データベースシステム 115－5
情 報 学 基 礎 49－5
（1998. 5. 14）

LSIの適用による大規模HTTPアクセスログからの情報抽出

相澤 彰子
学術情報センター

文献検索の分野で近年利用されている自動索引づけ手法LSI（Latent Semantic Indexing）を大規模HTTPログに適用して，ホストドメイン間の類似度を求める手法を検討する．特に膨大なログデータを扱うために，URL階層上でカルバック情報量を尺度としたデータの要約を試み，その効果を実際のログデータを用いて検証する．

Extracting Information from Large HTTP Logfiles using LSI

Akiko AIZAWA
National Center for Science Information Systems

This paper adopts LSI (Latent Semantic Indexing), an automatic indexing technique recently used in information retrieval, to large-scale HTTP logs and estimates the similarity between two Internet domains. In order to manipulate the huge amount of data, Kullback-Leibler information criteria is applied in the pre-processing stage to summarize the URL hierarchy. The effect of the summarization is demonstrated using actual HTTP log data.

1 はじめに

本稿では，大規模キャッシュサーバやプロキシーなどにおいて得られる膨大な量のアクセスログファイルを分析し，個々のURLに対するアクセス頻度を手がかりにクライアント間の類似度を計算するための手法を検討する．類似のクライアントは共通のURLを多く参照することから，このような情報はHTTPキャッシュサーバの配置などに活用できると考えられる．

ここで対象とするのは，HTTPログファイルに含まれる，アクセス毎の＜クライアントホスト＞と＜アクセスしたURL＞の情報である．これらの情報から，＜クライアントホスト＞対＜URL＞のアクセス頻度行列を作成し，文献検索の分野で近年利用されている自動索引づけ手法LSI [1]を適用する．ただし，HTTPログファイルの場合は文献検索の場合と比較して，クライアントホストを特徴づける「特徴素」であるURLの数が膨大な量になるため，文献検索と同様にしてLSIを適用すると有用な情報が失われてしまう恐れがある．そこで，URLの表記上の階層とカルバック情報量という基準を用いてアクセス頻度情報を要約することを試み，LSI適用における効果を実際のログファイルを用いて検証する [2]．具体的には`http://www.nacsis.ac.jp/Welcome-j.html`を`http://www.nacsis.ac.jp`に代表させるなどであり，本稿ではこのようなデータ要約を特に「URL要約」と呼んでいる．

以下まず**2.**で，LSIについて概観した後，大規模データに適用する際の問題点を述べ，提案する手法の概要を説明する．**3.**では，前処理段階でのURL階層の作成について述べる．**4.**では，URL階層上のノードについてカルバックの情報量を計算し，これに基づきLSIのための特徴素を選択する手順をまとめる．**5.**では，実際のログデータに対してURLの要約およびLSIを適用した結果を示す．最後に**6.**で今後の課題について述べる．

2 LSIによるHTTPアクセスログの分析

2.1 LSI

LSIは [1] で提案され，文献検索の分野で近年用いられている自動索引づけの手法である．通常，文献検索では文献中に出現する用語を索引

として用いる．これに対してLSIでは，重みづけした用語ベクトルを索引として用いる．用語ベクトルの求め方は基本的に主成分分析の場合と同様である．まず各文献中の各用語の出現頻度を求め，文献数 t，用語数 d に対する $t \times d$ の出現頻度行列 X を作成する．そして，双対尺度法と呼ばれる統計手法にしたがって X を以下の式(1)の形に特異値分解（Singular Value Decomposition）する．

$$X = T\,S\,D^t \qquad (1)$$

ここで，T，S，D はそれぞれ，$m = min(t, d)$ として，$t \times m$，$m \times m$，$m \times d$ 行列であり，I を単位行列とするとき $TT^t = I$，$D^tD = I$，S は対角行列である．

