Spreading Activation for Web Scale Reasoning: Promise and Problems

Nazihah Md. Akim[1], Alan Dix[1,2], Akrivi Katifori[3],
Giorgos Lepouras[4], Nadeem Shabir[2], Costas Vassilakis[3]

[1.] Computing Department
Lancaster University, Lancaster, UK

[2.] Talis, 43 Temple Row
Birmingham, UK

[3.] Department of Informatics & Telecommunications
University of Athens, Athens, Hellas (Greece)

[4.] Dept. of Computer Science and Technology,
University of Peloponnese, Tripolis, Hellas (Greece)

n.mdakim@lancaster.ac.uk , alan.dix@talis.com, vivi@mm.di.uoa.gr,
g.lepouras@uop.gr, nadeem.shabir@talis.com, costas@uop.gr

http://www.hcibook.com/alan/papers/websci2011-spreading-act/

ABSTRACT
Various forms of spreading activation has been used in a number of web systems, not least in the PageRank algorithm. In our own work we have been using this as a technique for managing context over small and large ontologies, and both our own work and that in LarKC suggests that spreading activation has the potential to aid in reasoning over web-scale data sets including the growing set of linked open data resources. Of particular importance is that spreading activation can be applied locally to a dynamic self-selecting working set of an (practically) unbound linked data collection, as well as globally to the entire collection. However, this potential does not come without problems, some concerning the nature of the algorithm on any large data set, and some more to do with the particular nature of linked open data.

Categories and Subject Descriptors
H.5.4 [**Information Systems**]: Hypertext/Hypermedia.
I.2.4 [**Computing Methodologies**]: Knowledge Representation Formalisms and Methods – *semantic networks*

General Terms
Algorithms, Human Factors

Keywords
spreading activation, reasoning, linked open data, context

1. SPREADING ACTIVATION AND WEB
Spreading activation was originally proposed as a model of the way associative reasoning proceeds in the human mind (Anderson, 1983). It has been used subsequently as a computational method in a number of areas (e.g. Crestani, 1997; Hasan, 2003; Liu, 2005), where some initial document/concept needs to be generalised to give a larger (weighted) number of related ones. We have been experimenting for a number of years with spreading activation over both small and large ontologies (Dix, 2006; Katifori, 2008, 2010; Dix, 2010), and are in particular interested in the way that it can be used for context inference over the, for practical purposes, unbounded datasets of the web of linked open data (LOD) (Bizer, 2009). As well as our own efforts, a spreading activation plug-in is available for LarKC (2011) as part of its data selection phases, and also PageRank (Page, 1998) is a form of spreading activation.

However, while spreading activation has many advantages and shows promise, there are various problems that arise: some concerning the nature of the algorithm on any large data set, and some more to do with the particular nature of linked open data. In this paper, we describe the range of types of spreading activation on ontologies with PageRank at one extreme (linear, global, input-independent) and our own work at another (non-linear, local, input-dependent); we also explore some of the problems and how these interact with particular types and choices in the algorithm using both real and simulated data.

WebSci '11, June 14-17, 2011, Koblenz, Germany.

2. WHAT IS SPREADING ACTIVATION
Spreading activation is similar to artificial neural networks, except instead of individual neurons, the level of analysis is larger; for example concepts, documents, or web pages. When applying spreading activation to web ontologies the unit is an entity and the connections relations between them.

The basic process is that each node receives some initial activation representing a stimulus (e.g. web page visited, or name in an email message). Each activated node then passes on some activation to connected nodes, usually with some weighting of connections determining how much gets spread to each. This is then iterated, either for some fixed time, or until the activations become stable.

Mathematically this looks like:

$$a_i' = \lambda u_i + \mu f(\Sigma w_{ij} \times a_j)$$

Here, a_i is the activation of node i, a_i' the activation at the next iteration, u_i is the initial activation, w_{ij} is the weight in the connection between node i and node j, and λ and μ are constants.

The function f is there because some form of threshold, or non-linear transformation (e.g. logistic) is often applied to the incoming activation for a node. A common value of w_{ij} is g/n_j where n_j is the number of outgoing connections from node j (fan out), and g is a 'gain' parameter.

