ELSEVIER

Applied Soft Computing 2 (2002) 11–23

Applied Soft Computing

www.elsevier.com/locate/asoc

Fast adapting value estimation-based hybrid architecture for searching the world-wide web

István Kókai, András Lőrincz*

Department of Information Systems, Eötvös Loránd University, Pázmány Péter sétány 1/D, H-1117 Budapest, Hungary

Accepted 21 May 2002

Abstract

The slogan that *information is power* has undergone a slight change. Today, *information updating* is in the focus of interest. The largest source of information is the world-wide web. Fast search methods are in need for this enormous source. In this paper a hybrid architecture that combines *soft* support vector classification and reinforcement learning for value estimation is introduced for the evaluation of a link (a document) and its neighboring links (or documents), called the *context* of a document. The method is motivated by (i) large differences between such contexts on the web, (ii) the facilitation of goal oriented search using context classifiers, and (iii) attractive fast adaptation properties, that could counteract diversity of web environments. We demonstrate that value estimation-based fast adaptation offers considerable improvement over other known search methods.

Keywords: Fast adaptation; Internet; Reinforcement learning; Search; Small world; SVM

1. Introduction

The number of documents on the world-wide web is way over 1 billion [1]. The number of new documents is over 1 million per day. The number of documents that change on a daily basis, e.g. documents about news, business, and entertainment, could be much larger. This ever increasing growth presents a considerable problem for finding, gathering, ordering the information on the web. The only search engine that may still warrant that the information it provides is not older than 1 month is AltaVista.[1] However, the number of indexed pages on AltaVista is about 250 million documents. Google,[2] on the other hand, is indexing about 1300 million pages, but Google does not warrant any refresh rate of these documents.

The problem is complex: these search engines are not up-to-date and information gathering is not always efficient with these engines. Search engines may offer too many documents; sometimes on the order of hundreds or many thousands. Many web pages have no value, e.g. by making use of a large set of keywords, or being simply huge collections of documents originating from broad sources.

Specialized, possibly personalized crawlers are in need. This problem represents a real challenge for methods of artificial intelligence and has been tackled by several research groups [2–10]. One of the first attempts in this direction was made by Chakrabarti et al. [11] who put forth the idea of *focused crawling*. To

* Corresponding author. Tel.: +36-1-463-3515; fax: +36-1-463-3490.
E-mail address: lorincz@inf.elte.hu (A. Lőrincz).

[1] http://www.altavista.com.

[2] http://www.google.com.

1568-4946/02/$ – see front matter © 2002 Elsevier Science B.V. All rights reserved.
PII: S1568-4946(02)00025-X

understand the idea, let us consider crawling in general. Assume that 'you are at a node' of the web. This node has been analyzed and you have to decide what to do next. It is very possible that relevant information can be found in the immediate neighborhood of this node. In turn, you download all the documents next to you and start to analyze those documents. Doing so, you may found relevant documents or may not. When you are done you have the option to download all the documents that are two steps away from you and to analyze those documents. This approach is well known in the AI literature and is called breadth first technique. However, the world-wide web is '*small*': the WWW had about 800 million nodes in 1999 and the number of minimal hops required to reach most documents from any particular document was 19 [12]. Such connectivity structure between units is called 'small world'. In turn, breadth first search incurs an enormous burden as a function of depth. At one point (at a given depth) breadth first search needs to be abandoned and a decision is to be made to which node to move next. To decide on that move, the values of the nodes need to be estimated from the point of view of the goal of the search. *Focused crawling* is based on this idea. Focused crawling makes an attempt to classify the content of the document. If the document falls into the search category then the document is downloaded and the links of the documents are followed.

Diligenti et al. [1] have recognized the pitfall of focused crawling: searched information on the web is typically *hidden*: sites of particular interest may have a lower number of directed links then sites of general interest. In turn, we might face the 'needle in the haystack' problem with the haystack being sites on general interest. The hidden property is thus the implicit consequence of our particular interest.

Let us consider sites dealing with support vector machines (SVMs). Sites about SVMs are not typical on the web. Not all sites dealing with SVM are linked. In turn, focused crawling could be rather inefficient and this direct search for SVM sites might fail. On the other hand, most of the SVM sites are within (i.e. linked to) academic environments, or within sites dealing with information technology. These topics are much more general and might have much more links and a much higher 'visibility'. In turn, searching for the environment of SVM sites, could be much more efficient. A hand-waving argument can be given as follows. Documents are linked to each other. Links are made by those for whom the document has value. These links form the one-step *context* of the document. The one-step context, in turn, may be characteristic to the document. The one-step environment of the document (i.e. documents that are one step away), documents that are two steps away, etc. form the 'context' of the document. When we search for a document, by definition, we shall encounter the environment of the document first. In turn, first we might search for the environment of the document. This is the idea behind 'context focused crawling' (CFC) [1]. This idea, which is trivial for graphs with high clustering probability (e.g. regular lattices), could be criticized for the case of 'small worlds', when documents—on average—are about as far as the environment of the document. However, the question is intriguing, because the visibility could be much less for searched documents than that of the environment of the searched documents.

CFC does not take into consideration the varieties on the web: environments may differ. For example, for small universities or for small research institutes 'one-step context' may correspond to 'two-step context' for large departments of large universities. If the order of contexts might change then CFC will go close and will miss the documents. In turn, the decision whether to 'stay and download' at a given site or 'not to download but move' can be seriously jeopardized. Fast adapting value estimation method may provide an attractive solution to this search problem where information is hidden within not-yet-experienced environments. The environment of high value documents can provide reinforcing feedback in a straightforward fashion. Interestingly, reinforcement learning (RL) has not been found particularly efficient for searching the world-wide web [13]. The efficiency of RL, however, depends strongly on feature extraction. It seems natural to explore the CFC idea as the initial feature extraction method for RL. Here we show how to combine CFC with RL to search on the web.