式(1)の右辺を求めたら次に，T の各行を検索のための用語ベクトル，対応する S の対角成分を各用語ベクトルに対する重みとして，各文献を行列積 $X^t\,T = D\,S$ で得られる特徴空間上の点に配置する．特徴空間の次元数 m が大きい場合には，重みが大きい順に上位 k 個の特徴ベクトルを選び，特徴次元の数を t から k に減らして用いる．

LSIではこのように特徴ベクトルに重みをつけることによって，同義語による影響などを避け，文献や用語どうしの位置関係を特徴空間上で見通しよく示す効果を狙っている．LSIを用いると，構築や更新に手間がかかる専門用語シソーラスを使わなくても，与えられたデータから自動的に特徴空間の生成が行える．LSIの詳細は[1]に，特異値分解の計算アルゴリズム例は[4]に示されている．

2.2 HTTPアクセスログへの適用の問題点

本稿では，文献をクライアントホストに，用語をURLに置き換え，LSIをHTTPログファイルの分析に適用することを試みる．

ここで，ホストの数を N_h，ログファイル中に出現するURLの総異なり数を N_u とすると，LSIの適用は，ホストごとのURLアクセス頻度を示す $N_h \times N_u$ 行列の特異値分解の問題に帰着される．しかし，現存するURLの数は，著名な検索エンジンに登録されているだけで数百万とされており，対するインターネットホスト数も大規模なネットワークのプロキシーにおいては数千以上であると考えられる．

一般に特徴素の数が大きくなると，計算量やメモリの問題からすべての特徴素をLSIに用いることは不可能で，何らか前処理を行って特徴素の数を減らす手順が必要になる．文献検索の分野では，用語数が多くなりすぎる場合に，すべての文献に共通して頻出するストップワード（英語の 'a' など）を除いた上で，頻度の大きい数百から数千程度の語を特徴素として選ぶ方法が一般的である．しかしHTTPログファイルの場合には，数千個のURLを選んだとしてもURL総数の1%にも満たず，情報の大半が失われてしまう恐れがある．また，すべてのホストが高頻度でアクセスする「ストップワード」に相当するURLは，文献検索の場合のように自明ではなく，文献検索と同様に頻度順にURLを選択する方法では，特徴素として意味のないURLばかりを選んでしまう可能性もある．

2.3 LSIの適用を前提とした分析処理プロセス

上記の問題点に対処するために，URLの表記上の階層を利用して，アクセス頻度のばらつきの小さいURLを上位の階層に要約しながらアクセス頻度の偏りが大きい階層を検出し，LSIに用いる特徴素を選択する方法を検討する．以下，ホストという用語はつねにクライアント側のホストを参照するものとし，クライアント側の `ws1.rd.nacsis.ac.jp` や `nacsis.ac.jp` などをまとめてホストドメインと呼ぶ．

一般に規模データからの情報抽出においては，単に適用可能な分析手法（データ発掘アルゴリズム）を見つけるだけではなく，対象とするデータの特性にあわせて処理プロセス全体をバランスよく設計することが重要であるとされている[6]．以下に[6][7][8]に示された情報抽出の一般的手順にしたがって本アプローチによる分析の流れを示す．

(1) データ集合の選択

分析の対象とするホストドメインの集合を選ぶ．HTTPログファイルから，(<ホストドメイン>，< URL>）の組を関連するアクセス毎に

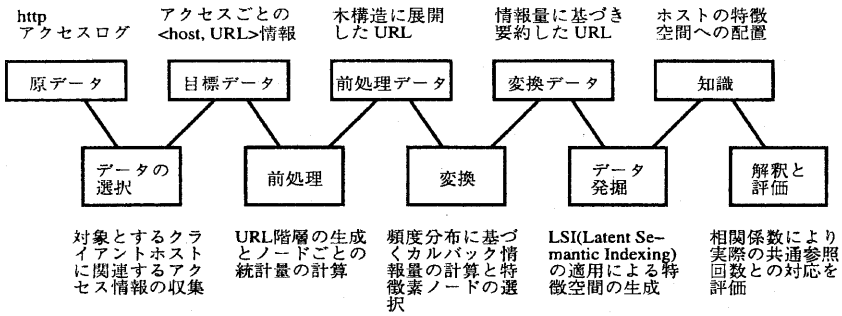


図 1: 本稿における情報抽出の処理プロセス

取り出し，目標データ集合とする．

(2) データの前処理

目標データ集合中の URL をドメインやディレクトリの階層にしたがって木構造に展開する．木構造上の各ノードについて，その下に含まれる URL のアクセス頻度の分布を調べる．

