D-006

不特定ユーザを対象とするAjaxウェブサイトのための コンテンツ先読み手法

楢崎修二† 田代純規‡ 野口翔† 平山陽†
Shuji Narazaki Junki Tashiro Sho Noguchi Akira Hirayama

1 はじめに

Webによる情報収集が日常生活で重要な位置を占めるにつれ,「使いやすさ」を意味するユーザビリティはWebサイトの設計にあたって重要な要素となっている．ユーザビリティを向上させるために様々な研究が行われているが，中でもAjax(Asynchronous JavaScript + XML)という技術が最近注目を集めている．この技術によりWebページ中の一部を書き換えたい場合には，その部分のデータのみを要求すればよく，通常のサイトよりも双方向性の高いWebページを作成することが可能になった．

しかしAjaxを用いたWebサイトにおいてもデータ取得はユーザのリクエスト(リンクのクリックなど)を契機としてWebサーバとの通信を始めることは変わらず，通信遅延によるユーザの待ち時間の存在は避けられない．

通信遅延を解決する手法の一つは先読みの実現である．JavaScriptを使えば，遷移確率の高いコンテンツを先に読むことによりユーザの体感速度を向上させることができる．しかし，サーバに多大な負荷を掛けないためには精度の高い先読みの実現が必要となる．Web推薦システムに関する研究 [2, 4] ではユーザの嗜好や行動を予測するために，クライアント上でのユーザの行動履歴やサーバのアクセスログなどが利用されるが，先読みを対象とした研究は少ない．

そこで，本論文ではユーザの行動履歴に基づく精度のよい先読み手法を提案し，評価する．

2 提案手法

提案手法はAjaxを使ったサイトにおいて，更新時刻とアクセスパターン，アクティブセッションとからユーザのアクセスを予測し，コンテンツ先読みを行うというものである．

ここでの更新時刻とは，Webサイト内の各コンテンツが更新された時刻情報を指し，アクセスパターンとはサーバに残ったアクセスログにおいてよく出現するアクセスの流れをパターンとして抽出した情報を指す．

アクセスパターンはその情報をアクセスログから求める処理が必要であるため頻繁に求めるとサーバの負担になってしまう．そのため更新時間とアクセスパターンとの両方を用いることで，更新時刻からは発見しにくいユーザのアクセス行動の偏りをアクセスパターンにより見つけ，アクセスパターンからは発見しにくい新しいコンテンツを更新時刻により見つけることが可能になると考える．

アクセスパターンからのアクセス予測では，アクセスパターンとアクティブセッションとのマッチングを行う．しかし通常アクセスログから求めるアクセスパターンはサーバ側で生成され，アクティブセッションはブラウザ側で行われるユーザのアクセスとともに変化するためマッチングには工夫が必要となる．

しかしAjaxを用いたWebサイトではページ全体の遷移が発生しないという利点から，ユーザが最初にサイトを訪問してからの行動履歴を全て記録し先読みにおいて利用することができるため，精度のよい推定が容易に行える．

なお，提案手法においてアクセスパターンはCookieで保存される一時的なIDやクライアントのIPアドレスなどによってユーザが区別されるアクセスシーケンスによって用いて作られるためユーザの特定は必要としない．アクティブパターンも永続的なID情報を基にするのではなく，単に訪問したユーザの現在までの履歴情報であるため，ユーザの同定は不要であるため，SNSなどだけではなくより広い範囲のwebサイトに適用可能である．

