Hebbian Algorithms for a Digital Library Recommendation System

Francis Heylighen
CLEA, Free University of Brussels
http://pcp.vub.ac.be/HEYL.html

Johan Bollen
Computer Science, Old Dominion University
http://www.cs.odu.edu/~jbollen

Abstract

This paper proposes a set of algorithms to extract metadata about the documents in a digital library from the way these documents are used. Inspired by the learning of connections in the brain, the system assumes that documents develop stronger associations as they are more frequently co-activated. Co-activation corresponds to consultation by the same user, and decreases exponentially with the time interval between consultations. The strength of activation is proportional to the user's interest for the document, either evaluated explicitly, or inferred implicitly from user actions or the duration of the consultation. Co-activation values are added, producing a matrix of associations. This matrix can be used to recommend the documents that are most strongly related to a given document, most relevant to the user's implicit interest profile, or most interesting to users overall. Moreover, it allows the calculation of document similarity values, which in turn can be used to cluster similar documents. The data needed to feed such a recommendation system are readily extracted from the usage logs of document servers, and can be processed either in a centralized or a distributed manner.

1. Introduction

Compared to traditional libraries, the World-Wide Web has some spectacular advantages: the range of documents it proposes is much wider, they are easier to consult, they are available always and everywhere, and their electronic format makes it easy to search for specific phrases or keywords. On the other hand, the web's disadvantages are obvious too: an almost total lack of organization of the material, and virtually no selection for quality or trustworthiness.

Distributed digital libraries hold the promise of combining the benefits of both the web and traditional paper libraries. Their electronic documents would be available to everybody via the Internet, yet a staff of editors, "cybrarians", or information scientists would guarantee that all documents fulfil minimum quality requirements, and that they are organized according to a coherent system of categories, keywords or more generally *meta-data*, so that documents on any specific subject can be transparently retrieved.

While quality control can in principle still rely on the traditional methods of peer-refereeing and evaluation by experts that work relatively well with paper documents, retrieval on the basis of metadata has some intrinsic shortcomings, which can only get worse as the number of documents in the library increases.

The first set of problems derives from the fact that metadata can never completely capture the subject or meaning of a document: traditional metadata merely provide a coarse and rigid categorization, which can never specify all potentially relevant characteristics [7]. Free-form keywords provide perhaps the most flexible kind of traditional metadata. Yet, keywords suffer from the problems of *synonyms* (the user may enter a keyword similar in meaning but different in form to the one by which documents are classified, and therefore fail to locate a relevant document), and *homonyms* (the users may enter a keyword similar in form but different in meaning, and therefore receive an inappropriate document). Moreover, when the subject is new or as yet unclear to the user, the user may not be able to formulate any relevant keyword. Combined with huge collections and limited selection this leads to queries showering the user with material of little relevance, in which perhaps a few nuggets of true value are hidden.

A second set of problems derive from the fact that assigning good (if by definition incomplete) metadata to documents requires a great effort and a special expertise: only people who know the domain well, and have studied the document well can determine the appropriate keywords or categories. This problem is to some extent mitigated for electronic texts, since IR algorithms can extract the most distinctive words from the text to be used as keywords, as is done by search engines on the web. However, this method is useless for non-text documents, such as pictures, movies or sounds.

The present paper proposes a general approach that seems able to tackle these problems. The idea is to extract metadata not from the content of the documents but from the pattern of document usage, assuming that users have an intuitive grasp of what a document is about and how valuable it is, and that this intuition guides their actions. As we will show in the next section, such metadata do not rely on fixed categories or keywords, but on the variable associations that exist between documents in their users' mind. Since interesting documents will anyway be consulted by users and this activity is stored in usage logs, analysing usage patterns allows us to collect metadata without requiring additional effort from either users or librarians.

It must be noted that recently a variety of techniques have been developed to mine knowledge from data about document usage (see e.g. [15,16]). However, these techniques constitute a rather *ad hoc* collection, with little integration or motivation and no underlying theory to explain how users navigate through a web of linked documents [17]. Moreover, most of these techniques are based on the clustering of navigation paths, user profiles, or documents into a static set of discrete categories, thus suffering from all the shortcomings of rigid categorization.

