Pre-sending Documents on the WWW: A Comparative Study

David Albrecht, Ingrid Zukerman and **Ann Nicholson**
School of Computer Science and Software Engineering
Monash University
Clayton, VICTORIA 3168, AUSTRALIA
{dwa,ingrid,annn}@csse.monash.edu.au

Abstract

Users' waiting time for information on the WWW may be reduced by pre-sending documents they are likely to request, albeit at a possible expense of additional transmission costs. In this paper, we describe a prediction model which anticipates the documents a user is likely to request next, and present a decision-theoretic approach for pre-sending documents based on the predictions made by this model. We introduce two evaluation methods which measure the immediate and the eventual benefit of pre-sending a document. We use these evaluation methods to compare the performance of our decision-theoretic policy to that of a naive pre-sending policy, and to identify the domain parameter configurations for which each of these policies provides a clear overall benefit to the user.

1 Introduction

Users typically have to wait for information they require from the World Wide Web (WWW). Excessive waiting increases user dissatisfaction. We propose to address this problem by means of a system placed on a single server site, which pre-sends documents to a user. A decision-theoretic approach is taken, where documents that yield the highest expected positive benefit are pre-sent. This requires the consultation of a predictive model that anticipates a user's document requests from the WWW site [Zukerman et al., 1999]. The calculation of the benefit to the user takes into account the increased cost of transmitting documents that are not requested versus the reduction in waiting time for documents that are requested.

In preliminary work [Nicholson et al., 1998], we used a simple Time Markov prediction model and evaluated the pre-sending system only in terms of its immediate benefit to the user. The contributions of this paper are: (1) the evaluation of the eventual benefit to the user as a result of pre-sending a document, for different operating conditions; (2) the comparison of the decision-theoretic pre-sending policy with a naive policy which pre-sends the document that is most likely to be requested [Bestavros, 1996]; and (3) the incorporation of a hybrid prediction model [Zukerman et al., 1999] into both pre-sending policies.

In the next section we discuss related research. We then consider the features of our domain, followed by a description of our prediction model. In Section 5, we describe the decision-theoretic model used for pre-sending documents. In Section 6, we consider our evaluation methods, followed by the presentation of our results, and concluding remarks.

2 Related Research

The recent growth in the WWW and on-line information sources has inspired research on agents that help users derive the most benefit from the vast quantities of available facilities and information. These agents may be broadly classified into *recommender systems*, which recommend facilities or information items likely to be of interest to the user, e.g., [Lieberman, 1995; Joachims et al., 1997], and *action systems*, which go one step further, performing actions on the user's behalf, e.g., [Bestavros, 1996; Balabanović, 1998; Nicholson et al., 1998]. Both types of systems use prediction models which anticipate a user's preferences, including documents of likely interest, e.g., [Maes and Kozierok, 1993; Lieberman, 1995; Bestavros, 1996; Joachims et al., 1997].

The action system described in this paper is most closely related to the system described in [Bestavros, 1996], which pre-sends documents to a user by consulting a prediction model obtained from the behaviour patterns of the general population. Our system differs from Bestavros' in two aspects: (1) we consult a hybrid prediction model which combines four Markov models [Zukerman et al., 1999], compared to Bestavros' simple Time Markov model; and (2) we use a decision-theoretic model for pre-sending documents, while Bestavros uses a naive strategy which pre-sends the document with the highest probability of being requested.

3 Domain Features

The most salient features of the WWW are its large size and constant variation. The first feature suggests that a pre-sending system, such as that developed here, should use approximate models to predict a user's requests. The second feature suggests that such a system should dynamically adapt to changes, or at least be easily modifiable. Further, since we are modeling a single server site, a feature particular to our system is that our observations of the user's document requests constitute a partial record of the user's movements through the internet. This is because not all the user's movements to external locations are observed, and requests for documents already in the client's cache are not observed.

The pre-sending system described in this paper requires a predictive model which anticipates a user's document requests on the WWW. The predictive model presented in the next section takes into account the above features as follows.

It is based on Markov models which approximate users' document requests on the WWW. These models represent external or unseen locations, and are trained from data collected over a period of time (and can be easily re-trained).

The training data was obtained by logging our web server over a 15 month period. The results presented in this paper are based on a 50-day time window of these logs.