† 長崎大学工学部
‡ 熊本計算センター

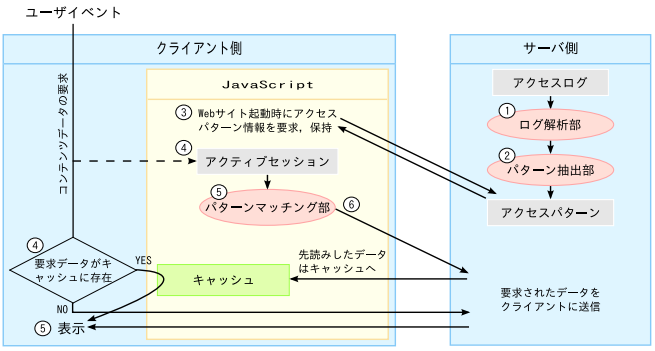


図 1: アクセスパターンを用いた先読みの概要図

2.1 提案システム

先読みを行う上で重要であることは，どのタイミングでどのくらいのコンテンツを先読みするかということである．先読みはユーザがコンテンツを閲覧中に行うのが好ましいと考え，今回先読みを行うタイミングはユーザが Ajax サイトを訪れた直後とユーザがあるコンテンツを要求した直後としている．このとき例えば一度に一つのコンテンツしか先読みしないのであればヒット率 (ユーザが訪問したコンテンツ中でそのコンテンツが先読みされたコンテンツであった割合) は低くなると予想され，逆に大量の先読みを行えば後でアクセスされる可能性を加味してもアクセス回数が少なかった場合に適合率 (コンテンツ先読みを行ったコンテンツ中で実際に訪問されたコンテンツの割合) は下がってしまうと予想される．そのため一度に先読みを行う最大コンテンツ数 $|p|$ というパラメータを設けることにする．

またコンテンツへのアクセスがあるたびに先読みを行ってしまうと適合率を下げてしまい Web サーバへも負担になると考え，一ユーザの Ajax サイト訪問につき先読みを行う最大コンテンツ数 $|P|$ というパラメータも設け，実験により値を設定することにした．

2.2 アクセスパターンを用いた先読み

アクセスパターンを用いた先読みの概要を図 1 に示す．この手法では最初にサーバ側でアクセスパターンを求めるところから始まる．図 1 のようにサーバ側では，アクセスログのデータをログ解析部とパターン抽出部とで処理することでアクセスパターンの情報を求めることができ，サーバではこの情報を保持する．一方，ユーザが Ajax サイトを訪れると，クライアントに相当する Web ブラウザ上で動く JavaScript がサーバにアクセスパターンの情報を要求し，それを保持する．

その後ユーザがコンテンツへのアクセスを行うたびに，JavaScript がユーザのアクセス順序を表すアクティブセッションに記録する．そしてアクティブセッションと保持しているアクセスパターンの情報とをパターンマッチング部で処理することで先読みすべきコンテンツを決定し，非同期通信によりサーバにリクエストを発行する．

以下では図 1 中の各処理部についての説明を行う．

2.2.1 ログ解析

ログ解析部では Web サーバのアクセスログから，不要なデータを捨てるなどの前処理を行ない，各ユーザのアクセス履歴を表わすアクセスシーケンスを求める．

ユーザごとのアクセスシーケンスを求めるためにはアクセスログに含まれるセッション ID 情報を用いる．このセッション ID とはサーバが与え，同じユーザでもログインごとに異なるランダムな値であり，ユーザを特定する必要はない．

2.2.2 パターン抽出

パターン抽出部ではログ解析部で求めた各アクセスシーケンス同士をマッチングさせ，出現頻度の高い部分アクセスシーケンスを求める．今回は Wu らが Web 推薦システムのために提案した手法 [3] を用いることにした．

この手法は Web ページそれぞれを文字と見做して二つのアクセスシーケンス間の編集距離を求めるものである．最小の編集距離を与える系列 (Shortest Edit Distance; SED) は二つのアクセスシーケンスに現われる Web ページを全て含むものとなる．

ログ中の全てのアクセスシーケンス対に対して SED を求め，出現頻度が高いものを抽出する．抽出した SED は共通の根をまとめることで木として表現することができる．Web サーバはクライアントである Web ブラウザ上の JavaScript からリクエストに対してこの情報をユーザによらない Web ページ間の遷移状況を現わす XML データとして返す．

2.2.3 パターンマッチング

パターンマッチング部では上のパターン抽出部で求めたアクセスパターンデータとアクティブセッションとの比較により先読みすべきコンテンツの決定を行なう．

木構造化された SED とアクティブセッションとのマッチングでは木構造の探索を行うことになるが，マッチン

グにおいて SED とアクティブセッションは完全に一致する必要はないため木の全探索を行うことになる．

パターンマッチングには 2 章で触れた [3] で使われている手法を用いることにした．図 2 で説明する．

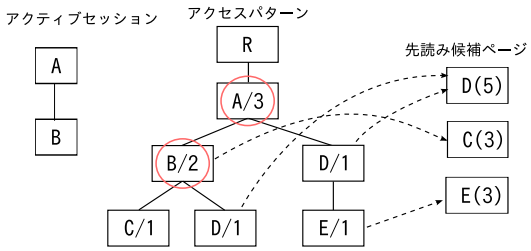


図 2: マッチングの例

図 2 ではユーザが A→B の順にアクセスを行った場合のマッチングの様子を表している．

まずアクセスパターンとアクティブセッションとの間で共通するページを見つける．ここでは丸で囲んだ A と B が共通ページとなる．そしてルートから共通部分最後のページまでを除去し，残ったページが先読み候補ページとなる．

