社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2006－DBS－138（17）
2006－GN－58（17）
2006／1／27

シーケンス間の効率的類似度算出方法と Web アクセスログへの適用

赤塚厚司[†]　鈴木　優[††]　川越恭二[††]

近年，Web サイトの構築にあたって利用者の利便性を向上させるために，Web サーバに蓄積された Web アクセスログから，利用者のアクセス動向の調査が行われている．そこで本稿では，疎なベクトル間の高速な計算が可能であることに着目し，Web アクセスログ間の類似度を高速に算出する手法を提案する．従来，Web アクセスログ間の類似度算出のための手法では，主に動的計画法が用いられてきたが，膨大な計算量が問題となっていた．一方提案手法では，Web アクセスログ間の類似度算出のために，Web アクセスログを疎なベクトルで表現し，疎なベクトル間の類似度を高速に計算する手法を用いることによって計算量の削減を行う．提案手法の有効性を確認するために，実際の Web アクセスログに対して，動的計画法を用いた従来手法と提案手法の Web アクセスログ間の類似度算出にかかる処理時間と算出された類似度の相関関係を調べる評価実験を行った．その結果，提案手法は従来手法と同等の精度を維持しながら，処理時間が削減されたことが示された．

A Method of Calculating Similarity Values for Web access logs

Atsushi Akatsuka ,[†] Yu Suzuki [††] and Kyoji Kawagoe[††]

Recently, Web access logs are widely analyzed by Web administrators to build user-friendly Web documents. In this paper, we propose a method for analyzing Web access logs using a method of calculating similarities of sparse vectors. In the proposed method in the past, the dynamic programming is widely used to calculate similarities of Web access logs. However, a calculation cost of dynamic programming method is high. Therefore, the Web administrators cannot input numbers of Web access logs into analyzers because of much calculation time. In this paper, we use a calculation method based on similarities of vectors, instead of dynamic programming method, to reduce the calculation costs of analyze of Web access logs. In our experiments, we confirmed that our proposed method can reduce the calculation costs from the method using dynamic programming method, where the outputs of these two methods are almost the same.

1. はじめに

近年，インターネットの急速な発展により，Web サイトの利用者は急速に増加している．これに伴い Web サイトの管理者にとって，Web サイトにアクセスする利用者の動向を知ることは，利用者の利便性を考えた Web サイトを構築するための重要な要素である．ある利用者が大量の利用者全体の中で，どのようなアクセスを行った利用者であるかを相対的に判断するためには，全ての利用者間のアクセスの比較を行う必要がある．そこで，全ての利用者間のアクセスの比較を行う方法として，利用者間の類似度を定義する方法が考えられる．

Web サーバに蓄積されている Web アクセスログは利用者の識別情報，その利用者がアクセスした Web ページであるアクセスページ，Web ページにアクセスしたアクセス時刻によって構成されている．このため，Web アクセスログからどの利用者がどのような順番で，どのような Web ページにアクセスしたかを把握することができ，これらの情報から利用者を比較することができる．ここで，Web アクセスログによって各利用者を表現するための方法として，利用者ごとのアクセス時刻順に並んだアクセスページを用い，これをシーケンスと呼ぶ．そして，利用者のアクセスページによって構成されるシーケンス間の類似度算出により，利用者間の比較を行うことができる．また，算出されたシーケンス間の類似度から階層的クラスタ分析[1),2)] を用いデンドログラムを作成することができる．これにより，全ての利用者間の類似性を判断することができ，特異な利用者の発見，および利用者の分類を行うことができる．

上記のように特異な利用者の発見，および利

† 立命館大学大学院 理工学研究科
Graduate School of Science and Engineering, Ritsumeikan Univ.
†† 立命館大学 情報理工学部
College of Information Science and Engineering, Ritsumeikan Univ.

用者の分類を行う場合，シーケンス間の類似度の算出方法が重要となる．シーケンス間の類似度の算出方法が異なると階層的クラスタ分析の結果および，全体の処理速度に大きな影響を与える．

シーケンス間の類似度を算出するための従来手法として，Pirjo Moen が提案する手法[3] や，A. Banerjee らの提案する手法[4] が存在する．しかし，これらの従来手法ではシーケンス間の類似度を算出する際に，動的計画法を用いるため，計算量が多いという問題点がある．そこで本稿では，この問題点を解決するために疎なベクトル間の高速な計算が可能であることに着目し，シーケンスをベクトルとして表現したシーケンス間の類似度の算出方法を提案する．

