

# Pure Spreading Activation is Pointless\*

Michael R. Berthold  
Martin Mader

Ulrik Brandes  
Uwe Nagel

Tobias Kötter  
Kilian Thiel

Department of Computer & Information Science, University of Konstanz  
Firstname.Lastname@uni-konstanz.de

## ABSTRACT

Almost every application of spreading activation is accompanied by its own set of often heuristic restrictions on the dynamics. We show that in constraint-free scenarios spreading activation would actually yield query-independent results, so that the specific choice of restrictions is not only a pragmatic computational issue, but crucially determines the outcome.

## Categories and Subject Descriptors

H.3.3 [Information storage and retrieval]: Information Search and Retrieval—*retrieval models, search process*

## General Terms

Algorithms, Theory, Performance

## Keywords

Spreading activation, cosine similarity, information retrieval

## 1. INTRODUCTION

Spreading activation was proposed by Quillian and Collins [7, 3] as a technique to query networks of information. It facilitates the extraction of subgraphs, nodes, or edges relevant to a given query. Upon activation of a number of specific nodes, their activation is spread iteratively to adjacent nodes until some termination criterion is met. The result is determined from the subset of activated nodes, their activation level, and induced subgraph.

While spreading activation techniques have initially been applied to semantic networks, associative retrieval [9] has paved the way for applications in information retrieval [2, 10, 4]. The fundamental idea of associative retrieval is that

\*Research supported in part by DFG under grant GRK 1042 (Research Training Group “Explorative Analysis and Visualization of Large Information Spaces”) and the European Commission in 7th Framework Programme (FP7-ICT-2007-C FET-Open, contract no. BISON-211898).

related information is connected in a network and that relevant information can be retrieved by considering associations of concepts that are either known to be relevant or specified by the user.

To our knowledge, all these methods share the usage of constraints to steer the spread of activation inside a network, such as distance constraints to terminate the spreading procedure after a certain number of iterations, or fan-out constraints to direct the spreading. In Section 3 we show that without such constraints, spreading activation converges to query-independent fixed states, rendering pure (i.e., constraint free) spreading activation an inadequate technique for answering queries. While this drawback is generally overcome by applying heuristic constraints that enforce convergence to query-specific states, their precise effects are difficult to analyze. In Section 4 we show that another approach to produce query-dependent activation levels is the accumulation of intermediate states. This approach is more amenable to theoretical analysis and can be combined freely with other commonly used heuristics.

## 2. PRELIMINARIES

Spreading activation is related to neural networks. What both methods have in common is that units can be activated by incoming activation, and edges spread outgoing activation to adjacent units. Based on the eight modeling aspects of neural networks defined in [8], the functionality of spreading activation can be specified in terms of the three components described below.

### 2.1 Framework

Activation is spread on a graph  $G = (V, E; w)$  with weights  $w : E \rightarrow \mathbb{R}$ . For ease of exposition we assume that  $V = \{1, \dots, n\}$  and that  $G$  is undirected, but our results easily generalize to directed graphs. We extend  $w$  to  $V \times V$  by letting  $w(u, v) = 0$  if  $(u, v) \notin E$ . The set of neighbors of  $v \in V$  is denoted by  $N(v) = \{u : \{u, v\} \in E\}$ . The activation state at time  $k$  is denoted by  $\mathbf{a}^{(k)} \in \mathbb{R}^V$ , where  $\mathbf{a}_v^{(k)}$  is the activation of  $v \in V$ . The state at time  $k > 0$  is obtained from the state at time  $k - 1$  via the following three families of functions.

**Output functions  $\text{out}_v : \mathbb{R} \rightarrow \mathbb{R}$**  An output function determines the outgoing activation  $\mathbf{o}_v^{(k)} = \text{out}_v(\mathbf{a}_v^{(k)})$  of a node  $v$  at time  $k$  based on its current activation level  $\mathbf{a}_v^{(k)}$ . Output functions can be used to normalize the output in certain ways, i.e. to ensure that the sum of activation in the whole network is constant.