このままでは候補ページが多く出現するため各候補ページに重みを付ける (括弧内の数値)．重み Weight(W) は候補ページの SW(図中の数字) とルートから候補ページまでの共通ページ数 (Number of Match(NM)) とにより決定する．木探索において複数の枝から同じ先読み候補ページが挙がった場合はその重み同士を足し合わせる．具体的な計算は SW と NM とを足し合わせた式

$$W = SW + \alpha \cdot NM \tag{1}$$

で行う．α は W に対する NM の影響度合を表すパラメータである．読み候補は重みでソートされ，上位 $|p|$ 個までを先読み対象とする．

2.3 更新時刻を用いた先読み

更新時刻を用いた先読み手法は，Ajax サイト内に存在する各コンテンツの更新時刻を基にコンテンツ先読みを行う手法である．実験に使用する授業用 Ajax サイトも含め，多くの Ajax サイトではトップページ内や各カテゴリー内のコンテンツはともに更新時刻の新しい順で並ぶためこの手法を用いる．

この方針の妥当性を検証するため，今回実験を行う Ajax サイト内コンテンツの更新時刻とそのアクセス回数との関係性について予備実験を行った．実験では Ajax サイトのトップページ，各カテゴリー内においてユーザが更新時刻順序で何番目のコンテンツにアクセスしたかを記録した．結果は図 3 に示す通りである．図に示すように，更新時刻の新しいコンテンツがアクセスの多くを占めていることが分かる．特に上位 2 個までのコンテンツで全体のアクセスの 40%近くを占めていた．これよりサーバ負荷と効率的な先読みとを両立するため，最大で上位 (最新)2 個程度のコンテンツを先読み対象とする．

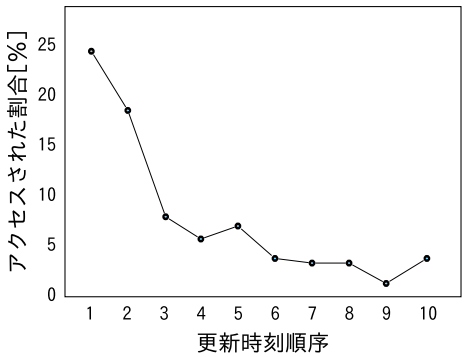


図 3: 更新時刻とアクセス回数の関係図

2.4 更新時刻とアクセスパターンを用いた先読み

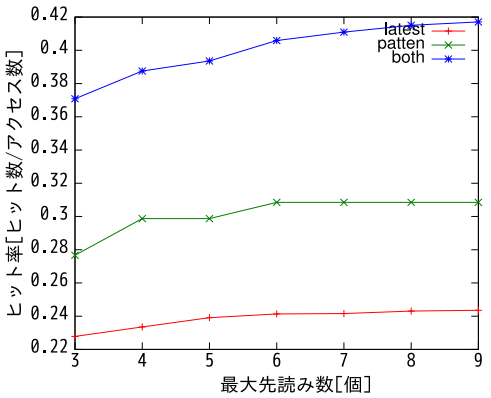
更新時刻とアクセスパターンを用いた先読みは上の二つの手法を組み合わせた手法であり，各手法で決定した先読み候補ページを先読みする．

この手法ではまずアクセスパターンを用いた先読み手法により先読みを試みる．その時一度の先読みタイミングで先読みを行う最大コンテンツ数 $|p|$ 以上の先読み候補ページが挙がった場合は，候補ページの重み上位 $|p|$ 個のみ先読みを行う．しかし $|p|$ 個未満であった場合には足りない分を更新時刻を用いた先読み手法によって先読みページを選出する．

