DEWS2004 2-B-03

多基準意思決定に基づくウェブ情報検索機能の改良

成　凱[†]　平野真太郎[††]　相　利民[†]　上林　彌彦[††]

† 九州産業大学情報科学部〒 813–8503 福岡市東区松香台 2–3–1
†† 京都大学大学院情報学研究科　〒 606–8501 京都市左京区吉田本町
E-mail: †{chengk,xiang}@is.kyusan-u.ac.jp, ††{shin,yahiko}@db.soc.i.kyoto-u.ac.jp

あらまし　ウェブ上の膨大な情報を使いこなすために，高機能の検索サービスが不可欠である。従来の情報検索システムでは，検索条件との類似度 (similarity) を基準として対象文書を評価し、その結果に基づいて，ランクが付けられる。ウェブの場合は，信頼性の低い情報が多く含まれるため，リンク構造に基づく重要度 (authority) を新たな評価基準として導入しランキングに反映させるのが一般的である。本稿では，このような考え方をさらに発展し，類似度，重要度に加え，情報の利用頻度 (frequency) を新たな評価基準としてランキングに反映させ，多基準意思決定に基づくウェブ検索の手法を提案する。さらに，動的な利用頻度を効率よく維持するため，新たなデータ構造，エージング・ブルーム・フィルタ（ABF–Aging Bloom Filters）を提案する。最後に提案手法を検証するためのプロトタイプシステムの実現，予備実験の結果について述べる。

キーワード　情報検索，ウェブ，多基準意思決定，近似データ構造，ランク集約

Improving IR Functions by Multicriteria Decision-Making Methods

Kai CHENG[†], Shintaro HIRANO[††], Limin XIANG[†], and Yahiko KAMBAYASHI[††]

† Faculty of Information Science, Kyusan University Matsukadai 2–3–1, Higashi-ku, Fukuoka, 813–8503
†† Graduate School of Informatics, Kyoto University　Yoshida Homachi, Kyoto, 606–8501
E-mail: †{chengk,xiang}@is.kyusan-u.ac.jp, ††{shin,yahiko}@db.soc.i.kyoto-u.ac.jp

Abstract　In this paper, we propose a framework that uses multicriteria decision–making techniques to improve effectiveness of web information retrieval. We suggest to add access frequency as a new criterion to evaluate the popularity of web resources. We develop a new synopsis data structure, called Aging Bloom Filters (ABF) for efficient management of frequency statistics of underlying web pages. We also describe the implementation of a prototype system and report preliminary experiments.

Key words　Information Retrieval, Web, multicriteria decision–making, approximate data structures, rank aggregation

1. はじめに

ウェブ上の膨大な情報にアクセスし使いこなすために，優れた情報検索 (IR) サービスが不可欠である。従来の IR システムでは，一般に，与えられた質問 (検索語の集まり) に対し，検索対象の「類似度」(similarity) を評価しその結果に基づいて，ランクが付けられる。このようなシステムの想定していた検索対象が，新聞記事や特許明細書などのような均質・静的なテキスト集合であり，検索対象を評価するには単一の基準を用いてもよい。一方，ウェブの場合は，検索対象が非均質・動的なハイパーテキスト集合であり，様々な言語・表現が用いられるため，従来の方法で適切な結果が出ない可能性がある [10], [11]。

（1） ほとんどの質問は１～２語しかなく，類似度を正確に計算するための情報に乏しい。

（2） 検索結果の数が大量にあるにもかかわらず，一度に表示可能な結果は限られているため，上位に現れない結果は利用者に無視される可能性が高い。

（3） 意図的にランキングの上位に位置する行為（スパム）が存在し，検索結果の公正性に問題が生じる。

（4） 不特定多数の利用者に利用されるため，全ての利用者に適した結果を提供することが難しい。

これらの問題を解決するため，類似度に加え，新たな評価基準を用いる必要がある。この数年間，Google をはじめ，数多くのウェブ検索システムでは，ウェブを有向グラフとしてモデル化し，ハイパーリンクの参照関係から情報の「重要度」(authority) を新たな基準として計算し結果をランキングに反映

させる仕組みが採用されている。例えば，Google では，リンク構造から PageRank と呼ばれるページの重要度を計算し，検索結果に反映されている。Google 成功の一つの要因は PageRank の導入と知られている [9]。

しかしながら，類似度や重要度は情報の「送り手」の意図しか反映せず，それに基づく検索の上位に現れる結果でも決して利用者が満足できるものではない。例えば，東京でアパートを探そうとしている人に対して，一般の賃貸関係のサイトより，首都圏の利用者のよく訪れるウェブサイトのほうがより有用な情報を持っていると考えられる。また，東京の利用者としても，時期によって，利用状況が変わるので，特定の地域や時期においてよく利用される情報はその地域の利用者にとって重要と思われ,「利用頻度」（frequency）の高い情報を検索結果の上位に現せるべきである。

利用状況を用いて検索効率を向上する研究は数多くあったと思われるが，プライバシーやスケーラビリティの問題で，汎用検索サービスに取り込む研究が少なく，個人や組織レベルのパーソナライゼーションに限られたことが多い。その原因の一つは巨大のウェブに対し利用頻度情報を効率的に維持するにはスケーラブルな方法が存在しない。もう一つは 複数の評価基準を検索結果に反映させる適切な方法が必要である。