**Input functions**  $\mathbf{in}_v : \mathbb{R}^n \rightarrow \mathbb{R}$  An input function aggregates incoming activation of a node  $v$  and defines its input  $\mathbf{i}_v^{(k)} = \mathbf{in}_v(\mathbf{o}^{(k-1)})$  at time  $k$ . A prototypical choice is the total weighted output activation of all neighbors,  $\mathbf{in}_v(\mathbf{o}^{(k-1)}) = \sum_{u \in N(v)} \mathbf{o}_u^{(k-1)} w(u, v)$ .

**Activation functions**  $\mathbf{act}_v : \mathbb{R} \rightarrow \mathbb{R}$  An activation function determines the level  $\mathbf{a}_v^{(k)} = \mathbf{act}_v(\mathbf{i}_v^{(k)})$  of activation resulting from the input, and thus in particular whether a node is activated, i.e. whether its activation will be spread in the next iteration. At this point non-linearity is often introduced into the system, e.g., using sign, threshold, or sigmoid functions.

The initial state  $\mathbf{a}^{(0)}$  is usually defined by activating those nodes that represent the query. An activation spreads across incident edges to adjacent nodes and activates these nodes as well. The process is usually terminated after a fixed number of iterations, activation of a given number of nodes, or similar criteria. The query response is determined from the subgraph induced by activated nodes. Often the desired form of result in information retrieval is a ranking of the nodes obtained from their final output activation (henceforth simply referred to as the activation).

## 2.2 Linear Standard Scenario

The general framework is most commonly instantiated with a linear input function and the identity function for activation and output. In this scenario, the spreading activation process is conveniently described in matrix notation. Given a graph  $G = (V, E; w)$  and an activation vector  $\mathbf{a}^{(k-1)}$ , the next activation step yields  $\mathbf{a}_v^{(k)} = \sum_{u \in N(v)} w(u, v) \cdot \mathbf{a}_u^{(k-1)}$  for all  $v \in V$ . With the weight matrix  $W \in \mathbb{R}^{n \times n}$  defined by  $(W)_{uv} = w(u, v)$  (recall that  $w(u, v) = 0$  if  $(u, v) \notin E$ ), a single iteration can be stated compactly as  $\mathbf{a}^{(k)} = W\mathbf{a}^{(k-1)}$ , and therefore

$$\mathbf{a}^{(k)} = W^k \mathbf{a}^{(0)}. \quad (1)$$

## 2.3 Constraints

In addition to termination criteria, the following heuristic constraints are common [4]:

**Distance constraints:** activation decreases with distance from initially activated nodes and finally stops at a certain distance, with the argument that the strength of the relation decreases with their semantic distance.

**Fan-out constraints:** nodes of high out-degree are not activated to avoid excessive spreading.

**Path constraints:** activation spreads across preferential paths that reflect specific inference rules.

**Activation constraints:** to avoid the activation of all nodes that receive activation at all, a threshold function can be applied. In this case a certain degree of input activation is required to activate the node.

Crestani [4] argues that constraints are necessary, because pure (constraint-free) spreading activation has three serious drawbacks:

1. Activation spreads over the entire network.
2. Semantics of relations, represented as edge labels can hardly be interpreted and considered.
3. The implementation of an inference procedure based on the semantics of relations (edge labels) is difficult.

We show that the standard approach described above exhibits a crucial fourth drawback, namely convergence to a single query-independent state.

## 3. QUERY INDEPENDENCE

From Eq. (1) it is obvious that pure linear activation spreading corresponds to power iteration with matrix  $W$ . From the Perron-Frobenius Lemma it is well known (see, e.g., [5]) that power iteration converges to the unique (up to scaling) principal eigenvector of  $W$ , if  $W$  is irreducible and aperiodic, or, in graph terminology,  $G$  is connected and not bipartite. These conditions are most often fulfilled in information retrieval scenarios.