2. 従来方式とその問題点

2.1 Edit distance

シーケンス間の類似度を算出するための従来手法として，Pirjo Moen が提案する手法がある．Pirjo Moen の手法は，Web アクセスログや，通信の際のアラームを対象としており，アクセスページやアラームタイプをイベントと定義している． そして，このイベントによって構成されるシーケンス間の類似度を Edit distance としている．Edit distance とは，あるシーケンスを他のシーケンスに変換する際の変換作業にかかる最小の総変換コストである．ここでシーケンスの変換作業とは，イベントの挿入，イベントの削除，イベントの移動である．

Pirjo Moen の手法において対象としているシーケンスは，イベントのみによって構成されるシーケンス，またはイベントとそのイベントの発生時刻を含むシーケンスである．前者のイベントのみによって構成されるシーケンス間の類似度を算出する際には，イベントの挿入，削除のみの変換作業を用いる．そして，後者のイベントとその発生時刻を含むシーケンス間の類似度を算出するためには，イベントの挿入，削除，移動の変換作業を用いる．また，各変換作業を行った場合のコストの定義が 2 種類提案されている．一つ目は，全ての変換作業を同一であるとし単一のコストを用いる方法で，一回の変換作業のコストを 1 としている．二つ目は，各イベントの発生回数を考慮し発生回数が少ないイベントに対する変換作業のコストを高くする方法である．この場合一回の変換作業のコストを，変換作業を行う対象となるイベントの発生回数の逆数としている．

あるシーケンスに対して複数回の変換作業を行うことにより，他のシーケンスへと変換することができる．しかし，他のシーケンスに変換するために必要な変換作業は様々であり，その変換作業の総コストも様々である．そのため，動的計画法を用い最小の総変換コストを算出し，その値を Edit distance としている．

2.2 Clickstream Clustering

Web アクセスログを利用者ごとにシーケンスとみなし，そのシーケンス間の類似度を算出するための手法が A. Banerjee らによって提案されている．A. Banerjee らの手法において対象としているシーケンスは，アクセスページとアクセスページの閲覧に費やした時間によって構成されている．A. Banerjee らの手法ではシーケンス間の類似度を算出するために，シーケンス間の最長で共通のサブシーケンス (LCS) を，動的計画法を用いて発見する．そして，類似度を算出するシーケンス間で LCS に含まれるアクセスページに対しての閲覧に費やした時間を比較することによって，類似度を算出している．

2.3 従来手法の問題点

2.1 節，2.2 節で説明を行ったシーケンス間の類似度を算出するための手法では，どちらの手法においても動的計画法を用いている．そこで，動的計画法を用いる場合の計算量について説明する．複数のアクセスページによって構成されるシーケンス S_1，S_2 が存在したとする．各シーケンスが持つアクセスページ数をシーケンスサイズとした時，S_1 のシーケンスサイズを $|S_1|$，S_2 のシーケンスサイズを $|S_2|$ とする．シーケンス S_1，S_2 に対して動的計画法を用いる場合，$(|S_1|+1)\times(|S_2|+1)$ の表を作成する必要がある．つまり，一回の類似度算出のために $(|S_1|+1)\times(|S_2|+1)$ 回の計算を行う必要があり，大量のシーケンス間の類似度を算出する際には，計算量が増加するという問題点がある．そこで提案手法では，シーケンスを疎なベクトルで表現し，疎なベクトル間の計算を高速に行うことにより計算量の削減を行う．

3. 提 案 手 法

3.1 提案手法の概要

3.1.1 基本的な考え方

従来手法では，シーケンス間の類似度を算出する際に動的計画法を用いる．これは，シーケンス間の共通のアクセスページを発見するためには有効な手法である．しかし，動的計画法は 2.3

節で説明したように，計算量の多さが問題である．そこで本研究では，疎なベクトル間の高速な計算が可能であることに着目し，シーケンス間の類似度を高速に算出するためにベクトルを用いる手法を提案する．提案手法では，シーケンス間の類似度を算出するための前処理として，シーケンスに含まれるアクセスページをベクトルに変換したシーケンスを作成する．ここで，変換を行ったシーケンスを Event Vector Sequence (EVS) とする．本章では，シーケンスの EVS への変換方法および EVS 間の類似度の算出方法を提案する．

従来手法の中には，アクセスページだけでなくアクセス時刻または，閲覧に費やした時間によって構成されたシーケンスを対象としている手法がある．しかし，現実の利用者は Web ページを閲覧している際に，他の作業をすることが頻繁にある．そのため，時間に関する情報はあいまいな情報である．そこで提案手法では，時間に関する情報を考慮せずアクセスページのみによって構成されたシーケンスを対象とする．

