# Continual Computation Policies for Utility-Directed Prefetching

Eric Horvitz
Microsoft Research
One Microsoft Way
Redmond, WA 98052
horvitz@microsoft.com

## ABSTRACT

People accessing documents via the Internet typically experience latencies in retrieving content. We discuss continual-computation policies that dictate strategies for prefetching into cache portions of documents that a user may wish to review later. These utility-directed prefetching strategies maximize the expected utility of idle network resources that are available frequently, yet sporadically during the review of documents accessed from the World Wide Web. We present policies based on alternate utility models for assigning value to having immediate access to content and discuss means for coupling the methods with probabilistic models that predict a user's interests and access behavior.

## Keywords

Prefetching, caching, network bandwidth, cost—benefit analysis, decision theory, continual computation.

## 1. INTRODUCTION

The experience of accessing information via the Internet is often colored by delays incurred as information flows through the bottlenecks of limited bandwidth connections. The latencies associated with low-bandwidth local communication links are exacerbated by the trend toward embedding increasingly rich multimedia content in internet-based documents. We have explored strategies for prefetching documents as a means for minimizing latencies associated with the local communication bottleneck. Prefetching strategies promise to diminish the average perceived wait for pages when information is explicitly requested by leveraging idle time for downloading content.

In *Seventh ACM Conference on Information and Knowledge Management* (CIKM '98), Bethesda MD. November 3-7 1998, pp. 175-184. ACM Press: New York.



Beyond the goal of minimizing latencies, theoretically sound prefetching policies promise to be important as a companion technology for background query systems such as the Letzia system of Lieberman (Lieberman 1995). Such systems attempt to automatically identify content of interest based on a user's browsing and searching activity and download content based on such guesses.

Opportunities for prefetching content are underscored by typical patterns of access which show intermittent bursts of downloading content amidst a sea of idle connection time while a user reviews documents or performs other tasks. We describe decision-theoretic policies for harnessing such idle time to prefetch information into a local cache. The methods have application on the client side as well as in client-server prefetching policies. The approach is best leveraged when coupled with probabilistic models that provide the relative likelihoods about future information access.

## 2. AN ECONOMICS OF PREFETCHING

We shall take an economic perspective on the marginal benefits of incrementally prefetching portions of the content of documents. This perspective provides a foundation for maximizing the expected utility of prefetching activity. In prior work on methods for guiding ongoing problem solving, termed *continual computation* (Horvitz 1997, Horvitz 1997b), results were developed on the ideal use of idle time for computational problem solving. We adapt these methods to the problem of ideal downloading of text and multimedia over limited bandwidth networks. The methods consider the probability that a user will access documents $D$ in the future and the value of making varying portions of the documents available immediately.

### 2.1 Ideal Policies for Prefetching Documents

Let us first explore policies for prefetching for the case where we wish to minimize the delay associated with accessing from the Internet the total content contained in documents that may be accessed shortly. Assume we have

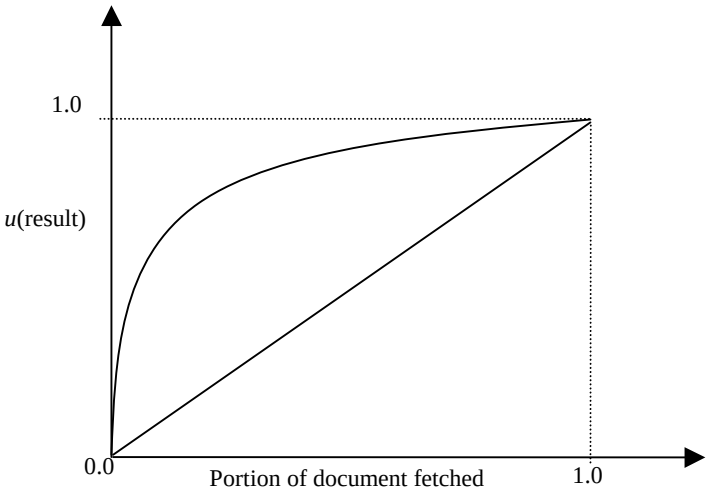


Figure 1. Linear and nonlinear utility models for representing the value of prefetching portions of documents into the local cache as a function of completeness of a download. The nonlinear curve captures the notion of decreasing marginal returns with additional fetched content.

access to exact or approximate information about the probabilities, $p(D|E)$, of the user accessing document $D$ in the next period, given some evidence $E$ about the user and current document. We decompose $E$ into sets of distinct classses of evidence, including evidence about the user's interest and access patterns, $E_u$, evidence about the link structure of a current document, $E_s$, and evidence about the user's dynamically changing view of the current document, $E_v$. Gaining access or modeling the likelihood of future instances can range from trivial to difficult depending on the application.

We shall first focus on the derivation of ideal policies that take as input likelihood information, $p(D|E)$, at any level of precision that is available, considering the knowledge-free case to be the situation where a uniform probability distribution is used to assign equal likelihood to all documents that may be accessed subsequently or within some time horizon.

