On Finding Power Method in Spreading Activation Search

Ján Suchal

Slovak University of Technology
Faculty of Informatics and Information Technologies
Institute of Informatics and Software Engineering
Ilkovičova 3, 842 16 Bratislava, Slovakia
suchal@fiit.stuba.sk

Abstract. Graph ranking algorithms such as PageRank and HITS share the common idea of calculating eigenvectors of graph adjacency matrices. This paper shows that the power method usually used to calculate eigenvectors is also present in a spreading activation search. In addition, we empirically show that spreading activation algorithm is able to converge on periodic graphs, where power method fails. Furthermore, an extension to graph ranking calculation scheme is proposed unifying calculation of PageRank, HITS, NodeRanking and spreading activation search.

1 Introduction

Rapidly growing information volume in world-wide web is forcing search engines to find new ways of computing better results. Recent works show that the usage of additional metadata, such as the link structure and usage data of the web [1], or even topics [2] can be used to improve search and recommendation results. While some works focus on exploring the colorful spectra of possible metadata features the core idea of the most popular and well known PageRank and HITS algorithms usually used as a basis still remains the same – in calculating eigenvectors of graph adjacency matrices. These computed eigenvectors usually represent an additional relevance measure of vertices in graph and are then used to alter (and hopefully improve) ordering in search and/or recommendation results. Such recommendations have shown usefulness in many domains, including web search [3], job offers [4], publications, or even e-learning.

After a short description of an iterative process called power method that can be used to calculate matrix eigenvectors in Section 2, we concentrate on an slightly different approach to graph ranking, which is based on a theoretical model of human semantic memory known as spreading activation search (Section 3). In Section 4 we show that recursive procedure of spreading activation search can be reformulated using matrix algebra resulting into a minor variation of classic power method. Section 5 empirically shows that the power method and spreading activation search can converge to identical results. Section 6 revisits PageRank, HITS and NodeRanking algorithms unifying them in a proposed scheme extension for iterative graph ranking calculation.

2 The Power Method

Power method is an iterative procedure wich can be used to calculate matrix eigenvectors. If $\mathbf{A}$ is a matrix[1], eigenvector $\mathbf{r}_\infty$ can be calculated simply by applying matrix multiplication in an infinite loop[2] starting from a randomly chosen vector $\mathbf{r}_0$.

$$\mathbf{r}_k = \mathbf{r}_{k-1}\mathbf{A} \tag{1}$$

$$\lim_{k\to\infty} \mathbf{r}_k = \mathbf{r}_\infty = \mathbf{r}_\infty\mathbf{A} \tag{2}$$

3 Spreading Activation Search

Spreading activation search [5] is based on a simple recursive procedure that distributes energy on vertices through graph edges. Similarity of graph vertices is then expressed by overall energy amounts accumulated on vertices that resulted from this energy distribution process.

This energy distribution process usually starts by energizing a starting vertex of graph with a given amount of energy and can be formulated as follows. Whenever vertex v is activated by energy e, energy e is added to overall activation vector $\mathbf{c}$, where component $\mathbf{c}_v = \mathbf{c}_v + e$. Subsequently all vertices directly connected to vertex v are activated by energy $e' = e/\rho(v)$, where $\rho(v)$ is the degree of vertex v. For convergence reasons vertex activation energy e' must also satisfy $e' > \theta$, where θ is a given small energy threshold value.

Algorithm 1 describes this process using a recursive function.

Algorithm 1 SPREAD-ACTIVATION$(v, e, \mathbf{c} \Leftarrow \mathbf{0})$

Require: Starting vertex v.
Require: Activation energy $e > 0$.
Require: Energy $\mathbf{c}$ accumulated on graph vertices.

```
1: c_v ⇐ c_v + e
2: e' ⇐ e/VERTEX-DEGREE(v)
3: if e' > θ then
4:    for all vertices t such as, there exists an edge from v to t do
5:       c ⇐ SPREAD-ACTIVATION(t, e', c)
6:    end for
7: end if
8: return c
```

[1] This matrix must satisfy some additional constraints such as being stochastic, irreducible and aperiodic. We explain and discuss these properties later.