2. Methods

2.1. Preprocessing of texts

There is a large variety of methods that try to classify texts [14–27]. Most of these methods are based on special dimension reduction. First, the occurrence, or sometimes the frequency of selected words is measured. The subset of all possible words (‘bag of words’ (BoW)) is selected by means of probabilistic measures. Different methods are used for the selection of the ‘most important’ subset. The occurrences (0’s and 1’s) or the frequencies of the selected words of the subsets are used to characterize all documents. This low—typically 100—dimensional vector is supposed to encompass important information about the type of the document. Different methods are used to derive ‘closeness measures’ between documents in the low-dimensional spaces of occurrence vectors or frequency vectors. The method can be used both for classification, i.e. the computation of decision surfaces between documents of different ‘labels’ [15,16,23,26] and clustering, a more careful way of deriving closeness (or similarity) measures when no labels are provided [14,17,22,24,28].

We tried several BoW-based classifiers on the ‘call for papers’ (CfP) problem.[3] CfP is considered a benchmark classification problem of documents: the ratio of correctly classified and misclassified documents can be automated easily by checking whether the document has the three word phrase ‘*call for papers*’, or not. Classifiers were developed for one-step, two-step environments, etc., for CfP documents. We found that these classifiers perform poorly for the CfP problem. In agreement with published results [16], supervised SVM classification was superior to other methods. SVM was simple and somewhat better than Bayes classification. However, SVM requires a large number of support vectors for the CfP problem.

2.2. SVM classification

The SVM classifier operates similarly to perceptrons. SVM, however, has better generalizing capabilities, see, e.g. the comprehensive book of Vapnik [29] a tutorial material [30], comparisons with other methods [31,32], improved techniques [33] and references therein.[4] The trained SVM was used in ‘soft mode’. That is, the output of the SVM was not a decision (yes, or no), but instead, the output could take continuous values between 0 and 1. A saturating sigmoid function[5] was used for this purpose. In turn, (i) the non-linearity of the decision surface was not sharp, (ii) for inputs close to the decision surface the classifier provides a linear output. The output of the sigmoid non-linearity can be viewed as the probability of a class. These probabilities for the different classes are distinct yardsticks working on possibly different features. The RL algorithm was used to estimate the *value* of these yardsticks.

2.3. Value estimation

There is a history of value estimation methods based on reinforcement learning: some of the important steps—judged subjectively—are in the cited papers: [34–45]. A thorough review on the literature and the history of RL can be found in [46]. In our approach, value estimation plays a central role. Value estimation works on states (s) and provides a real number, the *value*, that belongs to that state: $V(s) \in \mathrm{R}$. Value estimation is based on the *immediate rewards* (e.g. the number of hits) that could be gained at the given state by executing different actions (e.g. download or move). Value of a state (a node, for example) is the long-term cumulative reward that can be collected starting from that state and using a *policy*. Policy is a probability distribution over different actions for each state: policy determines the probability of choosing and action in a given state. Policy improvement and the finding of an optimal policy are central issues in RL. RL procedures can be simplified if all possible ‘next’ states are available and can be evaluated. This is our case. In this case one does not have to represent the policy. Instead, one could evaluate all neighboring nodes of the actual state and move to (and/or download) the one with the largest estimated long term cumulated reward, the estimated value. Typically one includes random choices for a few percentages of the steps. These random choices

[3] The CfP problem is defined by deleting the phrase ‘call for paper’ from the document, executing search on the internet and considering each document that contains the phrase ‘call for paper’ a ‘hit’.

[4] Note that SVM has no adjustable parameters.

[5] The output can be calulated as follows:

$$\text{output} = \frac{1}{1 + \exp(-\lambda \times \text{input})}$$

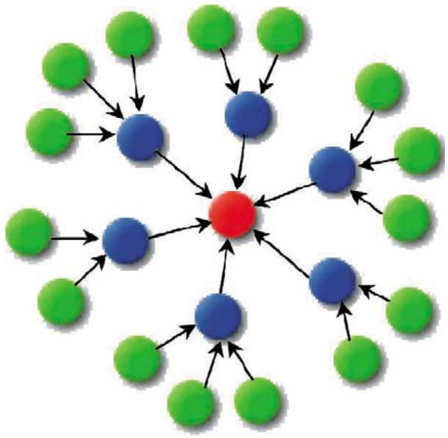

Fig. 1. Context of the document (document and its first and second 'neighbors').

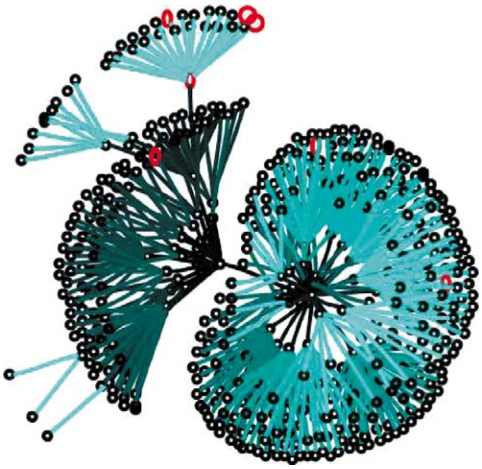

Fig. 3. Search pattern for breadth first crawler. Search was launched from neutral site. A site is called neutral if there is very few target document in its environment. Diameter of open circles is proportional to the number of target documents downloaded. Edges are color coded. There are two extremes. Dark blue: site was visited at the early stage during the search. Light blue: recently visited site (for further details, see text).

are called '*explorations*'. The estimated value-based greedy choice is called '*exploitation*'.

If the downloaded document contains the phrase 'call for papers' then the learning system incurs an immediate reward of 1. If a downloaded document does not contain this phrase then there is negative reward (i.e. a punishment) of −0.01. These numbers were rather arbitrary. The relative ratio between reward and punishment and the magnitude of the parameter of the sigmoid function do matter. These parameters influence learning capabilities. Our studies were constrained to a fixed set of parameters. One may expect improvements upon optimizing these parameters for a particular problem. In our case, search over the internet was time consuming and prohibited this optimization.