Given access to probabilities of future accesses, how should a computer make best use of network idle time that might be available as a user reviews content from the current document or performs another task? We focus our analysis first on the subset of models of continual computation that address minimizing the latency associated with the next access. We seek to identify efficiently the best allocation of resources and, more generally, to identify basic principles for harnessing idle time in information retrieval.

Idle time begins at the moment a document currently being viewed has been accessed completely. The expected delay associated with accessing the next desired document is a function of the actions the system takes to prefetch documents into a local cache and the duration of the idle-time period.

Assume that we have as a goal minimizing the expected delay for downloading a document completely. What is the best policy for expending idle time in this case? Intuitively, several different policies are candidates. These include the possibility that policies are a function of the expected amount of idle time, and such strategies as timesharing the idle-time resources among several documents by sequencing the allocation of a fraction of the expected idle time to downloading each document, dictated by the likelihoods of accessing each document. In the Appendix, we include a proof that, for any quantity of idle time, it is best to spend the idle-network resources on downloading documents completely in order of the probability that they will be accessed. This policy is insensitive to document length and to the amount of idle time.

## 2.2 Ideal Policies for Partial Prefetching

In Section 2.1, we considered documents as providing value to users only when they are downloaded completely. A more realistic preference model allows for value to be drawn from portions of documents that are immediately available. Let us now consider relaxing the restriction that an entire document must be prefetched and consider policies for prefetching partial content from a document.

Our approach to developing strategies for prefetching partial documents is related to earlier work on computational strategies with the ability to refine results incrementally with ongoing computation. Such *flexible computation strategies* (Horvitz 1987) are valuable for computing results that are employed in time-critical decision making, especially where there is variation and uncertainty in computation time. We shall explore analogous flexible strategies for text retrieval and their use in conjunction with continual prefetching policies.

The downloading of documents can typically be decomposed in a natural manner. Such straightforward strategies as simple, serial truncation can be employed to incrementally extend the completeness, and, thus, the value of transmitted documents. Beyond simple truncation of the latter portion of documents, a document can be pruned in a variety of ways. Partial documents can be created through various forms of summarization and the excision of specific classes of content. A form of the latter is widely available today as an option in web browsers that suppresses the downloading of complex bitmap graphics to speed the transmission of web pages.

We have investigated the use of utility functions to capture the value of having immediate access to progressively larger amounts of content of partially downloaded documents. Figure 1 displays linear and nonlinear utility functions that represent the value of having immediate access to portions of a document as a function of the completeness of the content. The linear model represents documents and associated tasks where the value of having immediate access to a portion of a document grows linearly

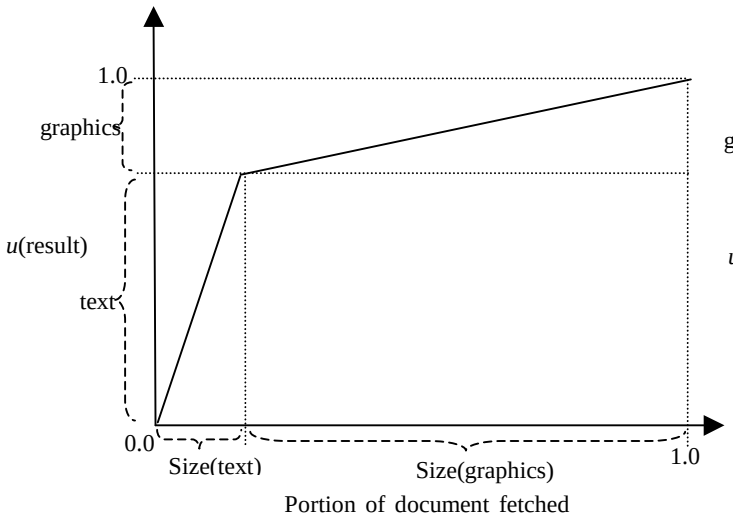


Figure 2. A piecewise linear utility model for the value of prefetching portions of content contained in network-based document. For this function, the flux associated with the download of text is greater than that of downloading the associated graphics.

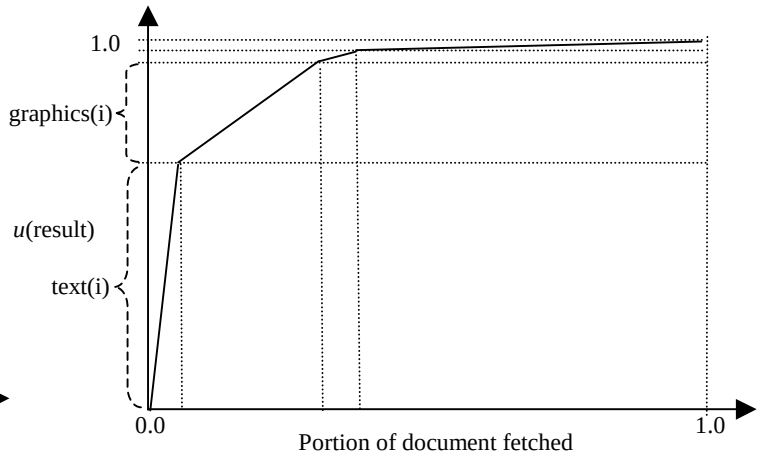


Figure 3. A finer-grained piecewise linear model. This piecewise utility model considers the diminishing returns associated with screenfulls *i* of content for prefetching text and graphics.

with the fraction of the document that is prefetched. The nonlinear model represents documents and associated tasks where having immediate access to the initial portion of the document is most valuable and where additional content is associated with positive, but decreasing marginal increases in value with continuing downloading.