Our approach, on the other hand, which builds on our experimental results with a website that adapts to or learns from the way it is used [1, 10, 17, 19], tries to discover the finely-graded, continuous associations between documents that trace the users' constantly changing focus of interest while browsing a document collection. Clustering in categories is one possible application of our approach, but should be seen only as an "afterthought" that is not fundamental, since different clusterings can be made in different contexts. Moreover, our approach, whose first development dates back to 1995 [19], is characterized by its unique, coherent paradigm, based on the analogy between the dynamic organization of a document network and the organization of the brain [18], where concepts or neurons are connected through variable strength associations or synapses, whose strength evolves according to the rule of Hebb [6].

2. Associative networks that learn

2.1. Association matrices

Rather than analysing a document's content or components, an alternative way to define meaning is through bootstrapping: a document's meaning consists of the whole of associations it has with other words or documents [8, 9]. An *association* between two documents d_i and d_j can be defined as a measure of the degree of relatedness between d_i and d_j, or the degree of "expectancy" of d_j, given d_i.

An *associative network* is a weighted, directed graph, whose nodes d_i represent documents, and whose weights represent the association between nodes. It can be represented as a matrix whose components a_{ij} [0, 1] correspond to the connection weights between nodes d_i and d_j. [9]

This matrix is generally sparse, since most associations will have value 0, which means that encountering node *i* does not in any way prepare the mind to encounter *j*. A maximum weight of 1 means that given d_i, everything is already known about d_j; d_j does not provide any additional information that isn't yet contained in d_i. This is an extreme case which is likely to be found only if document d_j is a copy, excerpt or summary of d_i.

In such an associative network, every node or document can now be represented by a vector:

$$\mathbf{d_i} = (a_{i1}, a_{i2}, ..., a_{in})$$

The assumption underlying the bootstrapping model is that this vector captures the essential meaning of the document relative to other documents. Therefore, the contents of this vector can be interpreted as representing the associative metadata about document *i*. If keywords and categories are seen as a discrete, symbolic representation of a document's meaning, then an association vector provides a continuous, subsymbolic representation, of the kind used in distributed or connectionist models of cognition.

2.2. A Hebbian learning rule

Associative networks are inspired by the functioning of the brain, where on the higher, abstract level concepts are connected by associations, and on the underlying, physical level neurons are connected by variable strength synapses. An association a_{ij} represents the degree or probability of activation of neuron/concept *j* following the activation of *i*.

In the brain, associations are learned through the rule of Hebb [6, 1]: concepts that are activated simultaneously (*co-activation*) become more strongly associated. This strengthening is proportional to the degree of activation A(*i*) and A(*j*) of each concept. Since we can assume that activation decays with the time that has passed since the initial stimulus that created the activation, the degree of co-activation will decrease exponentially with the time interval (t_j - t_i) between the activation of *i* and the subsequent activation of *j*.

This brings us to the following formula for the bonus or association strength added by a particular episode of co-activation, where *d* is a constant decay factor:

$$\mathrm{A}(i, t_i).\mathrm{A}(j, t_j).\exp\,(-\,d.(t_j - t_i)).$$

The total association strength is then merely the (possibly normalized) sum over all episodes of co-activation of the strength bonuses for each co-activation.

Note that a bonus can be negative if we allow for negative activation values. This means that association strength can decrease if a positive activation of *i* is followed by a negative activation of *j*, or vice-versa.

Note also that the different co-activations can be weighted in the calculation of the total association strength so that more recent co-activations make a larger contribution. This is useful in circumstances where the pattern of usage regularly changes depending on new developments or social or cultural changes in the group of users. Again, our brain paradigm would suggest an exponential decay factor *d*' that would lessen the impact of older contributions depending on the time that has passed.

An efficient way to compute the overal value would be to store only the total association strength *S* together with a time stamp (t_1) of when that strength was last updated. Whenever an updated association strength $S(t_2)$ is needed, this is calculated by multiplying the previously stored value with the decay factor: $S(t_2) = S(t_1)$. exp $(-d'(t_2–t_1))$. A new co-activation at t_2 is then simply added, without decay factor, to this reduced sum of all previous activations, and this new value is stored together with the new time stamp t_2. Each next time t_i that an update is to be made, the same procedure is applied recursively. In that way, the value at any moment will reflect the history of usage, so that older contributions weigh in with a gradually lower contribution, but without need for the system to store the full sequence of update episodes.