3.1.2 類似度算出の手順

提案手法を用いて類似度を算出する際の手順を説明する．

Step 1. アクセスページをベクトルに変換する際に，アクセスページの種類数をベクトルの要素数とする．そのため，シーケンス集合に含まれるアクセスページの定義を行う．

Step 2. シーケンス間の類似度を算出するための前処理として，全てのシーケンスを EVS に変換する．

Step 3. 全ての EVS 間の類似度を算出する．

3.1.3 本章で用いる式の定義

N 個のシーケンスによって構成されるシーケンス集合 S は $S = \{S_1, S_2, ..., S_N\}$ となる．そして，シーケンス集合 S を構成する M 種類のアクセスページの集合を，$E = \{E_1, E_2, ..., E_M\}$ とする．

シーケンス集合 S を構成する i 番目のシーケンスを，$S_i =< e_{i1}, e_{i2}, ..., e_{il} >$ とする．ここで，S_i に含まれる j 番目のアクセスページを $e_{ij} \in E$ とする．また，S_i のシーケンスサイズは l であり，$|S_i| = l$ とする．

次に，変換後の EVS に用いる式を定義する．S_i を EVS へ変換した場合，S'_i とし $S'_i =< \delta_{i1}, \delta_{i2}, ..., \delta_{il} >$ となる．ここで，δ_{ij} は，S_i の e_{ij} をベクトルに変換したものであり，構成要素は $\delta_{ij} = [\tau_{ij1}, \tau_{ij2}, ..., \tau_{ijN}]$ となる．

本節以降は，この定義を用いて説明を行う．

3.2 アクセスページの定義

提案手法では，シーケンスを構成するアクセスページを独立した存在であるとし，アクセスページをベクトルに変換する際に，アクセスページの種類数をベクトルの要素数としている．このため，本節では 2 種類のアクセスページの定義方法について説明する．

(1) Web アクセスログに存在する全ての Web ページへのアクセスを，シーケンスを構成するアクセスページとして定義する．

(2) Web ページの持つメタ情報に着目し，意味的に同等と考えることができる複数の Web ページを一つのアクセスページとして定義する[5]．つまり，ポータルサイトを例に用いた場合，各 Web ページはあらかじめ，買い物，本，映画などのカテゴリに分類されている．このようなカテゴリごとに，複数の Web ページへのアクセスを一つのアクセスページとする．

3.3 アクセスページのベクトルへの変換方式

本研究では，シーケンスを構成するアクセスページは独立した存在であると考える．そこで，全てのアクセスページをベクトルに変換する際には，シーケンスに含まれるアクセスページの種類数をベクトルの要素数とし，アクセスページごとに異なる要素に 1 を挿入する．そして，その他の要素には 0 を挿入する．

アクセスページ集合 $E = \{E_1, E_2, ..., E_M\}$ によって構成されるシーケンス S_i が存在する．S_i のアクセスページ e_{ij} をベクトル δ_{ij} に変換する場合，δ_{ij} を構成する τ_{ijp} は，式 (1) のように一般化できる．ただし，e_{ij} のアクセスページの種類を E_k とし，$1 \leq p \leq M$ とする．

$$\tau_{ijp} = \begin{cases} 0 & (p \neq k) \\ 1 & (p = k) \end{cases} \tag{1}$$

アクセスページ集合 $E = \{a, b, c\}$ に含まれるアクセスページによって構成されるシーケンス $S_1 =< a, b, c, b >$ を例に用いて，上記の変換を行う．

アクセスページ集合 E の構成数が 3 のため，全てのアクセスページを要素数 3 のベクトルに変換する．そして，アクセスページの種類ごとに異なる要素に 1 を挿入し，$S'_1 =< [1,0,0], [0,1,0], [0,0,1], [0,1,0] >$ へと変換する．