The value of allocating resources to prefetching portions of documents that may be accessed in the future can be characterized in terms of the *rate* at which the best methods can deliver expected value with downloading. We prove ideal policies for prefetching partial pages in terms of the rate of *change* of the expected value in the next period with additional bytes of transferred information. Key notions in our analysis is the prefetching *value flux*, $\psi(D_i,t)$, and *expected value flux*, $\psi(D_i,t)p(D_i|E)$. The value flux for a document $D_i$ is the instantaneous rate at which the downloading increases future expected value at $t$ seconds into the downloading of a document. The value flux depends on the speed of a connection as well as the utility associated with portions of a model. The expected value flux is computed as the product of the value flux and the probability that the document will be accessed next or within some time horizon, $p(D_i|E)$. The expected value flux is the instantaneous change in the total expected utility in the next period generated by prefetching activity that is contributed by continuing to download document $D_i$.

### *2.2.1 Prefetching Content with Constant Flux*

Let us first consider the case where computational strategies generate a constant value flux as content is downloaded. Given documents $D_i$, that may be accessed in the next period, and a constant value flux of $\psi(D_i,,t)$, we can show that, for any amount of idle time, the prefetching policy that maximizes the expected value at the start of the next period is to apply all resources to the problem with the maximal *expected value flu*x. The expected value flux is computed as the product of the value flux and the probability that the document will be accessed next or within some time horizon, $\psi(D,t)p(D|E)$.

We have shown that, for the case of linear expected value flux, networking resources should be dedicated to prefetching the document with the highest expected flux until it is completely downloaded, followed by the document with the next highest expected flux, and so on, until the cessation of idle time or downloading of all feasible documents that may be fetched in the next period. The proof of this policy is included in the Appendix.

### *2.2.2 Prefetching Content with Nonlinear Flux*

Moving beyond the case of linear value flux, a user's preferences for having immediate access to partial documents may often be described by a nonlinear function of the completeness of the available document. Nonlinearity captures several factors, including the likelihood that desired information is contained in increasingly larger portions of a document, that a user can determine quickly whether a document is relevant or not by reviewing initial content, and that latencies will be perceived only when a user attempts to page past the prefetched portion of a document. The mean time that passes before a user incurs the latency associating with fetching the additional content from a document grows with the content that has already been fetched—and this time can be used to fetch additional portions of a document. For

these factors, nonlinear value showing decreasing marginal benefit with additional content are good candidates for representing the value of prefetching portions of the content of web pages.

Let us first examine a special case of nonlinear value functions represented as piecewise-linear functions. Such models facilitate modeling separately the flux associated with the download of distinct components of a document. For example, we can decompose a document into the value flux associated with downloading portions of distinct screenfuls of text, of components of pages, or combinations of these two. Figure 2 displays a piecewise linear value function for a document as a function of the completeness of a partial download where different linear fluxes are associated with the text and graphics components. Figure 3 displays a similar decomposition, but with the addition of a consideration of the diminishing returns associated with prefetching subsequent screenfuls of text and graphics. In this model, pairs of linear segments represent the value of fetching text, followed by fetching the associated graphics into cache for successive screenfuls of content.

Ideal policies for downloading partial content can be derived by generalizing the earlier results for the case of partial prefetching of content in situations of constant expected flux. Rather than associate documents with constant levels of expected flux, we consider the flux associated with subcomponents of documents, and consider ideal strategies for downloading these components. As described in Theorem 3 in the Appendix, the ideal policy is to continue to download the portion of a document's content associated with the value segment with the greatest expected flux and to continue to download this segment until it is completely prefetched or idle time ends. When all of the content associated with the best segment is fetched, we consider whether the value flux of the next segment of the document we have been focusing on is greater than the greatest flux associated with other documents. If so, we fetch the content associated with the next segment of the current document. If downloading a component of another document has a greater expected flux, we switch the attention of prefetching to the other document. The monotonically decreasing flux of successive piecewise linear segments makes it necessary to only check the next available segments for each document under consideration during the current session. We continue to prefetch content in this manner until idle time ends or all documents under consideration are fetched.

When a user makes a new explicit access, we can employ the straightforward strategy of focus networking resources on the downloading of the specified document or on the completion of a partially prefetched document. More sophisticated strategies can be designed by considering the probability distribution over future idle time and trading off completing the explicitly requested document for prefetching potential future content. Details on this tradeoff for continual computation are described in Horvitz 1997.

For the case of continuous nonlinear utility functions that have a negative second derivative, we can show that it is optimal to prefetch documents by the maximal expected flux. We include the proof of this policy of following the maximum instantaneous derivative in the Appendix. The proof is a slight generalization of the proof for the piecewise linear model, shrinking the size of the segments to be of zero length in the limit. A simple way to understand this result is that the expected gain in future value for any amount of networking is optimized by continuing to prefetch the content with the highest expected flux. In prefetching sets of documents associated with concave-down functions, we know that the document that offers the maximal instantaneous expected flux will add the most to the overall expected utility and that all other sources of content will deliver less value now and at all other times in the future.