Value estimation makes use of the following upgrade

$$V^{+}(s_t) = V(s_t) + \alpha(r_{t+1} + \gamma V(s_{t+1}) - V(s_t)) \quad (1)$$

where α is the learning rate, $r_{t+1} \in R$ is the immediate reward, $0 < \gamma < 1$ is the discount factor, and

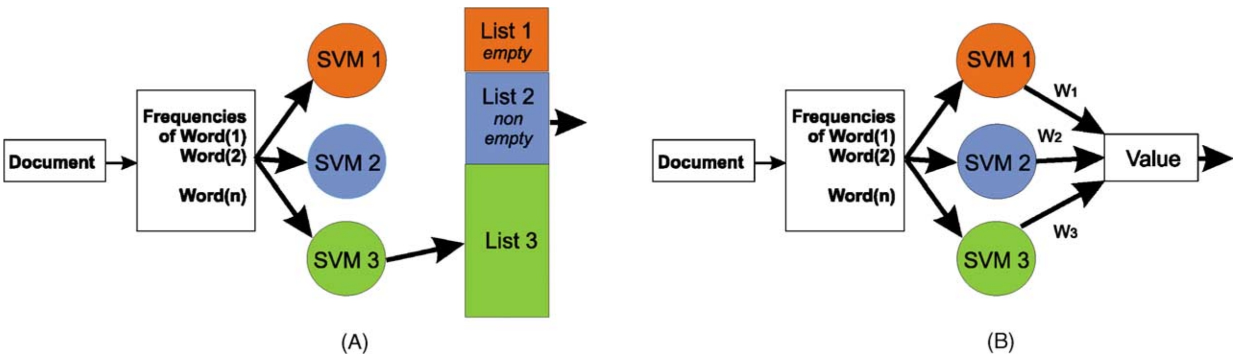

Fig. 2. SVM-based document classifiers. (A) Classification of distance from document using SVM classifiers. The CFC method maintains a list of visited links ordered according to the SVM classification. One of the links belonging to the best non-empty classifier is visited next. (B) Value estimation based on SVM classifiers. Reinforcement learning is used to estimate the importance of the different classifiers during search.

subscripts $t = 1, 2, \dots$ indicate action number (i.e. time). This particular upgrade is called temporal differencing with zero eligibility, i.e. the TD(0) upgrade. TD methods were introduced by Sutton [36]. An excellent introduction to value estimation, including the history of TD methods and description on the applications of parameterized function approximators can be found in [46]. Concerning details of the RL technique, (a) we used eligibility traces. (b) Opposed to the description given above, we did not need explorative steps because the environments can be very different and that diminished the need for exploration. (c) We did not decrease the value of α by time to keep adaptivity. (d) We approximated the value function as follows:

$$V(s) \approx \sum_{i=1}^{n} w_i \sigma(\mathrm{SVM}(i)) \tag{2}$$

where the output of the ith SVM (i.e. the ith component of the output without any 'signum' type of non-linearity) is denoted by $\mathrm{SVM}(i)$, $\sigma(\cdot)$ denotes the sigmoid function acting on the outputs of the SVM classifiers, w_i is the weight (or relevance) of the ith classifier determined by upgrade Eq. (1). If the quality of the upgrade is measured by the mean square error of the estimations then the following approximate weight upgrade can be derived for the weights (see, e.g. [46] for details):

$$\Delta w_i = \alpha(r_{t+1} + \gamma V(s_{t+1}) - V(s_t))\sigma(\mathrm{SVM}(i)). \tag{3}$$

This upgrade—extended with eligibility traces [36,46]—was used in our RL engine.

3. Features and learning to search

3.1. Breadth first crawler

A crawler is called *breadth first crawler*, if it first downloads the document of the launching site, continues by downloading the documents of all first neighbors of the launching site, then the documents of the neighboring sites of the first neighbor sites, i.e. the documents of the second neighbor sites, and so on.

3.2. Context focused crawler

A target document and its environment are illustrated in Fig. 1. The goal is to locate the document by recognizing its environment first and then the document within. The CFC method [1] was modified slightly—in order to allow direct comparisons between the CFC method and the CFC method extended by RL value estimation—and the following procedure was applied. First, a set of irrelevant documents were collected. The kth classifier was trained on (good) documents k-steps away from known target documents and on (bad) irrelevant documents. The classifier was trained to output a positive number ('yes') for good documents and to output a negative number ('no') for irrelevant documents. The outputs were scaled into the interval $(0, 1)$ by using the sigmoid function $\sigma(x) = (1 + \exp(-\lambda x))^{-1}$. If the kth classifiers output was close to 1—according to its decision surface—it suggests that there might be a target document k-steps away from the actual site/document. If more than one classifier outputs 'yes' then only the best classifier is considered in CFC. Other outputs are neglected. The CFC idea with SVM classifiers is shown in Fig. 2(A). CFC maintains a list of visited links ordered according to the SVM classification. One of the links belonging to the best non-empty classifier is visited next (this procedure is called backtracking).

The problem of the CFC method can be seen by considering that neighborhoods on the WWW may differ considerably. Even if the kth classifier is the best possible such classifier for the whole web, it might provide poor results in some (possibly many) neighborhoods. For example, if there is a large number of connected documents all having the promise that there is a valuable document in their neighborhood—but there is, in fact, none—then the CFC crawler will download all invaluable documents before moving further. It is more efficient to learn which classifiers predict well and to move away from regions which have great but unfulfilled promises.

It has been suggested that classifiers could be retrained to keep adaptivity [1]. The retraining procedure, however, takes too long[6] and can be ambiguous if CFC is combined with backtracking. Moreover, retraining may require continuous supervisory monitoring and supervisory decisions. Instead of retraining, we suggest to determine the relevance of the classifiers during the search.

[6] Training may take on the order of a day or so on 700 MHz Pentium III according to our experiences.

3.3. CFC and RL: fast adaptation during search

Reinforcement learning offers a solution here. If the prewired order of the classifiers is questionable then we could learn the correct ordering. There is nothing to loose here, provided that learning is fast. If prewiring is perfect then the fast learning procedure will not modify it. If the prewiring is imperfect then proper weights will be derived by the learning algorithm.

The outputs of the SVMs can be saved. These outputs can be used to estimate the value of a document at any instant. Value is estimated by estimating weights for each SVM and adding up the SVM outputs multiplied by these weights. In turn, one can compute value-based ordering of the documents with minor computational effort and this reordering can be made at each step. This reordering of the documents replaces prewired ordering of the CFC method. The new architecture is shown in Fig. 2(B).