3. Collecting data from usage

3.1. Document activations

In the context of libraries or document collections we can say that a document is activated each time it is being consulted (opened, downloaded, borrowed or bought) by a user. In the simplest model, every consultation event produces a fixed activation unit of, say, 1. In a more sophisticated model, we can assume that more interesting documents are used more intensively than others, and therefore activation values can vary. There are basically two ways to evaluate activation strength: *explicitly* or *implicitly*.

Explicit evaluation would require the users to indicate how interesting or relevant the document they are consulting is. This could be done e.g. with a five point scale going from “useless” to “just what I needed”. This can be recomputed to an activation value varying between 0 (or -1) and 1.

The disadvantage of explicit evaluation is that it demands additional effort from the user, which many users might not be inclined to perform, especially if they are browsing through long lists of documents. Explicit evaluation is likely to be done in practice only for documents that somehow stand out, because they are particularly interesting or disappointing.

3.2. Implicit evaluation

Implicit evaluation tries to estimate the degree of relevance of a document indirectly from the way the user acts on the document. Different actions such as browsing, saving, bookmarking, printing, or buying indicate different degrees of interest [12]. The most straightforward way to derive activation values from these actions would be to correlate them with explicit evaluations. E.g., a large sample of user data might indicate that documents that are bookmarked get an average evaluation of 3.7 on a 5 point scale, while documents that are printed get a 4.1 evaluation.

Implicit activation values can be derived even more simply from the time spent consulting the documents. Several studies [5,12] have found a strong correlation between duration of consultation and explicit ratings of value (while—surprisingly—there was no correlation between duration and size of the document). It must be noted, though, that the relation between duration and value will not be strictly proportional or linear: there will be less difference in value between documents consulted for 90 minutes, respectively 95 minutes, than between documents consulted for 5, respectively 10 minutes.

A plausible relation might take the form of a sigmoid or logistic function, which initially increases very slowly to absorb noise fluctuations due to differences in connection speed with the document server, then increases almost linearly, and finally slows down gradually in order to reach a plateau where further increases in duration produce virtually no increases in activation. Like in the case of user actions, the specific shape and parameters of the function can be derived by determining the best match with explicit evaluation data.

3.3. Co-activation of documents

Now that we know how to get activation values, we need to determine co-activation. The basic principle is that documents are co-activated if they are consulted by the *same* user, since that user can be assumed to be looking for mutually relevant documents rather than a random assortment of unrelated documents. The exponential decay factor expresses the fact that the more time passes, the more likely it is that the user has directed his/her attention elsewhere and has started exploring a different subject. Still, the fact that people generally have a stable personality and occupation would imply that two documents consulted by the same user, even with a ten year interval, are more likely to be related than two documents consulted by randomly chosen users. Therefore, the exponential decay factor might be complemented by a constant term *b* so that the co-activation formula becomes:

$$\mathrm{A}(i, t_i).\mathrm{A}(j, t_j).(a.\exp(-d.(t_j - t_i)) + b)$$

The situation $b = 0$ would bring us back to the previous, purely Hebbian case.

On the other hand, $a = 0$ would bring us to the case of *collaborative filtering* [3,14,10]. This method is used e.g. by Internet bookshops, such as Amazon.com, which recommend books on the basis that they have been bought by the same users, without taking into account the time interval between the different purchases. Again, the values of the parameters *a*, *b* and *d* can be determined by minimizing the difference between the recommendations derived from the association matrix and the explicit evaluations by the users.

4. Applications

Given the co-activation values derived above, we can compute a matrix of associations between documents by adding together all the collected values for the different users, documents and moments in time. This matrix can be used to guide further users in several different ways:

4.1. Listing related documents

The most straightforward application is to append to each document *i* a list of the documents that are most strongly associated with it (i.e. that form the largest components in the document vector $\mathbf{d_i}$). These are the documents that not only were frequently consulted by the same users, but consulted within a relatively short time interval, and (implicitly or explicitly) evaluated to be most interesting. In that way, a user who discovers one document that looks particularly relevant will immediately get to know all the documents that are most likely to be relevant as well. To most efficiently guide the user, the documents can be listed in the order of their degree of association, the strongest associations first, perhaps with a graphical indication of that degree.