[2] In an actual implementation an infinite loops would of course take infinite time to calculate, thus a small convergence threshold is usually used to stop computation resulting into an acceptable eigenvector approximation.

4 Rewriting Spreading Activation Search

Let us start with a simple adjacency matrix $\mathbf{B}$ of some graph G defined as

$$\mathbf{B}_{ij} = \begin{cases} 1 & \text{if } G \text{ contains an edge from vertex } i \text{ to } j \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Stochastic matrix $\mathbf{A}$ can be created from $\mathbf{B}$ by dividing each matrix element $\mathbf{A}_{ij}$ by corresponding vertex degree $\rho(i)$. In matrix algebra this is equivalent to matrix row normalization, where k is the total number of vertices.

$$\mathbf{A}_{ij} = \frac{\mathbf{B}_{ij}}{\rho(i)} = \frac{\mathbf{B}_{ij}}{\sum_{k} \mathbf{B}_{ik}} \tag{4}$$

Such stochastic matrix can be now easily exploited to calculate one energy distribution step of spreading activation search.

Simple matrix multiplication of a starting vector $\mathbf{r}_0$ by matrix $\mathbf{A}$ results into a new vector $\mathbf{r}_1$. This vector represents energy on each vertex of graph which has been distributed by neighboring edges, while $\mathbf{r}_0$ consists of starting activation energies for each vertex of graph G. By applying these matrix multiplications in an iterative fashion, successive energy distributions steps can be calculated.

$$\mathbf{r}_k = \mathbf{r}_{k-1}\mathbf{A} \tag{5}$$

In spreading activation search, final vertex ranks are obtained by accumulating energies, which were propagated through neighboring edges. Since we have shown that $\mathbf{r}_0, \mathbf{r}_1, \ldots, \mathbf{r}_\infty$ correspond to these energy propagation steps, final ranks can be easily calculated by summing up all propagated energies. After k steps final ranks $\mathbf{c}$ of graph vertices are computed as follows.

$$\mathbf{c}_k = \sum_{i=0}^{k} \mathbf{r}_i \tag{6}$$

With a silent assumption of discarding threshold a comparison of equations 1 and 5 reveals that power method (in its purest form) is also present in spreading activation search.

Thresholding can be included back into this calculation by adding a thresholding function τ_θ with threshold θ at each energy propagation step.

$$\mathbf{r}_k = \tau_\theta(\mathbf{r}_{k-1}\mathbf{A}) \tag{7}$$

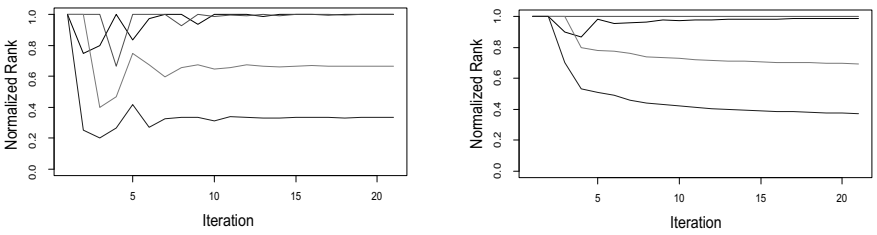
Where τ_θ is defined as

$$\left(\tau_\theta(\mathbf{r})\right)_i = \begin{cases} \mathbf{r}_i & \text{if } \mathbf{r}_i > \theta \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

5 Comparison of Rank Computation

The main difference between power method and spreading activation search lies in final rank calculation. While ranks computed by power method use only the last step (an eigenvector approximation), spreading activation search ranks sum up the history of the whole process of finding an eigenvector by power method.