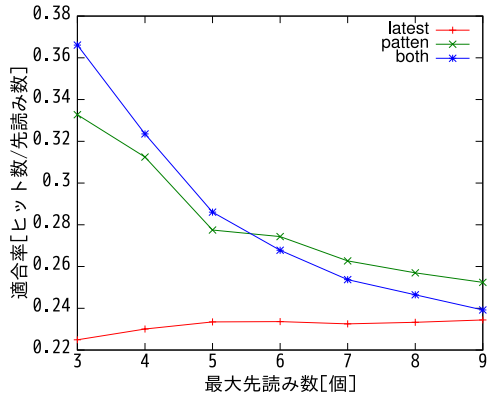
これにより，ユーザがパターンに現れていない新しいページにアクセスを行う場合や，十分なユーザプロファイルが蓄積できていない場合でもある程度の精度での先読みが実現できると考えられる．

3 実験と評価

前章で提案した手法の有用性を検証するために当研究室で運用している授業用資料配布サイトにおいて提案システムの実験を行った．対象システムは我々が提案・開発を行なっている，Web コンテンツを P2P 型の複製サーバ群で保有し JavaScript で負荷に応じて動的なサーバ選



(a) ヒット率



(b) 適合率

図 4: 最大先読み数 $|P|$ に対する評価

択を実現するシステム [1] であり単一サーバよりも多くのリクエストに応えることが可能である．

実験は曜日による影響を除くため 2008 年 1 月 28 日～2 月 2 日の一週間に行い，前処理後の有効セッション ID 数は 107 であった．なお，この実験において先読み候補ページの重み付けのパラメータである α は予備実験により適当と判断した 10 と設定している．また先読みタイミング 1 回に対して先読みを行う最大コンテンツ数 $|p|$ は 2 とし，一ユーザの Ajax サイト訪問に対して先読みする最大コンテンツ数 $|P|$ を変化させて実験を行った．

図 4 において latest は更新時刻を用いた先読み手法，pattern はアクセスパターンを用いた先読み手法，both は更新時刻とアクセスパターンを用いた先読み手法を示す．またヒット率・適合率はともに全ユーザのアクセスにおける平均の値である．

図 4(a) に表われているように，ヒット率に関して提案手法である更新時刻とアクセスパターンを用いた先読み手法 (both) は他の二つの手法に比べて非常によい結果が得られた．これは更新時刻情報の持つアクセスしそうなコンテンツ情報と，アクセスパターンの持つカテゴリー移動情報とのどちらも考慮できたことによる結果と思われる．一方いずれの手法でも，最大先読み数に対してするヒット率の上昇はゆるやかである．

適合率に関しては図 4(b) から分かるように，アクセスパターンを用いる pattern と both とは最大先読み数が増えるにつれて適合率は減少している．これは最大先読み数が増えたことであまり効果的でないパターンからも先読みを行うようになったことが原因と考えられる．ヒット率とのバランスを考慮する必要があることがわかる．

以上より，この実験においては，サーバに対するリクエスト数 (負荷) が 3 倍になるが約 4 割の先読みのヒット率を実現できる $|P| = 3 \sim 4$ がもっともよいと判断した．半数近いアクセスを隠蔽することができるため，提案手法には十分な効果があると考えられる．

4 まとめ

ページ全体の遷移が発生しないという Ajax の特性を活かしながら，単純に新しいコンテンツを先読み，もしくはパターンから予測したコンテンツを先読みする手法に比べてヒット率と適合率との両方を改善する先読み手法を提案した．今後は先読み候補ページの重み付け方法，更新時刻とアクセスパターンとの優先順位決定方法などの改善を行なう予定である．

参考文献

[1] 楢崎修二, 田代純規. Web ブラウザ上で動的サーバ選択を行なう JavaScript エージェント．合同エージェントワークショップ&シンポジウム JAWS2007, ウェブエージェント (2)-4, 2007.

[2] 大塚真吾, 喜連川優. Web アクセスログとその利活用. 人工知能学会誌, Vol. 21, No. 4, pp. 410–415, 2006.

[3] Sun Wu, Udi Manber, and Gene Myers. An $O(NP)$ sequence comparison algorithm. *Information Processing Letters*, Vol. 35, No. 6, pp. 317–323, sep 1990.

[4] 山元理絵, 小林大, 小林隆志, 横田治夫. Web アクセスログの LCS を用いた Web ページの推薦手法 (履歴応用). 電子情報通信学会技術研究報告．DE，データ工学, Vol. 106, No. 148, pp. 109–114, Jul 2006.