These links to further documents function like shortcuts for the otherwise extended exploration sequences that help users to find other related documents. From these related documents, users will be offered shortcuts to further related documents. This may lead the system to create even shorter shortcuts, from the first document directly to the third or fourth in the sequence ("transitivity"). Thus, the use of already learned connections will be assimilated further into the learning system to create even more direct connections, creating a positive feedback loop which in our first experiments was shown to spectacularly enhance performance [1, 19].

4.2. Personalized recommendations

The recommendation of mutually relevant items can be taken a step further. Users browsing through a library database generally won't settle on a single, most interesting document, but find several documents d_i that are relevant in different ways and to varying degrees A(*i*), while none of them actually captures the main focus of interest. This determines an "interest profile" which can be represented by the activation vector (A(1), A(2), ...) (this activation vector can also take into account the decrease of interest of the user with time passing by incorporating an exponential decay factor, see [9]).

This vector can now be multiplied with the association matrix to produce a new "recommendation" vector (r_1, r_2, ...) with:

$$r_j = \sum_i a_{ij} A(i)$$

This recommendation adds together the contributions from the previously visited documents in proportion to their relevancy. This procedure can be repeated, multiplying the recommendation vector iteratively with the matrix to get *indirectly* associated documents, that perhaps have never been consulted by the same user, but that are associated to other documents that have been co-consulted.

This implements the general retrieval technique of *recurrent spreading activation* [1, 2, 17]: the initial activation represented by the interest vector is allowed to spread iteratively through the associative network, so as to activate all documents that have strong direct or indirect associations with one or more of the initially selected documents. Note that the most well-known applications of spreading activation in information retrieval [13], which produce rather disappointing results, are not recurrent: they only allow activation to spread for one or two steps, in one direction only (i.e. without the possibility of activation flowing back to previously activated nodes, which would allow the non-linear accumulation of activation in the most interesting regions). There exist many different variations on this spreading activation algorithm, depending on parameters such as number of iterations, relative contributions of each iteration phase, etc. Again, fine-tuning of the result may be obtained by repeated experiments where recommendations are compared with explicit evaluations.

The advantage of spreading activation is that the user may have found only poor examples of relevant documents, but still receive good recommendations through indirect association. The only requirement is that the user be able to distinguish better from worse options. Thus, with each recommended document that the user checks, the interest profile and therefore the further recommendations will be refined, since the system will now know in how far this additionally consulted document is really relevant to the task.

4.3. Determining overall interestingness

Recurrent spreading activation has more benefits than fine-tuned, individual recommendations. Associations are in general asymmetric ($a_{ij} \neq a_{ji}$) since they reflect the particular sequence in which a user has moved from one document to another one. Since users will typically move from less relevant to more relevant documents, the most interesting documents will tend to reside at the end of the association sequence. This means that as activation spreads further it will encounter documents that are more and more interesting generally, albeit less directly associated with the initial preference profile.

If the matrix multiplication is iterated indefinitely, the output vector will converge to the largest *eigenvector* of the matrix. This eigenvector, or "attractor" of the spreading activation dynamics, represents the equilibrium distribution of activation. The degree of activation of each component of that vector can be interpreted as the global "attractiveness", "interestingness" or "authority" of that component, independent of the initial query.

Such "authority" is equivalent to the PageRank measure that lies behind the surprising effectiveness of the Google search engine [4], although PageRank starts from a binary connection matrix (link, no link) rather than a continuous association matrix, and thus is likely to produce less fine-grained results. As demonstrated by Google, such overall ranking is very useful when ordering query results before the user has had the time to express preferences for one document over another.

Generalizing from this observation, we may argue that the number of iterations is an important parameter that would allow us to control the generality of the recommendation: the larger that number, the wider the public for which the most highly activated documents will be relevant, but the less direct their relation to the initial preference profile.

4.4. Clustering documents

By multiplying the (asymmetric) matrix with its transpose we can create a new, symmetric matrix:

$$s_{ij} = \sum_k a_{ik} a_{kj} \quad .$$

s_{ij} represents the degree of similarity between the components *i* and *j*. Indeed, s_{ij} is the dot product between the vectors $\mathbf{d_i}$ and $\mathbf{d_j}$ that represent all the associations that the documents *i* and *j* have with other documents (see 2.1). The more the association vectors overlap, and thus the more *i* and *j* resemble each other in the way they relate to other documents, the larger the dot product, and therefore s_{ij}. This similarity measure can now be used as an input to a variety of *clustering* algorithms that put documents together in classes depending on how similar/dissimilar they are from each other.