本稿では，我々は，多基準意思決定 (multicriteria decision making) の枠組みに基づいて，類似度と重要度に加え，利用頻度を新たな評価基準として導入し，複数の評価基準を用いたウェブ検索の改良方法を提案する。それを実現するために，我々は

- 利用頻度の新鮮さを扱う可能なデータ構造，エージング・ブルーム・フィルタ（ABF–Aging Bloom Filters）を考案した。
- 複数の基準の結果を集約するため，スコアベースの加重総和法と，ランク集約に基づく合意法とボルダ法の適用を考察した。
- プロトタイプシステムの実装と予備実験を行った。

本稿の構成は次のとおりである。第 2 節では，本論文に扱う問題を正式に述べる。3 節では，提案する利用頻度を効率よく扱うための近似データ構造について説明する。4 節は複数の検索基準を統合するための技術を考察する。5 節と 6 節はそれぞれ提案方法を検証するためのプロトタイプシステムと実験評価の結果について述べる。

2. 問　　題

情報検索システムの目標は，対象文書から利用者がもっとも満足できる部分を見つけ出すことである。この目標を達成するために，複数の評価基準を用いて，総合的に評価を行う必要がある。本節では，複数の基準に基づく意思決定の枠組みと，この枠組みをウェブ情報検索の問題に適用するための要素を説明する。

2.1 多基準意思決定の枠組み

多基準意思決定問題は次のように定式化できる。

（1） n 種類の選択肢 $a_j(j = 1, 2, \cdots, n)$ によって構成される選択肢集合 A とする

（2） m 種類の評価基準 $g_i(i = 1, 2, \cdots, m)$ によって構成される基準集合 G とする

（3） $g_i(a_1) > g_i(a_2)$ ならば第 i 基準に基づくと a_1 は a_2 よりも好ましい

2.2 ウェブ情報検索における評価基準

ウェブ検索の場合は，検索対象となるウェブページの集合は選択肢集合 A である。評価基準集合 G には，以下の三つが考えられる。

利用頻度 (frequency) ウェブリソースはすべて同じように使われているわけではなく，特定の時期，特定の地域と特定の利用者・利用者団体によって大いに違っている。この違いは利用者の興味や好みを反映していると期待できるため，利用頻度を検索結果のランキングに反映させるべきである。利用頻度，ウェブサーバのログデータや，セキュリティ関係で集められた履歴デーアから抽出できる。

類似度 (similarity) 検索条件と検索対象の類似度はその特徴を表わすベクトル間の距離で計測する。類似度の計算は特徴ベクトル構成の仕組みや，距離関数の選択によって異なる。これは情報検索分野の主要な研究課題のため，様々なモデルが考えられてきた [12]。

重要度 (authority) ページ p の PageRank のような重要度 $pr(p)$ は次のように定義する。多くの価値のあるページからリンクされているページは，価値があるページである。この定義に従って，あるページの重要度を再帰的に計算できる。

3. 利用頻度の効率的計算方式

本節では，ウェブページの利用頻度を効率的に計算する方式について述べる。最新の利用頻度を素早く算出しサーチエンジンに渡すため，我々はエージング・ブルーム・フィルタ (ABF–Aging Bloom Filters) を提案する。

3.1 ウェブログ：利用頻度を抽出するための基礎データ

インターネットは様々なサーバで構成され，すべてのサーバに一定の期間の利用履歴をログファイルに格納している。ウェブと直接関連するプロキシサーバとウェブサーバでは，利用者のウェブ利用履歴を保存している。この履歴情報には利用者もしくは利用者集合の興味や特性が現れている。この履歴情報はウェブの利用頻度を計算する基礎データである。利用履歴は契約するプロバイダに保有しているため，次のような特徴を持つ。

- **地域特徴**　特定の地域の利用者は通常その地域のプロバイダと契約することが普通なので，地域のプロバイダに保有している利用履歴はその地域の特性を持つわけである。例えば，関西地区のプロバイダが保有している利用履歴は関西を中心とする地域の利用者がどのようなウェブサイトをアクセスしていたかを反映したものと考えられる。
- **時期特徴**　同じ地域の利用状況が時期によって異なる。例えば，人気になるスポーツに関する内容として，冬はスキー，夏は海水浴のように違うはずである。

このような特徴を生かしたら，地域・時期に適した検索が期待できる。

利用履歴はサーバの性質によって異なり，同じタイプのサーバでも設定によってカスタマイズすることができる。ここで，プロキシサーバのログについて説明する。プロキシログは利用者のウェブ上での活動をレコード時系列の形で保持したものであり，各レコードは主要に以下のとおりである。

(Timestamp, IP, Size, Code, URL, MIME-Type)

- **Timestamp** リクエストされた時刻，単位はミリセカンド
- **IP アドレス** クライアントの IP アドレス。プライバシー保護のため，正確に記録していない可能性がある。
- **Code** レスポンスの状態を表わすコード
- **URL** リクエストされた URL
- **Size** リクエストされた URL のデータを取得した際のサイズ，単位はバイト