(3) データの変換

後述するカルバック情報量を基準にしてして特徴素として用いる URL 階層を選択する．選択した URL 階層に対する各ホストのアクセス頻度を計算してアクセス頻度行列を得る．

(4) データ発掘アルゴリズムの適用

アクセス頻度行列に対して LSI を適用する．

(5) 知識の解釈と評価

得られた特徴ベクトルからホスト間の距離を計算し相関行列を得る．また (1) のデータから直接，2 つのホストが共通の URL を参照する比率を求めて相関行列との一致性を評価する．

以下では，LSI を適用するための前段プロセスである (2) および (3) の手順を詳細に説明した後，実際のデータからの情報抽出結果に基づき，後段プロセスである (5) により評価を行う．

なお本稿は URL を特徴素とするホストの分析を目標としているが，双対尺度法という呼び名が表すように，LSI で用いる統計分析手法はホストと URL に関して双対的であり，ホストを特徴素とする URL の分析も同様にして実現できる．

3 前処理：URL 階層の作成

3.1 URL 階層

まず目標とするログファイル中に含まれる各 URL について，ファイル中での出現回数を求めてアクセス頻度とする．また URL に含まれるサーバ名，ポート番号，ファイル名の情報から，図 2 の形の URL 階層を作成する．URL 階層上のノード i について，その下に含まれる URL の数 x_i，アクセス総数 y_i を求める．

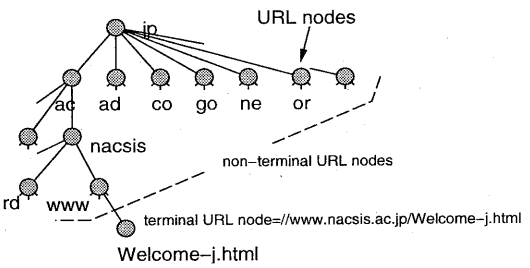


図 2: URL 階層の例

ここで URL がディレクトリを参照している場合には，新たに文字列 "/" を参照する終端ノードを作成して実際の URL と対応させることにすると，この階層における終端ノードは個々の URL に対応しており，アクセス頻度を n_0 として，$x_i = 1, y_i = n_0$ である．以後，URL 階層上のノードを URL ノード，URL ノードのうち終端ノードであるものを単に URL と呼ぶ．

3.2 アクセス頻度データの分布

任意の URL ノード i において，そのノードから下位に属する URL（ノード自身を含む終端ノード）のうちアクセス頻度が n であるようなものの総数 $h_i(n)$ を分析データから求める．次に，こうして得られるアクセス頻度のヒストグラムをアクセス総数で正規化して，URL ノード i のアクセス頻度分布に関する確率密度関数 f_i を以下の式で定める．

$$f_i(n) = \frac{h_i(n)}{\sum_{n=1}^{\infty} h_i(n)} \qquad (2)$$

ノード i が終端ノードである場合，その URL のアクセス頻度を n_0 として，$f_i(n) = 1$ （$n = n_0$ のとき），$f_i(n) = 0$ （$n \neq n_0$ のとき）となる．