One example of such a clustering can be found in the HITS algorithm developed by Kleinberg [11], that clusters web pages starting from the product of a connection matrix with its transpose, by finding the different orthogonal eigenvectors of that matrix and by considering components that load strongly on a particular eigenvector as members of the same cluster. This allowed Kleinberg to e.g. distinguish "pro life" from "pro choice" pages on abortion, or pages on the animal "jaguar" from pages on the car and the sports team with the same name, thus tackling the problem of homonyms.

More generally, a clustering algorithm should allow us to automatically create categories of documents, even when these categories haven't been formally recognized yet, thus catching emerging new domains from the very beginning. The categories can be labelled by extracting the keywords that appear most frequently in that category relative to the overall collection. The PageRank or HITS algorithms can moreover be used to list the documents most authoritative for each category, which are likely to be classic papers, general reviews or introductory texts about the subject.

4.5 System evaluation and optimization

An essential step in the development of the system that we envisage is an evaluation of its effectiveness. This can be easily built into the system itself. If the system allows for explicit evaluation of recommended documents by users, then the average score given by users can be compared with the strength of the recommendation as calculated by the system. The correlation between the two scores can be taken as a measure of the system's effectiveness. This applies as well to document-centered recommendations, recommendations based on a user-profile as to estimates of overall interestingness. (It would seem that an evaluation of the quality of clusters will have to be made by domain experts rather than by everyday users).

To make sure that the recommendations are doing more than just stating the obvious, system recommendations can be compared with recommendations collected from independent sources, such as randomly selected documents, author-provided references, or simply lists of the most frequently used documents. In order to provide an unbiased test, system-generated recommendations can be randomly interspersed with recommendations from these other sources, in a way unknown to the user. The system will have unambiguously proven its worth if its recommendations get a systematically higher score than these other possible recommendations.

Such evaluation can be used to continuously optimize and fine-tune the system. It suffices to consider the correlation between system-calculated strength and average user evaluation as a function to be maximized, and then vary the different parameters used in the algorithms (e.g. strength of exponential decay, number of iterations in spreading activation, ...) so as to achieve the largest possible value for the correlation. In that way, the system will not only learn better relevancy judgments from its users, but moreover learn how to improve its own learning functions, i.e. it will undergo *metalearning* towards ever greater effectiveness.

Moreover, metalearning will allow the system to adapt to specific contexts: different types of document collections (e.g. songs vs. lecture notes) will be used in different ways, and thus require different parameters for the learning and recommendation algorithms (e.g. the duration of consultation is likely to be lower for pictures than for technical documents or movies, and information search in well-structured databases is likely to be more focused than in more "associative" collections of artistic photos, and thus require less iterations during spreading activation).

5. Implementation

The data necessary for the Hebbian algorithm that we outlined are easy to collect. The document server will normally maintain a log of all consultations, including the identity of the user, the documents requested, and the precise date and time at which each document was requested [2]. This is sufficient to calculate the activation of each document on the basis of the time spent between requesting a document and requesting the next document, which indicates the time spent reading the document and thus, as we have seen, provides an implicit measure of interest. The exponential decay factor can be calculated from the time interval between requests for the two documents (which may be several steps away from each other in the request sequence) between which co-activation needs to be calculated, independently of any requests in between.

It must be noted that server logs tend to contain a lot of noise, such as consultations made by webrobots rather than true users, users whose IP address changes during the session, different users with the same IP address, sessions interrupted e.g. because the user went to drink a cup of coffee, consultations made through backtracking or bookmarks rather than following sequences of links, etc. Various techniques have been developed to preprocess such log data so as to extract only the meaningful navigation paths (see e.g. [2, 15, 16, 17]). Obviously none of these will ever be perfect. Yet, we don't expect the remaining errors to have a great impact on the results, because our general algorithms appear quite robust, based on principles of self-organization that are able to extract strong patterns from a noisy background [10].

Moreover, the effect of any amount of noise can be attenuated through the law of large numbers: if a sufficiently large number of contributions is collected, summation will drown out any random deviations from the underlying signal [10]. Because any log file, which for a typical active webserver contains millions of lines for a few weeks worth of use, can be used as input, it seems that in most cases there will be sufficient data to kickstart the system and quickly produce a usable list of recommendations.