しかし，上記で説明を行った変換方式では，変換の対象となっているアクセスページの前後関係が考慮されていない．そこで，前後関係を考

慮した拡張方式を以下に示す．

Step 1. 前後関係を考慮するアクセスページ数をウインドウサイズ w として設定する．ここで w は $w \geq 2$ とする．

Step 2. 式 (1) を用いた変換を行う．

Step 3. 変換の対象となっているアクセスページの h 個前のアクセスページに対応しているベクトルの要素に $(w-h)/w$ を足す．

Step 4. 変換の対象となっているアクセスページの h 個後のアクセスページに対応しているベクトルの要素に $(w-h)/w$ を足す．

ここで，w，h はともに自然数である．また，Step 3，Step 4 の作業は $h < w$ となる全ての h 対して行う．

上記と同じシーケンス $S_1 =< a, b, c, b >$ を例に用いて，拡張方式の各 step ごとに説明する．

Step 1. ウインドウサイズ 2 として設定する．

Step 2. 式 (1) によって，$S_1' =< [1,0,0], [0,1,0], [0,0,1], [0,1,0] >$ となる．

Step 3. 変換の対象となっているアクセスページの 1 個前のアクセスページに対応している要素に，$(2-1)/2$ を足す．$S_1' =< [1, 0.5, 0], [0, 1, 0.5], [0, 0.5, 1], [0, 1, 0] >$ となる．

Step 4. 変換の対象となっているアクセスページの 1 個後のアクセスページに対応している要素に，$(2-1)/2$ を足す．$S_1' =< [1, 0.5, 0], [0.5, 1, 0.5], [0, 1, 1], [0, 1, 0.5] >$ となる．

3.4 類似度の算出方法

アクセスページを変換したベクトルの各要素は独立した要素である．このため，独立した要素を持つベクトル間の距離の算出に適しているユークリッド距離を用いて，ベクトル間の類似度を算出する．類似度を各ベクトル間のユークリッド距離を用いて算出する際に，類似度を対応するベクトルごとのユークリッド距離の総和とすることができる．しかし，この方法ではシーケンスサイズが異なる場合，適切な類似度の算出が行うことができない．そこで，シーケンスがどのようなアクセスページによって構成されているかに着目し，シーケンス間の類似度を算出する．

どのようなアクセスページが存在しているかをシーケンス全体から判断するため，EVS の各ベクトルの和を算出する．そして，算出された和ベクトルを **Terminal Vector** と定義し，Terminal Vector 間のユークリッド距離を算出する．

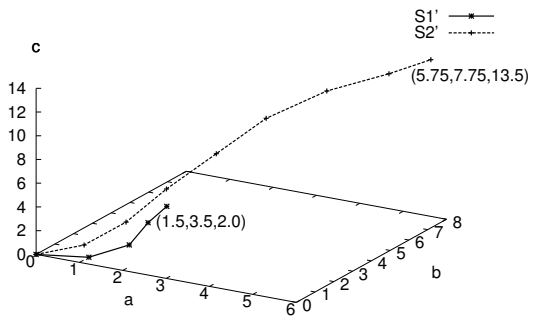

図 1 EVS のベクトル遷移と Terminal Vector
Fig. 1 Transition of vectors in EVS and Terminal Vector.

例えば，アクセスページ集合 $E = \{a, b, c\}$ に含まれるアクセスページによって構成される二つのシーケンス $S_1 =< a, b, c, b >$，$S_2 =< a, c, c, b, c, b, a, c >$ が存在したとする．それぞれのシーケンスを EVS に変換した場合，以下のように変換することができる．ここで，ウインドウサイズは我々の過去の研究[6] よりシーケンスサイズの 50% に設定する．

$S_1' =< [1.0, 0.5, 0], [0.5, 1.0, 0.5], [0, 1.0, 1.0], [0, 1.0, 0.5] >$

$S_2' =< [1.0, 0.25, 1.25], [0.75, 0.5, 2.0], [0.5, 1.0, 2.25], [0.5, 1.5, 2.0], [0.5, 1.5, 2.0], [0.75, 1.5, 1.5], [1.0, 1.0, 1.25], [0.75, 0.5, 1.25] >$

この二つの EVS の各ベクトルを和を算出することにより，アクセスの傾向を読み取る．そこで，各シーケンスの和を算出し 3 次元の座標にプロットしたものを，**図 1** に示す．3 次元の座標を用いるのは，アクセスページ集合 E に含まれるアクセスページの種類数が 3 であるためである．図 1 は，各シーケンスのアクセスページの変遷を意味している．そして，座標にプロットされている各 EVS の終点が各シーケンスの特徴を示しているものと考え，この値が Terminal Vector となる．S_1'，S_2' の Terminal Vector は，各々 $[1.5, 3.5, 2.0]$，$[5.75, 7.75, 13.5]$ となる．そして，算出された Terminal Vector 間のユークリッド距離を，シーケンス S_1，S_2 間の類似度とする．

しかし，このように類似度を算出した場合，類似度はシーケンスサイズに大きく依存することとなる．このため，Terminal Vector を正規化する必要がある．Terminal Vector の正規化は各要素を Terminal Vector のユークリッドノルムで除算することによって可能である．