4. Results and discussion

The CfP problem has been studied. Search pattern at the initial phase for the breadth first method is shown in Fig. 3.

Search patterns for the context focused crawler and the crawler using RL-based value estimation are shown in Figs. 4 and 5. The launching site of these searches was a 'neutral site', a relatively large site containing few CfP documents (http://www.inf.elte.hu). We consider this type of launching important for web crawling: it simulates the case when mail lists are not available, traditional search engines are not satisfactory, and breadth first search is inefficient. This particular site was chosen because breadth first search could find very few documents starting from this site.

'Scales' on Figs. 4 and 5 differ from each other and from that of Fig. 3. 'True surfed scale' would be reflected by normalizing to edge thickness. Radius of open circles is proportional to the number of downloaded target documents. The CFC is only somewhat better in the initial phase than the breadth first method. Longer search shows that CFC becomes considerably better than the breadth first method when search is launched from this neutral site.

Quantitative comparisons are shown in Fig. 6. According to the figure, upon downloading 20,000 documents, the number of hits were about 50, 200, and 1000 for the breadth first, the CFC and CFC-based RL crawlers, respectively. These launches were conducted at about the same time. We shall demonstrate that the large difference between CFC and CFC-based RL method is mostly due to the adaptive properties of the RL crawler.

There are two site types that have been investigated. The first site type is the neutral site that has been described before. The other site was a mail server on conferences. Also, for some examples there are runs separated by 1 month (March 2001). A large number of summer conferences made announcements during this month.

First, let us examine the initial phase of the search. This initial phase of the search (the first 200 downloaded documents) is shown in Fig. 7. According to this figure downloading is very efficient from the mail server site in each occasion. The (non-adapting) CFC based RL crawler utilizing averaged RL weights is superior to all the other crawlers—almost all downloaded documents are hits. Close to this site there are many relevant documents and the 'breadth first crawler' is also efficient here. Nevertheless, the non-adapting CFC based RL crawler outperforms the breadth first crawler in this domain. Launching from neutral sites is inefficient at this early phase. Breadth first method finds no hit close to the neutral site (not shown in the figure).

Middle phase of the search is shown in Fig. 8. Performance in the middle phase is somewhat different. Sometimes, launches from the neutral site can find excellent regions. The non-adapting CFC based RL crawler is still competitive if launched from the mail server. Launches from the mail list spanning 1 month looked similar to each other; conference announcements barely modified the results.

Search results up to 20,000 documents are shown in Fig. 9. This graph contains results from a subset of the runs that we have executed. These runs were launched from different sites; the neutral site and the mail list, as well as a third type, the 'conference' site: http:// www.informatik.uni-freiburg.de/index.en.html. This latter is known to be involved in organizing conferences. Adapting RL crawlers collected a large number of documents from all site types and during the

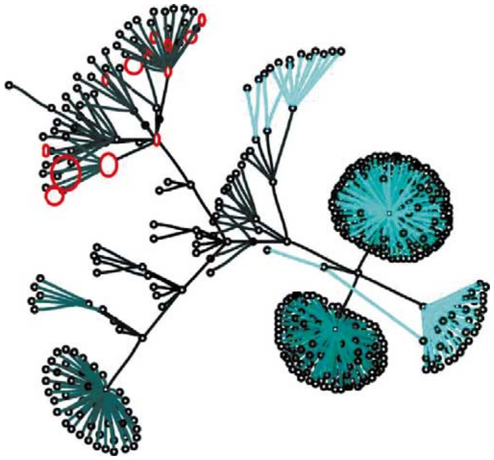

Fig. 4. Search pattern for context focused crawler. Search was launched from neutral site. Diameter of open circles is proportional to the number of target documents downloaded. Edges are color coded. There are two extremes. Dark blue: site was visited at the early stage during the search. Light blue: recently visited site.

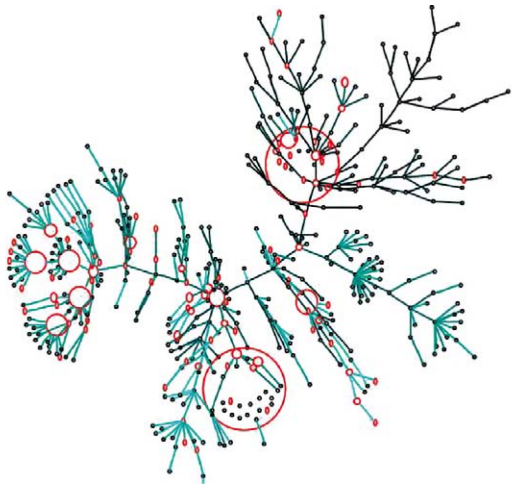

Fig. 5. Search pattern for CFC *and* reinforcement learning. Search was launched from neutral site. Diameter of open circles is proportional to the number of target documents downloaded. Edges are color coded. There are two extremes. Dark blue: site was visited at the early stage during the search. Light blue: recently visited site.

whole (1 month) time region. The rate of collection was between 2 and 5%. In contrast, although the collection rate is close to 100% for the non-adapting CFC based RL crawler launched from the mail list site up to 200 downloads, the lack of adaptation prohibits this crawler to find new target documents in circa 17,000 downloads at later stages. Taken together:

1. Identical launching conditions may give rise to very different results 1 month later.

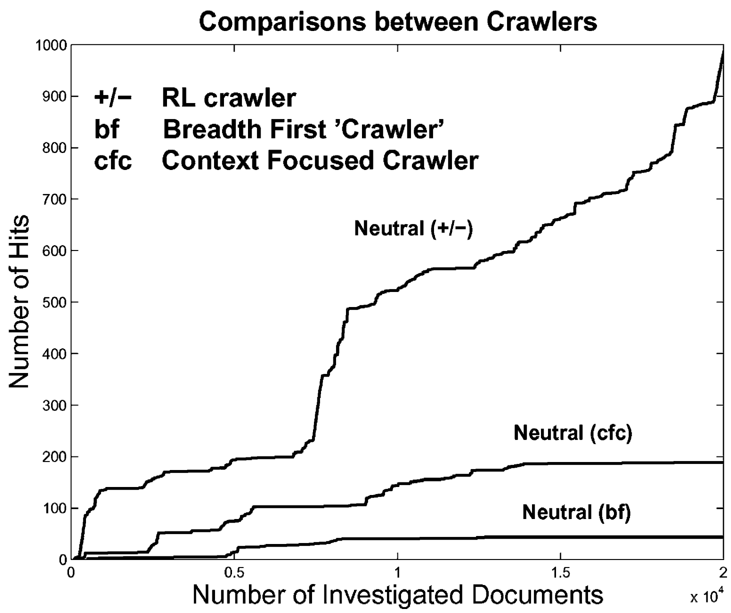

Fig. 6. Results of breadth first, CFC and CFC-based RL methods.

