Which One to Choose: Random Walks or Spreading Activation?

Serwah Sabetghadam, Mihai Lupu, and Andreas Rauber

Institute of Software Technology and Interactive Systems
Vienna University of Technology
Vienna, Austria
{sabetghadam,lupu,rauber}@ifs.tuwien.ac.at

Abstract. Modeling data as a graph of objects is increasingly popular, as we move away from the relational DB model and try to introduce explicit semantics in IR. Conceptually, one of the main challenges in this context is how to "intelligently" traverse the graph and exploit the associations between the data objects. Two highly used methods in retrieving information on structured data are: Markov chain random walks, as is the basic method for page rank, and spreading activation, which originates from the artificial intelligence area. In this paper, we compare these two methods from a mathematical point of view. Random walks have been preferred in information retrieval, while spreading activation has been proposed before, but not really adopted. In this study we find that they are very similar fundamentally under certain conditions. However, spreading activation has much more flexibility and customization options, while random walks holds concise mathematics foundation.

Keywords: Information retrieval, Graph, Spreading activation, Random Walks

1 Introduction

The data available today is becoming more and more connected. For example via platforms like Semantic Web, providing linkage between data objects semantically - leading to a connected network, or through social networks (text, images, videos on LinkedIn, Facebook or the like). Connected data poses structured IR as an option for retrieving more relevant data objects.

How to investigate relations between data objects and exploit the links available, is one of the main challenges in finding related information in the graph of linked data. There are different methods to traverse the graph, like different models of random walks or spreading activation.

Crestani [7] explains spreading activation as a method of associative retrieval to identify relevant information based on the associations between the information elements. Random walks is a sequence of independent, distributed discrete random path selection in a graph of objects. There are numerous works on utilizing random walks in order to find related data objects as well [5,1,15].

In this article we investigate these two approaches from a theoretical point of view. We categorize the routing in a graph of related data objects in IR,

D. Lamas et al. (Eds.): IRFC 2014, LNCS 8849, pp. 112–119, 2014.

as *query dependent* and *query independent* routing. In this article, we compare spreading activation and random walks according to these categories and show the similarities and differences in their behaviour.

The article is structured as follows: background of the work is described next, Section 3 describes the basic mathematic concepts behind the query independent graph traversal for the two methods. Section 4 follows with the more interesting case when we take into account the query, with the purpose of ranking the set of nodes based on some understanding of relevance. Finally, a discussion and conclusions are presented in Section 5.

2 Background

2.1 Spreading Activation

The idea of application of Spreading Activation in IR, originates from the works on Associative Retrieval in which it is possible to retrieve relevant information by retrieving information related to the query. This information is either already retrieved and is static, like the relation between the objects of information and indexings, or is dynamically achieved like based on user behaviour in the search session [7].

Spreading activation has various utilizations, for instance, Salton and Buckley [14] leverage spreading activation for identifying related terms and documents to improve the retrieval process. Rocha et al. [11] propose a model utilizing spreading activation for search in Semantic Web. Hussein and Ziegler use spreading activation in determining important elements in an ontology according to user's current context and past interactions [9].

2.2 Random Walks

PageRank is one of the most prominent examples leveraging random walks. It ranks websites based on the authority weights given to each page of a web graph. This authority weight is independent of textual content and is based on the web hyperlink structure. PageRank is based on a random surfer model that can be viewed as a stationary distribution of a Markov chain [1]. Another application of random walks is in Craswell and Szummer work, who model queries and clicked images in a graph [6]. They use random walks to retrieve more relevant images for each textual query. Furthermore, Clements et al. [4] use random walks through a social annotation graph, comprising of people, tags and content. They identify the influence of the design choices in social annotation systems on ranking tasks. Random walks has also been used in query expansion modeling. Collins et al. [5] identify weather a potential expansion term reflects aspects of the original query in the stationary distribution of their model.

2.3 Baseline Graph

We will be referring to a directed graph of objects, which we denote by $G = (V, E; W)$ where $V \in \mathbb{R}^n$ and $E \subset V \times V$ are the sets of vertices and edges

respectively, and the matrix $W \in \mathbb{R}^{n \times n}$ defines any weights in the graph (could be transition probabilities, in the case of random walks). In the usual way, $W_{u,v} = 0$ iff $(u, v) \notin E$.

3 Query Independent Graph Traversal

Random walks most famous instance is PageRank, which is query independent, while for spreading activation it has been shown that "*pure spreading activation is pointless*" [2]. What is the difference?