3.5 従来手法との計算量の比較

動的計画法を用いた従来手法により，類似度を算出する際の問題点として，2 章で計算量の多さを挙げた．そこで本節では，提案手法と従来手法の計算量の比較を行う．

まず，全てのシーケンス間の類似度を算出するために必要な計算量に着目する．アクセスページ集合 $E = \{E_1, E_2, E_3, ..., E_M\}$ によって構成されるシーケンス集合 $S = \{S_1, S_2, ..., S_N\}$ が存在するとする．シーケンス集合 S に含まれる全てのシーケンス間の類似度を算出する際に必要な計算回数は，提案手法，従来手法ともに $(N \times (N-1))/2$ 回である．つまり，計算量は $O(N^2)$ となる．次に，提案手法の前処理であるシーケンスの EVS への変換に必要な計算量について説明する．前処理では，全てのシーケンスに対して変換を行うため，変換回数は N 回であり，計算量は $O(N)$ となる．類似度を算出するために必要な計算量と，前処理に必要な計算量を比較した場合，類似度の算出にかかる計算量に対し，前処理に必要な計算量は非常に少ないといえる．

次に，シーケンス集合 S に含まれる個々のシーケンス間の類似度を算出するために必要な計算量に着目する．シーケンス集合 S に含まれる二つのシーケンス S_α，S_β が存在したとする．ここで，S_α，S_β のシーケンスサイズは各々$|S_\alpha|$, $|S_\beta|$ である．従来手法では，2.3 節で説明したように S_α，S_β 間の類似度を算出するために，$(|S_\alpha| + 1) \times (|S_\beta| + 1)$ の表を作成する．つまり，計算量は $O(|S_\alpha||S_\beta|)$ である．提案手法では，S_α，S_β 間の類似度を算出するために M 個の要素を持つ Terminal Vector 間のユークリッド距離を算出する必要があり，計算量は $O(M)$ となる．

提案手法と従来手法の計算量を比較した場合，どちらの場合に計算量が増加するかどうかは明確でない．そこで，計算回数の比較を行う．上記の説明の通り，従来手法の計算回数は $(|S_\alpha| + 1) \times (|S_\beta| + 1)$ 回，提案手法の計算回数は M 回である．ここで，M はシーケンス集合 S を構成するアクセスページの種類数である．そのため種類数が増加した場合，従来手法の計算回数を上回る可能性がある．

しかし，この提案手法の計算回数は最も非効率な計算を行った場合のものである．実際の Web アクセスログに本手法を適用した場合は，Terminal Vector を構成する多くの要素は 0 となる．このため，0 と 0 の計算を無視することにより，提案手法の計算回数を大幅に削減することが可能となる．具体的には，Terminal Vector の 0 以外の要素数は S_α，S_β に含まれるアクセスページの種類数である．そのため，計算回数は $|S_\alpha| + |S_\beta|$ 以下になると考えることができる．これにより，提案手法の計算回数は従来手法よりも少なくなるといえる．

4. 実　　験

4.1 実験概要

提案手法の有用性を確認するために試作システムを作成し，提案手法と従来手法の比較実験を行った．実験には，Pentium4 プロセッサ 3 GHz と 1GB のメモリを搭載し，OS には Windows XP を使用した．

提案手法では，アクセスページのみで構成されたシーケンス間の類似度を算出する．そのため従来手法には，2.1 節で説明を行ったアクセスページのみで構成されたシーケンス間の類似度である Edit distance を算出する手法を用いる．そして提案手法では，シーケンスを構成するアクセスページの存在回数を考慮していないため，Edit distance を算出する際の一回の変換コストは 1 とする．また，シーケンスサイズの違いによって Edit distance は大きな影響を受けてしまうため，実験では正規化された Edit distance を用いる．正規化された Edit distance は，一方のシーケンスに含まれる全てのアクセスページを削除するために必要な変換コストと，もう一方のシーケンスに含まれる全てのアクセスページを挿入するために必要な変換コストの和で，シーケンス間の Edit distance を除算することによって算出される．これにより，正規化された Edit distance は，0 から 1 の範囲で算出される．一方，提案手法を用いて実験を行う際にはウインドウサイズを設定する必要があり，この値はシーケンスサイズの 50% とした．比較実験では，類似度の算出にかかる処理時間の比較，および算出された類似度の性質の比較を行う．性質の比較を行うために 2 種類の方法を用いた．一つ目は，両手法によって算出された類似度間の相関関係を，相関図と相関係数を用いて調べた．二つ目は，両手法によって算出された類似度から，クラスタ分析を用いてデンドログラムの作成，およびクラスタを作成した．そして，作成されたデンドログラムとクラスタの比較を行った．