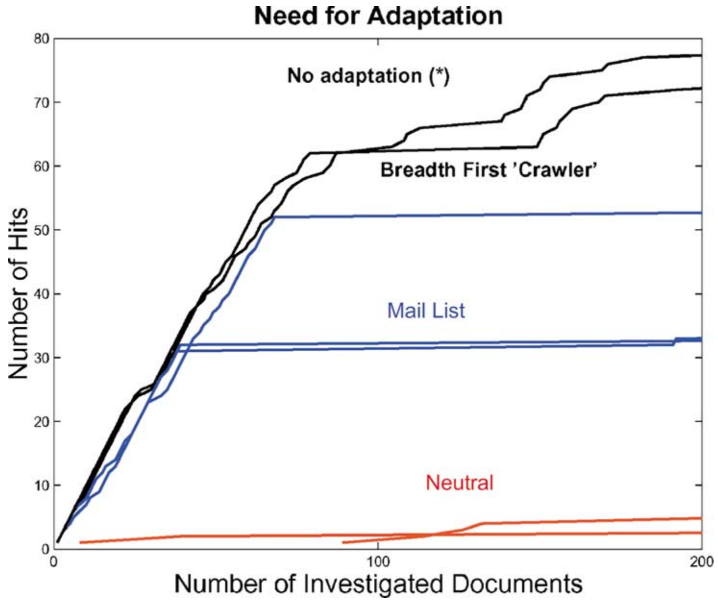

Fig. 7. Comparisons between 'neutral' and mail server sites in the initial phase. Reward and punishment are given in the legend of the figure. Differences between similar types are due to differences in launching time. The largest time difference between similar types is 1 month. Neutral site (thin lines): http://www.inf.elte.hu. Mail list (thick lines): http://www.newcastle.research.ec.org/cabernet/events/msg00043.html. Search with 'no adaptation' (dotted line) was launched from mail list and used average weights from another search that was launched from the same place.

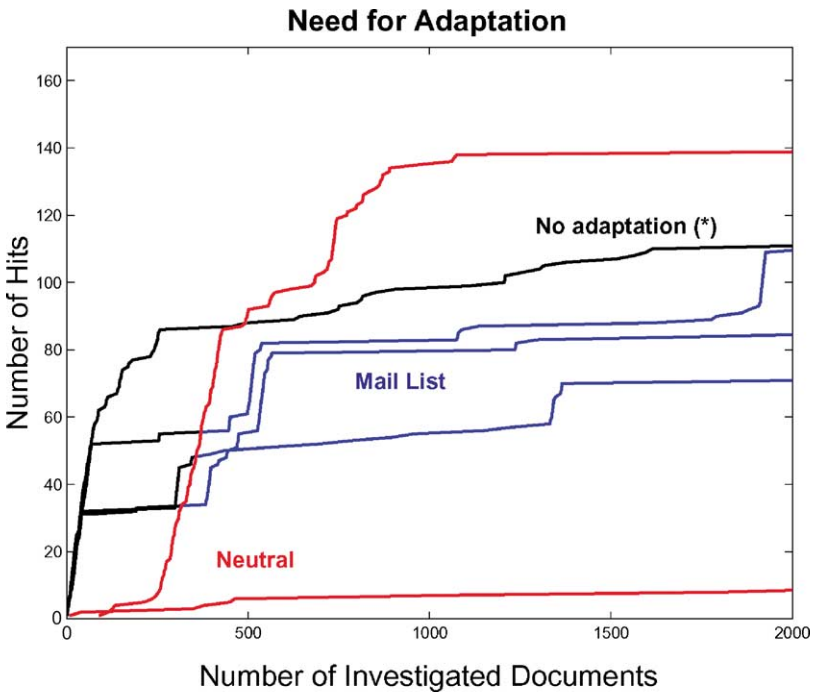

Fig. 8. Comparisons between 'neutral' and mail server sites up to 2000 documents. Same conditions as in Fig. 7.

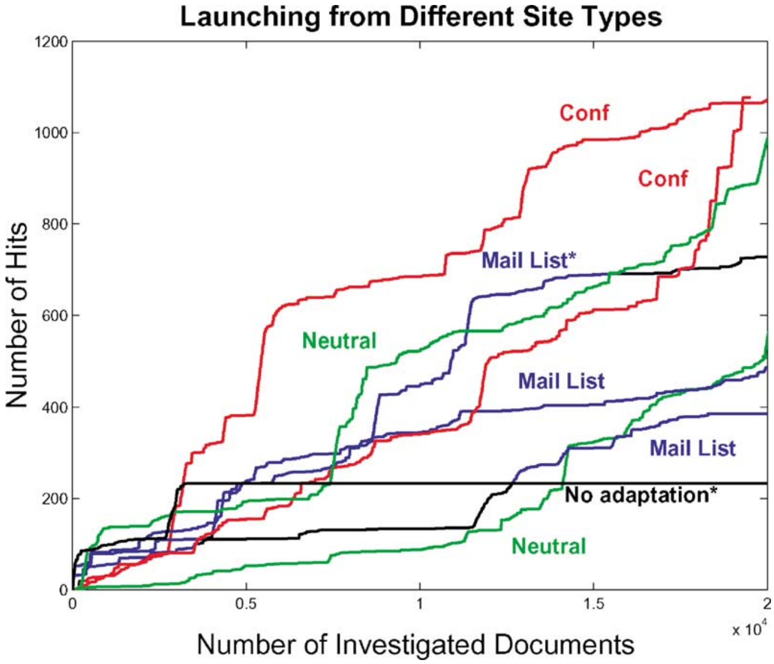

Fig. 9. Comparisons between different sites up to 20,000 documents. Same conditions as in Figs. 7 and 8. Search with 'no adaptation' used average weights from another search that was launched from the same place (denoted by ∗).