The absolute value of the largest eigenvalue of  $W$  is called the spectral radius,  $\rho(W)$ . In network analysis, its associated eigenvector is commonly known as eigenvector centrality [1]. Its entries are guaranteed to have the same sign and induce a reasonable ranking reflecting a global notion of each node's importance in the graph structure.

Consequently, after a sufficient number of iterations, the activation vector will approach eigenvector centrality. While this may actually be a reasonable default answer, it is not at all related to the query.

## 4. AVOIDING QUERY INDEPENDENCE

Clearly, query-independence is not desirable in information retrieval. While the constraints used in typical applications of spreading activation avoid this problem, their effects are difficult to assess theoretically.

### 4.1 Iteration with Memory

To avoid query independence we pursue a different approach here. The iterations are modified to take more than just the directly preceding state into account. Thereby the intermediate states following the initial state (the query) are able to influence the final result. Three seemingly natural such attempts are discussed below.

#### Accumulation.

A straightforward approach whereby the entire iteration history is taken into account is to define the final activation as the sum of all intermediate states. While the iteration converges, the sum of intermediate states will not unless we introduce a vanishing series of weights. Hence consider the accumulated activation

$$\mathbf{a}^* = \sum_{k=0}^{\infty} \lambda(k) \cdot \mathbf{a}^{(k)} = \left( \sum_{k=0}^{\infty} \lambda(k) W^k \right) \mathbf{a}^{(0)},$$

where  $\lambda(k)$  is a decay function ensuring convergence.

Note that choosing  $\lambda(k) = \alpha^k$  for a constant  $\alpha > 0$  yields a variant of another well-known centrality index: Katz' status [6] is defined as  $c_{\text{Katz}}(W) = \sum_{k=1}^{\infty} (\alpha W^T)^k \mathbf{1}$ , where  $\mathbf{1}$  is the vector of all ones and  $\alpha < \rho(W)^{-1}$  guarantees convergence. The status attribution matrix  $\sum_{k=1}^{\infty} (\alpha W^T)^k$  thus equals  $\sum_{k=1}^{\infty} \lambda(k) W^k$  and our accumulated activation  $\mathbf{a}^*$  can be interpreted as a global importance ranking biased by a query-defined  $\mathbf{a}^{(0)}$ .

Usually,  $n$  is large and  $W$  is sparse, so that precomputing the fully populated limit matrix is prohibitive. The computation can nevertheless be accelerated by noting that

$\mathbf{a}^* = (I - \alpha W)^{-1} \mathbf{a}^{(0)}$  and applying methods for sparse matrix inversion.

### Activation renewal.

A seemingly different approach is the constant renewal of the initial activation as in  $\mathbf{a}^{(k)} = \mathbf{a}^{(0)} + W \mathbf{a}^{(k-1)}$ . Recursive substitution shows, however, that  $\mathbf{a}^{(k)} = \left( \sum_{i=0}^k W^i \right) \mathbf{a}^{(0)}$ , therefore this is actually the special case  $\lambda(k) = 1$  of the previous approach, and convergence is guaranteed only for  $\rho(W) < 1$ .

### Inertia.

A frequently applied smoothing method adds inertia to an iteration process by partially retaining the previous state as in  $\mathbf{a}^{(k)} = \mathbf{a}^{(k-1)} + W \mathbf{a}^{(k-1)}$  or in closed form  $\mathbf{a}^{(k)} = (I + W)^k \mathbf{a}^{(0)}$ . We are thus only modifying the influence weight matrix in a way that corresponds to adding self loops of unit weight to each node. Consequently, the final activation is an eigenvector corresponding to the dominant eigenvalue of  $I + W$ , and therefore continuous to be the principal eigenvector of  $W$ . This is because adding  $I$  adds one to every eigenvalue without changing the associated eigenvectors. Moreover, convergence of power iteration may be slowed down, as the ratio of the largest and second-largest absolute eigenvalue may be reduced. On the other hand, the potential problem of bipartiteness is avoided.