### *2.2.3 Mixtures of Utility Models with Non-positive Second Derivatives*

We can generalize decision making for utility models with nonpositive second derivatives, including documents associated with linear, piecewise linear, and smoothly varying utility models. We continue to pick the policy with the highest instantaneous derivative. We know that any document under consideration offers constant or diminishing returns with additional effort. Thus, we need to continue to pick the document associated with the content with the greatest instantaneous derivative of expected flux and to either continue to download the same document or switch to another with an initial derivative that becomes greater than the current document, or when the current document is completely downloaded (for cases of linear expected flux).

### *2.2.4 General Nonlinear Value Models*

For the general case of nonlinear flux under uncertain idle time, we are typically forced to perform general optimization to identify ideal policies and consider the probability distribution over future idle time. However, we employ allocation strategies that take advantage of the results we described earlier, but employ a greedy, myopic approximation; we consider the best allocation of small slices of the usable idle-time, $\Delta t$. At each stage, we consider the contribution to the total value of prefetching for the allocation of $\Delta t$ resources to downloading additional portions of each document.

We allocate all of the time to the instance with the greatest product of likelihood of seeing the problem instance, $p(D_i|E)$ and the change in value of the allocation of $\Delta t$, and continue to apply this greedy procedure as more time is available. We justify allocating all of the resource in the $\Delta t$ slice to the solution of a single instance by arguing that the flux in small $\Delta t$ regions is approximately constant and adapt earlier results on linear value flux. However, for value functions of expected flux associated with nonmonotonic second derivatives, such myopic policies can lead to expected utilities that are less optimal than a global

optimization that considers the probability distribution over future idle time.

## 2.3 Considering Costs of Shifting Attention

We have considered ideal policies without considering the cost of shifting attention from one document to another. We note that costs typically arise in networking based in the overhead of shifting attention from one file or server to another. Such costs can influence decisions about shifting to problems with greater value flux. Given the presence of costs of shifting attention, idle-time should be switched to refining an instance that yields a greater expected flux only if the expected benefits of the switch are greater than the costs of the shift. Details of consideration of the cost of shift of attention for the general case of computational problem solving are presented in Horvitz 1997.

## 2.4 Cost-Benefit Analysis to Limit Prefetching

Prefetching activity during idle time should not be considered to be cost free. Prefetching taxes the overall network as clients or servers allocated resources to speculatively access additional information much of which may go unused. We can limit prefetching by expressing simple limits on the rate at which bytes are transferred for prefetching. Alternatively, we can employ an active cost—benefit analysis. The latter models can be employed with mechanisms that associate a cost per byte for downloaded content or that manipulate an explicit measure of cost capturing the estimated cost of client or server prefetching activity on the overall network. Such a model can be made sensitive to the load sensed on the local or more global network.

We harness the expected value flux in procedures for limiting the extent of prefetching. Control procedures trade the expected savings in reduced latencies for the monetary cost of downloading bytes, or for the estimated cost of the additional prefetching activity on the Internet. For content represented by value functions with marginal decreasing utility, the expected value of continuing to prefetch content with the continual computation policies drops with ongoing prefetching. In operation, we continue to compare the current expected flux associated with prefetching with a specified (or sensed) cost per downloaded kilobyte. When the expected value of prefetching falls below the cost, prefetching activity should be ceased. The cost assigned per bits of downloaded content can be made sensitive to dynamically changing context. A user entering a time-critical, directed information-gathering mode of operation may wish to pay the higher cost for a larger cache of content in return for reduced access latencies.

## 2.5 Assessment and Application in Practice

Let us consider the case of the use of the piecewise linear representation of value in practice. We allow software engineers or users to specify thresholds on the size of the local cache and on the maximum number of screens of content or maximum bytes of information that can be prefetched from each document. For the sake of assessing value we assume that context where document will be accessed in the future, and assume that the value of prefetching the maximal content for the document to be 1.0, and that the value derived from each component (*e.g.*, successive text or graphics components) increases linearly with the portion of the component as specified by the utility model. Let us assume that the size of a component (*e.g.*, the first screenful of text) is *Size*(*Component*) bytes and that the

$$Expected\ \psi(\text{segment}) = \left[\frac{Value(\text{Component})}{Size(\text{Component})} \times R\right] p(D \mid E)$$

value derived from the component ranges linearly from 0 for no content fetched to *Value*(*Component*) for all of the component. Given a transmission rate of *R* baud, the expected value flux associated with downloading any component associated with a segment in a piecewise linear value function is:

# 3. PROBABILISTIC MODELS OF ACCESS

The prefetching policies that maximize expected utility described in Section 2 can take as input quantitative or qualitative information about the likelihoods of the next documents accessed when such information is available. The prefetching policies do not require highly accurate probabilities. Indeed, the policies apply to any probabilistic information at any level of resolution. The economic prefetching policies provide motivation for developing methods for estimating the probability, $p(D|E)$, that a user will access documents *D* in this session, given evidence, *E*, about the user's behavior and the link structure of a document.