3.1 Spreading Activation

Spreading Activation is inspired by simulated neural networks without any training phase. Edge weights are based on the semantics of the modeled domain. The spreading activation procedure always starts with an initial set of activated nodes. Different values can be given to the initial nodes according to the task being solved. They are usually the result of a first stage processing of the query, e.g. a distance measure between the objects and the query. During propagation, other nodes get activated and ultimately, a set of nodes with respective activation is obtained. Here we explain how to compute the activation values of the nodes after some steps in the graph, independently of the query.

We denote the initial activation of the nodes as $a^{(0)}$ and the activation in t-th iteration as $a^{(t)}$. Three phases are commonly defined: preprocess, spreading and postprocess [2]. The preprocess consists of calculating an input value in_v for each node v by aggregating output values of its neighbours:

$$in_v^{(t)} = \sum_{u \in V} o_u^{(t-1)} \cdot W_{u,v} \tag{1}$$

where o_u is the output value of node u. Based on the input value, different functions are used to determine the activation value: linear, sigmoid, threshold, step function, etc. [7]. We denote any of these functions as *act*. Based on it, we calculate the activation value of each node:

$$a_v^{(t)} = act(in_v^{(t)}) \tag{2}$$

Finally, an output function *out* determines how much output of a node is passed to its neighbours. We define it as

$$o_v^{(t)} = out(a_v^{(t)}) \tag{3}$$

This function avoids retention of activation from previous iterations and helps control the overall activation in the graph [7]. Putting all these equations together, we obtain the following general formula to calculate the activation at the next step:

$$a_v^{(t+1)} = act\left(\sum_{u \in V} out(a_u^{(t)}) \cdot W_{uv}\right) \tag{4}$$

Weighting in Spreading Activation There is no specific constraint in spreading activation on weight definition or weight values on the edges, and it is application dependent. For example, Rocha et al. [11] define the edge weight based on the number of relations via neighbours between the two connected nodes u and v, to the number of relations of node u to its neighbours. Hussein and Ziegler [9] define the weighting based on the context defined by an expert in the preliminary step of system definition.

3.2 Random Walks

Different variants of random walks exist, but the Markov Chain random walks is by far the most commonly used in the IR literature, and we focus on it here. Markov Chain is a mathematical system for transitions between a finite set of states. The transition probabilities between the nodes form a matrix. In our case they are represented by the W matrix of transition weights. By $W_{u,v}$ we understand $P(v|u)$, the probability of moving from node u to node v. The matrix W is row-stochastic, e.i. the probabilities on one row sums up to one. The objective of random walks is to reach a probability distribution over the set of nodes V. If we view this as a vector $p \in \mathbb{R}^n$, we can denote by $p^{(0)}$ and $p^{(t)}$ the initial probability distribution, and respectively the probability distribution over the set of nodes at time t, which is computed by:

$$p^{(t)} = p^{(t-1)}W = p^{(0)}W^t \tag{5}$$

Weighting in random walks Weighting the edges for random walks presents as much flexibility as in spreading activation, with the one constraint that the matrix must be row-stochastic. For example, Craswell and Zsummer [6] define the weight based on the normalized value of the number of clicks between two nodes: $P_{t+1|t}(k|j) = C_{jk}/\sum_i C_{ji}$. Also they utilize the factor of *self-transitivity* (s). This parameter helps significantly to control the walk pace. It can be interpreted as the importance of staying in nodes. Using random walks to calculate the probability of transition between two nodes shows the volume of the path [15], which increases with the number of paths with the same length between the two nodes.

3.3 Discussion

Based on Equation 3, the output of a node in spreading activation is the result of applying the activation and output functions on the input of the node. If the input function is defined as linear combination and the output and activation functions are identity functions, then the Equation 4 in spreading activation can be written as $a_v^{(t+1)} = \sum_{u \in V} a_u^{(t)} \cdot W_{uv}$, which in compact form is:

$$a^{(t+1)} = a^{(0)} \cdot W^{t+1} \tag{6}$$

Comparing Equations 5 and 6 we observe that in a query independent case, both random walks and spreading activation perform the same scenario. In this

case convergence of the weighting matrix is important since there is no limit to stop the propagation or walk in the graph. It is from here that the difference in utility of the two methods, mentioned at the beginning of this section, stems.

According to Perron-Frobenius Lemma [8], the power iteration of a matrix W converges to its stationary distribution if the matrix is ergodic (irreducible and aperiodic). In graph terminology, ergodic refers to a connected and not bipartite graph. Based on this lemma for nonnegative matrices, eigenvalue 1 is the largest eigenvalue.