## 4.2 Normalization

Of the approaches above, accumulation turned out to be the only candidate worth investigating, since the final ranking of inertia is equal to that of pure spreading activation and the convergence of activation renewal is not guaranteed. Choosing an appropriate decay function, however, is difficult. Even for  $\lambda(k) = \alpha^k$  we need to know the spectral radius  $\rho(W)$  for a reasonable choice of  $\alpha$ .

In contrast, accumulating normalized activation vectors is much easier, because a converging series is obtained for any  $\alpha \in (0, 1)$ , and it is indeed appropriate to normalize the total activation after each iteration to avoid numerical problems during power iteration.

Consider therefore the normalized power iteration  $\mathbf{a}^{(k)} = W \mathbf{a}^{(k-1)} / \|W \mathbf{a}^{(k-1)}\|$ , and assume that  $\|\cdot\|$  is the  $l_2$  norm (our arguments apply to other norms as well). The following theorem implies that the normalized iteration not only simplifies the choice of a decay function for accumulation, but also removes the need for conditions on  $W$ .

**THEOREM 1.** *Given any matrix  $W \in \mathbb{R}^{n \times n}$  and  $\alpha \in [0, 1)$ , a vector  $\mathbf{a}^* \in [-(1 - \alpha)^{-1}, (1 - \alpha)^{-1}]^n$  exists such that*

$$\mathbf{a}^* = \lim_{m \rightarrow \infty} \sum_{k=0}^m \alpha^k \mathbf{a}^{(k)} \quad \text{with} \quad \mathbf{a}^{(k)} = \frac{W \mathbf{a}^{(k-1)}}{\|W \mathbf{a}^{(k-1)}\|}.$$

**PROOF.** Since  $\|\mathbf{a}^{(k)}\| = 1$ ,  $|\mathbf{a}_i^{(k)}| \leq 1 \forall i \in V$ . Therefore  $\mathbf{a}_i^* \leq \sum_{k=0}^{\infty} \alpha^k = (1 - \alpha)^{-1}$  and  $\mathbf{a}_i^* \geq -\sum_{k=0}^{\infty} \alpha^k = -(1 - \alpha)^{-1}$ . Since  $\lim_{k \rightarrow \infty} \alpha^k \mathbf{a}_i^{(k)} = 0$  this concludes the proof.  $\square$

## 5. ALTERNATING COSINE SIMILARITY

As an illustrative application we consider relatedness queries in a database of term-document relationships. The example is also interesting because it renders iterated cosine similarity a special case of spreading activation.

## 5.1 Scenario

Consider a graph  $G = (V, E; w)$  in which the nodes  $V = D \uplus T$  are partitioned into documents  $D$  and terms  $T$ . Edges  $E = \{\{d, t\} : d \in D, t \in T, w(d, t) > 0\}$  with weights  $w : D \times T \rightarrow \mathbb{R}_{\geq 0}$  describing the relations between documents and terms. A common choice would be tf-idf (term frequency, inverse document frequency) values. Without loss of generality, we assume that  $G$  is connected.

As noted in [11], spreading activation can be instantiated for this bipartite network in such a way that activation after a single step is equal to the standard cosine similarity measure between a virtual query document and all documents in the network. This is achieved by defining a spreading function for each class of the bipartition as follows.

Let every iteration consist of two phases, one in which documents activate terms, and another in which terms activate documents. The spreading function for terms in  $T$  is a normalized weighted sum of document activations,

$$\mathbf{a}_t^{(k)} = \frac{\sum_{d \in N(t)} \mathbf{a}_d^{(k-1)} w(d, t)}{\left( \sqrt{\sum_{d \in N(t)} w(d, t)^2} \cdot \|\mathbf{a}_d^{(k-1)}\| \right)}.$$