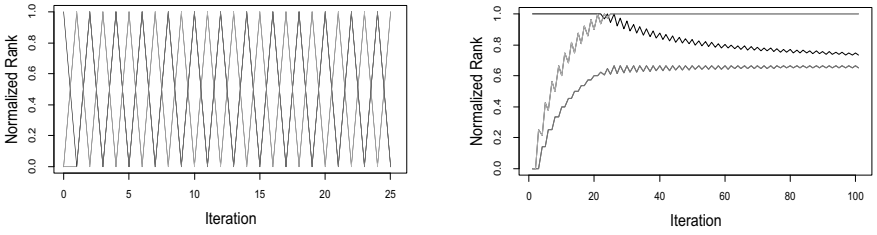
Figure 1 shows an example of eigenvector approximation using the power method (a) and a calculation of ranks by spreading activation search (b) on a small graph (4 vertices, 6 edges). Normalized values of ranks for both methods clearly converge to the same result of normalized eigenvector approximation.



(a) Power method. (b) Spreading activation search.

Fig. 1: Convergence of ranks computation on a small graph (4 vertices, 6 edges).

Figure 2 shows another example of rank calculation on a small periodic graph (5 vertices, 12 edges). By definition power method fails to converge on periodic graphs and starts to oscillate (a). However, spreading activation search successfully converges even on such periodic graph (b).



(a) Power method. (b) Spreading activation search.

Fig. 2: Comparison of ranks computation on a periodic graph (5 vertices, 12 edges).

A large adjacency matrix constructed from CiteSeer citation dataset consisting of approximately 3300 vertices has been used for further experimentation. This matrix has been transformed into PageRank form with damping factor

0.85 to ensure convergence of power method. Figure 5 shows the convergence of mean vertex rank difference of spreading activation calculation and power method calculation. Mean vertex rank difference is calculated as

$$\text{mean vertex rank difference} = \sum_{i=1}^{N} \frac{|\mathbf{a}_i - \mathbf{b}_i|}{N} \tag{9}$$

where N is number of vertices.

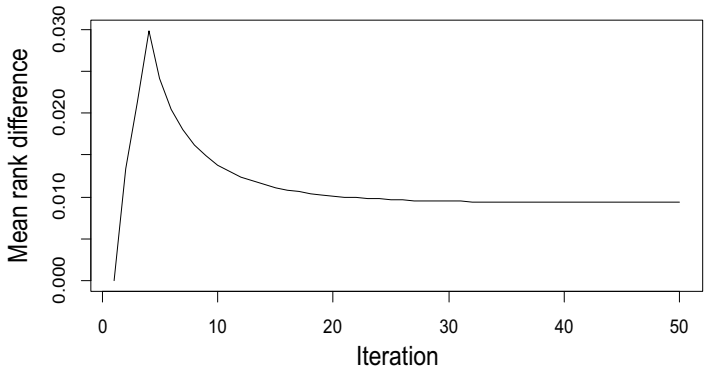


Fig. 3: Convergence of mean vertex rank difference of spreading activation search and power method calculation on CiteSeer dataset.

6 Revisiting PageRank, HITS and NodeRanking

The process of calculating ranks using spreading activation search starts by initializing a starting vector $\mathbf{r}_0$ which represents starting activation energies on vertices, then the power method is used to calculate energy distribution steps which are finally summed up resulting into final ranks. Table 1 shows the power method, PageRank [3], HITS [6], NodeRanking [7] and spreading activation search ranking algorithms rewritten into this initialization, iteration and ranking scheme.

Table 1: Graph ranking algorithms comparison.