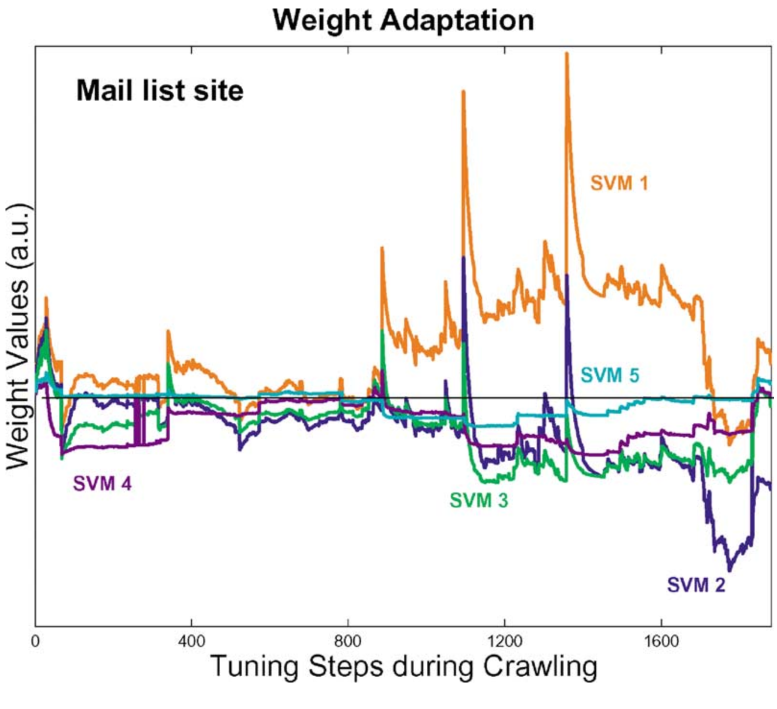

Fig. 10. Change of weights of SVMs upon downloads from mail site. Horizontal axis: occasions when weights were trained.

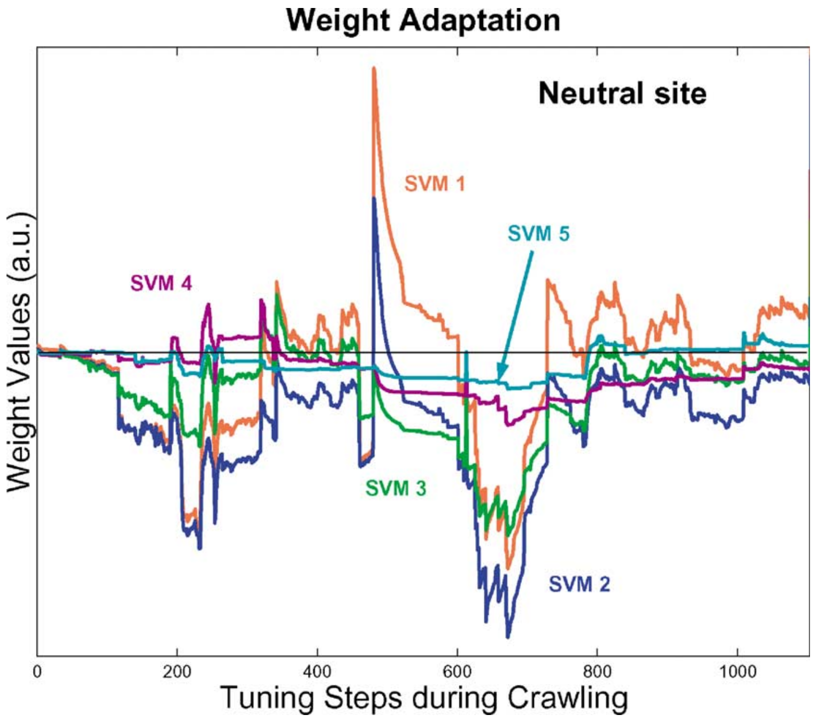

Fig. 11. Change of weights of SVMs in value estimation for 'neutral' site. Horizontal axis: occasions when value estimation was erroneous and weights were trained.

2. Starting from a neutral site can be as effective as starting from a mailing list for the adaptive RL crawler.
3. The lack of adaptation is a serious drawback even if the crawler is launched from a mailing list.

The importance of adaptation is also demonstrated by the RL weights assigned during search. These weights are shown in the following figures. Fig. 10 depicts the weights belonging to the different SVMs launched from the mail list site. At the beginning of the search the weights are almost perfectly ordered; the largest weight is given to the SVM that predicts relevant document 'one step away' whereas the fourth and the fifth SVMs have the smallest weights. That is, RL 'pays attention' to the first SVM and pays less attention to the others. This order changes as time goes on. There are regions (at around tuning step number 1700 on the horizontal axis) where most attention is paid to the fifth SVM and smaller attention is paid to the others. This means that the crawler will move away from the region. The order of importance changes again when a rich region is found; the importance of the first SVM recovers quickly and, in turn, crawling is dominated by the weight of the first SVM: the crawler 'stays' and downloads documents.

'Weight history' is different at the neutral site (Fig. 11). Up to about 100 downloads very few relevant documents were found at this site. The value of weight of the fifth SVM is slightly positive, whereas the values of the others are negative. The first and the second SVMs are weighted the 'worst'; weights belonging to these classifiers are large negative numbers. At this site, the order of SVMs that were trained at around target documents is not appropriate. Situation changes quickly when a rich region is found. In such regions the first SVM takes the lead. It is typical that the weight of the 5th SVM is ranked second. That is, the adaptation concerns mostly whether the crawler should stay or if it should move 'far away'. In turn, information contained by the 'context' is relevant and can be used to optimize the behavior of the crawler.