In practice, the number of iterations to reach to stationary distribution as a fixed state is important. This is defined as mixing time (Mt). The smaller the second largest eigenvalue is, the faster the convergence rate is. In more detail, mixing time is proportional to the inverse of difference between the largest eigenvalue and the second eigenvalue $Mt \propto (1 - \lambda_2)^{-1}$.

Stationary distribution in random walks, providing the probability distribution over all nodes after convergence, has its own applications (e.g. in PageRank). However, spreading activation is mainly utilized to provide highly customized solutions using heuristic restrictions and different activation and output functions in various applications. As Berthold et al. [2] proved that pure spreading activation is meaningless.

Self-transitivity. One of the factors affecting convergence speed is the self-transitivity value (s): a high value slows down the walk while a low value speeds up. For the probabilities of node u it will be modified as:

$$P_{t+1|t}(v|u) = \begin{cases} (1-s)W_{uv} & v \neq u \\ s & v = u \end{cases} \tag{7}$$

In compact form the transition matrix becomes $sI + (1 - s)W$ which has $s+(1-s)\lambda$ as eigenvalue of a matrix. Therefore, applying s value, does not change the stationary distribution, as the eigenvalue of the new matrix is the same as matrix W. We know the largest eigenvalue of W is 1. Then the largest eigenvalue for this combination also remains 1 and does not change the convergence property of the matrix.

In spreading activation, self-transitivity is referred to as "inertia". It can be used to retain the previous state partially during iteration: $a^{(t)} = a^{(t-1)} + Wa^{(t-1)}$. In a closed form is $a^{(t)} = (I + W)^t a^{(0)}$ [2] with the same eigenvector of W. Using inertia, the weight matrix is changed to add a self loop of unit weight to each node.

We see that self-transitivity factor is applicable in both methods without affecting the eigenvector of the weighting matrix.

4 Query Dependent Routing

It is potentially desirable in IR that the graph traversal be dependent on the query. We look now at how this has been done in the literature, for the two methods studied.

4.1 Spreading Activation

In order to avoid pure spreading activation, leading to query independent results, common heuristic constraints are defined:

- Distance constraint [7]: imposes a hard limit on the number of iterations the activation can traverse
- Fan-out constraint [7]: to cease activation in very high fan-out nodes (indicating common nodes)
- Path constraint [7]: some edges are preferred to others in transferring the activation energy
- Concept type constraint [11]: some nodes are not traversed in the activation process
- Accumulation [2]: as a form of *iteration with memory*, this approach modifies the iterations to take into account not only the last state (of the activation propagation), but the sequence from the beginning. As a closed formula, we have:

$$a^* = \sum_{t=0}^{\infty} \lambda(t) \cdot a^{(t)} = \sum_{t=0}^{\infty} \lambda(t) W^t a^{(0)} \tag{8}$$

  where λ is decaying factor used to make the sum convergent.

4.2 Random Walks

One of the methods making random walks query dependent is *query-dependent probability computation*. Richardson and Domingos [10] modify the random surfer model used in PageRank by considering the query in the probability computation. Assuming $R_q(u)$ as a measure of relevancy between query q and page u, they suggest: $P_q(v|u) = R_q(u)/\sum_{k|u} R_q(k)$ where $P_q(v|u)$ is the probability of going from node u to v, and k goes through neighbours.

Another way of query dependent random walks is to employ the Metropolis-Hastings method. Provided with a reasonable estimation for the probability of relevance, this method could provide a better approximation to the true relevancy probability distribution.

We know that Metropolis-Hastings algorithm can be used to generate a sequence of samples from a probability distribution π which is difficult to sample from directly [3]. The algorithm steps are as follows:

- Consider a stochastic matrix W
- Consider an initial state $v_0 \in V$
- for $i = 0, 1, 2, ...,$
  - sample v from W_{v_i}
  - sample x from uniform $(0, 1)$
  - if $x < \widetilde{\pi}(v)/\widetilde{\pi}(v_i)$ then $v_{i+1} = v$ else $v_{i+1} = v_i$

After the mixing time, the probability of the walker to be in any specific node v will be $\pi(v)$.

Mapped to our problem, the proposed matrix W is our stochastic transition matrix. As a stationary distribution over the set of nodes, we would like to have the *true* relevance probability distribution by the indexing ranked results. This is the $\pi(v)$ distribution from which we cannot directly sample. Instead, we have the $\widetilde{\pi}(v)$ which could be a relevance scoring function (e.g. a BM25 score between the data object v_i and the query). Metropolis-Hastings would formally provide us with a method to sample from the probability distribution, if the approximate probability $\widetilde{\pi}$ is properly chosen.