If we wish to use other data than duration to estimate user satisfaction, we will need to establish a protocol that signals specific user activities, such as printing, saving or bookmarking, to the server collecting the data. This is most straightforward for explicit evaluations, where a user can click on an evaluation bar, and thus pass on the coordinates of the click to the server. Another approach is to have a Java applet loaded into the user's browser when the server is first contacted, which registers the activity within the browser and sends this information back to the server [15].

When there is a single, centralized server for all documents, this basically completes the information gathering, since the association matrix can now be directly extracted from the log of that server, while the related documents, "authority" measures and clustering can be computed off-line using the matrix, after which the results are fed back into the document system, e.g. in the form of a recommendation list at the bottom of each document summary, together with a taxonomy of subjects and a list of the most important pages on the entry page.

Individual recommendations based on spreading activation are somewhat more involved as they require the maintenance of a constantly updated interest vector for each user, which must be multiplied with the matrix to provide tailored recommendations in real time. One method is to keep a "cookie" on the user's browsing application that keeps track of the user's sequence of activities. When desired, this cookie can then be transferred to a central server to be used as input for a spreading activation algorithm.

With a truly distributed library system, running on a variety of independent servers, the main additional step is the establishment of a protocol for the exchange of data about user activities between the different machines. Each time a user moves to a new server, previously consulted servers should receive a trace of the user's activities on that server. Thus, they can update—or create—associations from the documents kept locally to the documents kept on other servers.

With many documents spread over many different servers, the danger is that the information to be kept on any one server would explode. This can be controlled by limiting the trace's extent in time or in number of consultations, so that e.g. information is no longer sent to servers whose documents were consulted more than *x* days or requests ago. Moreover, each server could locally decide to maintain not more than *y* associations for each document. This can be done by periodically removing from memory the weakest associations, or—if the system moreover keeps track of the time each bonus was added—the associations that received their last bonus the longest time ago.

The disadvantage of such a distributed implementation is that there isn't any single place where the complete association matrix is stored. In principle, the association matrix can be reconstructed from the association data that are kept locally on each server, but this will require complicated distributed protocols if global computations must be performed—such as calculating PageRanks or global clusterings. Local recommendations, such as proposing documents related to a given document or spreading activation with few iterations, should be produced easily.

An alternative for a fully distributed implementation would be a central server that maintains and manipulates the overall association matrix. However, since the complete matrix, while being sparse, will contain a huge amount of data, an in-depth application will require extensive computing power together with sophisticated algorithms for sparse matrix manipulation. Still, similar matrices, albeit probably less fine-grained, are successfully being used by databases such as Google or the Alexa recommendation service on the web, demonstrating the feasibility of the project.

6. Conclusion

The present paper has sketched a general family of algorithms to extract meta-data about documents from the way these documents are consulted by users. Implementing such a system in a digital library would automatize much of the hard work that would otherwise need to be performed by highly trained information scientists.

However, the results of this system are envisaged to *complement* or support traditional methods rather than fully replace them. The reason is that the proposed system focuses on otherwise difficult to formalize properties of documents, namely the subjective associations that exist in the mind of the users between their different subjects and contents. The advantage is that these associations allow us to build a system that emulates human *intuition*, so that it can anticipate the desires of its users and provide them with the information they would find most interesting, even when these users cannot explicitly formulate what they are looking for. This is particularly useful for multimedia documents, which do not contain any searchable keywords, and for queries that are as yet ill-defined.

The disadvantage of associative networks is that they are intrinsically fuzzy, ambiguous, and constantly shifting [9]. However, the clustering approach that we sketched might help us to extract discrete categories, which can be automatically labelled with keywords, although here it is likely that the system would still need the assistance of a human operator in order to build a coherent taxonomy.

Another advantage is that the system is designed from the start to *learn*, so that its recommendations become better the more it is used. This applies at the collective level, where the association matrix becomes more precise as more usage data are collected, but also at the individual level, where every (explicit or implicit) evaluation of a document made by a user helps the system to produce a more individually tailored recommendation, and even at the metalevel, where the system adapts its own learning functions to the circumstances.