PageRank is a special case of this where f is linear, g=1 and $\mu=1-\lambda$. Because of these values, PageRank is a *conservative* variant of spreading activation – the total amount of activation is always the same. Being conservative makes an activation network well behaved, but it is limiting and our own brains are not conservative: for example, small initial inputs can cause large effects (e.g. a sudden movement at the corner of your eye may cause momentary panic). A non-linear f is usually critical in non-conservative networks otherwise they often have 'run away' feedback with some nodes getting higher and higher activation at each iteration (in linear case, and Eigenvalue greater than 1).

Most choices of f have a finite asymptote creating a maximum value for the final activation of any node. However, the behaviour at near zero varies. Figure 1.i shows a classic sigmoid curve as found in many forms of neural network. As well as having a maximum value it attenuates small values of activation, making nodes with small activation even less activated, rather like a increasing the contrast in an image. In contrast, 1.ii is near linear for small values of input activation. Figure 1.iii shows the same function as in 1.ii but with addition of a sharp threshold cut-off, which means that if the level of input activation is low enough the node does not become activated at all. A similar effect can be obtained by shifting the curve to the right.

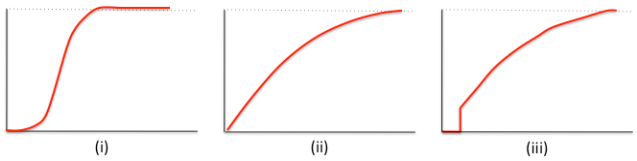


Figure 1. Different shapes of the transformation function f.

3. WEB SCALE REASONING

Some efforts in web-scale reasoning are focused on *global* application, using the whole (or significant portions) of the web (e.g. PageRank itself). More often some level of selection is necessary whereby only some subset of the total web resources are used to tackle a potential query or problem. This selection maybe deterministic based on the problem instance (Qasem, 2007), or probabilistic (Fensel, 2007). Usually this selection is an additional stage on an existing algorithm.

Spreading Activation has the advantage that it can also be used *locally*. The running of the algorithm naturally defines a working set, which is in principle unbounded (it could access anything), but in practice finite (only some is actually accessed for any specific query/problem). This is rather like a scholar in a large library, where all the books are available, but only those required for a particular purpose are accessed.

To see how this works, imagine we have a small number of entities in memory, some of which receive initial activation. We apply spreading activation for a few iterations, so that the different entities in memory receive differing amounts of activation. We then take all entities that are sufficiently highly activated and access suitable web data resources to fetch entities related to them. These new entities are then subject to further iterations of spreading activation, which might lead to yet more entities being fetched.

Note that the shape of the transformation function f is critical here. If there is no threshold (Fig 1.i, Fig 1.ii) then while there is an effective working set of highly activated entities, in addition virtually all entities (those that have any connectivity whatsoever) will receive some, albeit very tiny, activation.

It is necessary to limit this form of low-activation spreading on web-scale graphs. Because of this, whether or not the transformation function f used internally in memory has a threshold, we effectively add a threshold for externally fetched entities. Furthermore, even when f has a threshold internally, we usually apply a *higher* threshold for external fetching to reduce the working set and thus decrease network accesses and improve efficiency. In substantial experiments on the 20 million triple BBC data set (Dodds, 2009), we found that this additional threshold did not significantly affect the behaviour of the algorithm for normal ranges of parameters (Dix, 2010).

4. ISSUES AND PROBLEMS

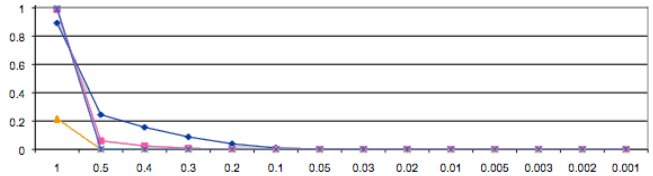
While spreading activation shows potential, and has proved valuable in several application areas, there are several problems that need to be tackled.