このようなレコードから，URL や利用時刻を取り出して，以下の技術でページの利用頻度を計算する。

3.2 ブルーム・フィルタ (BF–Bloom Filters)

集合要素の帰属関係を効率よく扱うため，Burton Bloom はブルーム・フィルタ (BF–Bloom Filters) と呼ばれる近似的なデータ構造と提案した [6]。このデータ構造は k 個のハッシュ関数と長さ m のビットマップ構造によって構成される。ここで，ハッシュ関数 $h_i(i = 1, \cdots, k)$ は集合 S の領域 U から $[0, \cdot, m-1]$ へ写像する。まず，ビットマップを初期化し全てのビットを 0 にしておく。次に，集合の各要素 s のハッシュ値 $h_i(s), (i = 1, \cdots, k)$ を求める。最後に，ビットマップのこれらのハッシュ値に相当する箇所のビットを 1 にする。

このようにビットマップを構成すれば，任意の要素 x が集合 S の要素であるかについて，高い確率で正しく答えることができる。もし，ビットマップの $h_i(x)$ ビットがすべて 1 ならば，x が S の要素と思われる。そうでなければ，x が S の要素でないことが確信できる。ブルーム・フィルタを介して，元の集合データよりはるかに少ないメモリサイズで，高い確率で要素の帰属関係が判定できる。

3.3 スペクトル・ブルーム・フィルタ (SBF–Spectral Bloom Filters)

多重集合の多重性に関する問合せを答えるため，Saar Cohen ら [7] は上記の基本 BF を拡張し，スペクトル・ブルーム・フィルタ (SBF–Spectral Bloom Filters) を提案した。領域 U の多重集合 S の SBF は，SBF は k 個のハッシュ関数 $hash = (h_1, h_2, \cdots, h_k)$，$m$ 個のカウンタ $C = (C_1, C_2, \cdots, C_m)$ によって，次のように構成される。

まず，全てのカウンターを 0 にしておく。要素 s が入るたびに，s の各ハッシュ下の値に対応するカウンター $C_{h_1(s)}, C_{h_2(s)}, \cdots, C_{h_k(s)}$ をそれぞれ 1 増やす。多重集合の全要素に対して，上記の挿入操作を行い，SBF を構築できる。

このように出来上がった SBF を用いて，要素 x の出現頻度 f_x は $C_{h_1(s)}, C_{h_2(s)}, \cdots, C_{h_k(s)}$ の最小値 m_x として，推定できる。最小値を選ぶのは，大きい値にはハッシュ衝突のため，x 以外の要素がそのカウンターに当たった値が含まれているからである。

[**定理** 1] 任意の $x \in U$ に対して，$f_x \leqq m_x$。しかも，$f_x \neq m_x$ の確率 $E_{SBF} \approx (1 - e^{-kn/m})^k$。ここで n は S に含まれる異なる要素の数である。[7]

上記の確率の最小値を求めると，$k = \ln(2)(\frac{m}{n})$ の時，R_{SBF} の最小値 $= (1/2)^k = (0。6185)^{m/n}$。 $m/n = 8$ の場合は，回答が誤る確率は 2% しか過ぎず，正解率は 98% になる。

3.4 エージング・ブルーム・フィルタ (ABF–Aging Bloom Filters)

利用履歴データは上記の多重集合と異なり，古いデータが利用状況を正しく反映されていないため，利用頻度の値から古くなった部分を徐々に減らしていくことが望ましい。例えば，数年前によくアクセスされたページは今ほとんど意味がなくなる可能性がある。そのページの利用頻度は古い情報しかなく，それを検索結果に反映すれば，無意味な結果が上位に出てくる恐れがある。従って，利用頻度情報の新鮮さを維持する「新陳代謝」機能を持つブルーム・フィルタが必要となる。本節は，このような機能を有する「エージング・ブルーム・フィルタ, Aging Bloom Filters(ABF)」を提案する。

3.4.1 実系列データの移動平均–λ-エージング

ある時系列データ $\{y_0, y_1, \cdots, y_{t-1}, y_t, \cdots\}$ に対し，ある時刻 t に，新規の値と過去の値の加重平均はその時系列の移動平均と呼ばれる。時刻 $(t-1)$ までの移動平均値 w_{t-1} と，時刻 $[t-1, t]$ の間の新規値を y_t から，時刻 t の移動平均値 w_i を計算することができる。

$$w_t = (1 - \lambda) \cdot y_t + \lambda \cdot w_{t-1}, \quad 0 \leqq \lambda \leqq 1, \quad w_0 = y_0 \quad (1)$$

λ は $[0, 1]$ の間の実数で, 過去の値を減らせるスピードをコントロールするパラメターである。λ が小さいほど，古い値の減るスピードがはやい。$\lambda = 0$ なら，古い値は完全に考えず，平均は時系列の実際の数値になる。逆に，$\lambda = 1$ ならば，最初値に固定する。このような極端なケース以外は，古い値が減ってゆくと共に，新規の値が取り入れる。このため，パラメター λ を持つ移動平均は「λ–Aging」と呼ばれる。

