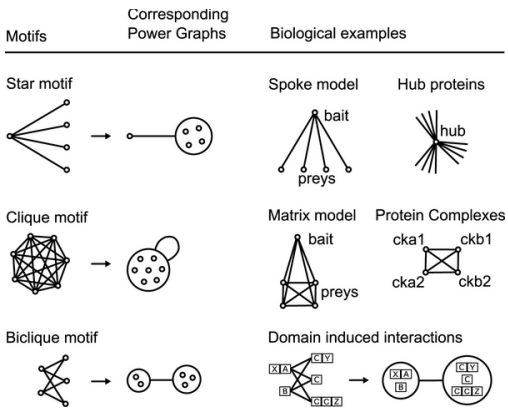

Fig. 1. Power Graph Analysis summary (image by Royer et al.)

be merged into a single power edge without loss of that information if they have exactly identical weights. Fortunately, despite this loss, it is still possible to use power graph analysis to gain insight into the underlying relationships and find groups and clusters of related documents.

Because the construction of power graphs from regular graphs is an NP-complete problem (it encompasses the maximum edge bi-clique problem, which was established as NP-complete by Peeters [3]), Royer et al. use a series of heuristics to establish candidate power nodes - sets of nodes that could potentially become power nodes. They then select the actual power node by finding the candidate with the highest edge reduction[1].

When creating a power graph from a weighted graph, rather than judging each power node only by the number of edges removed, we use the weights to determine which power node is actually the best candidate, its value determined by the total weight of the edges removed. This means that strongly correlating nodes (with a high weight on the edge between them) are more likely to be grouped into power nodes together.

3 Experiment Setup

One of the key features of power graph analysis mentioned by Royer et al. is that it can reveal known underlying biological patterns in the data. Royer et al. evaluate their results on various protein networks by comparing them to randomly generated networks of the same size and edge density. Their hypothesis is that power graphs will have a lower edge reduction for randomly wired graphs than they would for the graphs with real data, as good power edges should be more easily created based on the underlying structure.

Our method for using power graph analysis on weighted graphs has a potentially sub-optimal total edge reduction to allow for better grouping of closely

[1] Edge reduction is a measure for the number of edges replaced by power edges.

related nodes. Though power nodes with a high edge reduction are still likely to have a high combined weight, we expect that the total edge reduction will be lower than for unweighted graphs because candidates with high correlation will be preferred over candidates with more, lower valued edges. However, using the same hypothesis as Royer et al., we still to see the total edge reduction remain above the average of random graphs.

To compare to their work, we used Royer et al.'s method of establishing the random baseline by means of 1000 randomly rewired networks of the same size and edge density to estimate the variance of the edge reduction and establish a z-score[2]. Additionally, we wanted to compare the use of weighted graphs to unweighted graphs, so we processed each generated dataset both in its original form and with all weights stripped.

4 Construction of the Dataset

To construct a test set, we took a corpus of thirty related articles from a single category in Wikipedia and converted these to plain text. Only articles with more than 1000 words were selected. For each article, we calculated the tf-idf values [4] of all words in the corpus. We then used the sum of the difference of these values to determine the relationship between each pair of articles:

$$D(d_x, d_y) = \sum_{n=1}^{w} |tfidf(d_x, W_n) - tfidf(d_y, W_n)| \quad (1)$$

where D(d_x, d_y) is the distance between the documents d_x and d_y, W is the set of words 1...w in the corpus and *tfidf(d_i, W_j)* is the term frequency / inverse document frequency of the word W_j in the document d_i.

We established this relationship between all pairs of documents, removing all but the top 100 links between articles to keep only relevant relations. This process was repeated five times with different sets of articles, each time resulting in a different graph of 30 nodes and 100 edges with different weights.

5 Results

Table 1 lists the edge reduction and conversion rates [3], which were calculated in the same way as Royer et al. to allow for easier comparison. The z-score was calculated by comparing the datasets to the randomly rewired baseline samples.

At almost 90%, the edge reduction is higher than even the best datasets of Royer et al., both for weighted and unweighted graphs, with matching high conversion rates around 12. The corresponding z-score is low compared to Royer et al. Notably all scores for the unweighted graphs are lower than for the weighted graphs and both are higher than the random baseline.

[2] The z-score or 'standard score' is the number of standard deviations an observation is above or below the mean.

[3] The conversion rate is an a ratio between the edge reduction and the number of power nodes, indicating the average reduction per created power node.

	Edge Reduction	Conversion Rate	z-score e.r.
Weighted graph average performance	89.8%	12.319	5.764
Unweighted graph average performance	88.2%	11.437	4.910
Random rewired baseline	84.9%	6.208	-
Royer et al. data sets best	85%	13	242.7
Royer et al. data sets average	55.8%	6.0	43.4
Royer et al. data sets worst	38%	3.8	2.2

Table 1. Results

6 Conclusions and Future Work

Though our sample size is small and further research in this area is required, our results suggest that like protein networks, document categorisation may indeed benefit from using power graph analysis to uncover hidden information, as power graph analysis clearly performed better than the random baseline.

The worry that using weighted graphs might harm the total edge reduction appears to be unfounded, with the weighted versions of the graphs actually performing slightly better than the unweighted versions, lending further credence to the hypothesis that power graphs identify underlying patterns in the data.

Comparing our results to Royer et al.'s results on 13 Protein Interaction Networks, the edge reduction and conversion rates for our data were high, some even above all of Royer et al.'s networks. This is likely to be, at least in part, a result of the slightly smaller network size (30 nodes, 100 edges) compared to the protein networks. When comparing the z-score the results are on the low side, but this is to be expected with the smaller network size.

Additional work is required but the results so far suggest that power graph analysis may indeed become a useful tool in finding the underlying patterns in sets of documents. We are currently investigating larger networks to see if the results continue to hold true there.

References

1. Andreopoulos, B., An, A., Wang, X., Faloutsos, M., Schroeder, M.: Clustering by common friends finds locally significant proteins mediating modules. Bioinformatics, 23(9):1124-1131. (2007)
2. Bader, G.D., Hogue, C.W.: An automated method for finding molecular complexes in large protein interaction networks. BMC Bioinformatics, 4:2 (2003)
3. Peeters, R.: The maximum edge biclique problem is NP-complete. Discrete Applied Mathematics 131(3) (2003)
4. Ramos, J.: Using TF-IDF to Determine Word Relevance in Document Queries. Proceedings of the First Instructional Conference on Machine Learning (iCML), Piscataway NJ (2003)
5. Royer, L., Reimann, M., Andreopoulos, B., Schroeder, M.: Unraveling Protein Networks with Power Graph Analysis. PLoS Comput Biol 4(7) (2008)