3P－7

データマイニングを用いたウェブ設計支援*

小川浩司　溝口文雄†

東京理科大学 理工学部‡

1 はじめに

ウェブ利用者の増加と共に、情報発信やサービスの提供などあらゆるものがウェブベースのシステムに移行してきている。ウェブは、企業、大学、政府等多くの組織にとって重要なものとなった。このような状況のなかで、ウェブサイトの適切な分析を行い、ウェブの構築をいかに効果的なものにするかが問題となっている。

そこで、本論文は、ウェブサイトのアクセスログからサイト内におけるユーザの振る舞いや、その頻繁なパターンを発見し [2]、それをウェブ設計者に提供するウェブ設計支援システムを提案する。

2 既存の WWW 分析ツール

既存の WWW 分析ツールとしては、analog や wwwstat などが有名である。しかし、これらには次のような問題がある。

まず、これらのツールはアクセス情報などページ単位の分析しか行うことができない点である。分析は、アクセスログの単純なカウントによるものである。しかし、サイト内でのユーザはリンクからリンクへと連続的な動きをするのである。単に通過しただけのページなのか、本当に興味あるページなのか、カウント情報だけでは適切な分析は困難である。

つぎに、これらのツールはユーザレベルの分析ができない点である。分析は、IP もしくはドメインレベルで行われる。しかし、このレベルではユーザがプロキシサーバやゲートウェイを経由してサイトにアクセスしてきたような場合に対処できない。このような場合、アクセスログにプロキシサーバーやゲートウェイ IP が記録され、同時にすべて同じ訪問者と認識されてしまう。

3 ユーザレベルのセッション認識

本研究では、ユーザ個々のセッションを認識し、ページ単位ではなくサイト内でのユーザの行動を分析の対象とする。ユーザレベルのセッション認識は、cookie を利用することで可能にする。cookie とは、ウェブサーバが訪問者のハードディスク上に保存する小さな情報のことをいう。ブラウザを通して、cookie を保存したり、読み出したり、消去したりすることが出来る。本研究では、訪問者がサーバにアクセスしたときに、この cookie を利用してサーバ側からクライアント (ユーザ) 側に ID を発行する。もし、以前にそのサイトにアクセスしたことがあるならば、ID を発行せずクライアント側に保存されている cookie を使用する。この発行した ID は、ユーザのセッション情報とともにログファイルに記録される。

4 マイニングプロセス

データから知識を発見するためには、複数のステップを踏む必要がある。ここでは、サイト内でのユーザの振る舞い (Traversal Path) とその中で頻繁にユーザが通過する部分パス (Frequent Path) を発見していく。順次、それらのプロセスを述べていくことにする。

4.1 データの前処理

4.1.1 フィルタリング

画像ファイルやアプレットに関するレコード、リロードによる重複レコード、そして失敗セッションのレコードを元のアクセスログから削除する。

4.1.2 フォーマット変換

日付: [日、月、年、時、分、秒] のフォーマットで記録されている日付を"1970 年 1 月 1 日 00:00:00 GMT からのミリ秒数" を計算した整数フォーマットに変換する。これは、セッションの時間などの計算を行いやすくするためである。

URL、フレーム: フィールドによって異なる URL フォーマットを同一のフォーマットに変換する。また、フレームによって複数のページから構成されるページ名を一つのページ名にする。

前処理されたレコードは以下のようになる。左から順に「クライアント cookieID」「日付 (GMT)」「参照元 URL」「参照先 URL」を示している。

*The support system of Web design using datamining
†Koji OGAWA, Fumio MIZOGUCHI
‡Faculty of Sci. and Tech. Science University of Tokyo

133.19.25.67.11955910497346526 910497363402
Thesis/index.html Thesis/Graduate/index.html

4.2 Traversal Path の発見

Traversal Path というのは、ユーザのサイト内での振る舞い、つまり、移動の軌跡のことをいう。[2] サーバはユーザがページからページへ移動するたびに、「参照元 URL」と「参照先 URL」を記録する。しかし、すべてのユーザのセッションが記録されるわけではない。ブラウザやプロキシサーバーによってユーザが一度通過したパスはキャッシュされてしまう。よって、アクセスログにはユーザの後方への記録は存在しない。よって、完全な Traversal Path を形成するには、欠けた後方のパスを補間する必要がある。例えば、$(- \rightarrow a),(a \rightarrow b),(b \rightarrow c),(\mathbf{c} \rightarrow \mathbf{b}),(\mathbf{b} \rightarrow \mathbf{a})$, と行動した場合でも、ログには、$(- \rightarrow a),(a \rightarrow b),(b \rightarrow c)$ しか記録されない。