The collected data points were pre-processed (see [Zukerman et al., 1999] for details) and divided into sessions. Each session contains the temporal sequence of requests from a single client, where a request takes the form {`referer requestedDoc time size`}. The `referer` is the current internet location (http address) of the user. This location may be a local (previously requested) web page on the server site, an external web page on another internet site, or `'-'` (empty) when the information has not been provided. The `requestedDoc` is the http address of the document being requested by the client. The `time` is a time stamp (in seconds) indicating when the request was received. The `size` is the number of bytes in the requested document.

After pre-processing, our data consisted of 1,095,730 document requests, where 59,486 clients at 21,692 referer locations requested 17,332 different documents (one session per client); 14,023 of the referers were requested documents, and there were 103,972 different referer/document combinations.

4 Prediction Model

We estimate $P(D_{R_1}, T_{R_1} | \text{previous requests})$, where D_{R_1} is the next document requested and T_{R_1} is the time of this request. To make the prediction problem computationally tractable, we assume that the distribution of the time for requesting a document is independent of the document that is requested, that the next document requested depends only on the previous documents, and that the time of the next request depends only on the time of the last request, T_R. This last assumption over-simplifies our domain, since the size of a document affects both its transmission time and the user's reading time, thereby influencing the time of the next request. In the future, we intend to factor the size of a document into the estimation of the time of the next request.

According to our assumptions,

$$P(D_{R_1}, T_{R_1} | \text{previous requests}) = P(D_{R_1} | \text{previous documents}) \times P(T_{R_1} | T_R) \ .$$

The estimation of $P(T_{R_1} | T_R)$ is described in Section 4.1, and that of $P(D_{R_1} | \text{previous documents})$ in Section 4.2.

4.1 Next document is requested at time t

For our current database (based on 50 days of data), the time between successive requests from a client ranges from 0 to 4,100,910 seconds (∼ 47 days): $0 \leq T_{R_1} - T_R \leq 4,100,910$.

Figure 1 shows the cumulative frequency distribution of the inter-arrival time between consecutive requests (plotted against a log scale). This distribution indicates that approximately 90% of document requests from a client are made within 122 seconds of the previous request, 95% are made within 874 seconds, and 99% within 343,412 seconds. As shown in Figure 1, a combination of three functions provides a good fit for the data (these functions were found using a weighted least-squares method). Therefore, we use the following fitted probability function to estimate the probability of receiving a request at a particular time.

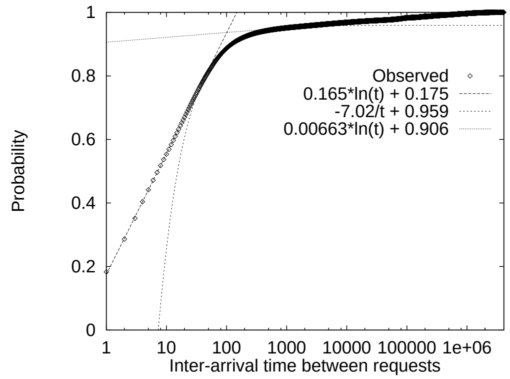


Figure 1: Cumulative frequency distribution of document requests plotted against a log scale of the inter-arrival time between requests $(T_{R_1} - T_R)$ and fitted with three functions.

$$\Pr(T_{R_1} - T_R < t) = \begin{cases} 0.165 \times \ln(t) + 0.175 & 1 \leq t \leq 45 \\ -7.02/t + 0.959 & 45 < t \leq 960 \\ \min\{0.00663 \times \ln(t) + 0.906, 1\} & t > 960 \end{cases}$$

4.2 A particular document is requested next

To predict the document requested next, we use a hybrid model called `maxHybrid`, which combines four basic Markov prediction models: *Time, Second-order Time, Space* and *Linked Space-Time*. The time-based models consider temporal information only. The Time Markov model predicts a user's next request based only on the document that was requested last, and the Second-order Time Markov model makes this prediction based on the last two requested documents. The Space Markov model, which was motivated by the observation that normally people follow links on web pages, adds structural constraints to the Time Markov model. In the Space Markov model, the probability of a document being requested depends only on the referring document, which has a link to the requested document. The Linked Space-Time Markov model also combines temporal and structural information. In this model, the probability of a client requesting a document depends on both the last requested document and the referring document of the last requested document. A detailed description of these Markov models and their training procedure appears in [Zukerman et al., 1999].