4.1 For Spreading Activation in General

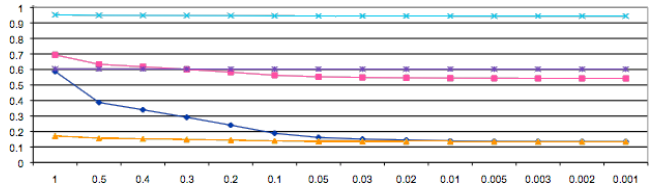
One key generic problem is that of 'greedy' nodes, portions of the graph of data, which become activated no matter what the input – a form of the runaway feedback possible in non-conservative nets. This is part of a general trade-off: on the one hand we expect the output of the algorithm to be determined by the initial set of activated nodes, whilst on the other we also expect the graph geometry to matter. In PageRank there is no (or to be precise uniform) initial activation, it is purely the graph structure that matters, but in other cases we need to work to ensure these intrinsic structures do not swamp the initial input.

In fact, this is a problem with neural-related algorithms in general and, in particular, led to the use of simulated 'dreaming' and 'unlearning' in neural networks (Crick, 1983; Hopfield, 1983). The idea is that you give the network random inputs and apply learning algorithms as normal expect instead of increasing weights one decreases them; thus nodes that get activated through random (meaningless) inputs are penalised.

There are various solutions to this problem. Some applications of



(i) transform non-linear near zero



(ii) transform linear near zero

Figure 2. Greedy node collection.

spreading activation use a limited, and small, number of iterations,

this preventing runaway feedback, others include limits on the direction or distance activation can spread.

The shape of the transformation function is again critical. Figure 2 shows the activation of a small number of 'problem' nodes in two simulation experiments. In each case a single node is initially activated and the final activation (after a large number of iterations) of the nodes measured. The axis at the bottom (logarithmic) shows in initial activation level, from 1.0 to 0.001, the lines are the final activation levels for particular nodes. In the first case, Fig 2.i, there is a non-linear transform function as in Fig 1.i; in this case as the initial activation decays so does the final activation of all nodes in the network. However, in Fig. 2.ii, the transform function is linear near zero as in Fig 1.ii; in this case the final activation of these nodes stay high even when the initial activation is vanishingly small. This does not mean that curves such as Fig 1.ii are wrong, but that other mechanisms may be need (in this case the weights used were non-conservative).

4.2 For Web-Scale SA

The greedy node problem was expected, but some problems only come to light in large-scale experiments. As noted in section 3, we performed experiments with different threshold values for loading in new entities for web-scale spreading activation and that for *normal ranges of parameters*, this performed robustly.

The means used to assess this was to measure final activation of entities with no (or very small) threshold and do the same with the target threshold and then plot the before and after activation. For most applications what matters is the more highly activated entities, so the crucial thing is that adding thresholds does not affect these more highly activated entities. In Figure 3 this is the top right-hand portion of the graph, which is nearly linear as desired.

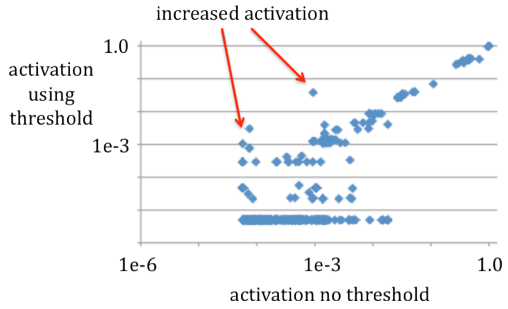


Figure 3. Unexpected increase in activation.

At the bottom of Figure 3 are entities that had low (<0.02) activation in the baseline (no threshold) condition, but had zero activation when the threshold is applied, because they are not part of the working set. This is again as expected. What is surprising are the entities indicated by the arrows, which had *higher activation* in the condition where the threshold was applied.

At first this seems counter-intuitive, swopping, for example, the function in Fig 1.ii for the one in Fig. 1.iii reduces the value of activation at each step and so surely would do so when applied iteratively? The reason turns out to be due to the calculation of weights based on the fan-out of nodes.

First of all take the condition with no threshold, so that we are effectively spreading activation over everything. Imagine a case where a highly activated node A is of class C. The class then becomes activated itself (we treat the *is_a* relationship in a similar way to other links). The class C then passes on some activation to other members of the class, but as there are many such members only a tiny amount of activation is passed on.