3.4.2 エージング・ブルーム・フィルタ (ABF) の構成

λ-エージングのメカニズムを SBF に取り組むことによって，新しい BF を構築することが可能である。しかし，ウェブページの移動平均値を計算するための古い平均値を保つに，余分のメモリスペースを追加する必要がある。

幸いに，我々の環境において，利用頻度は正確でない近似値でも構わないので，余分のメモリスペースが要らないで，近似の利用頻度を維持することができる，エージング・ブルーム・フィルタ (ABF) を提案する。ABF の構成は SBF の要件のほか，パラメター λ，λ-エージングの計算周期 T を指定することが必要である。時間を計るためのタイマ τ を用意する。

3.4.3 ABF への要素の追加

λ-エージングによって，要素の新しい出現回数の一部 (式 1 の $(1 - \lambda) \cdot w_t$ の部分) しか計上しない。この結果を得るため，以下の確率的なプロセスを用いる。

まず，全てのカウンターを 0 にし，タイマ tau を初期化する。それから，要素 s が入るたびに，$(1 - \lambda)$ の確率で，カウ

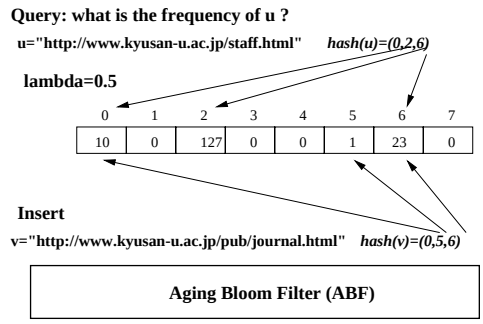

図 1　ABF:エージング・ブルーム・フィルタ
Fig. 1　ABF: Aging Bloom Filters

ンター $C_{h_1(s)}, C_{h_2(s)}, \cdots, C_{h_k(s)}$ を増加させる。$[0,1]$ から実数 r をランダムに選ぶ。$r \geqq \lambda$ の場合は，カウンターの値を増加させる。$r < \lambda$ であれば，何もしない。τ が時間 T を超えるまで このプロセスを続ける。τ が T を超えたら，$C_i = \lambda \cdot C_i$ $(i = 1, 2, \cdots, m)$ にしてから，$1 - \lambda$ の確率でカウンターの値を変えるプロセスを繰り返し行う。

［**補題** 1］　上記の要素追加法で構築された ABF において，任意の時刻 t に要素 $x \in U$ の移動平均値は

$$\hat{f} = \sum_{i=0}^{b} \lambda^{b-i} \cdot \hat{f}_i$$

ここで，$b = t/T$ はこれまで移動平均計算の回数である。$\hat{f}_i$ は第 i 回と $i+1$ 回移動平均計算の間の利用頻度の見積であり，$\hat{f}_i$ カウンターの増やし方から分かるように，期待値は $(1-\lambda) \cdot f_i$

3. 4. 4　ABF による利用頻度管理

ABF による利用頻度管理の動作例を図 1 を用いて説明する。ハッシュ関数 $k = 3$ 個とカウンター数 $m = 8, C_0, C_1, \cdots, C_7$, $\lambda = 0$。5。例えば，v=”http://www。is。kyusan-u。ac。jp/pub/journal。html ”が利用されたとする。v を ABF に挿入するために，まず，v のハッシュで対応するカウンターを特定する。$hash(v) = (0, 5, 6)$ のため，0。5 の確率でこれらのカウンターを 1 増やす。 もし，移動平均計算のタイムが来る場合は，全てのカウンターを半分 (0。5 倍) に減らす。

u=”http://www。is。kyusan-u。ac。jp/pub/staff。html ” の利用頻度を調べる。まず，u のハッシュ値を計算し $hash(u) = (0, 2, 6)$ が分かる。続いて，カウンターの値を調べる。$C_0 = 10, C_2 = 127, C_6 = 23$ のうち，最小値が 10 なので，u の利用頻度は高々10 であることが分かる。

4.　多基準評価結果の集約

前節では利用頻度を ABF を用いて効率よく管理する方式について述べた。その以外の評価基準の類似度，重要度の情報は数多くの検索があるので，詳しい議論を省く。詳しく知りたい方は [9]〜[12] へご参照してもらいたい。

本節では，これらの検索基準の集約する方法について検討する。表 1 で示すように，この問題は二つのタイプに分類できる。TYPE I は各基準，検索条件との類似度，ページの重要度と実際の利用頻度，のそれぞれにおける検索対象のスコアを考えた場合の集約問題と，TYPE II は Google などの商用に用いられている検索エンジンでは，企業秘密としてページのスコアは公開されていない場合の集約問題である。TYPE II の場合には，ランクをスコアに変換することによって対処することが可能である。これにより，スコアが分からずランクのみしか分からない場合でも利用状況を反映させることができる。ここでは数値スコアが分かっている場合の利用状況の集約方法と，ランクしか分からない場合の利用状況の集約方法について順に述べる。