The `maxHybrid` model was built based on empirical evidence obtained from the performance of these four basic models. Its performance in predicting the next requested document was compared with that of the basic models and other hybrid models, producing significantly more accurate predictions than any of these models [Zukerman et al., 1999].

After receiving a request for document D_R, the `maxHybrid` model consults the four Markov models, and makes its prediction using the model which made a prediction with the highest probability (this may be a different model after each observation). The decision-theoretic model then uses the probabilities obtained from the selected model to calculate the expected benefit from pre-sending a document.

5 Decision-theoretic Model

The decision-theoretic model selects for pre-sending the document whose transmission has the highest positive *expected immediate benefit*. This benefit is the difference between the

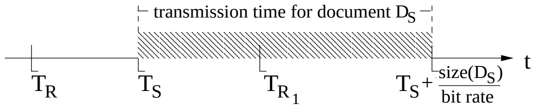

transmission time for document D_S
T_R
T_S
T_{R_1}
$T_S + \frac{size(D_S)}{bit\ rate}$
t

Figure 2: Time line for a request–pre-send sequence.

expected additional cost of pre-sending a document that is not requested next and the expected reduction in waiting cost due to the pre-sending of a document that is requested next.

Our decision-theoretic model considers for pre-sending documents that may be requested *next* by a user, rather than documents that may be requested subsequently. This may be justified by examining the circumstances under which the benefit of pre-sending a subsequently requested document is higher than the benefit of pre-sending a next requested document. This would happen when there is high uncertainty regarding the document to be requested next, but many subsequent requests converge on the same document. A preliminary inspection of our WWW site shows that only 16% of the pages can be reached by more than one path of length 2 or 3. Hence, our decision-theoretic model (which considers only paths of length 1) is justified as a promising initial approach. In the future, we intend to investigate policies which consider pre-sending documents that may be requested later by a user, and to compare their performance with that of our current policy.

Expected additional transmission cost

Let D_R be the document requested by the client at time T_R, and D_S the document selected for pre-sending.[1] The actual pre-sending is done at time T_S, and the next document D_{R_1} is requested at time T_{R_1} (Figure 2). The additional cost of pre-sending an unnecessary document is a function of the document size (in bytes) and the cost per byte, *cpb*. Thus, the additional cost of pre-sending a document D_S at time T_S is[2]

$$C(D_S, D_{R_1}, T_S, T_{R_1}) = \begin{cases} cpb \times size(D_S) & \text{if } D_{R_1} = \emptyset \text{ or } D_{R_1} \neq D_S \\ 0 & \text{if } D_{R_1} = D_S \end{cases}$$

The top line of this formula reflects a situation where no further requests are made by the user ($D_{R_1} = \emptyset$) or the document which was pre-sent is not the one requested next. The second line reflects a situation where the pre-sent document is requested next, hence no unnecessary costs are incurred.

Therefore, the *expected* additional cost of pre-sending a document D_S at time T_S is

$$EC(D_S, T_S) \;=\; cpb \times size(D_S) \times [\Pr(D_{R_1} = \emptyset) + \Pr(D_{R_1} \neq D_S \;\&\; D_{R_1} \neq \emptyset)] \;,$$

where the probabilities are obtained from the `maxHybrid` prediction model (Section 4.2).

Expected reduction in waiting cost

If the system pre-sends the document the client requests next, then the waiting time is reduced or even removed. Since the benefit of pre-sending a document is formulated in terms of cost, we multiply the formulas for the reduction in waiting time by a cost per second (*cps*), which reflects the inconvenience caused to the user from having to wait for a document.