4.2 実験データ

実験には Microsoft Anonymous Web Data[7] をテストデータとして用いる．このデータは，一週

表 1 Microsoft Anonymous Web Data に前処理を施した実験データ
Table 1 The Microsoft Anonymous Web Data that gives preprocessing.

	data1	data2
シーケンス数	9544	50
アクセスページの種類数	280	90
最長シーケンスサイズ	35	28
最短シーケンスサイズ	4	4
平均シーケンスサイズ	6.03	6.78

間分の www.microsoft.com へのアクセスを無作為に抽出した Web アクセスログである．格納されている情報は，利用者の匿名化された識別情報と，その利用者のアクセスページである．Microsoft Anonymous Web Data では，アクセスページをその Web ページが存在するカテゴリごとに記録している．つまり，3.2 節で説明を行った二つ目のアクセスページの定義を用いているものである．

実験を行う際には，Microsoft Anonymous Web Data に以下のような前処理を施した 2 種類のデータを用いる．2 種類のデータの詳細は，**表 1** に示す．

data 1. シーケンスサイズが 3 以下のデータを削除したシーケンス集合．

data 2. data1 からランダムに 50 シーケンスを抜き出したシーケンス集合．

4.3 類似度の算出にかかる処理時間の比較

本節では，本手法と従来手法の類似度を算出するためにかかる処理時間の比較を行う．また本手法を用いる場合，シーケンスを EVS に変換する必要があり，この変換にかかる処理時間に関しても説明を行う．

data 1 に対して，提案手法と従来手法を用いて類似度を算出し，その処理時間の結果を **図 2** に示す．図 2 から，提案手法の処理時間は従来手法の処理時間の約半分程度であることが分かる．

本実験では，提案手法を用いて シーケンス間の類似度を算出する際に，各 Terminal Vector の要素が 0 と 0 の場合に計算を無視している．このため，3.5 節で説明したように Terminal Vector 間の類似度を算出するために必要な計算回数は，類似度を算出する対象となる二つのシーケンスサイズの和よりも少ない回数となる．

具体的に，本実験でテストデータとした data 1 の場合について説明する．data 1 のアクセスページの種類数は 280 で，Terminal Vector の要素数は 280 となる．このため，0 と 0 の計算を無視しない場合は，シーケンス間の類似度を算出するためには 280 回の計算が必要である．しかし，data 1 の平均シーケンスサイズはおよそ 6 で，Terminal Vector の 0 以外の平均の要素数はおよそ 6 となる．このため，0 と 0 の計算を無視した場合，シーケンス間の類似度を算出するために必要な平均計算回数は 12 回以下となる．

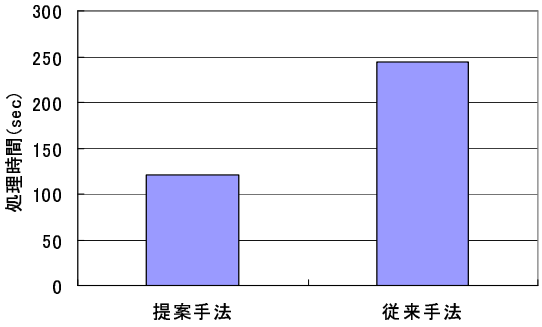

図 2 処理時間の比較
Fig. 2 Comparison of processing times.

提案手法の前処理である シーケンスの EVS への変換にかかった処理時間についても実験を行った．EVS への変換にかかった処理時間は 656*msec* であり，これは提案手法を用いた類似度算出にかかる処理時間の 0.5% 程度である．これにより，3.5 節で説明したように，変換にかかる処理時間は問題にならないといえる．

4.4 算出された類似度の比較

本節からは，提案手法と従来手法によって算出された類似度の性質についての評価を行う．まず，両手法によって算出された類似度の相関関係を調べる．

data 1，data 2 に対して，提案手法と従来手法を用いて類似度を算出し，算出された類似度の相関係数を算出する．また data 2 に関しては，算出された類似度の相関関係を相関図によって表す．結果は**図 3** に示す．

data 1 から算出された類似度の相関係数は 0.9083 で，data 2 から算出された類似度の相関係数は 0.9080 であり，どちらの場合も同じ程度の相関関係を持っていることが分かる．つまり，図 3 に示されているような強い相関関係を，どちらの場合も持っていることが分かる．これにより，EVS によって算出された類似度は，正規化された Edit distance と同じ程度の精度を維持しているといえる．