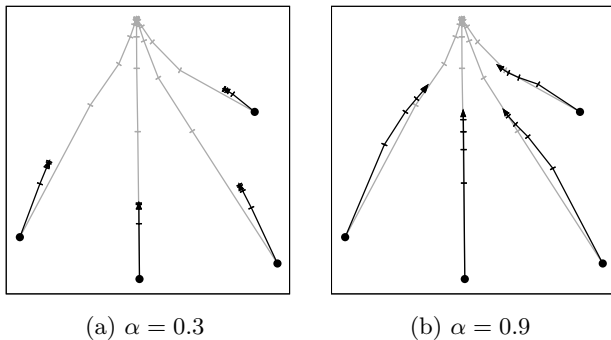
The activation of documents is updated analogously, but by using term activations from within the same iteration,

$$\mathbf{a}_d^{(k)} = \frac{\sum_{t \in N(d)} \mathbf{a}_t^{(k)} w(d, t)}{\left( \sqrt{\sum_{t \in N(d)} w(d, t)^2} \cdot \|\mathbf{a}_t^{(k)}\| \right)}.$$

Given a set of query terms  $d_q$  (which may be seen as a virtual document representing the query terms), let  $\mathbf{d}_q$  be the corresponding normalized term-vector space representation. That is  $(\mathbf{d}_q)_t = 1/\|\mathbf{d}_q\|$ , if term  $t$  is part of the query and otherwise 0. Furthermore let  $\mathbf{d} = (w(d, t_1), \dots, w(d, t_{|T|}))^T$  be the vector representation of document  $d \in D$ . Then the cosine similarity of  $d_q$  to every document  $d$  in the network is calculated by activating the terms contained in  $d_q$ , i.e.  $\mathbf{a}_t^{(0)} := (\mathbf{d}_q)_t$ , and spreading their activation to the documents. The above spreading functions are of course chosen such that  $\mathbf{a}_d^{(0)} = \cos(\mathbf{d}, \mathbf{d}_q)$ . When the next step is executed, i.e. the activation is spread from documents to terms, their new activation represents a new virtual document and after spreading this back to documents, their activation represents their cosine similarity to this new virtual document, and so on.

## 5.2 Analysis

Let  $W \in \mathbb{R}^{D \times T}$  be the weight matrix of  $G$ , i.e.  $(W)_{d,t} = w(d, t)$  for all  $d \in D$  and  $t \in T$ . In addition, let  $W_D \in \mathbb{R}^{D \times T}$  be the  $l_2$ -row-normalization of  $W$ ,  $(W_D)_{d,t} = w(d, t)/\|\mathbf{d}\|$  and let  $W_T \in \mathbb{R}^{T \times D}$  be the  $l_2$ -row-normalization of its transpose,  $(W_T)_{t,d} = w(d, t)/\|\mathbf{t}\|$ , with  $\mathbf{t}$  as the vector representing term  $t \in T$  analog to the document vectors. Furthermore, let  $\mathbf{a}_D^{(k)}$  and  $\mathbf{a}_T^{(k)}$  denote the activation of documents and terms after  $k$  iterations. In matrix-vector notation, the above spreading functions become  $\mathbf{a}_D^{(k)} = W_D \cdot \mathbf{a}_T^{(k)} / \|\mathbf{a}_T^{(k)}\|$  and  $\mathbf{a}_T^{(k)} = W_T \cdot \mathbf{a}_D^{(k-1)} / \|\mathbf{a}_D^{(k-1)}\|$ . Combined this gives a



**Figure 1: 2D-projections of state trajectories in the alternating cosine similarity application with accumulated spreading activation.**

power iteration representing

$$\mathbf{a}_D^{(k)} = \frac{W_D(W_T W_D)^k \mathbf{a}_T^{(0)}}{\|(W_T W_D)^k \mathbf{a}_T^{(0)}\|}.$$

Since  $G$  is connected, the graph underlying  $W_T W_D$  is connected and not bipartite, therefore the power iteration converges and the considerations of the previous sections apply in this scenario.

### 5.3 Illustration of Results

To illustrate the analytical results we applied the above described method to the TIME Magazine dataset of the SMART test collection.<sup>1</sup> This dataset consists of 425 documents and 83 queries together with relevance judgements for the documents.

