頻出系列パターンマイニング手法を用いた Web 利用パターン発見に関する研究
和田・犬塚 研究室　早川 潤一
No. 17417574　Hayakawa Junichi

1　序論

本研究は，Web ブラウジングにおける閲覧者のアクセスログを収集し，閲覧者の Web アクセスパターンに対する頻出パターンの抽出を行う．**(1)Web アクセスログに関する複数の種類の情報と系列の形をまとめる統一的方法の提案**，また，アクセスログの性質を検討し，定義した系列データに対してこれらを扱えるように拡張した **(2) 頻出系列マイニングアルゴリズムの一般的拡張の提案**を行う．

Web マイニングは，Web ページのコンテンツに注目する **Web 内容マイニング**，Web ページ間を結ぶハイパーリンクのグラフ構造に注目する **Web 構造マイニング**，Web ページのアクセスログデータに注目する **Web 利用マイニング**の 3 つに分類される [2]．

本研究では，Web アクセスログを用いて（Web 利用マイニング），クリックによる HTML のハイパーリンク構造をたどって得られる情報（Web 構造マイニング），HTML ソースを解析して得られる Web ページ内部の情報（Web 内容マイニング）を統一的に扱うことで，各分野を融合したマイニングを提案する．

2　Web アクセスログからの情報抽出

本研究では上述の 3 つに分類された Web 情報を系列形式に変換し，頻出系列問題に定式化する．Web 上にあるドキュメントをノード，ハイパーリンクを枝として作られるドキュメントのグラフの中で，一つのユーザのクリック系列に現れる部分グラフを考える．このとき，クリックストリームの初期ページから，この部分グラフの葉接点までを一つのドキュメント系列とみなせる．この系列にはすでに Web 構造情報，及び Web 利用情報が含まれており，さらに系列を構成する各ドキュメントにそのドキュメントの属性を付加することで Web 内容情報を加味する．また，この部分グラフ上で，クリックストリームが重複するドキュメントは分岐点である，などの特徴を系列に含めて考える．

2.1　系列データの統一的形式

本研究で扱う系列データに対して，Web アクセスログに関する複数の情報を統一的に扱える方法を提案する．

Web アクセスログをログレコードの集合とする．すべてのログレコードの ID の列を R とすると，$R = \{1, 2, 3, \cdots, n\}$ と表せる．ログレコードはアクセスログへのエントリを表し，エントリ項目をログアイテムと呼ぶ．ログアイテムのエントリ項目を属性とみなすこととする．例えば，$i \in R$ のエントリ項目 *remotehost* が atelier.elcom.nitech.ac.jp であるとき，属性値 $remotehost(i) = atelier.elcom.nitech.ac.jp$ と表記する．

また，属性（エントリ項目）a の取り得る属性値の集合を v_a と書くことにする．

次に，クリックストリーム抽出の手順を説明する．あるユーザ（=*remotehost*）x に関するログレコード ID の列 R_x は，以下の通り．

$$R_x = \{i \in R \mid remotehost(i) = x,\ i = 1, 2, \cdots, n\}$$

この R_x を用いて，ノード R_x，エッジ E_x となる有向グラフ $G_x : (R_x, E_x)$ を考える．ここで，E_x は，

$$E_x = \{(i, j) \in R_x^2 \mid (referer(i) = request(j)) \wedge (referer(k) \neq request(j)),\ i < k < j\}$$

となる辺である．*referer(n)* は n 番目のログレコードのリンク元 URL，*request(i)* はリクエスト URL を表す．有向グラフ $G_x : (R_x, E_x)$ のノード i は，グラフの構造から例えば以下のような属性をもつ（例：ルートなるノード）．

$$Initial(i) = \begin{cases} True & : \text{リンク元 } URL \text{ が } NULL \\ False & : \text{上記以外} \end{cases}$$