There has been growing interest in probabilistic models of user behavior for Internet searching and browsing. Statistical analyses of user behavior in reviewing and accessing documents highlight opportunities for gaining access to useful insights about user behavior through collection of data about user actions. For example, Morita and Shinoda showed a correlation between the duration of users' dwell time on documents and their interest in the content contained in the documents (Morita and Shinoda, 1994). More recently, Tauscher and Greenberg have explored patterns of revisitation and bookmarks of web users (Tauscher and Greenberg, 1997). We have been exploring two complimentary approaches to modeling the probability of information access: (1) analysis of log files of user access and (2) constructing Bayesian or parametric user models of access, taking into consideration statistical information when data is available. We shall review these approaches.

## 3.1 Statistical Analysis from Server Logs

We performed experiments from data about user activity gathered from real-world Internet servers to probe the value of continual prefetching. In the studies, we analyze

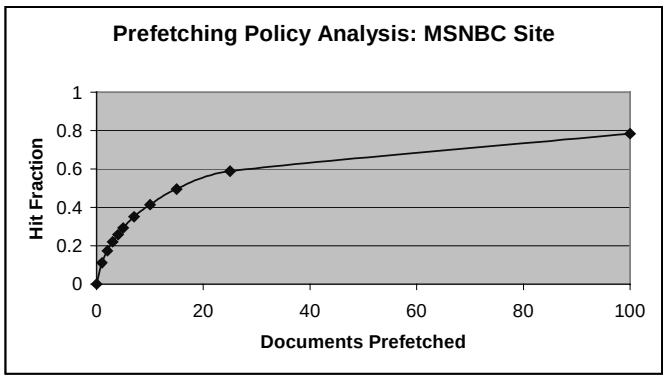


Figure 4. Analysis of the value of prefetching using a Markov model built from a training set of log data. The graph displays the probability of content from the next document accessed being in the cache as a function of the number of documents prefetched.

anonymous user log data to generate the probability that a user will access content given the current document being reviewed. In a representative experiment, we examined a log file of anonymous user activity at the MSNBC Internet site for news and information. We process this file to build a Markov model of the probability, $p(D_i \mid \text{current document})$, for all pages in the log. The training data for this data set consists of 95,000 transitions among pages. For each page in the training set, the probability of a transition to other pages is computed by looking at all pages that had been visited next by users. The learned Markov transition probabilities provide the likelihood of users accessing the next page, and can be used to compute the probability of different sequences of pages.

Figure 4 displays an analysis of the value of prefetching activity for the MSNBC statistical analysis on a test set of 6,000 transitions gathered from the server. If we consider documents has having identical utility models, and fetch by the probability of the document. $p(D_i \mid E)$, we obtain a 0.5 probability of having content in the cache for the next accessed if we prefetch content from approximately 15 documents during recent idle time. This analysis indicates that the expected latencies are halved for this quantity of cached content. The number of documents and completeness of the content in the cache, the idle time required to generate the cache, and the size of the increase in expected value associated with such prefetching activity depends on the details of the specific utility model employed.

## 3.2 Constructing User Models

As a complement to our analysis of log data, we have been studying the value of constructing probabilistic user models which can be used to infer $p(D|E)$ based on a consideration of the structure of links on a page, a user's behavior, and notions of the inferred *context* of an information retrieval or browsing session.

There has been growing interest in the use of Bayesian models of the goals and needs of users as they work with software applications (Horvitz, 1997c, Heckerman & Horvitz, 1998; Horvitz, et al., 1998). Some of this work has focused on the use of the Bayesian network representation (Pearl 1989; Heckerman et al, 1992). Bayesian models can be built from expert knowledge or from data sets drawn from logs or instrumented browsers.

Our work on user modeling has been focused on models for identifying interests and for predicting access of content based on user activity and document structure. Our work to date on monitoring events has highlighted the potential value of observing the following distinctions:

- the links and classes of links displayed on the page being viewed and their configuration
- a user's browsing activity
- the standard interface options available to the user
- the links being presented to the user based on the relevance ordering of a recent search or the results of inference strategies such as collaborative filtering
- the inferred similarity of link content to the page being viewed

Browsing actions with relevance to the probability of the next access include the length of time of a dwell on the current page following a new access or scroll, the dwell or access history, pattern and direction of scroll, and the mouse cursor position and patterns of motion. We have been building and refining models that relate a user's access patterns to a user's actions and document structure. We are especially interested in events and structure that might be monitored with an instrumented browser. Models taking into account such information include simple functional parameterizations of such observations as the order of links in a screen of content. That is, we build functions to assign a set of probabilities to links given the nature and configuration of the links and update the probabilities with information on the time-dependent status of the displayed content.

A Bayesian network representing dependencies among observations and user access is displayed in Figure 5. The model highlights dependencies among such variables as link structure and user browsing behavior in predicting the next access. As captured in the model, links displayed to users and the search context influence the probability that the user will attend to different links. In turn, the user's attention to explicit links influences scroll, dwell, and mouse behavior, as well as the probability distribution over the next access.