The example is not intended to be a performance benchmark, but rather an illustration of the influence and convergence behavior of accumulation decay factor  $\alpha$ . Therefore, in contrast to other experimental evaluations, the precise characteristics of the dataset are not important here. Instead, we compare the state trajectories of pure spreading activation and spreading activation with accumulation on randomly chosen queries.

Figure 1 shows the trajectories of the iteration processes on four sample queries. Intermediate states are projected onto two dimensions using multidimensional scaling of the dissimilarities  $\text{dis}(\mathbf{r}_1, \mathbf{r}_2) = 1 - \cos(\mathbf{r}_1, \mathbf{r}_2)$ . These angular dissimilarities were chosen mostly to support geometric interpretation of the iterative progression. Clearly, for comparison of two rankings, other metrics such as rank inversion distance would be more appropriate. This is because activation vectors resulting in equal rankings would be projected onto the same coordinates, whereas cosine-based dissimilarities differentiate between different activation vectors implying the same ranking.

Trajectories of the intermediate states of pure spreading activation are depicted in gray. Black arrows indicate state trajectories of spreading activation with accumulation. Consecutive intermediate states  $\mathbf{a}_d^{(k)}$  are connected by a line, starting at  $\mathbf{a}_d^{(0)}$ , which is represented by a dot. This initial activation equals the cosine distance of the documents  $D$  and the query document  $\mathbf{d}_q$ . It is evident from the trajectories

that all results converge very quickly to the same default result, i.e. the principal eigenvector of the system. With increasing  $\alpha$  the accumulation approach yields activations resembling the query-independent one of pure spreading activation. This suggests that the choice of  $\alpha$  represents what can be interpreted as the trade-off between narrow and more generalist responses to relatedness queries.

## 6. CONCLUSION

Commonly-used spreading activation strategies constitute linear systems with additional constraints. Without these constraints, the approaches are equivalently described by power iteration converging to a query-independent state. Constraints therefore crucially influence the actual output, and their effects should be investigated more thoroughly.

Based on weighted accumulation of normalized activation states, we propose an alternative strategy to introduce query-dependence in a controlled way. Results can be geared toward the initial query or a global response of pure spreading activation. This method is independent of other constraints and can be combined with them freely.

## 7. REFERENCES

- [1] P. Bonacich. Factoring and weighting approaches to status scores and clique identification. *Journal of Mathematical Sociology*, 2:113–120, 1972.
- [2] P. Cohen and R. Kjeldsen. Information retrieval by constrained spreading activation in semantic networks. *Information Processing and Management: an International Journal*, 23(4):255–268, 1987.
- [3] A. Collins and E. Loftus. A spreading-activation theory of semantic processing. *Psychological Review*, 82(6):407–428, 1975.
- [4] F. Crestani. Application of spreading activation techniques in information retrieval. *Artificial Intelligence Review*, 11(6):453–482, 1997.
- [5] G. H. Golub and C. F. van Loan. *Matrix Computations*. Johns Hopkins University Press, 3rd edition, 1996.
- [6] L. Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.
- [7] M. Quillian. Semantic memory. In M. Minsky, editor, *Semantic Information Processing*. MIT Press, 1968.
- [8] D. Rumelhart and J. McClelland. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. MIT Press, 1986.
- [9] G. Salton. *Automatic Information Organization and Retrieval*. McGraw Hill, 1968.
- [10] G. Salton and C. Buckley. On the use of spreading activation methods in automatic information retrieval. In *Proc. 11th Ann. Intl. ACM SIGIR Conf. Research and Development in Information Retrieval*, pages 147–160, 1988.
- [11] R. Wilkinson and P. Hingston. Using the cosine measure in a neural network for document retrieval. In *Proc. 14th Ann. Intl. ACM SIGIR Conf. Research and Development in Information Retrieval*, pages 202–210, 1991.

<sup>1</sup>ftp://ftp.cs.cornell.edu/pub/smart