グラフの属性 b の取り得る属性値の集合を v_b と書くことにする．

request の属性値，つまりリクエスト URL に注目する．リクエストに応じてサーバから送信されてきた HTML ソースから HTML タグを抽出し，属性とする．例えば以下のような値をもつ．

$$table(i) = \begin{cases} True & : table \text{ タグをもつ} \\ False & : table \text{ タグをもたない} \end{cases}$$

抽出した HTML タグの属性 c の取り得る属性値の集合を v_c と書くことにする．

以上から，本研究で扱うアイテム集合 I は以下のように表すことができる．

$$I = v_{a_1} \oplus \cdots \oplus v_{a_n} \oplus v_{b_1} \oplus \cdots \oplus v_{b_m} \oplus v_{c_1} \oplus \cdots \oplus v_{c_l}$$

3　頻出系列パターン抽出アルゴリズム

I を，**アイテム**集合とする．その要素，つまりアイテムは i_1, i_2 等と表記する．アイテム集合の部分集合 $e \subseteq I$ を**エレメント**といい，$(i_1 i_2 \cdots i_m)$ と表記する．もしエレメントが，一つのアイテムだけで構成されるなら，“()” は省略して表記する．

系列とはエレメントの列であり，$\langle e_1 e_2 \cdots e_l \rangle$ と表記する．系列の長さとは，系列中のアイテムの延べ数である．また，ある系列 s の長さは，$len(s)$ と表記する．

系列データベースD とは，タプル $\langle sid, s \rangle$ の集合である．sid は系列識別子，s は系列である．系列 α の系列データベース D における**サポート（支持度）**とは，D 中のすべての系列のうち，系列 α を部分系列とする系列 s を含むタプルの数であり，以下のように定義される．

$$support_D(\alpha) = |\{\langle sid, s \rangle \in D \mid \alpha \sqsubseteq s\}|.$$

頻出系列マイニング問題とは，サポート（支持度）がユーザが定義する最小閾値以上出現する系列をすべて抽出するタスクである．

3.1　PrefixSpan

頻出系列マイニング問題において，有効な手法であると考えられている PrefixSpan[1] について説明する．PrefixSpan は深さ優先探索で短い頻出系列からより長い頻出系列へと系列を成長させていく射影を繰り返すアルゴリズムである．

定義１（射影データベース）系列データベース D に対し，系列 α により射影して作成するデータベースを α-射影データベース $D|_\alpha$ という．

$$D|_\alpha = \{\langle sid, s\rangle \in D \mid \alpha \sqsubseteq s\}$$

3.2 パターンによる系列マイニングの一般化

従来法において判別できない系列のパターンを**系列パターン**を定義することで判別可能にした手法を提案する．

定義２（系列パターン）パターンは，エレメント記号と区切り記号から成り，

$$\epsilon_1\sigma_1\epsilon_2\sigma_2\cdots\sigma_{n-1}\epsilon_n$$

の形式で表す．$\epsilon_1,\cdots,\epsilon_n$ はエレメント記号，$\sigma_1,\cdots,\sigma_{n-1}$ は区切り記号を示す．

エレメント記号 ϵ とエレメント e について，ϵ が e を表しているかどうかが決定できるとき，$\epsilon \prec e$ と表記することにする．

区切り記号 σ は添字間の関係を表しており，添字 i と j が σ の表す関係にあることを $i\sigma j$ と表記する．このとき，パターン $p = \epsilon_1\sigma_1\epsilon_2\sigma_2\cdots\epsilon_{n-1}\sigma_n$ が系列 $\alpha = e_1,\cdots,e_m$ を表す $\epsilon \prec e$ とは，添字の列 i_1, i_2, $\cdots$, i_n があり，

$$\begin{cases} \epsilon_k \prec e_{i_k} & (k = 1,\cdots,n) \\ i_k\sigma_k i_{k+1} & (k = 1,\cdots,n-1) \end{cases}$$

を満たすことである．例えば，エレメント記号はエレメントと同様に，アイテムの集合として，$\prec$ として $\subseteq$ を用いる．つまり，$\epsilon \prec e$ は，$\epsilon \subseteq e$ とする．区切り記号 σ をいつも，$i\sigma j$ は $i < j$ と考えると，従来の PrefixSpan と同様であるといえる．また，区切り記号 σ を”,” と”-” として考えると，

σ =”,”のとき　　$i\sigma j$ つまり, i, j は $i < j$
σ =”-”のとき　　$i\sigma j$ つまり, i-j は $i + 1 = j$

となり，エレメントの直後に現れるかどうかを判別することができる..