表 1　多基準評価結果集約のタイプ
Table 1　Types of Aggregation Tasks

	類似度	重要度	利用頻度
TYPE I	数値	数値	数値
TYPE II	順序	数値	数値

4. 1　TYPE I：数値スコアベースの集約

4. 1. 1　数値スコアの獲得

検索条件と対象文書の類似度スコアが分かる場合における利用状況の集約方法について述べる。情報検索においてよく用いられる TF/IDF 法 [4] で，対象文書と検索条件ともに特徴ベクトルで表現する。TF/IDF 法で，検索条件 q と文書 p の類似度 $W_q(p)$ を次のように計算する。

$$W_q(p) = \sum_{t \in q} (tf(t,p) * idf(t)) \tag{2}$$

$tf(t,p)$ は，検索 t の文書 p における出現頻度を，$idf(t)$ はキーワード t の文書集の中での希少さを表している。検索の対象となる文書集を TF/IDF 法を用いてベクトル空間モデルで表現する。

一方，PageRank 型の重要度スコアを計算するには，次式を繰り返し適用する。

$$R(p) = \frac{\epsilon}{n} + (1-\epsilon) \cdot \sum_{(q,p) \in G} \frac{R(q)}{outdegree(q)} \tag{3}$$

ここで，$R(p), R(q)$ はそれぞれページ p, q の現在の重要度スコアを表わす。n は対象とするグラフ G（ウェブページをノードとし，ウェブページ間のリンクをエッジとしたグラフ）のノード総数（ウェブページ数），$outdegree(q)$ はページ q から外向きリンク数である。$\sum_{p \in G} R(q) = 1$，ϵ は通常 0。1 0。2 の間に設定されたファクタである [9]。

最後に，ページ p の利用頻度スコアは ABF の p に対応する各カウンターの値の最小値である。

$$F(p) = m_p = \min \{C_{h_1(p)}, C_{h_2(p)}, \cdots, C_{h_k(p)}\} \tag{4}$$

4. 1. 2　数値スコアの標準化

方程式 2,3,4 で得られたスコアは数値スケールが大きく違うことを避けるため，標準化する必要がある。$W_q(p), R(p), F(p)$ はいずれも正の基準（値の大きい方が好ましい）であるため，以下の式で $[0, 100]$ 間の値に標準化する。

$$\bar{v_i} = \frac{v_i - \min_j v_j}{\max_j v_j - \min_j v_j} \times 100, \quad \bar{v_i} \in [0, 100]$$

$W_q(p), R(p), F(p)$ の標準化値はそれぞれ $\bar{W_q}(p), \bar{R}(p), \bar{F}(p)$ と表わす。

4.1.3 各基準下評価結果の統合

加重総和法 (Weighted summation) 各検索対象 p の各基準に適当なウェイトをかけて，総和を計算して比較する

$$\hat{W}_q(p) = w_1 \cdot \bar{W}_q(p) + w_2 \cdot \bar{R}(p) + w_3 \cdot \bar{F}(p) \quad (5)$$

$w_i \in [0,1], (i = 1,2,3)$，かつ $\sum_i w_i \leqq 1$。従って，$\hat{W}_q(p) \in [0,100]$

4.2 TYPE II：順序（ランク）が含まれる集約

既存の検索エンジンの結果に利用頻度を加えるメタ検索の場合，順序と数値が両方含まれる。そのため，集約に少し手間がかかる。これは順序結果を数値に統一する必要があるからであるランクをスコアに変える方法についての研究（[1]）が盛んである。これらはメタ検索エンジンにおける，複数の検索エンジンの検索結果 (ランク) を集約する研究である。

（1） 逆数法

ランクの逆数をスコアとし，現れたすべてのランクによるスコアを足し合わせる

$$W_q(p) = \frac{K}{Rank(p)^c} \quad (6)$$

K と c は定数であり，c はランクのスコアへの変換の際における重みである。c が大きいほど上位と下位のスコアの差は大きくなり，小さいほど上位と下位のスコアの差は小さくなる。

（2） ボルダ法

ボルダ法 [17] は l のランクをそのまま利用する方法で，利用状況を反映させた式は次のようになる。

$$W_q(p) = Max - Rank(p) \quad (7)$$

Max は順序リストの長さ，あるいは最大ランクである。
この場合は重要度基準はリンク構造が分からないため計算できないこともあるし，利用する検索エンジンにすでに重要度評価が行われたこともあるので，ここで，省略する。また，順序 (ランク) から得た数値スコアも標準化し，結果を $\bar{W}_q(p)$ とする。

$$\hat{W}_q(p) = w_1 \cdot \bar{W}_q(p) + w_2 \cdot \bar{F}(p) \quad (8)$$

$w_i \in [0,1], (i = 1,2)$，$\sum_i w_i \leqq 1$。従って，$\hat{W}_q(p) \in [0,100]$

5. プロトタイプシステムの構成

提案した多基準検索方式を検証するため，我々は現在プロトタイプシステム MCDM を実装している。図 2 はこの検索システムのユーザインタフェースを示している。このシステムは普段のキーワード入力を受けるほか，利用状況，利用地域，利用時期を指定することができる。利用状況は「考慮しない」,「よく利用」の項目が指定できる。「よく利用」を選べば，利用履歴から抽出した ABF に維持している利用頻度情報を検索結果に反映させる。