4.3 Discussion

Comparing the two methods, we find that simple constraints like distance threshold are applicable in both methods to make the traversal query dependent, for instance stopping random walks after a number of steps [6], or applying distance threshold in spreading activation [6,11]. The path and concept type constraints in spreading activation (as applied in [11]), make the graph traversal domain or context dependent, rather than strictly query dependent. Translated to a random walks understanding, type and path constraint, would assign zero probability to certain edges to nodes of that type.

By defining different types of constraints, spreading activation provides more options to customize the traversal. This appears much more arbitrary than the approach in random walks, which assigns probabilities based on assumed relevance in the context of IR.

It is worth to notice that we have a model for multimodal information retrieval under evaluation, named Astera [13], which models the data as a graph of information objects. We have chosen the spreading activation method to manage the graph traversal.

The reason goes back to the relation types and weight definitions in Astera. We define four relation types between information objects: semantic, part-of, similarity and facet relations [12]. The definition of part-of relation (e.g. an image part of a document) is containment, which transfers the whole activation value from the parent node to the part-of node. Weight of value 1 is defined on this type of relation which does not comply with normalized weighting definition, in which the *sum* of all edge weights should be 1.

Further, we leverage different types of constraints defined in spreading activation, e.g. distance constraint to stop the process after limited number of steps, or fan-out constraint to avoid energy transfer to high fan-out nodes.

5 Conclusion

We investigated two methods of graph traversal, namely spreading activation and random walks in the context of IR. These two methods are highly comparable in query independent routing, holding the same underlying mathematics. Their different behavior are only due to convergence properties.

In query dependent case, we noticed that giving the variety of options to spreading activation method to customize the routing, makes the graph traversal highly customizable and domain dependent. Of course, some constraints like distance threshold is applicable on both methods. On contrary, random walks is more clear and less flexible in making the walk dependent to the query. However, it provides the option of defining the probability based on relevancy in IR context - providing query dependent routing.

We conclude that with some limitations on spreading activation, these two methods are in essence the same, however, spreading activation provides more flexibility, leading to more complexity and less tractable scenario in comparison to random walks.

References

1. Berkhin, P.: A survey on pagerank computing. Internet Mathematics 2 (2005)
2. Berthold, M.R., Brandes, U., Kötter, T., Mader, M., Nagel, U., Thiel, K.: Pure spreading activation is pointless. In: Proc. of CIKM (2009)
3. Chib, S., Greenberg, E.: Understanding the metropolis-hastings algorithm. The American Statistician 49(4) (1995)
4. Clements, M., De Vries, A.P., Reinders, M.J.T.: The task-dependent effect of tags and ratings on social media access. ACM Trans. Inf. Syst. 28(4) (2010)
5. Collins-Thompson, K., Callan, J.: Query expansion using random walk models. In: Proc. of CIKM (2005)
6. Craswell, N., Szummer, M.: Random walks on the click graph. In: Proc. of SIGIR (2007)
7. Crestani, F.: Application of spreading activation techniques in information retrieval. Artificial Intelligence Review 11 (1997)
8. Golub, G.H., Van Loan, C.F.: Matrix computations, 3rd edn. Johns Hopkins University Press (1996)
9. Hussein, T., Ziegler, J.: Adapting web sites by spreading activation in ontologies. In: Int. Workshop on Recommendation and Collaboration (in Conjunction with IUI 2008) (2008)
10. Richardson, M., Domingos, P.: The Intelligent Surfer: Probabilistic Combination of Link and Content Information in PageRank. In: Proc. of NIPS (2002)
11. Rocha, C., Schwabe, D., Aragao, M.P.: A hybrid approach for searching in the semantic web. In: Proc. of WWW (2004)
12. Sabetghadam, S., Lupu, M., Rauber, A.: Astera - a generic model for multimodal information retrieval. In: Proc. of Integrating IR Technologies for Professional Search Workshop (2013)
13. Sabetghadam, S., Bierig, R., Rauber, A.: A hybrid approach for multi-faceted IR in multimodal domain. In: Kanoulas, E., Lupu, M., Clough, P., Sanderson, M., Hall, M., Hanbury, A., Toms, E. (eds.) CLEF 2014. LNCS, vol. 8685, pp. 86–97. Springer, Heidelberg (2014)
14. Salton, G., Buckley, C.: On the use of spreading activation methods in automatic information. In: Proc. of SIGIR (1988)
15. Szummer, M., Jaakkola, T.: Partially labeled classification with markov random walks. In: NIPS (2001)