Now consider the same example, but with a threshold, high enough that the members of C do not get loaded purely because of C's activation. Suppose that due to the activation of some other part of the graph a node B is loaded, which is a member of C. Without additional meta-information, it appears that C has just 2 members (low-fan out) and therefore it passes on a substantial amount of its activation to both. In such a case B can have substantially higher activation than when there was no threshold.

In fact this condition only arose with particularly extreme values of other parameters and very high threshold, so while not a concern immediately, does suggest a situation that could potentially arise in other, more normal, situations.

4.3 Asymmetry and Diversity

In an RDF graph the links are directed and typed. This creates more choices when applying spreading activation techniques.

First for directedness, we effectively assume that there is an implicit inverse relationship and apply spreading in both directions. Due to the different fan-out in each direction the weight for the two directions will usually be different. This makes sense for context related applications; if person A is initially activated and person B is part of the same team, then we would expect B to have activation. The alternative, to only follow links in one direction, would only activate the team, but not the member of the team. However, following relationships in both directions has some drawbacks: it makes feedback loops more likely, and double counts when there is already an explicit inverse predicate. The latter could be detected, but the former so far seems an inevitable consequence of the desired behaviour.

The typing of the links can simply be ignored and all the links treated equally. However, this does not lead to sensible behaviour. For example, the UK has 60 million citizens and one Queen. If we ignore the typing of links then if the UK is initially activated the Queen would get equally activated as every other citizen. Another option is to first divide the available activation by the number of types, and then to divide the activation within each type. In the above scenario the Queen would get 1/2 of the spread activation and each other citizen 1/(120,000,000). In other examples, it appears this biases a little too much towards the minority types and we are still experimenting with different alternatives.

4.4 Linked Data

The implicit working set of spreading activation means we can safely apply it to web-scale graphs and in particular the graph of the Linkng Open Data cloud (Bizer, 2009). Linked data guarantees that information about any entity can be found by dereferencing it. However, the web of linked open data, is 'biased' towards subject to object relationships – dereferencing a URI is more like doing a SPARQL DESCRIBE, that is the triples for which it is the subject, and does not usually yield the triples for which it is the object. Because many repositories can hold references to a single URI in general it can be hard to know when one has a complete view of the object.

This bias interacts in different ways with spreading activation depending on the variant of the algorithm (e.g. one-way vs. two-way spreading through links). The problem noted in 4.3 arose when using full SPARQL access, but would be likely to be worse if using Linked Data dereferencing. We do not have a solid

answer to this at present, but assume we will need to include heuristics that effectively assume that knowledge of back links is partial, and correspondingly reducing their weight.

Also, while the principles of linked open data (LOD) suggest that linkage should be through dereferenceable URIs, in practice one often has to look at values and some level of instance matching. In early Protégé implementations we in fact had activation on literals as well as entities. This was removed, but it may be that it should be reinstated; this would mirror real life: when we think of a person we may well be reminded of someone completely different who has a similar name. However, the problems of incomplete information would be if anything greater for literals than for URI resources.

Some of the services appearing for back links and linked data search may begin to ease some of these issues, and mean that the LOD cloud can be treated more like a single graph that can be queried about any resource or literal.

On the other hand spreading activation can potentially be used to aid search by selecting appropriate search services, even if these are not crawled by search engines (Dix, 2011). Whilst existing search engine personalise search based on previous search history, more dynamically calculated context could allow web search and other applications to be tuned to the particular moment.

5. DATA FORENSICS AND SIMULATION

In section 4.2, we needed to make sense of an unexpected result. In an emergent system this may be difficult. For example, following its televised win in *Jeopardy*, the developers of IBM Watson expect that it will take a year to work out why it answered certain questions wrong. New visual analytics tools are required for this kind of task.

As one way to investigate the issues we are combining in-the-wild applications to real large-scale data with simulations. The real datasets allow us to find new issues (such as the fan-out related issue in section 4.2), whilst the simulations allow us to then study these issues in a more constrained environment. This often involves having a small realistic core (taken from personal ontologies, for which we have known expected behaviour for given initial activations), and then adding an artificial structured extension. The extension has a large number of nodes, but organised in a systematic and well understood topology, hence aiding empirical and formal analysis.