また，利用の地域を指定するならば,「日本関東」,「日本関西」,「米西海岸」,「米東海岸」などを選ぶことができる。地域を選択すれば，その地域の ISP の履歴データを使うことになる。最後に，利用の時期も選択することができる。時期を季節にすれば，春，冬，夏，秋が選べる。時期によって，履歴データを

図 2 MCDM 検索システムの利用者インターフェース
Fig. 2 Interface for MCDM Prototype System

指定された時期のデータを使う。

このシステムは以下のような部分から構成される (図 3)。

（1） 検索インターフェース。上に説明した検索条件を入力する画面を提供する。このインターフェースを通して，検索式，使うべき利用履歴の区分を指定する。

（2） 関連性条件処理部分。この部分はシステムの実装によって役割が異なる。既存の検索システムを利用する場合，この部分は外部の（検索）システムとのインターフェース役になる。外部からの検索結果を受けとった後，集約バッファーに送り込む。もし，システム自体で関連性検索を行う場合は，この部分はそれを実現し中間結果を集約バッファーに送る。

（3） コンテンツ索引管理部分。関連性次元，並びに信頼性次元を効率よくするための索引データの作成，更新。

（4） 履歴管理部分。ウェブログから ABF によって必要な利用状況が抽出される。候補結果を与えれば，素早く ABF に利用頻度を算出し，集約バッファー管理部分に渡す。

（5） 集約バッファー部分。複数の次元からの結果や中間結果を取りまとめて，集約した結果をクライアント（利用者）に返す。

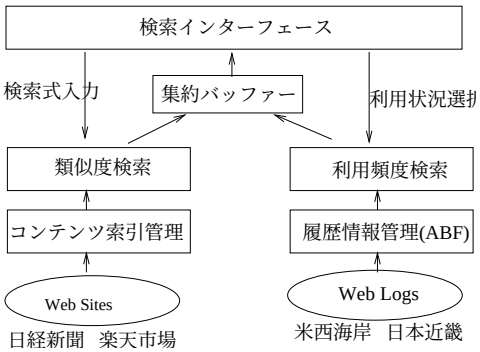

図 3 MCDM 検索システムの構成
Fig. 3 Architecture for MCDM Search System

6. 実験評価

本節では、提案する多基準検索の実験評価を行う。実験に使われるデータは，京都市の運営する大規模 ISP，Kyoto I-net のプロキシログを利用した。発信元を表わす電話番号の局番からみると、ほとんどの利用者は京都市に住んでいることが分かった。このデータは 2002 年 1 月 15 日から 2 月 14 日にかけておよそ 1ヶ月分のもので。1 日のリクエスト数が約 140 万回程度である。そのうち HTML リクエストは 20%程度であり約 30 万回である。ユニークな HTML ページ数は 5,234,099 があった。

表 2 「スポーツ」に関する検索結果

検索方式	利用度考慮せず	利用度考慮	
ランク	検索結果	検索結果	ランク変化
1	EC ポータル	サッカー	+1
2	サッカー I	競馬	+4
3	占い	スキー	+8
4	EC メガネ	プロ野球 II	+12
5	スポーツ用品	EC ポータル	-4
6	競馬	プロ野球 II	+13
7	EC プロレス	トピック	+13
8	EC ゴルフ用品	MLB	+1
9	MLB	ニュース	+13
10	サッカー II	芸能	+14

表 3 「プレゼント」に関する検索結果

検索方式	利用度考慮せず	小さな λ 値 (0.2)	
ランク	検索結果	検索結果	ランク変化
1	香水 I	カニ I	+2
2	香水 II	香水 I	+1
3	カニ I	宝石 I	+5
4	カニ II	宝石 II	+12
5	携帯用グッズ	玩具	+14
6	美容品	酒	-1
7	酒	食品	+6
8	宝石 I	洋食器	+12
9	ブランド用品	ダイエット	+14
10	カニ III	香水 II	-8

6.1 実験 I：トップページに対する多基準検索

実験 I は、上記の利用履歴データに現れたページを収集し、インデックスを付けて検索を行った。利用履歴に現れたページは一部なくなったり、変更されたりすることが可能なので、我々はさらに利用者がブラウザに登録していたトップページを中心に、32,938 ページを集めた。その中、有効なトップページが 4,192 ページであった。これらのページに対して、Namazu でインデックスを作成し得られたキーワードの数は 503,843 個であった。

実験用システムは Namazu をベースにして構築されている。ABF による利用頻度情報の抽出と利用機能と、TYPEI 型集約関数の実装を加えた。利用履歴の新鮮さが検索結果に対する影響を確認するため、異なる λ 値を使って、実験を行った。表 2、表 3 と表 4 はそれぞれの結果を示している。