4.5 階層的クラスタ分析による分類結果の比較

提案手法と従来手法によって算出された類似

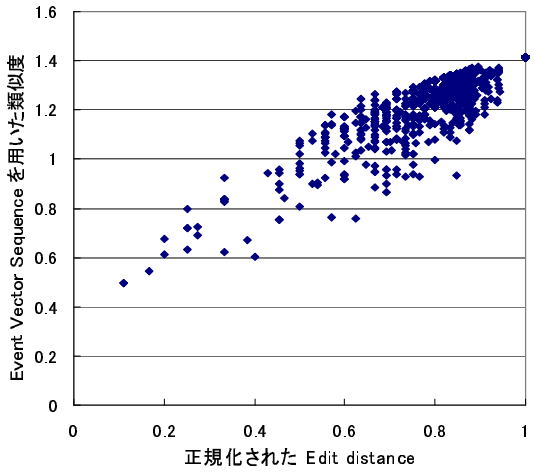

図 3 EVS を用いた類似度と正規化された Edit distance の比較
Fig. 3 Comparison of similarities of EVS between normalized Edit distances.

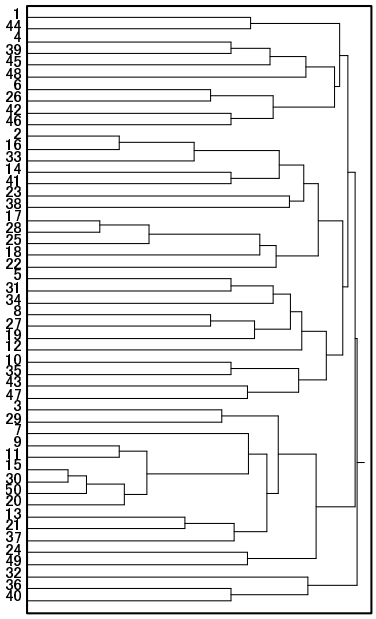

図 4 EVS を用いた類似度から作成したデンドログラム
Fig. 4 Dendrogram with similarities of EVS.

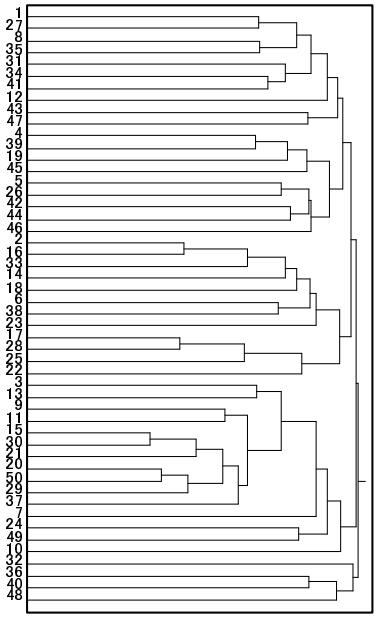

図 5 正規化された Edit distance から作成したデンドログラム
Fig. 5 Dendrogram with normalized Edit distances.

度から，階層的クラスタ分析の一手法である群平均法を用いてクラスタを作成する．本実験では，群平均法を用いるために統計解析アドイン[8]を利用した．

data 2 に対して，提案手法と従来手法を用いて算出した類似度から，群平均法を用いてデンドログラムを作成した．その結果を**図 4**，**図 5** に示す．図 4，図 5 の左の数字は，シーケンスを表している．そして，デンドログラムから作成した五つのクラスタを **表 2**，**表 3** に示す．表中の evs_1，ed_1 から evs_5，evs_5 は，それぞれ EVS を用いた類似度，正規化された Edit distance を用いて作成したクラスタを意味し，各クラスタの構成シーケンスを数字で示している．また，両手法の類似度から作成したクラスタを比較した結果を，**表 4**，**表 5** に示す．表で用いている正解率について以下に説明する．正解となるクラスタの要素数を R，正解率を算出する対象となるクラスタと，正解となるクラスタの共通要素数を C とした場合，正解率は，$\frac{C}{R}$ となる．

表 4，表 5 から evs_1 は ed_1 と ed_2 が結合したものであるといえる．これは evs_4 がシーケンス 32 のみのクラスタになり，クラスタ数が決定されているため，ed_1 と ed_2 が結合したといえる．しかし，シーケンス 32 は図 4 のデンドログラムから，evs_5 に類似していることが分かる．このため，デンドログラムの構成を意識した分類を行った場合，同じようなクラスタを作成することができる．