5. Conclusions

We have suggested a novel method for web search. The method makes use of combinations of two popular AI techniques, support vector machines (SVM) and reinforcement learning (RL). The method has a few adapting parameters that can be optimized during the search. This parameterization helps the crawler to adapt to different parts of the web. The outputs of the SVMs, together, formed a set of 'yardsticks' for the estimation of the distance from target documents. The value (the weight) of the different yardsticks may be very different at different neighborhoods. The point is that (i) RL is efficient with good features (the as k-step SVMs in this case), (ii) if there are just a few parameters for RL then these parameters can be trained quickly by rewarding for target documents. RL has many different formulations all of which could be applied here. Most promising are the approaches that can take into account (many) different criteria in the search objective [47–49]. Alas, RL methods are capable of extracting features [50] that may complement the prewired SVM features.

Acknowledgements

This work was supported by the Hungarian National Science Foundation (Grant OTKA 32487) and by EOARD (Grant F61775-00-WE065). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the European Office of Aerospace Research and Development, Air Force Office of Scientific Research, Air Force Research Laboratory. We are grateful to Dr. Ata Kabán for helpful discussions and careful reading of the manuscript.

References

[1] M. Diligenti, F. Coetzee, S. Lawrence, C.L. Giles, M. Gori, Focused crawling using context graphs, in: Proceedings of the 26th International Conference on Very Large Databases, VLDB 2000, Cairo, Egypt, http://www.neci.nec.com/lawrence/papers/focus-vldb00/focus-vldb00.ps.gz.

[2] J. Cho, H. García-Molina, L. Page, Efficient crawling through URL ordering, Computer Networks and ISDN Systems 30 (1–7) (1998) 161–172, http://www-db.stanford.edu/pub/papers/efficient-crawling.ps.

[3] J. Dean, M. Henzinger, Finding related pages in the world wide web, WWW8/Computer Networks 31 (11–16) (1999) 1467–1479, http://www.research.compaq.com/SRC/personal/monika/papers/monika-www8-1.ps.gz.

[4] S. Chakrabarti, D. Gibson, K. McCurley, Surfing the Web backwards, in: Proceedings of the 8th World Wide Web Conference, Toronto, Canada, 1999, http://www8.org/w8-papers/5b-hypertext-media/surfing/.

[5] A. McCallum, K. Nigam, J. Rennie, K. Seymore, Building domain-specific search engines with machine learning techniques, AAAI-99 Spring Symposium on Intelligent Agents in Cyberspace, 1999, http://www.cs.cmu.edu/mccallum/papers/cora-aaaiss98.ps.

[6] R. Kolluri, N. Mittal, R. Ventakachalam, N. Widjaja, Focussed crawling, 2000, http://www.cs.utexas.edu/users/ramki/datamining/fcrawl.ps.gz.

[7] S. Lawrence, Context in web search, IEEE Data Eng. Bull. 23 (3) (2000) 25–32.

[8] A. McCallum, K. Nigam, J. Rennie, K. Seymore, Automating the construction of internet portals with machine learning, Information Retrieval 3 (2) (2000) 127–163, http://www.cs.cmu.edu/afs/cs/user/kseymore/html/papers/cora-journal.ps.gz.

[9] S. Mukherjea, WTMS: a system for collecting and analyzing topic-specific web information, in: Proceedings of the 9th World Wide Web Conference, 2000, http://www9.org/w9cdrom/293/293.html.

[10] J. Murdock, A. Goel, Towards adaptive web agents, in: Proceedings of the 14th IEEE International Conference on Automated Software Engineering, 1999, http://www.cc.gatech.edu/morale/papers/ase99.ps.

[11] S. Chakrabarti, M. van der Berg, B. Dom, Focused crawling: a new approach to topic-specific Web resource discovery, in: Proceedings of the 8th International World Wide Web Conference (WWW8), 1999, http://www.cs.berkeley.edu/soumen/doc/www1999f/pdf/www1999f.pdf.

[12] R. Albert, H. Jeong, A.-L. Barabási, Diameter of the world-wide web, Nature 401 (1999) 130–131.

[13] J. Rennie, K. Nigam, A. McCallum, Using reinforcement learning to spider the web efficiently, in: Proceedings of the 16th International Conference on Machine Learning, Morgan Kaufmann, San Francisco, CA, 1999, pp. 335–343, http://www.cs.cmu.edu/ mccallum/papers/rlspider-icml99s.ps.gz.

[14] A. McCallum, Bow: a toolkit for statistical language modelling, text retrieval, classification and clustering, 1996, http://www.cs.cmu.edu/ mccallum/bow.

[15] A. Blum, J. Mitchell, Combining labeled and unlabeled data with co-training, in: Proceedings of the Workshop on Computational Learning Theory (COLT), Morgan Kaufmann, San Francisco, CA, 1998, http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-11/www/wwkb/colt98_final.ps.

[16] S. Dumais, J. Platt, D. Heckerman, M. Sahami, Inductive learning algorithms and representations for text categorization, in: Proceedings of the 7th International Conference

on Information and Knowledge Management, ICIKM'98, ACM, 1998, http://robotics.stanford.edu/users/sahami/papers-dir/cikm98.pdf.

[17] S. Kaski, Dimensionality reduction by random mapping: fast similarity computation for clustering, in: Proceedings of the International Joint Conference on Neural Networks, Vol. 1, IEEE Service Center, Piscataway, NJ, 1998, pp. 413–418, http://websom.hut.fi/websom/doc/ps/kaski98ijcnn.ps.gz.

[18] S. Chakrabarti, B. Dom, P. Indyk, Enhanced hypertext categorization using hyperlinks, in: L.M. Haas, A. Tiwary (Eds.), Proceedings of the SIGMOD-98, ACM International Conference on Management of Data, ACM Press, New York/Seattle, US, 1998, pp. 307–318, http://www.almaden.ibm.com/cs/k53/irpapers/sigmod98.ps.

[19] T. Kolenda, L. Hansen, S. Sigurdsson, in: M. Girolami (Ed.), Independent Components in Text, 2000, http://eivind.imm.dtu.dk/publications/1999/kolenda.nips99.ps.gz.

[20] T. Mitchell, The role of unlabeled data in supervised learning, in: Proceedings of the 6th International Colloquium on Cognitive Science, 1999, http://www.ri.cmu.edu/pub_files/pub1/mitchell_tom_1999_2/mitchell_tom_1999_2.pdf.