疑似コードを以下に示す．

Input: 系列データベース D，サポート閾値 min_supp
Output: すべての頻出系列パターンの集合

call $PrefixSpan(\langle\rangle, D)$
procedure $PrefixSpan(\alpha, D|_\alpha)$
　$F \leftarrow \{\sigma\epsilon \prec \alpha \mid (support_{D|_\alpha}(\sigma\epsilon) \geq min\_supp)\}$
　foreach $\sigma\epsilon \in$ (区切り記号, F)
　　$\alpha\sigma\epsilon, support_{D|_\alpha}(\alpha\sigma\epsilon)$ を出力
　　$D|_{\alpha\sigma\epsilon} \leftarrow \{\beta \in D|_\alpha \mid \alpha\sigma\epsilon \prec \beta\}$　// 射影データベースを作成
　　call $PrefixSpan(\alpha\sigma\epsilon, D|_{\alpha\sigma\epsilon})$

3.3 アイテム間の決定性を考慮したマイニング

アイテム間の決定的関係に着目し，より効率よく頻出系列を抽出できるよう拡張した手法を提案する．

定義３（決定的関係）任意の系列に対して，アイテム A を含むエレメントは必ずアイテム B を含むとき，A は B を決定する．

つまり，系列 α の末尾のエレメント内のあるアイテムと決定的なアイテムを付加した系列 α' に関して，$D|_\alpha = D|_{\alpha'}$，もしくは $D|_\alpha = \phi$ となる．

4 実験

KDnuggets（http://www.kdnuggets.com）のサーバログを用いて，PrefixSpan と提案した手法との比較を行った．提案手法の方がより多くのパターンを抽出していることを確かめた．図１は，最小サポート値ごとの頻出パターン数の推移を示す．

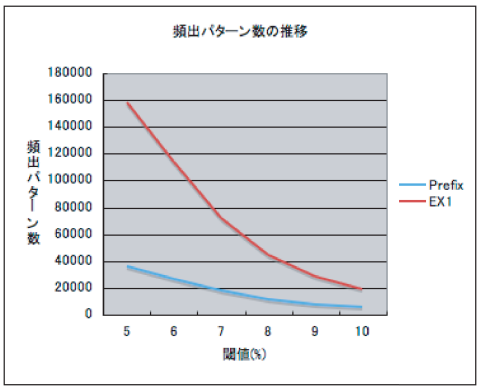

図１：頻出パターン数の推移

5 結論

本研究では，1)Web アクセスログに関する複数情報を統一的に系列データとして扱う方法を提案し，2)アクセスログの性質を検討し，定義した系列データに対してこれらを扱えるように拡張した頻出系列マイニングアルゴリズムの一般的拡張を提案，実験を行い有用性を示した．今後の課題として，提案したアルゴリズムをシステムとして運用し，評価を行うことが挙げられる．

参考文献

[1] J. Han, J. Pei, B. Mortazavi-Asl, Q. Chen, U. Dayal, and M. Hsu. Mining Sequential Patterns by Pattern-Growth: The Prefixspan Approach. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 16, No. 11, pp. 1424–1440, 2004.

[2] Kosala and Blockeel. Web mining research: A survey. *SIGKDD: SIGKDD Explorations: Newsletter of the Special Interest Group (SIG) on Knowledge Discovery & Data Mining, ACM*, Vol. 2, , 2000.

発表論文

- 早川 潤一, 中野 智文, 犬塚 信博, ” 頻出系列パターンマイニング手法を用いた Web 利用パターン発見”, 人工知能学会　データマイニングと統計数理研究会,2007.（発表予定）