We have studied the integration of notions of a user's information-gathering context in Bayesian user models for prefetching. We have included a variable labeled *Search Context* in the Bayesian network in Figure 5. This variable encodes the probability distribution over a set of alternate access scenarios linked to distinct informational goals. Informational goals include the situation where an Internet search has recently been performed and a user is iteratively visiting a list returned by a search engine. In such a situation, we can expect the user to display a pattern of visiting links from a list composed from the search and return to the search page. Other information goals include the state of "browsing without a specific goal," and "seeking out specific information on a topic related to the last *x* documents." Browsing without a specific goal increases the likelihood that goals are shifting with information being reviewed, and are likely to be influenced more strongly by the current page structure and long-term profile of interests rather than by a topic that characterizes documents that have been recently reviewed.

We have also pursued the value of other classes of evidence. For example, we have done preliminary studies of the relevance of patterns of gaze for predicting the access of documents. Studies of user gaze during Internet browsing in the Microsoft usability labs (using an ASL-6000 corneal reflection gaze tracking system) suggests that this data may be used in conjunction with other observations about activity to provide valuable additional evidence about a user's attention, interests, and next access.

We are currently continuing our investigation of the value and costs of building and using probabilistic user models to inform a prefetching system.

## 4. SUMMARY

We presented decision-theoretic policies for guiding the ideal prefetching of network-based content in situations of limited or costly bandwidth. We considered the case of prefetching of complete documents, as well as more flexible strategies for downloading portions of documents. We then discussed means for controlling a tradeoff in the cost versus the benefits of continuing to prefetch content. After presenting prefetching policies, we analyzed issues with the development of models of user document access, including statistical analysis and Bayesian modeling. Utility-directed continual-computation policies for prefetching hold opportunity for enhancing the experience of users browsing content on the Internet by minimizing latencies. Our ongoing research centers on refining Bayesian models that diagnose a users' interests and that forecast future information access based on a user's actions and the structure of the content. As part of this endeavor, we are pursuing the construction and use of models that include richer representations of the context of an information-retrieval setting.

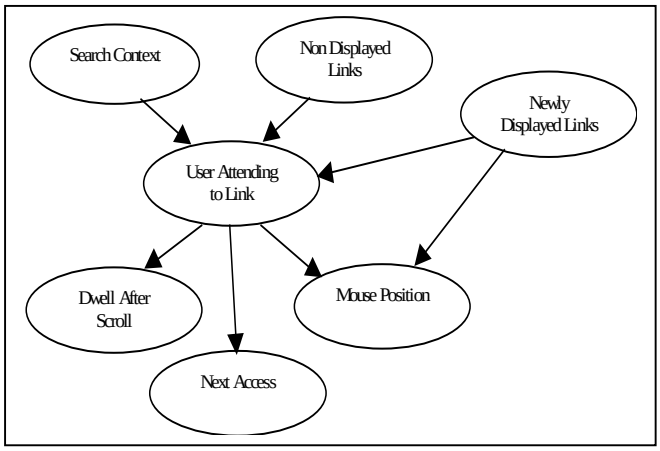


Figure 5. A Bayesian network for predicting the next access based on a consideration of document structure and user behavior.

## 5. ACKNOWLEDGMENTS

Carl Kadie assisted with the analysis of patterns of user access from server log data.

## 6. REFERENCES

(Heckerman et al., 1992) D.E. Heckerman, E.J. Horvitz, and B.N. Nathwani. Toward normative expert systems: Part I. The Pathfinder project. *Methods of Information in Medicine* 31:90-105, 1992.

(Heckerman and Horvitz, 1998) D. Heckerman and E. Horvitz, Inferring Informational Goals from Free-Text Queries: A Bayesian Approach, *Fourteenth Conference on Uncertainty in Artificial Intelligence, Madison, WI*, July 1998. http://research.microsoft.com/~horvitz/aw.htm

(Horvitz, 1987) E. Horvitz, Reasoning about Beliefs and Actions under Computational Resource Constraints, *UAI-87: Third Conference on Uncertainty in Artificial Intelligence, Seattle, WA*, AUAI. July 1987. 429-444. http://research.microsoft.com/~horvitz/u87.htm

(Horvitz, 1997a) E. Horvitz, Models of Continual Computation, *AAAI-97: National Conference on Artificial Intelligence, Providence, Rhode Island*, AAAI Press: Menlo Park, California. July 1997. 286-293 http: //research.microsoft.com/~horvitz/cc.htm

(Horvitz, 1997b) E. Horvitz, Continual Computation, *Microsoft Research Technical Report*. July 1997.