4.3 Frequent Path の発見

Frequent Path とは、Traversal Path のうち前方に進んだ分のパス (Maximum Forward Path) だけに注目し、さらにそのパスから抽出した頻繁な部分パスのことである。その中でも特に、ユーザが指定した最小サポート値を満たすパスは LargePath(LP) と呼ばれる。[2] ここで、サポート値とは FrequentPath を含む MaximumPath の数のことである。LargePath は、アプリオリアルゴリズム [1] を利用して求めることができる。以下にそのアルゴリズムを示す。

```
00:LP_1 = size1- largepaths
01:for(k = 2;LP_{k-1} != 0; k++){
02: CLP_k; //new Candidate LP
03:for each F_i //a set of MaximumForwardPath in Session,
04:   for each {x_1, x_2, ...x_m} in F_i{
05:     if(m >= k){
06:       for(j = 1;j < m-k+1;j++){
07:       subpath = {x_j,...,x_{j+k-1}};
08:         if( subpath ∈ CLP_k)
09:         subpath.count++;
10:         else if({x_j,...,x_{j+k-2}} ∈ LP_{k-1} )and
11:              ({x_{j+1},...,x_{j+k-1}} ∈ LP_{k-1} ))
12:              add subpath to CLP_k;
13:              subpath.count++;
14:       }
15:     }
16:   }
17: }
18: LP_k={subpath ∈ CLP_k | subpath.count > minimum support}
19:}
```

4.4 適用事例

ここで、実際に適用した事例を紹介する。対象としたのは、本研究室のウェブサイト[1] における 1998.10.30〜1998.12.27 のアクセスログである。ここでは、サポート値が 0.1%でパスの長さが 4 の時の LargePath を示す。

[1]http://mizo-www.ia.noda.sut.ac.jp/

表 1: 長さが 4 の時の LargePath(LP_4)

順位	サポート数	LargePath
1	32	/index.html;/Thesis/index.html; /Thesis/JConference/index.html; /Thesis/JConference/ipsj.html
2	29	-;/index.html;/Thesis/index.html; /Thesis/JConference/index.html
3	27	http://iaws-20.ia.noda.sut.ac.jp; /index.html;/People/index.html; /People/98/master.html

5 設計支援システム

設計支援システムは、先のマイニングプロセスを実行し、発見した知識を視覚化しウェブ設計者にレポーティングする. 本研究では、従来のグラフやテーブルによる表示だけではなく, サイト全体のページとリンク構造を視覚化し、発見されたパスを頻度毎に色分けする等、特別に表示することで設計者を支援する。図 1 は実行例であり、発見されたパスは色分けされその各ページ名が表示されている。

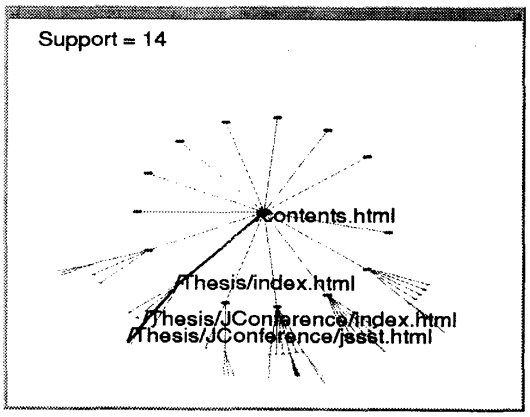

図 1: レポーティング

6 おわりに

ウェブサイト内でのユーザの振る舞いと、頻繁なパターンを発見し、レポートする支援システムを提案した。ウェブ設計者は、サイト内におけるユーザの一連の行動を理解することでページの更新や構成変更を効果的に行うことが可能となる。

参考文献

[1] R.Agrawal and R.Srikant. Mining sequential patterns . Research Report RJ 9910,IBM Almanden Research Center,San Jose,California, October 1994

[2] k.-L.Wu,P.S.Yu,A.Ballman SpeedTracer.A Web usage mining and analysis tool IBM Systems Journal Reprint Order No.G321-5665