4 変換：特徴素とする URL ノードの選択

4.1 URL ノード毎のカルバック情報量の計算

URL 階層から特徴素を選ぶ場合に問題になるのは，階層全体を通してノード数やアクセス頻度の偏りが大きいことである．たとえば国別やドメイン別の統計情報などは階層の深さが等しい例であるが，その中には極端にアクセスが集中するノードとほとんどアクセスがないノードが混在し，ノード間の差が激しいことがわかる．また，階層の深さによる比較は，ファイル名のレベルでは意味をなさないことは明らかである．

本稿では，各 URL ノードが分析の対象となる URL 全体のアクセス傾向から見てどの程度特徴的であるかを数量的に表現することによりこの問題に対処する．具体的には，URL 全体のアクセス頻度分布に対する URL ノード i のアクセス頻度分布のカルバック情報量（Kullback-Leibler information, KLI）[9][10] を計算して用いる．カルバック情報量は 2 つの確率密度分布の距離を表す非負の情報尺度であり，その値は両者が等しい場合にゼロとなる

いま，URL 全体のアクセス頻度分布を $f^*(n)$，注目する URL ノード i に関する URL のアクセス頻度分布を $f_i(n)$ とするとき，$f^*(n)$ に対する $f_i(n)$ のカルバック情報量は次式で与えられる．

$$k_i = -\sum_{n=1}^{\infty} f_i(n) log \frac{f^*(n)}{f_i(n)} \qquad (3)$$

全体のアクセス頻度分布 $f^*(n)$ は，すべての URL の情報を集計しているため，$f_i(n) > 0$ なる n に対してつねに $f^*(n) > 0$ となり，上式は必ず計算できる．

$f_i(n)$ が $f^*(n)$ に等しい場合，すなわち最上位の URL ノードについて計算した場合には，カルバック情報量の値は最小値 0 となる．また i が終端ノードである場合，そのアクセス頻度を n_0 とすると，カルバック情報量は $-log f^*(n_0)$ となり，n_0 の自己情報量そのものである．

図 3に，典型的な URL ノードのアクセス分布とそれに対するカルバック情報量の計算例を示す．アクセス頻度が低い場合と高い場合のいずれについても，その分布が平均から偏るほどカルバック情報量の値が大きくなることがわかる．

ここでカルバック情報量の持つ意味を考察するため式 (3) を変形すると以下の形になる．

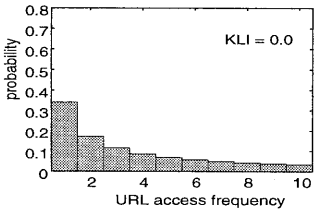
$$k_i = -\sum_{n=1}^{\infty} f_i(n) log f^*(n) + \sum_{n=1}^{\infty} f_i(n) log f_i(n) \qquad (4)$$

上式の第 1 項はノード i を展開して得られる URL の自己情報量の平均であり，第 2 項はノード i の分布 $f_i(n)$ に関する自己情報量である．直感的には前者は平均アクセス頻度の偏り，後者はノード内でのアクセス頻度のばらつきを表しており，カルバック情報量が両者を同時に評価していることがわかる．

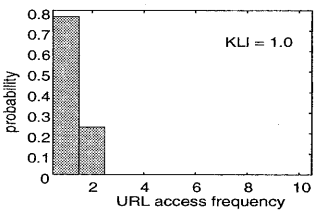
4.2 カルバック情報量に基づく特徴素の選択

カルバック情報量の値が大きい URL ノードは，すでに全体の分布と比較して十分に特徴的であるため，そのノードに含まれる URL のアクセス頻度を平均値で代表させても失われる情報は少ないと考えられる．そこで，カルバック情報量に関する下限値 K_l をパラメタとして，以下の手順 (1)(2) で特徴素とする N 個の URL ノードを選択する．

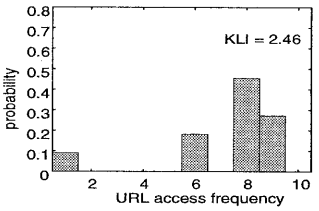
- [手順 (1)] まず，URL 階層を最上位のノードから深さ優先でたどる．$k_i < K_l$ であ