6.2 利用頻度の有効性

利用頻度の有効性を検証するために、検索条件を ”スポーツ ”とし，利用状況を ”考慮しない ”と ”よく利用（利用頻度を考慮する）”で検索を行いその検索結果の比較を行う。2 つの検索結果の順位と内容，EC サイトでは取扱内容を表 2 に示す。表 2 はそれぞれの検索結果に現れたページの順位と，コンテンツの説明を載せている。表の右端には，利用状況を考慮しないで検索した時の順位からの差分を載せている。

利用頻度を考慮しない検索では，上位 10 位にはスポーツ用品を対象とした EC や，サッカー，占い等に関するページが現れている。とりわけ EC 関連が多いことが分かる。利用頻度を考慮した検索結果は，EC が順位を下げ，スキー，競馬と言った季節に相応しいスポーツに関するページが順位を上げている。サッカーはワールドカップや選手の移籍の話題，競馬はこの時期に地元の京都競馬場で開催されるレースが多いことから影響が出たと推測される。これは履歴ウェブより計算した利用者集合の ”地域性 ”や ”時期 ”の特性が正しく反映した結果と考えることができる。多くのキーワードをメタデータに含む EC サイトを，ノイズとして除去できていることも注目すべき点である。

6.3 λ 値の影響

λ の値の評価を行うために，利用状況を ”考慮しない（$\lambda \to 1$）”と ”最近の利用頻度を考慮する（$\lambda \to 0$）”場合の検索を行う。

表 4 小さな λ 値 ($\lambda = 0.2$)

検索日	1 月 31 日	2 月 7 日	2 月 14 日
1	香水 I	香水 I	カニ I
2	香水 II	香水 II	香水 I
3	カニ I	下着	宝石 I
4	携帯	玩具	宝石 II
5	美容品	カニ I	玩具
6	酒	カニ II	酒
7	宝石 I	カニ III	食品
8	ブランド用品	鞄	洋食器
9	カニ II	魚	ダイエット
10	食品	洋服	香水 II

$\lambda \to 1$ の場合の結果を比較するため、1 月 31 日，2 月 7 日，2 月 14 日の三日において，利用状況を ”最近利用 ”として検索を行った。最近に利用されたものを重視する小さな λ 値の場合に相応しいキーワードで評価するのが適当である。今回は，キーワードを ”プレゼント ”として検索を行った。それぞれの日の検索結果はに表 4 で示されている。ここではそれぞれの検索結果のコンテンツの説明を表に記すのみとする。検索結果に表れるコンテンツは，ほとんどが EC に関するものであったため，表におけるコンテンツの説明は，その EC サイトで扱っている商品についての説明である。

表 3 によると 2 つの検索結果に違いがあることが確認できる。表 4 をよると，3 日間における URL の順位の変動は激しい。これにより 1 週間単位でも検索結果に違いが表れることが確認できた。

2 月 14 日における検索において，宝石に関するコンテンツが 3 位と 4 位と 2 つも上位に現れているが分かる。1 月 31 日においては 7 位に，2 月 7 日においては宝石に関するコンテンツは 10 位以内表れていないことから，2 月 14 日付近になって利用した利用者がいることが分かる。香水に関するコンテンツが上位に現れているが，普段 EC でよく購入されるものとは考えにくい。宝石や香水と言ったコンテンツが上位に現れていることは，バレンタインデーの影響と推測される。

”プレゼント ”のような、利用者の興味の中でも，時期による変化が激しい検索条件を用いて検索を行う時は，λ 値が小さな時、最近の利用傾向が多く反映されたと見える。

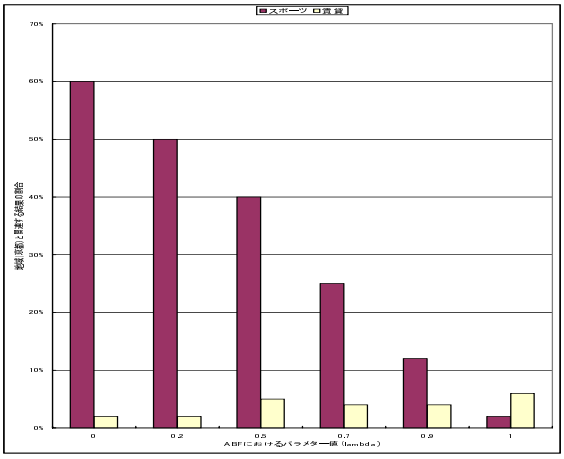

図 4　利用の地域（京都）と検索結果の関連度

Fig. 4　The percentage of top 10 results related to region(Kyoto)

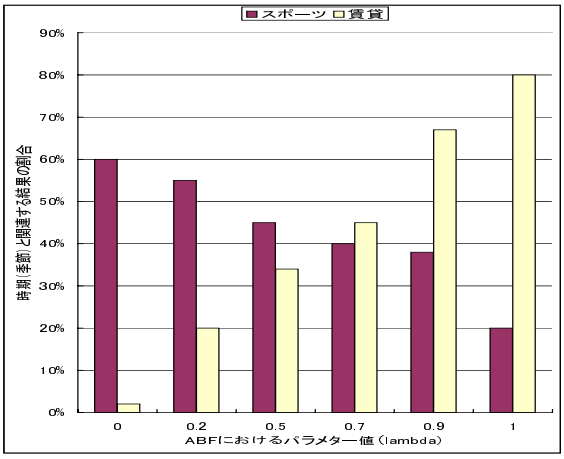

図 5　利用の時期（季節）との検索結果の関連度

Fig. 5　The percentage of top 10 results related to season(winter)