[21] A. McCallum, Multi-label text classification with a mixture model trained by em, 1999, http://www.cs.cmu.edu/mccallum/papers/multilabel-nips99s.ps.gz.

[22] T. Hofmann, Learning the similarity of documents: an information-geometric approach to document retrieval and categorization, in: Neural Information Processing Systems, Vol. 12, MIT Press, Cambridge, MA, 2000, pp. 914–920,

[23] K. Nigam, A. Mccallum, S. Thrun, T. Mitchell, Text classification from labeled and unlabeled documents using em, Machine Learning 39 (2/3) 103–134, http://www.cs.cmu.edu/ knigam/papers/emcat-mlj99.ps.gz.

[24] A. Kabán, M. Girolami, Clustering of text documents by skewness maximisation, in: Proceedings of the 2nd International Workshop on Independent Component Analysis and Blind Source Separation, ICA'2000, Helsinki, 2000, pp. 435–440.

[25] A. Vinokourov, M. Girolami, A probabilistic hierarchical clustering method for organizing collections of text documents, in: Proceedings of the 15th International Conference on Pattern Recognition, Vol. 2, IEEE Computer Press, Silver Spring, MD, 2000, pp. 182–185, http://cis.paisley.ac.uk/vino-ci0/vinokourov_ICPR00.ps

[26] T. Joachims, Estimating the generalization performance of an SVM efficiently, in: Proceedings of the 17th International Conference on Machine Learning, Morgan Kaufmann, San Francisco, CA, 2000, pp. 431–438, http://www-ai.informatik.uni-dortmund.de/DOKUMENTE/joachims_99e.ps.gz.

[27] S. Dominich, A unified mathematical definition of classical information retrieval, J. Am. Soc. Information Sci. 51 (7) (2000) 614–624.

[28] A. Vinokourov, M. Girolami, A probabilistic hierarchical clustering method for organizing collections of text documents, Technical Report ISSN 1461–6122, Department of Computing and Information Systems, University of Paisley, Paisley PA1 2BE, UK, http://cis.paisley.ac.uk/vino-ci0/hplsa_kais.ps.

[29] V. Vapnik, The Nature of Statistical Learning Theory, Springer Verlag, New York, 1995.

[30] A.J. Smola, B. Schölkopf, A tutorial on Support Vector Regression, Technical Report NC2-TR-1998-030, NeuroCOLT2, 1998.

[31] I. Guyon, V.N. Vapnik, B.E. Boser, L.Y. Bottou, S.A. Solla, Structural risk minimization for character recognition, in: J.E. Moody, S.J. Hanson, R.P. Lippmann (Eds.), Advances in Neural Information Processing Systems, Vol. 4, Morgan Kaufmann, San Mateo, CA, 1992, pp. 471–479.

[32] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines, Cambridge University Press, Cambridge, UK, 1999.

[33] S. Keerthi, S. Shevade, C. Bhattacharyya, K. Murthy, Improvements to Platt's SMO Algorithm for SVM Classifier Design, Technical Report CD-99-14, Department of Mechanical and Production Engineering, National University of Singapore, http://guppy.mpe.nus.edu.sg/ mpessk/smo_mod.ps.gz.

[34] R.E. Korf, Learning to Solve Problems by Searching for Macro-Operators, Pitman Publishers, Boston, 1985.

[35] S. Minton, Learning Search Control Knowledge: An Explanation-Based Approach, Kluwer Academic Publishers, Dordrecht, 1988.

[36] R. Sutton, Learning to predict by the method of temporal differences, Machine Learning 3 (1988) 9–44.

[37] C. Watkins, Learning from delayed rewards, PhD thesis, King's College, Cambridge, UK, 1989.

[38] J. Schmidhuber, Neural sequence chunkers, Technical Report FKI-148-91, Technische Universitat Munchen, Munchen, Germany, 1991.

[39] S. Mahadevan, J. Connell, Automatic programming of behavior-based robots using reinforcement learning, Artif. Intelligence 55 (1992) 311–365.

[40] P. Dayan, G.E. Hinton, Feudal reinforcement learning, in: Advances in Neural Information Processing Systems, Vol. 5, Morgan Kaufmann, San Mateo, CA, 1993, pp. 271–278.

[41] L.P. Kaelbling, Hierarchical learning in stochastic domains: preliminary results, in: Proceedings of the Tenth International Conference on Machine Learning, Morgan Kaufmann, San Mateo, CA, 1993

[42] G.A. Rummery, M. Niranjan, On-line Q-learning using connectionist systems, Technical Report, Cambridge University Engineering Department, 1996.

[43] M.L. Littman, A. Cassandra, L.P. Kaelbling, Learning policies for partially observable environments: Scaling up, in: A. Prieditis, S. Russell (Eds.), Proceedings of the 12th International Conference on Machine Learning, Morgan Kaufmann, San Mateo, CA, 1995

[44] M.J. Mataric, Behavior-based control: Examples from navigation, learning, and group behavior, J. Exp. Theoret. Artif. Intelligence 9 (1997) 2–3.

[45] T.G. Dietterich, Hierarchical reinforcement learning with the MAXQ value function decomposition, J. Artif. Intelligence Res. 13 (2000) 227–303.

[46] R. Sutton, A. Barto, Reinforcement Learning: An Introduction, MIT Press, Cambridge, MA, 1998.

[47] N.M. Fraser, J.W. Hauge, Multicriteria approval: application of approval voting concepts to MCDM problems, J. Multi-Criteria Decision Anal. 7 (1998) 263–273.

[48] Z.Gábor, Z. Kalmár, C. Szepesvári, Multi-criteria reinforcement learning, in: Proceedings of the 15th International Conference on Machine Learning, 1998.

[49] D. Dubois, M. Grabisch, F. Modave, H. Prade, Relating decision under uncertainty and multicriteria decision making models, Int. J. Intelligent Syst. 15 (2000) 979.

[50] T. Thrun, A. Schwartz, Finding structure in reinforcement learning, in: Advances in Neural Information Processing Systems, Vol. 7, Morgan Kaufmann, San Mateo, CA, 1995.