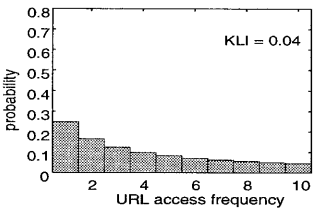
(a) URL 全体のアクセス頻度分布



(b) 低頻度 URL ノードの例



(c) 高頻度 URL ノードの例



(d) 上位 URL ノードの例

図 3: URL ノードに対するカルバック情報量の計算例

る間は下位ノードへの展開を続け，$k_i \geq K_l$ となった時点で展開を終了して現在のノードを新たな終端ノードとし，上位ノードに戻る．

- [**手順 (2)**] 手順 (1) により得られた終端ノードを総アクセス頻度 ($\sum h_i(n)$) の順に並べ，上位 N_f 個をとることによって最終的な特徴素を得る．

K_l の値を小さくとればとるほど URL 階層上でのノードの展開は浅くなり，上位の URL ノードが特徴素として選択されることになる．ノード展開の例を図 4に示す．

なお特徴素の選択に用いるパラメタ K_l の設定については，いったん URL 階層を作成してしまえばノードに関する統計量の計算は短時間で終了することから，現在の実装ではその値を対話的に定めるようにしている．

4.3 アクセス頻度行列の作成

まず，N_h をホストドメイン数，N_f を上記で選択した URL の数として，$N_h \times N_f$ のアクセス頻度行列 Mを作成する．すなわち，M の成分 m_{ij} $(1 \leq i \leq N_h, 1 \leq j \leq N_f)$ は，ホストドメイン i の URL j に対するアクセス頻度を表す．

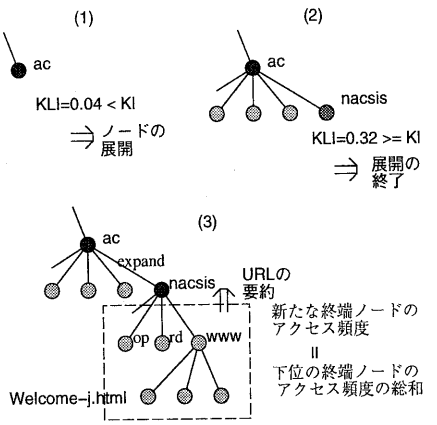


図 4: カルバック情報量に基づく URL 要約の例 (ただし $K_l = 0.2$ とする)

次に，ホストドメインごとに M を総アクセス頻度で正規化して，最終的に LSI の適用対象となるアクセス頻度行列 $\widehat{M} = \{\hat{m}_{ij}\}$ を得る．文献検索では多くの場合，文献中の出現頻度を，出現した文献数で正規化した「tf/idf 尺度」が用いられるが，ここでは特徴として選択する URL ノードがほぼすべてのホストドメインからアクセスされる場合も多いことから，ホストドメイン数による正規化は行っていない．

5 LSI 適用によるホスト間の類似度分析の例

5.1 jpドメインの分析

まず $N_h = 6$ とし，{ac, ad, co, go, ne, or}.jp の 6 つのホストドメインを対象として LSI の有効性とカルバック情報量を用いた URL 要約の効果を調べた．

評価用データは，電子技術総合研究所の中継サーバ ringer.etl.go.jp:80 における 1997 年第 23 週から第 42 週の 20 週間分のアクセスログから抽出し，アクセス絶対数の違いによる影響を除くため，各ドメインについてアクセス総数が等しくなるようにサンプリングをした．用いたデータ中の アクセス総数は 1,144,658，ホスト数は 13,789，URL 数は 459,429 であった．

まず，URL 要約におけるカルバック情報量の閾値 K_l の影響を調べるため，$K_l = 0.002, 0.02, 0.2, 0.5, \infty$ の各値について，要約により得られる URL ノードの階層の深さを調べた．その分布を表 1に示す．