Figure 4 shows two forms used to investigate greedy nodes and to generate the data shown in Figure 2. In real simulation the 'lollipop' of additional nodes may have many thousands of nodes around the central hub. Common sense would suggest that this large, but loosely connected group should have little effect on the relevance of elements on the core, the simulations allow us to stress test such hypotheses on different algorithm variants.

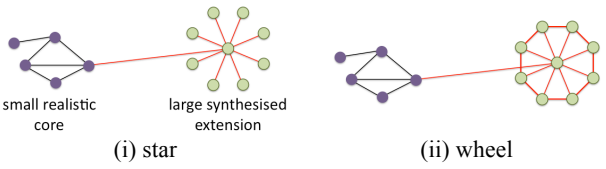


Figure 4. Simulated graphs.

6. REFERENCES

[1] Anderson, J. 1983. A Spreading activation theory of memory, *Journal of Verbal Learning and Verbal Behaviour.* 22, 261–295.

[2] Bizer, C., Heath, T. and Berners-Lee, T. 2009. Linked Data — The Story So Far. International. *Journal on Semantic Web and Information Systems,* 2009.

[3] Crestani, F. 1997. Retrieving documents by constrained spreading activation on automatically constructed hypertexts, in *Proc. of EU- FIT 97- Fifth International Congress on Intelligent Techniques and Soft Computing*, pp. 1210–1214.

[4] Crick, F. and Mitchison, G. 1983. The function of dream sleep, *Nature*, 304, 111–114.

[5] Dix, A. 2006. The brain and the web – intelligent interactions from the desktop to the world. keynote at IHC 2006, Natal Brazil. http://www.hcibook.com/alan/papers/brazil2006/

[6] Dix, A., Katifori, A., Lepouras, G., Vassilakis, C. and Shabir, 2010. Spreading Activation Over Ontology-Based Resources: From Personal Context To Web Scale Reasoning. *Int. Jnl of Semantic Computing*,. 4(1) pp.59-102.

[7] Dix, A. 2011. Context and Action in Search Interfaces, New Trends In Search Computing, LNCS. 6585, 39–49.

[8] Dodds, L. 2009. *Understanding the Big BBC Graph*, 11th June 2009. http://blogs.talis.com/n2/archives/569

[9] Fensel, D. and van Harmelen, F. 2007. Unifying reasoning and search to web scale, *IEEE Internet Comp* 11, 2, 94–96

[10] Hasan, M. 2003. A spreading activation framework for ontology-enhanced adaptive information access within organisations, in *Proc. of the Spring Symposium on Agent Mediated Knowledge Management (AMKM 2003)*.

[11] Hopfield, J. Feinstein, D. and Palmer, R. 'Unlearning' has a stabilizing effect in collective memories, *Nature*, 304, 158–159.

[12] Katifori, A., Vassilakis, C. and Dix, A. 2008. Using spreading activation through ontologies to support personal information management, in *Proc. of Common Sense Knowledge and Goal-Oriented Interfaces,* (at IUI 2008).

[13] Katifori, A., Vassilakis, C. and Dix, A. 2010. Ontologies and the Brain: Using Spreading Activation through Ontologies to Support Personal Interaction. *Cognitive Systems Research*, 11 (2010) 25–41.

[14] LarKC (2011). *The Large Knowledge Collider*, accessed 25/2/2011. http://www.larkc.eu/

[15] Liu, W., Weichselbraun, A., Scharl, A. and Chang, E. 2005. Semi-automatic ontology extension using spreading activation, *Jal of Universal Know. Mngmnt,* 0, 1, 50–58.

[16] Page, L., Brin, S., Motwani, R. and Winograd, T. 1998. *The pagerank citation ranking: Bringing order to the web*. Tech. Report, Stanford Digital Library Technologies Project.

[17] Qasem, A., Dimitrov, D. and Heflin, J. 2007. Effecient selection and integration of data sources for answering semantic web queries, in *Proc. of the First Int. Workshop "New Forms of Reasoning for the Semantic Web: Scalable, Tolerant and Dynamic,"* Co-Loc. with ISWC 2007 and ASWC 2007, Vol. 291. http://CEUR-WS.org/Vol-291