(Horvitz, 1997c) E. Horvitz, Agents with Beliefs: Reflections on Bayesian Methods for User Modeling, *Proceedings of the Sixth International Conference on User Modeling Chia Laguna, Sardinia*, A. Jameson, C. Paris, and C. Tasso, editors. June 1997. 441-442. http://zaphod.cs.uni-sb.de/~UM97/abstract horvitz.html

(Horvitz et al., 1988) E.J. Horvitz, J. Breese, and M. Henrion. Decision theory in Expert Systems and Artificial Intelligence. *International Journal of Approximate Reasoning*, Special Issue on Uncertainty in Artificial Intelligence, 2:247-30.
http: //research.microsoft.com/~horvitz/dt.htm

(Horvitz et al., 1998) D. Heckerman and E. Horvitz, The Lumiere Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users, *Fourteenth Conference on Uncertainty in Artificial Intelligence, Madison, WI*, July 1998.
http://research.microsoft.com/~horvitz/lumiere.htm

(Lieberman, 1995) Letizia: An Agent That Assists Web Browsing, *International Joint Conference on Artificial Intelligence*, Montreal, August 1995

(Morita and Shinoda, 1994) M. Morita and Y. Shinoda, Information Filtering Based on User behavior Analysis and Best Match Text Retrieval. *SIGIR '94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, Dublin Ireland*, B. Croft and C.J. van Rijsbergen, eds., Springer-Verlag, London, 1994. 273-282.

(Pearl, 1988) J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann Publishers: San Francisco, 1991.

(Tauscher and Greenberg, 1997) L. Tauscher and S. Greenberg, How people revisit web pages: empirical findings and implications for the design of history systems, *International Journal of Human-Computer Studies* (1997). 47(1) 1-222

# APPENDIX

## 1. Proof of Ideal Prefetching Policy for Complete Documents

The expected delay associated with accessing the next desired document is a function of the actions the system takes to prefetch documents and the duration of the idle-time period. We use $t(D_i)$ for the time required to completely download the next document, $D$, $T$ for the total usable idle network time, and $t^f$ for the idle-time fraction allocated to downloading a document ahead of time. The maximal time that can be allocated to downloading a future document is the time needed to access the document, $t(D)$, and the total usable idle time $T$ is less then or equal to the maximal idle time, $T^m=\Sigma_i t(D_i)$, sufficient to download all potential documents under consideration.

The expected delay before completely downloading a document is,

$$\sum_j p(T=T_j)\sum_i p(D_i \mid E)[t(D_i)-T_j t_{i_i}^f] \quad (1)$$

where idle-time $T$ is indexed by its magnitude and where

$$T_j t_{i_i}^f \le t(D_i)$$

The expected savings gained from prefetching activity is

$$\sum_j p(T=T_j)\sum_i p(D_i \mid E) T_j t_i^f \quad (2)$$

What can we say about optimizing an assignment of the set of fractions of total usable idle-time resource $T$ to alternate problems? Let us consider the case for some constant amount of idle time, $T$. We can rewrite the expected savings in the next period as

$$T\sum_i p(D_i \mid E) t_i^f$$

which will be maximized for any value of $T$ by maximizing the quantity

$$\sum_i p(D_i \mid E) t_i^f$$

**Theorem 1. Partition of Idle Network Resources.** *Given an ordering over the probability of future access, $p(D_1|E) > p(D_2|E) > \ldots > p(D_n|E)$, representing the likelihood that these problems will be accessed in the next period, the partition of network idle-time that minimizes the expected time in the next period is to apply all network resources to the most likely problem until it is solved, then the next most likely, and so on, until the cessation of idle time or solution of all possible accesses in the next period.*

Proof

Assume an ordering over the probability of the next accesses

$$p(D_1|E) > p(D_2|E) >,\ldots,> p(D_n|E)$$

Consider the case where network resources are only applied to the most likely document. The total savings are maximized when we maximize,

$$T\sum_i p(D_1 \mid E) t_i^f = Tp(D_1 \mid E) \quad (3)$$

Now consider the case where some resource fraction $x$ is diverted from downloading document $D_1$, and is applied to downloading one or more of the other documents $D_2$ ,. . .,$D_n$. We show that directing some portion $x$ of the idle-time resource fraction from the most likely instance to the other documents must be less optimal than allocating all of the network bandwidth to the most likely problem. That is, we show that

$$T\ p(D_1 \mid E) > Tp(D_1 \mid E)(1-x) + T\sum_{i=2}^{n} p(D_i \mid E)t_i^f \quad (4)$$

Our goal reduces to showing that

$$xp(D_1 \mid E) > \sum_{i=2}^{n} p(D_i \mid E)t_i^f$$

We know that $\sum_{i=2}^{n} t_i^f = x$ so we need to show that

$$p(D_1 \mid E)\sum_{i=2}^{n} t_i^f > \sum_{i=2}^{n} p(D_i \mid E)t_i^f \quad (5)$$

As $p(D_2|E) > p(D_{i>2}|E)$, we know that

$$p(D_2 \mid E)\sum_{i=2}^{n} t_i^f > \sum_{i=2}^{n} p(D_i \mid E)t_i^f$$

By definition, $p(D_1) > p(D_2)$. Thus, we know that

$$x\ p(D_1 \mid E) > \sum_{i=2}^{n} p(D_i \mid E)t_i^f$$

for any $x$.

## 2. Ideal Prefetching Policies for Partial Documents

The value of allocating resources to prefetching portions of documents that may be accessed in the future can be characterized in terms of the *rate* at which the best strategies can deliver value with downloading. We use *networking value flux* to refer to the rate of *change* of the expected value of downloading portions of documents with download time. The networking value flux, $\psi(D,t)$, for a document $D_i$ is the instantaneous rate, at which the downloading delivers value at $t$ seconds into the downloading of a document. The value flux depends on the speed of a connection as well as the utility associated with portions of a model. In the general case, a downloading strategy applied to a document may deliver value as a nonlinear function of networking effort. Let us consider the special case where computational strategies generate a constant networking value flux.