Let $W(D_S, D_{R_1}, T_S, T_{R_1})$ represent the reduction in the cost of waiting for the desired document D_{R_1} requested at time T_{R_1}, after document D_S was pre-sent at time T_S.

$$W(D_S, D_{R_1}, T_S, T_{R_1}) = \begin{cases} 0 & \text{if } D_{R_1} = \emptyset \\ cps \times (T_{R_1} - T_S) & \text{if } D_{R_1} = D_S \text{ and} \\ & \quad T_S < T_{R_1} < T_S + \frac{size(D_S)}{bps} \\ cps \times \frac{size(D_S)}{bps} & \text{if } D_{R_1} = D_S \text{ and} \\ & \quad T_S + \frac{size(D_S)}{bps} \leq T_{R_1} \\ 0 & \text{otherwise } (D_{R_1} \neq D_S \text{ or } T_{R_1} \leq T_S) \end{cases}$$

where *bps* is the transmission rate, expressed in bytes/sec. That is, if no further requests are made by the user, then the reduction in waiting cost is zero. If the pre-sent document is requested next, but has not arrived in its entirety when the request is made, the user will have to wait only for the portion of the document that still remains to be sent. If the pre-sent document is requested next and is fully in the cache at the time the request is made, then the user will save the time it takes for the entire document to arrive. Finally, if the pre-sent document is not the one that is requested next or the next request arrives before the system decides which document to pre-send, then the user will have to wait for the entire document, so there is no reduction in waiting time.

Therefore, the *expected* reduction in the cost of waiting for D_{R_1} after document D_S was pre-sent at time T_S is

$$EW(D_S, T_S) =$$
$$cps \times \int_{t=T_S}^{T_S + size(D_S)/bps} (t - T_S) \times p(t) \mathrm{d}t \times \Pr(D_{R_1} = D_S) +$$
$$cps \times \frac{size(D_S)}{bps} \times \Pr(T_{R_1} \geq T_S + \frac{size(D_S)}{bps}) \times \Pr(D_{R_1} = D_S) \;,$$

where p is the density function for requesting a document at time t, derived from the probability function described in Section 4.1, and the document-request probabilities are obtained from the `maxHybrid` prediction model (Section 4.2).

Expected immediate benefit

The system pre-sends the document which has the highest *expected immediate benefit*, provided it is positive (doing nothing has an expected benefit of 0).

$$\textit{Expected-Immediate-Benefit}(D_S, T_S) = EW(D_S, T_S) - EC(D_S, T_S) \;.$$

6 Evaluation Methods

We consider two methods for the comparative evaluation of our decision-theoretic model versus the naive pre-sending policy: *Immediate Benefit* and *Eventual Benefit*. The Immediate Benefit method operates under the `no-memory/next-request` scenario, while the Eventual Benefit method operates under the `8-hours/cache` and ∞`/cache` scenarios. The first scenario, which was also used in [Zukerman et al., 1999], was designed to assess the performance of a prediction model regarding the next requested document only. The second and third scenarios assume that the client has a cache and the server keeps track of the cache's contents. In the second scenario, a pre-sending action is considered successful if the

[1] In principle, more than one document may be selected for pre-sending. However, at present we consider only a single document.

[2] $T_S - T_R$ was empirically found to be about 33 milliseconds (Section 7).

pre-sent document is requested within 8 hours of being pre-sent (8 hours approximates one work day; 84.5% of the sessions last up to 8 hours). In the third scenario, a pre-sending action is considered successful if the pre-sent document is requested at any time after it was pre-sent.

These scenarios also affect the documents considered for pre-sending. Since for the `no-memory/next-request` scenario the server does not keep track of previous events, a pre-sending policy may decide to pre-send a just-visited page, which adversely affects its performance. In contrast, a memory of 8 hours indicates that it is unnecessary to pre-send documents that were sent (either requested or pre-sent) in the last 8 hours, since they are still in the client's cache. Similarly, a memory of ∞ indicates that any previously sent document should not be pre-sent. It is important to note that documents which are not considered for pre-sending are not ignored, in the sense that the probabilities of the remaining documents are not normalized. This is because normalization would artificially increase the probability that a document will be requested, which in extreme cases may result in the pre-sending of documents which have a slim chance of being requested.