表 1: URL 要約におけるパラメタ値 K_l の影響

URL レベル	$K_l =$ 0.002	$K_l =$ 0.02	$K_l =$ 0.2	$K_l =$ 0.5	全 URL ∞
1	551	550	499	405	53
2	57	3370	4304	3852	684
3	−−	2457	3097	3195	2323
4	−−	48	4707	11732	15803
5	−−	510	11187	26520	49418
6	−−	4767	16690	44477	96982
7	−−	−−	40126	114032	183454
8	−−	−−	3060	17195	66841
9	−−	−−	1331	7996	27831
10	−−	−−	189	2211	10181
11	−−	−−	83	765	3054
12	−−	−−	10	719	1562
13	−−	−−	6	89	756
14	−−	−−	−−	73	248
15	−−	−−	−−	7	128
16	−−	−−	−−	−−	48
17	−−	−−	−−	−−	48
18	−−	−−	−−	−−	10
19	−−	−−	−−	−−	4
20	−−	−−	−−	−−	0
21	−−	−−	−−	−−	0
22	−−	−−	−−	−−	1
合計	608	11702	85290	233268	459429

K_l の値を変化させることによって，URL 階層の展開の深さを調節できることがわかる．K_l の値を小さくすれば，階層上位の少数の URL ノードだけが選ばれるのに対して，K_l の値を大きくすると URL 階層の展開は下位まで進み，得られる URL ノードの数は増大する．$K_l = \infty$ ではすべての URL ノードが展開されることから，この場合は要約を行わない場合と等価である．

次に LSI の有効性を調べるため，対象とするログファイルからアクセス頻度行列を作成して LSI を適用した．LSI では，式 (1) の形の特異値分解を用いて，行列積 $D\,S^2\,D^t$ によりホストドメイン間の距離（非類似度）を求めることができる．そこで評価のために，実際にログデータからホストドメイン間で共通に参照された URL の異なり数を数え上げてアクセス総数で割った値を「真の」類似度とし，LSI 適用の結果得られる距離行列との一致性を調べた．以下，このようにして求めた「真の」類似度を URL 共通参照比率と呼ぶ．URL 共通参照比率による類似度の定義は文献 [3] でも用いられており，2 つのドメイン間で協調キャッシュを行った場合のヒット率にも対応している．

表 2では，単純に LSI を適用した場合（方法 (A)），URL 要約を行った上で LSI を適用した場合（方法 (B)）の 2 つについて，LSI 適用の結果得られるホストドメイン（HD）対の距離と URL 共通参照比率の一致度を比較している．

方法 (A) では，個々の URL をアクセス頻度順に並べ，上位の 30 個の URL を特徴素として選んだ．方法 (B) では，$K_l = 0.2$ として **4.** で述べた手順により URL を要約して総アクセス数の順に並べ，上位の 30 個の URL ノードを特徴素として選んだ．ここではホストドメイン数が少ないことから特徴素の数を $N_f = 30$ としたが，この値を変化させても同様の結果が得られることを確認している．

URL 共通参照比率による類似度行列との一致度は，順位相関係数および積率相関係数により評価した [11][12]．スピアマンの順位相関係数は，得られるデータ対（この場合はすべてのホストドメイン対 $6 \times 5/2 = 15$ 個）に対して類似度の高いものから順位をつけた場合の，順位上の相

関をみるものである．ピアソンの積率相関係数は，データ対に関して計算した類似度の数値的な相関をみるものである．どちらの相関係数も，$-1 \leq r \leq 1$ なる値 r をとり，絶対値が大きいものほど相関が高いことを示す．ただし積率相関係数では類似度と距離を比較しているため，相関係数は負の値となる．