5. おわりに

本稿では，シーケンス間の類似度を高速に算出するためにシーケンスを EVS へと変換し，Terminal Vector 間のユークリッド距離を算出する手法を提案した．提案手法の有効性を確認するた

表 2 EVS を用いた類似度から作成されたクラスタと構成シーケンス

Table 2 Clusters with similarities of EVS and consisting sequences.

evs_1	1,4,5,8,12,19,26,27,31,34,35,39,41,42,43,44,45,46,47
evs_2	2,6,14,16,17,18,22,23,25,28,33,38
evs_3	3,7,9,10,11,13,15,20,21,24,29,30,37,49,50
evs_4	32
evs_5	36,40,48

表 3 正規化された Edit distance から作成されたクラスタと構成シーケンス

Table 3 Clusters with normalized Edit distances and consisting sequences.

ed_1	1,4,6,26,39,42,44,45,46,48
ed_2	2,14,16,17,18,22,23,25,28,33,38,41
ed_3	5,8,10,12,19,27,31,34,35,43,47
ed_4	3,7,9,11,13,15,20,21,24,29,30,37,49,50
ed_5	32,36,40

表 4 正規化された Edit distance のクラスタを正解集合にした場合の EVS を用いた類似度のクラスタの正解率

Table 4 Correct rate of clusters with similarities of EVS when clusters with normalized Edit distances are correct.

	evs_1	evs_2	evs_3	evs_4	evs_5
ed_1	0.8	0.1	0	0	0.1
ed_2	0.083333	0.916667	0	0	0
ed_3	0.909091	0	0.090909	0	0
ed_4	0	0	1	0	0
ed_5	0	0	0	0.333333	0.666667

表 5 EVS のクラスタを正解集合にした場合の 正規化された Edit distance のクラスタの正解率

Table 5 Correct rate of clusters with normalized Edit distances when clusters with similarities of EVS are correct.

	ed_1	ed_2	ed_3	ed_4	ed_5
evs_1	0.421053	0.052632	0.526316	0	0
evs_2	0.083333	0.916667	0	0	0
evs_3	0	0	0.066667	0.933333	0
evs_4	0	0	0	0	1
evs_5	0.333333	0	0	0	0.666667

め，正規化された Edit distance を算出する従来手法と EVS を用いて類似度を算出する提案手法との比較実験を行った．比較実験において，実際の Web アクセスログをテストデータとして用いた結果，提案手法は従来手法の約半分の時間で類似度を算出することができた．また，提案手法を用いるための前処理であるシーケンスの EVS への変換にかかる処理時間は，全ての EVS 間の類似度を算出するための処理時間に比べて非常に短く，無視できる程度であることを示した．また，提案手法と従来手法によって算出された類似度を比較した結果，相関関係から提案手法は従来手法と同程度の精度を維持していることが証明された．また，両手法によって算出された類似度から，階層的クラスタ分析を行った結果，似た特徴を持ったデンドログラムを作成した．これらの比較実験の結果から，本手法の有用性が確認された．

今後の課題としては，Web アクセスログだけでなく，音声などの他のシーケンスへの応用を考える必要がある．また，シーケンスに適用可能な他のマイニング手法[9] との比較および，組合せについて検証を行う必要がある．

参 考 文 献

1) R., A. M.: クラスター分析とその応用, 内田老鶴圃 (1988).
2) マイケルJ.A. ベリー, ゴードン・リノフ: データマイニング手法, 海文堂 (1999).
3) Moen, P.: *Attribute, Event Sequence, and Event Type Similarity Notions for Data Mining*, PhD Thesis, University of Helsinki, Finland (2000).
4) Banerjee, A. and Ghosh, J.: Clickstream clustering using weighted longest common subsequences, *Proceedings of the Web Mining Workshop at the 1st SIAM Conference on Data Mining*, pp. 33–40 (2001).
5) Banerjee, A. and Ghosh, J.: Concept-based clustering of clickstream data, *Proc. 3rd Intl. Conf. on Information Technology*, pp. 145–150 (2000).
6) 赤塚厚司, 鈴木優, 川越恭二: Web アクセスログ分類のための Event Vector Sequence 法とその評価, 情報処理学会研究報告, 2004-DBS-134(I), Vol.2004, pp. 1–8 (2004).
7) UCI: . UCI KDD Archive http://kdd.ics.uci.edu/.
8) 早狩進: . EXCEL アドイン工房 http://www.jomon.ne.jp/ hayakari/.
9) Agrawal, R. and Srikant, R.: Mining Sequential Patterns, *Proceedings of the Eleventh International Conference on Data Engineering*, pp. 3–14 (1995).