Immediate Benefit

The Immediate Benefit method computes the difference between the savings due to a reduced waiting time for documents that are requested next and the cost of pre-sending documents that are not requested next. To compute this benefit we assume that the system receives a sequence of document requests $\{D_{R_1}, D_{R_2}, \ldots, D_{R_N}\}$ from a client at times $\{T_{R_1}, T_{R_2}, \ldots, T_{R_N}\}$. After receiving and satisfying a user's request for document D_{R_i}, the system may pre-send a document D_{S_i} at time T_{S_i}.

$$\textit{Immediate-Benefit} = \sum_{i=1}^{N} [W(D_{S_i}, D_{R_{i+1}}, T_{S_i}, T_{R_{i+1}}) - C(D_{S_i}, D_{R_{i+1}}, T_{S_i}, T_{R_{i+1}})] \ ,$$

where the calculation of $W(D_{S_i}, D_{R_{i+1}}, T_{S_i}, T_{R_{i+1}})$ and $C(D_{S_i}, D_{R_{i+1}}, T_{S_i}, T_{R_{i+1}})$ is as described in Section 5.

Eventual Benefit

The Eventual Benefit method computes the difference between the savings due to a reduced waiting time for documents requested eventually during their lifetime in the cache, and the cost of pre-sending documents that are never requested during their lifetime in the cache. To compute this benefit we assume that the client has a cache of virtually infinite capacity, and consider the above-mentioned `8-hours/cache` and ∞`/cache` scenarios.

$$\textit{Eventual-Benefit} = \sum_{i=1}^{N} [W(D_{R_i}, T_{R_i}, T_{S_{R_i}}) - C(D_{S_i}, T_{S_i})] \ ,$$

where $T_{S_{R_i}}$ is the time when document D_{R_i} was pre-sent, $C(D_{S_i}, T_{S_i})$ is the additional cost due to pre-sending a document that was never requested during a particular time span, and $W(D_{R_i}, T_{R_i}, T_{S_{R_i}})$ is the reduction in the cost of waiting for a pre-sent document that the client requested later.

$$C(D_{S_i}, T_{S_i}) =$$
$$\begin{cases} cpb \times size(D_{S_i}) & \text{if } D_{S_i} \text{ is never requested or} \\ & \quad T_{R_{S_i}} < T_{S_i} \text{ or} \\ & \quad T_{R_{S_i}} \geq T_{S_i} + \textit{MemorySpan} \\ 0 & \text{otherwise} \end{cases}$$

where $T_{R_{S_i}}$ is the time D_{S_i} is requested, and *MemorySpan* is a particular time span since a document was pre-sent (we consider two values for *MemorySpan*, 8 hours and ∞, depending on the scenario). According to this formula, the user incurs an unnecessary expense when a pre-sent document is never requested or when it is requested either before it is actually pre-sent or after a time which is not realistically considered part of the session.

Event	doc	Expected Benefit	Actual Benefit	Immed. Benefit (Cum.)	Eventual Benefit (Cum.)
`Req`	D3	D2 0.5 270 D5 0.5 150			
`Pre`	D2		-122	-122	-122
`Req`	D7	D4 1.0 -10			
`Req`	D2	D6 1.0 50	314+122	-122	314
`Pre`	D6		-40	-162	274
`Req`	D6	D1 1.0 -60	100+40	-22	414

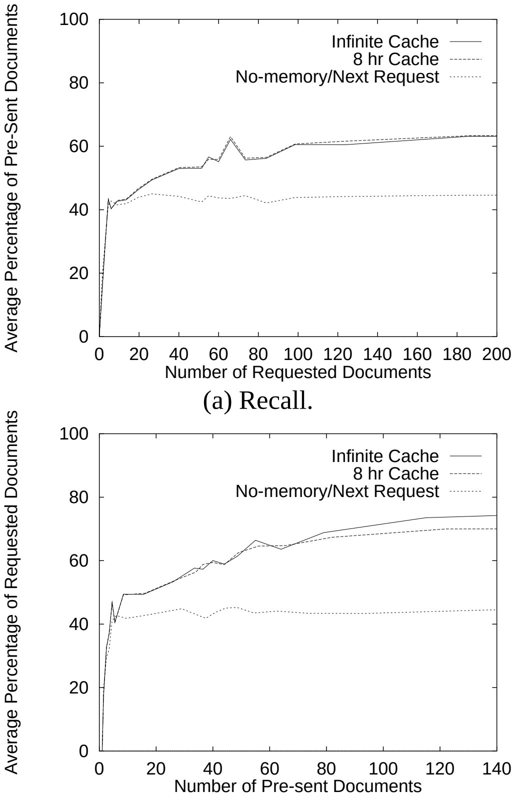
Figure 3: Constructed example showing an event sequence.