The biggest unresolved issue until now is the implementation of such a system at the level of a distributed library system. (Smaller scale implementations have already been made [1, 2, 8, 10, 17], or are under development.) While a single server centralizing and processing all incoming and outgoing data seems straightforward, albeit computationally intensive, the more interesting challenge will be to distribute both the database and the processing over a peer-to-peer document server network.

7. References

[1] Bollen, Johan and Heylighen, Francis (1998) A system to restructure hypertext networks into valid user models, *New Review of HyperMedia and Multimedia,* 189-213.

[2] Bollen, Johan, Herbert Van de Sompel, and Luis M. Rocha (1999) Mining associative relations from website logs and their application to context-dependent retrieval using spreading activation, in *Workshop on Organizing Web Space* (WOWS), ACM Digital Libraries 99, August 1999, Berkeley, California.

[3] Breese J.S., Heckerman D. and Kadie C. (1998), Empirical Analysis of Predictive Algorithms for Collaborative Filtering, *Proceedings 14th Conference on Uncertainty in Artificial Intelligence*, Madison WI: Morgan Kauffman.

[4] Brin S. and L. Page (1998): The Anatomy of a Large-Scale Hypertextual Web Search Engine, *Proceedings of the 7th International World Wide Web Conference*, April 1998.

[5] Claypool, Mark, Phong Le, Makoto Waseda and David Brown (2001): Implicit Interest Indicators, In *Proc. ACM Intelligent User Interfaces Conference (IUI),* Santa Fe, New Mexico, USA

[6] Hebb, D. O. 1967 *The organisation of behavior: a neuropsychological theory.* Science Editions, New York.

[7] Heylighen F. (1991): Design of a Hypermedia Interface Translating between Associative and Formal Representations, *International Journal of Man-Machine Studies* 35, p. 491-515.

[8] Heylighen F. (2001): Bootstrapping knowledge representations: from entailment meshes via semantic nets to learning webs, *Kybernetes* 30 (5/6), p. 691-722.

[9] Heylighen F. (2001): Mining Associative Meanings from the Web: from word disambiguation to the global brain, in: *Proceedings of the International Colloquium: Trends in Special Language and Language Technology,* R. Temmerman and M. Lutjeharms (eds.) (Standaard Editions, Antwerpen), p. 15-44.

[10] Heylighen, Francis (1999) Collective Intelligence and its Implementation on the Web: algorithms to develop a collective

mental map, *Computational and Mathematical Theory of Organizations* 5(3), 253-280.

[11] Kleinberg J. (1998): Authoritative sources in a hyperlinked environment, *Proc. 9th ACM-SIAM Symposium on Discrete Algorithms*.

[12] Nichols, D.M. (1998) Implicit Rating and Filtering, *Proc. Fifth DELOS Workshop on Filtering and Collaborative Filtering*, Budapest, Hungary, 10-12 November 1997, ERCIM, 31-36.

[13] Salton G. and Buckley C. (1988). On the Use of Spreading Activation Methods in Automatic Information Retrieval, *Proc. 11th Ann. Int. ACM SIGIR Conf. on RandD in Information Retrieval* (ACM), 147-160.

[14] Shardanand U. and Maes (1995), Social information filtering: Algorithms for automating 'word of mouth', *Proceedings of CHI'95 -- Human Factors in Computing Systems*, 210-217.

[15] Cooley, R. *Web Usage Mining: Discovery and Application of Interesting Patterns from Web Data*. Ph.D. Thesis, University of Minnesota, May 2000.

[16] Shahabi, C., Zarkesh, A.M., Adibi, J., and Shah, V. Knowledge Discovery from User's Web-page Navigation, in Proc. *7th IEEE Intl. Conf. On Research Issues in Data Engineering* (1997), 20-29.

[17] Bollen, Johan: *A Cognitive Model of Adaptive Web Design and Navigation: A Shared Knowledge Perspective*. PhD Thesis, Vrije Universiteit Brussel, 2001.

[18] Heylighen F. & Bollen J. (1996) "The World-Wide Web as a Super-Brain: from metaphor to model", in: *Cybernetics and Systems '96* R. Trappl (ed.), (Austrian Society for Cybernetics).p. 917-922.

[19] Bollen J. & Heylighen F. (1996) "Algorithms for the Self-organisation of Distributed, Multi-user Networks. Possible application for the future World Wide Web", in: *Cybernetics and Systems '96* R. Trappl (ed.), (Austrian Society for Cybernetics), p. 911-916.