**Theorem 2: Partition of Networking Resources for Constant Value Flux**. Given documents $D_i$, that may be accessed in the next period, and a networking value flux $\psi(D_i,,t)$ for the solution of each instance that is constant with time, the networking resource partition policy that maximizes the expected value at the start of the next period is to apply all resources to the problem with the maximal product of probability and networking value flux. That problem should be refined until a document is completely downloaded, then the result with the next greatest product should be analyzed, and so on, until the cessation of idle time or solution of all problems possible in the next period.

Proof

We assume that allocation of networking time to each document instances provide constant networking value fluxes $\psi(D_i,t) = \psi_i$ for each document based on the refinement of a sequence of documents. The expected value of prefetching is,

$$\sum_j p(T = T_j)\sum_i p(D_i \mid E)\,\mathrm{u}(D_i, T_j t_i^f) \quad (9)$$

and that,

$$\mathrm{u}(D_i, T_j t_i^f) = \int_0^{T_j t_i^f} \varphi_i dt = \varphi_i T_j t_i^f \quad (10)$$

Thus, the expected value of prefetching can be rewritten as,

$$\sum_j p(T = T_j)\sum_i p(D_i \mid E)\varphi_i T_j t_i^f \quad (11)$$

For any amount of idle networking time, $T_j$, less than the time required to download all of the future documents under consideration, the fastest that the expected value of prefetching can grow is by the instance that maximizes $p(D_i|E)\ \psi_i$. The ideal policy is to apply all resources to downloading the document with the highest value of $p(D_i|E)\ \psi_i$. Citing the same argument used in Theorem 1, any amount of time $x$ re-allocated to another document would diminish the total expected value of prefetching because it would be multiplied with smaller valued products. When document $D$ associated with the largest flux is downloaded completely, it is removed from consideration and the same argument is made with the remaining $n$-1 documents.

**Theorem 3: Partition of Networking Resources for Piecewise Linear Value with Marginally Decreasing Flux**. Given documents $D_i$, that may be accessed in the next period, and a networking value flux described by $\psi(D_i,,t)$ that is piecewise linear with successive segments associated with decreasing flux, the networking resource partition policy that maximizes the expected value at the start of the next period is to continue to allocate resources to the linear segment drawn from the set of next available linear segments of documents that has the maximal product of probability and value flux, and to continue to download content associated with that segment until the segment is completed, and then to move to the segment with the next highest expected value flux until all segments of all problems are solved or the cessation of idle time.

Proof

We extend Theorem 2 on ideal allocation of resources for constant flux to each piecewise linear segment associated with the current state of a document. By considering portions of content associated with the linear value segments instead of entire documents, the policy for maximizing the contribution to the expected value in the next period, will be to continue to pick the segment associated with the highest product of probability of document access and flux and to continue to download content until that segment is completed, and then move on to download content associated with the segment with the next highest expected flux, regardless of document. We can employ the same arguments in Theorem 2 to show that choosing any other segments would lead to a diminishment of the overall expected value at the start of the next period as compared with this policy. We know that, for any document, downloading content associated with earlier segments will necessarily have higher expected value flux than downloading content associated with later segments. Thus, we need only to check the next available segment in each incompletely downloaded document to identify the best sequence of segments.

**Theorem 4: Partition of Networking Resources for Smoothly Decreasing Marginal Flux**. Given documents $D_i$, that may be accessed in the next period, and a networking value flux described by $\psi(D_i,,t)$ for the solution of each instance associated with $d\psi(D_i,,t)/dt < 0$ for the solution of each instance associated, the networking resource partition policy that maximizes the expected value at the start of the next period is to allocate resources to the problem with the maximal product of probability and instantaneous value flux, until all problems are solved or until the cessation of idle time.

Proof

We adapt Theorems 2 and 3 by simply reducing the size of the segments in the piecewise linear utility models in the limiting case to zero while increasing the number of segments to infinity. As we shrink the unit of resource allocation to zero in the limit, we maximize the expected value at the start of the next period by continuing to access information by choosing to download content by the document with the greatest product of the probability of document and the instantaneous flux. The marginal decreasing flux for downloading the content of any document tells us that any other policy will lead to a smaller integrated expected value at the end of idle time.

In real-world computing, we must minimally allocate some *finite* amount of networking resource. For situations where the expected flux is monotonically decreasing for all documents, the policy for maximizing the contribution to the expected value in the next period will be to continually pick the problem associated with the highest mean expected flux for that small quantity of expenditure. Because each document has an expected networking flux that is monotonically decreasing with allocation of resources, the greatest currently available flux must be greater than the future expected flux associated with this or any other document. Thus, guiding allocation by the local mean expected flux leads to ideal expected overall expected value of prefetching. Unfortunately, the number of tests and potential frequency of switching among documents increase as the quantum of networking resource being considered for allocation is decreased. Thus, manipulating the allocation of resources of an extremely fine grain size can be costly if the cost of switching is significant.