$$W(D_{R_i}, T_{R_i}, T_{S_{R_i}}) =$$
$$\begin{cases} 0 & \text{if } D_{R_i} = \emptyset \\ cps \times (T_{R_i} - T_{S_{R_i}}) & \text{if } T_{S_{R_i}} < T_{R_i} < T_{S_{R_i}} + \frac{size(D_{R_i})}{bps} \\ cps \times \frac{size(D_{R_i})}{bps} & \text{if } T_{S_{R_i}} + \frac{size(D_{R_i})}{bps} \leq T_{R_i} \text{ and} \\ & \quad T_{R_i} \leq T_{S_{R_i}} + \textit{MemorySpan} \\ 0 & \text{otherwise} \end{cases}$$

That is, if no more documents are requested by the client, the waiting cost is not affected. If D_{R_i} is in transit, the user will not have to wait for the portion of the document that has already arrived at the time the request is made. If the requested document is in the cache (and *MemorySpan* has not lapsed), then the user will save the time (and cost) corresponding to waiting for the entire document. Finally, if the requested document is neither in the cache nor in transit, or its transit time is larger than *MemorySpan*, or it is requested after *MemorySpan* has lapsed, then there is no reduction in waiting time.

Example

We now illustrate the operation and evaluation of our pre-sending system with a simple constructed example. Consider the sequence of events in Figure 3. The client requests (`Req`) document D3, which is then sent. The decision-theoretic system is given two candidate documents for the next request, D2 and D5, each with probability 0.5, and calculates the expected benefits of these documents (270 and 150 respectively). D2, the document with the highest expected benefit, is pre-sent (`Pre`), which immediately incurs a transmission cost (-122) that reduces both the cumulative immediate and eventual benefits. The next request is for document D7. The only candidate for pre-sending this time is D4, but it has a negative expected benefit (-10), so nothing is pre-sent. Next, D2 is requested. Since it was previously pre-sent, the eventual benefit is incremented by the reduction in waiting cost (314) and by the transmission cost (122 – to cancel the previous cost, since the transmission proved necessary). The system then pre-sends D6 (the transmission cost yields -40 benefit), which is the next request, so both cumulative benefits increase by 100 – the reduced waiting cost, plus 40 – to cancel the transmission cost. The final total benefits are -22 using immediate benefit and 414 using eventual benefit.



(a) Recall.

(b) Precision.

Figure 4: Performance of the `maxHybrid` model.

7 Results

As indicated above, the results in this section were obtained from 50 days of data logged by our server. All the models were tested using 80% of the sessions for training and 20% for testing. Differences noted in the results for the various prediction models are significant at the 5% level.

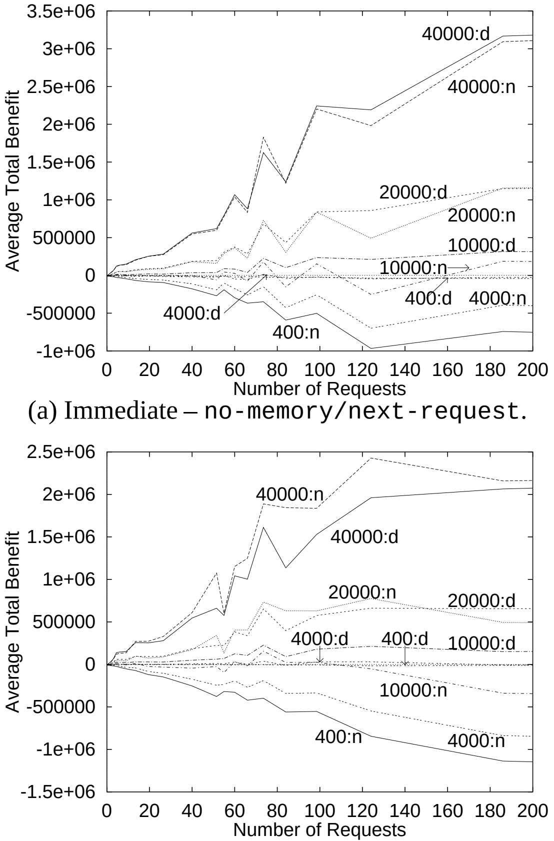
We are interested in two aspects of predictive performance: (1) *recall* – the percentage of requested documents that were previously pre-sent; and (2) *precision* – the percentage of pre-sent documents that are subsequently requested. Figure 4(a) depicts the recall predictive performance of the `maxHybrid` model for each of our three scenarios, under the assumption that the system pre-sends the document with the highest probability of being requested next (this is effectively the behaviour of the naive pre-sending policy described in [Bestavros, 1996]). The x-axis shows the number of documents requested by a client during a session.[3] The y-axis shows the average percentage of requested documents that were pre-sent within the event memory span of each scenario (0, 8 hours or ∞). For example, when 40 documents are requested, the `maxHybrid` model has an average recall of about 53% for both cached scenarios (8 hours and ∞), and an average recall of 44% for the `no-memory/next-request` scenario. After 4 requests, the performance of the pre-sending policy under this scenario is independent of the number of requested documents. As expected, the recall performance of the `maxHybrid` model improves when the evaluation is in terms of its eventual benefit rather than its immediate benefit. However, its performance for the ∞`/cache` and the `8-hours/cache` scenarios is essentially equivalent.