表 2: LSI の有効性と URL 要約の効果

順位	URL 共通参照比率 HD 対	類似度	方法 (A) (URL 要約なし) HD 対	距離	方法 (B) (URL 要約あり) HD 対	距離
1	co-ne	0.157	ad-co	0.003	co-or	0.004
2	co-or	0.155	ad-or	0.004	ne-or	0.011
3	ad-co	0.133	co-or	0.004	co-ne	0.016
4	ad-or	0.130	ad-go	0.011	ad-co	0.025
5	ne-or	0.130	co-go	0.014	ad-or	0.028
6	ad-ne	0.123	go-or	0.017	ad-ne	0.035
7	co-go	0.102	ne-or	0.017	go-ne	0.063
8	go-or	0.091	ad-ne	0.020	ad-go	0.077
9	ac-ad	0.089	co-ne	0.021	co-go	0.085
10	go-ne	0.088	go-ne	0.032	go-or	0.087
11	ad-go	0.080	ac-go	0.128	ac-go	0.133
12	ac-or	0.080	ac-ad	0.154	ac-ne	0.266
13	ac-ne	0.074	ac-or	0.166	ac-ad	0.306
14	ac-go	0.067	ac-ne	0.169	ac-or	0.322
15	ac-co	0.059	ac-co	0.179	ac-co	0.332
	スピアマン順位相関係数		0.75		0.92	
	ピアソン積率相関係数		-0.78		-0.87	

表 2からわかる通り，方法 (A)(B) いずれの場合も URL 共通参照比率との相関は高く，LSI の有効性が示されているといえる．また方法 (A) と (B) を比較すると (B) の方が相関係数の値がよいことから，URL 要約による効果を確認できる．

上記において，URL 共通参照比率を求めるためには，${}_{N_h}C_2$通りのすべてのホストドメインの対について，共通に参照した URL の数を数え上げる必要があるが，LSI では，ホストドメインごとに独立に URL の参照回数を調べるだけである．これにより高い相関が得られるのは，同様のアクセス傾向を持つホストドメインは同じ URL を参照する確率が高いことによる．実際には現状の HTTP によるインターネット情報システムでは著名で一般的な内容の URL にアクセスが集中しており利用方法が単調であることから，単純な LSI の適用でも高い相関が得られたものと考えられる．

5.2 類似ホストドメインの抽出

前節の実験では，URL 共通参照比率の数え上げによる比較を行うためにホストドメインの数を少なく設定したが，LSI そのものは多数のホストドメインに対しても適用可能である．そこで次に，与えられたホストドメインの集合に対してまず LSI を適用して距離行列を計算し，これより類似ホストドメインの対を取り出した上で共通参照回数を数え上げて評価を行った．

ここでホストドメイン間でアクセス総数にばらつきが大きいと，アクセス頻度の少ないホストドメインについて評価のための十分な統計データが得られないという問題がある．そこでホストに対しても，URL の要約と同様にドメイン階層を利用した要約を行い，分析対象とする集合を自動生成することにした．

この方法により選んだ $N_h = 64$ 個のホストドメインに対して特徴素の数を $N_f = 320$ とし，**5.1** で述べた方法 (A)（URL 要約を行わない LSI）および方法 (B)（URL 要約を行う LSI）をそれぞれ適用して距離行列を求め，類似度が高い上位 5 組のホストドメイン対を抽出した．その結果を表 3に示す．対象データとしては 1997 年第 24 週の 1 週間分のログデータを用いた．ホストドメインの選択では $K_l = 5$ とし，得られた 7,759 個ホストノードのうち総アクセス数が上位の 64 個を分析対象として選んだ．これらのホストドメインには合計で 805 個のホストが

表 3: 類似ホストドメインの抽出結果

順位	URL 共通参照比率 方法 (A) URL 要約なし	方法 (B) URL 要約あり
1	0.013	0.229
2	0.013	0.112
3	0.061	0.163
4	0.102	0.113
5	0.043	0.228
平均	0.047	0.170

含まれており，関連するログデータのアクセス数は 2,387,197，URL 数は 1,013,503 であった．方法 (A) では，これらの URL のうちアクセス頻度順に 320 個を特徴素として選んだ．方法 (B) では $K_l = 0.2$ とし，得られた 94,767 個の URL ノードのうち総アクセス数が上位の 320 個を特徴素として選んだ．表から明らかなように，LSI の要約を行った場合には類似ホストドメインの抽出がうまく行われていることがわかる．