Algorithm	*Initialization*	*Iteration*	*Ranking*
Power method	$\mathbf{r}_0$	$\mathbf{r}_k = \mathbf{r}_{k-1}\mathbf{A}$	$\mathbf{r}_k$
PageRank[3]	$\overbrace{(1-d)\frac{\mathbf{E}}{n} + d\mathbf{A}}^{\mathbf{M}}$ $\mathbf{r}_0$	$\mathbf{r}_k = \mathbf{r}_{k-1}\mathbf{M}$	$\mathbf{r}_k$
HITS[4]	$\mathbf{a}_0$, $\mathbf{h}_0$	$\mathbf{a}_k = \mathbf{a}_{k-1}\mathbf{A}^T\mathbf{A}$ $\mathbf{h}_k = \mathbf{h}_{k-1}\mathbf{A}\mathbf{A}^T$	$\mathbf{a}_k, \mathbf{h}_k$
NodeRanking[5]	$\overbrace{\mathbf{J}\frac{\mathbf{E}}{n} + (\mathbf{E}-\mathbf{J})\mathbf{A}}^{\mathbf{M}}$ $\mathbf{J}_{ii} = \frac{1}{\sigma(i)+1}$ $\mathbf{r}_0$	$\mathbf{r}_k = \mathbf{r}_{k-1}\mathbf{M}$	$\mathbf{r}_k$
Spreading activation with threshold θ	$\mathbf{r}_0$	$\mathbf{r}_k = \tau_\theta\left(\mathbf{r}_{k-1}\mathbf{A}\right)$	$\sum_{i=0}^{k} \mathbf{r}_i$

7 Future Work

We have shown that the power method used to calculate eigenvectors is also present in spreading activation search. Since fast eigenvector calculation is still a demanding and open problem we focus our further research to exploitation of various spreading activation search speedup techniques such as constrained spreading activation [8] or caching spreading activation [9] which can be used in distributed enviroment. By reusing these techniques we believe to find an alternative method for eigenvector calculation, however convergence characteristics for large graphs are still being explored.

8 Conclusion

By rewriting spreading activation search using matrix algebra we have shown that this graph ranking approach contains a minor variation of power method which is used to calculate eigenvectors of matrices. We have unified the notation of PageRank, HITS, NodeRanking and spreading activation search algorithms in a simple scheme consisting of three steps (initialization, iteration and ranking) demonstrating a significant similarity between them. We have empirically shown

[3] d is damping factor (probability of jumping to a random vertex), $\mathbf{E}$ is a matrix of all ones, n is number of all vertices in graph.

[4] $\mathbf{a}$ is authority rank, $\mathbf{h}$ is hubness rank.

[5] $\mathbf{J}$ is a probability matrix of a random jump from a vertex.

that normalized ranks calculated by power method and spreading activation search can converge to the same result, thus hopefully opening a new possibility to calculate eigenvectors by exploiting speedup techniques originally designed for spreading activation search.

Acknowledgment. This work was partially supported by the Slovak Research and Development Agency under the contract No. APVT-20-007104, the State programme of research and development "Establishing of Information Society" under the contract No. 1025/04 and the Cultural and Educational Grant Agency of the Slovak Republic, grant No. KEGA 3/5187/07.

References

1. Liu, B.: Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data (Data-Centric Systems and Applications). Springer-Verlag New York, Inc., Secaucus, NJ, USA (2006)
2. Wu, B., Goel, V., Davison, B.D.: Topical trustrank: using topicality to combat web spam. In: WWW '06: Proceedings of the 15th international conference on World Wide Web, New York, NY, USA, ACM Press (2006) 63–72
3. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank citation ranking: Bringing order to the web. In: Proceedings of the 7th International World Wide Web Conference, Brisbane, Australia (1998) 161–172
4. Návrat, P., Bieliková, M., Rozinajová, V.: Acquiring, organising and presenting information and knowledge from the web. Communication and Cognition **40**(1–2) (2007) 37–44
5. Maciej Ceglowski, A.C., Cuadrado, J.: Semantic search of unstructured data using contextual network graphs (2003)
6. Kleinberg, J.M.: Authoritative Sources in a Hyperlinked Environment. Journal of the ACM **46**(5) (1999) 604–632
7. Pujol, J.M., S.R., Delgado, J.: A Ranking Algorithm Based on Graph Topology to Generate Reputation or Relevance. In: Web Intelligence. Springer Verlag (2003)
8. Crestani, F., Lee, P.L.: WebSCSA: Web search by constrained spreading activation. In: ADL '99: Proceedings of the IEEE Forum on Research and Technology Advances in Digital Libraries, IEEE Computer Society (1999) 163
9. Suchal, J.: Caching Spreading Activation Search. In Bieliková, M., ed.: IIT.SRC 2007: Student Research Conference, Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava (April 2007) 151–155