As for recall, the precision performance of the `maxHybrid` model is higher for the eventual benefit evaluation method than for the immediate benefit method (Figure 4(b)). In sessions where more than 71 documents were pre-sent (which constitutes 3% of the data), the precision under the ∞`/cache` scenario rises over the precision under the `8-hours/cache` scenario. This happens because in the 60% of these sessions which take longer than 8 hours, the decision-theoretic policy pre-sends more documents under the `8-hours/cache` scenario than under the ∞`/cache` scenario (where the cache holds every previously sent document).

Since modelling a cache over more than 8 hours does not improve the recall predictive performance at all, and improves the precision predictive performance only slightly for a small portion of the data, we now compare the performance of our two pre-sending policies (naive and decision-theoretic) only for the `no-memory/next-request` and `8-hours/cache` scenarios. We assess these policies in terms of their total benefit to a client over a session under these scenarios, while taking into account different configurations of the domain parameters described in Section 5. Figure 5(a) shows the average total benefit (y-axis) achieved by the pre-sending policies in terms of the number of requests in a session (x-axis) for the `no-memory/next-request` scenario, and Figure 5(b) displays these results for the `8-hours/cache` scenario (the data points on the x-axis are grouped as described for the results in Figure 4). The parameter configurations were chosen to enable a comparison between the reduction in waiting (which depends on cps/bps) and the cost of unnecessary pre-sending (which depends on cpb). This is achieved by fixing cpb and bps to 1 and 4000 respectively (4000 bps is a common transmission rate), and varying only cps from 400 (cps/bps = 1/10 cpb) to 40000 (cps/bps = 10 cpb). Each line in Figure 5 is labelled with a cps value and a tag that indicates the pre-sending policy (`d` for decision-theoretic and `n` for naive). For example, in Figure 5(a) for 66 requests, when cps = 40000, the average total benefit for the naive pre-sending policy is 834,117, compared to 881,802 using the decision-theoretic policy; when cps = 400, the corresponding average total benefits are -367,232 and -15,404 respectively.

We are interested in two inter-related factors: (1) the relative performance of the decision-theoretic and naive pre-sending policies, and (2) the impact of the domain parameters. For the `no-memory/next-request` scenario, the decision theoretic policy consistently outperforms the naive policy. For the `8-hours/cache` scenario, the naive policy performs better than the decision-theoretic policy when the relative cost of waiting becomes high enough (e.g., cps = 40000). This is because for this cost, the naive policy, which pre-sends a document after every request, sometimes achieves a large eventual reduction in waiting cost, which offsets its losses from its unnecessary transmissions. In contrast, the decision-theoretic policy, which is more conservative, does not always pre-send these large-payoff documents. For a lower

[3] To smooth the graph, each point on the x-axis represents a group of clients, such that the clients in each group have requested a similar number of documents. Each of the first nine groups consists of 10% of the clients, while each of the remaining ten groups has 1% of the clients. The x-value for each data point is the midpoint of the range of numbers of documents requested by the clients in a group. The final data point, x=6143, has been excluded from the graph in order to view the data more clearly; this still leaves 99% of the data.



(a) Immediate – `no-memory/next-request`.