6 まとめと今後の課題

本論文の実験では，協調キャッシュのための類似ドメインの発見を意識して URL の共通参照比率に基づき評価を行った．現在はログファイル中に出現するすべての URL を対象データとして分析を行っているが，今後はキャッシュ可能なものを区別して URL を選別するなど，具体的なアプリケーションを意識した評価を行うことが考えられる．

本論文の手法を用いて要約した URL 階層は情報量的にバランスよく展開されており，その他にもログファイルの圧縮やキャッシュ管理への適用も考えられる．ただしこの場合には統計処理のリアルタイム化が課題となる．また実験では，アクセス頻度順に上位から特徴素となる URL を選択したが，文献検索の分野では中頻度語を選ぶ方法も一般的である．URL に関しては「中頻度」の基準を定めるのは容易ではなく，このために本手法において特徴素として選択する URL ノードのカルバック情報量に上限値を設定し，極端に偏ったノードをふるい落とすことが考えられる．

最後に文献 [13] では，World-Wide Web からの情報発掘を Web Mining と呼び，その技術を「資源発見（Resource Discovery）」，「情報抽出（Information Extraction）」，「一般化（Generalization）」の 3 つに分類している．本論文で述べた類似ホストの発掘手法はこのうちの「一般化」技術に属するが，URL とホスト名だけを手がかりに類似度の計算を行っている点で，問題設定としては前提知識を必要としない単純なものになっている．より高度な手がかりとして，利用履歴の時系列分析，文書の構文あるいは意味上の特徴解析，利用者との対話等により付与されるラベルを利用することも考えられ，これらを用いた情報フィルタリングへの適用も今後の課題である．

謝辞

本研究を行うにあたり，実験データの提供と論文に関する貴重なコメントを頂きました電子技術総合研究所の佐藤豊氏に謝意を表します．

参考文献

[1] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman : "Indexing by Latent Semantic Analysis," Journal of the American Society of Information Science, Vol,41, No.6, pp.391-407 (1990).

[2] 相澤彰子: "LSI の適用による HTTP アクセスログからのデータ抽出", 情報処理学会第 56 回全国大会 (1998) (発表予定).

[3] 早川和宏，鶴巻宏治，浜田洋: "ユーザの利用履歴に基づく WWW サーバの類似検索", 情報処理学会研究会報告，95-IM-21, pp.11-17 (1995).

[4] Press, W. H., Flannery, B. P., Teukolsky, S. A. and Vetterling W. T. : "Numerical Recipes in C," Cambridge University Press (1988).

[5] 佐藤豊: "多目的 proxy サーバ DeleGate", インターフェース, 95 年 9 月号 (1995).

[6] 河野弘之: "データベースからの知識発見の現状と動向", 人工知能学会誌, Vol.12, No.4 (1997).

[7] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth: "The KDD Process for Extracting Useful Knowledge From Volumes of Data," Communications of the ACM, Vol.39, No.11, pp.27-34 (1996).

[8] 寺野隆雄: "KDD ツールの動向と課題", 人工知能学会誌, Vol.12, No.4, pp.521-527 (1997).

[9] 鈴木義一郎 : "情報量基準による統計解析入門", 講談社サイエンティフィック (1995).

[10] 中川聖一: "情報理論の基礎と応用", 近代科学社 (1992).

[11] 芝祐順，渡辺洋，石塚智一編: "統計用語辞典", 新曜社 (1984).

[12] 応用統計ハンドブック編集委員会編: "応用統計ハンドブック", 養賢堂 (1978).

[13] Oren Etzioni: "The World-Wide Web: Quagmire or Gold Mine ?" Communications of the ACM, Vol.39, No.11, pp.65-68 (1996).