(b) Eventual – `8-hours/cache`.

Figure 5: Effect of the pre-sending policy and domain parameter configuration on the average total benefit.

cost of waiting (e.g., $cps = 20000$), the two pre-sending policies give similar results. When the cost of waiting decreases further (e.g., $cps < 20000$), the decision-theoretic policy gives a greater average total benefit than the naive policy. In some cases (e.g., $cps = 4000$), the decision-theoretic policy gives a positive average total benefit, while the naive policy yields an overall negative benefit. For our lowest waiting cost ($cps = 400$), the decision-theoretic policy gives a small negative total benefit in both scenarios, compared to much larger negative total benefits for the naive policy for $cps < 10000$. Since our decision-theoretic policy does not pre-send when it computes a negative expected benefit, this overall small negative total benefit can be explained by the fact that our prediction model is only an approximation.

The effect of the pre-sending policy, the scenario and the domain parameters can also be seen in the average percentage of requests for which the system pre-sends a document. Under the `no-memory/next-request` scenario, the naive pre-sending policy pre-sends a document 99.5% of the time (it fails to pre-send only when the request was unseen in the training data). Under the `8-hours/cache` and ∞`/cache` scenarios, it pre-sends only 86.1% and 76.3% of the time respectively (nothing is pre-sent when all the candidates are already in the client's cache). The decision-theoretic policy pre-sends much less often than the naive policy, becoming more conservative as the importance of the waiting time decreases. For example, for cps=40000, documents are pre-sent 63.9% of the time for the `no-memory/next-request` scenario and 35.4% for the `8-hours/cache` scenario, dropping down to 10.6% and 2.6% respectively for cps = 400.

For the test data used to generate these results, the decision-theoretic pre-sending system makes a decision in about 33 milliseconds of CPU time on a SGI Indy R5000, compared to about 5 milliseconds for the naive pre-sending system (due to the extra time taken to compute the benefits). The off-line training time to build the four Markov models used by the hybrid prediction model is about 1 millisecond per request.

8 Conclusion

We have presented two systems for pre-sending documents on the WWW, one based on a decision-theoretic model, and another based on a naive approach. Both systems consult a Markov-based model which predicts the next document request. We have compared the performance of these systems using two evaluation methods, immediate benefit and eventual benefit, and considering several domain parameter configurations. Our evaluation shows that the decision-theoretic approach generally outperforms the naive approach, except when the penalty for waiting for a document is extremely high ($cps/bps = 10\,cpb$) and the evaluation is done using the eventual benefit method. In addition, it is better to use the decision-theoretic approach for pre-sending documents (rather than doing nothing) in all situations where the waiting time is relatively important to the user ($cps/bps > cpb$).

Acknowledgments

This research was supported in part by grant A49600323 from the Australian Research Council.

References

[Balabanović, 1998] Balabanović, M. (1998). Exploring versus exploiting when learning user models for text recommendation. *User Modeling and User-adapted Interaction*, 8(1-2):71–102.

[Bestavros, 1996] Bestavros, A. (1996). Speculative data dissemination and service to reduce server load, network traffic and service time in distributed information systems. In *Proceedings of the 1996 International Conference on Data Engineering*.

[Joachims et al., 1997] Joachims, T., Freitag, D., and Mitchell, T. (1997). WebWatcher: A tour guide for the World Wide Web. In *IJCAI97 – Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, pages 770–775, Nagoya, Japan.

[Lieberman, 1995] Lieberman, H. (1995). Letizia: An agent that assists web browsing. In *IJCAI95 – Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 924–929, Montreal, Canada.

[Maes and Kozierok, 1993] Maes, P. and Kozierok, R. (1993). Learning interface agents. In *AAAI-93 – Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 459–465, Washington D.C.

[Nicholson et al., 1998] Nicholson, A. E., Zukerman, I., and Albrecht, D. W. (1998). A decision-theoretic approach for pre-sending information on the WWW. In *PRICAI'98 – Proceedings of the Fifth Pacific Rim International Conference on Artificial Intelligence*, pages 575–586, Singapore.

[Zukerman et al., 1999] Zukerman, I., Albrecht, D., and Nicholson, A. (1999). Predicting users' requests on the WWW. In *UM99 – Proceedings of the Seventh International Conference on User Modeling*.