上記の通り利用状況を考慮することで，地域性や時期が考慮されより効果的な検索結果が得られることが分かった。このことは提案する方式による専用検索が，従来の（専用）検索システムよりも優れていることを示している。

6.4　実験 II：メタ検索エンジンとしての多基準検索

既存の検索エンジンを用いて、そこから最初の検索結果を受け取りる。その結果に利用頻度を適用して、最終の結果を出力する実験を行った。特に、Google の検索結果（ランク）に利用状況を反映させその結果がどのように変化するかを確かめてみた。Google の検索結果は 2003 年 5 月 19 日に行ったものを利用した。キーワードは ”スポーツ ”と ”賃貸 ”で上位 300 の結果を収集し，それらのランクに TYPEII 型のランク集約手法で利用状況を反映させ評価を行った。Google の検索結果において親子の関係で同時に表示されるサイトは親のみを利用し子サイトは無視した。

キーワードを「スポーツ」と「賃貸」として検索を行った時の上位 10 の検索結果は利用地域（京都）、利用時期（冬季）との関連度を確かめた。

図 4 は検索結果と利用の地域性の関連度を示している。この図を見れば、「スポーツ」に関する検索結果は地域と密接に関連していることが分かるが、「賃貸」に関しては、検索結果が地域京都との関係が非常に薄く、予想以外の結果になった。しかし、ちょっと考えると、実は 1 月中旬から 2 月初旬にかけて、引越しする人が少なく、賃貸関係のウェブサイトはあまり利用されていないことが原因と考えられる。

図 5 は検索結果と利用の時期性（季節）の関連度を示している。「スポーツ」においては、最近の利用頻度を無視する（$\lambda \to 1$）ことにつれ、冬との関連度が弱くなってしまう。その反面、「賃貸」に関する検索結果は冬との関連度が強くなる。これは「スポーツ」に関する話題は最近の動きに影響を受けやすい。最近の利用状況を無視してしまうと、季節との関連度が弱くなる。逆に、「賃貸」の検索結果は最近の利用状況に依存しないので、最近の利用状況を無視することで、季節との関連度も感じられるようになった。

7.　結論と今後の研究

本稿では検索条件との類似度、情報の重要度に加え、情報の動的な利用頻度を検索基準に使う多基準検索のアプローチを提案した。利用頻度情報を効率よく扱うため、エージング・ブルーム・フィルタ（ABF）を開発し、時間的に変化する利用頻度情報を検索に反映させることができた。提案するシステムは利用する ISP によって検索結果が異なる．ISP を一つのコミュニティーと見れば，これは検索のパーソナライゼーションといえる．実験では，トップページ検索とメタ検索の 2 つのプロトタイプ検索システムを用いて，その提案する検索の優位性と可能性を示した．今後、より大規模のログデータと検索対象となるウェブページを使って、より全面的に評価を行う予定である。また、ABF の効率性の理論証明と実験検証にもいくつかの課題が残っており、これから取り組む予定である。

謝　　辞

本研究の一部は CREST プロジェクト「デジタルシティのユニバーサルデザイン」よりサポートをいただいております。一ヶ月前に逝去されました故上林先生のご指導・ご助言をいただき，誠に感謝いたします。先生のご冥福をお祈りいたします。

文　　献

[1] C. Dwork ,R. Kumar ,M. Naor and D. Sivakumar, Rank Aggregation Methods for the Web. In Proceedings of the Tenth International World Wide Web Cpnferece,2001.

[2] L. Page, PageRank: Bringing order to the Web. Stanford Digital Libraries Working Paper, 1997.

[3] J. M. Kleinberg, Authoritative sources in a hyperlinked environment. In Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms, pp 668–677, 1998.

[4] G. Salton, A. Wang, and C. Yang, A vector space model for information retrieval. In Journal of the American Society for Information Science. vol 18, pp 613–620, 1975.

[5] M. Hansen and E. Shriver, Using Navigation data to improve IR functions in the context of Web search. In Proceedings of the Tenth International Conference on Information and Knowledge Management (CIKM 2001), pp 135–142, 2001.

[6] B. H. Bloom, Space/time tradeoffs in hash coding with allowable errors. Communications fo the ACM, vol 13, no 7,

pp. 422-426, 1970

[7] Saar Cohen and Yossi Matias, Spectral Bloom Filters, SIGMOD Conference 2003: 241-252

[8] Y. Kambayashi and K. Cheng, Capacity Bound-free Web Warehouse, First Biennial Conference on Innova-tive Data Systems Research, 2003.

[9] 山名早人, 近藤秀和, サーチエンジン Google, 情報処理 42 巻 8 号 (2001 年 8 月), pp.775-780

[10] 原田昌紀, 道しるべ：WWW サーチエンジンの作り方, 情報処理 41 巻 11 号 (2000 年 11 月), pp.1280-1283

[11] 原田昌紀, サーチエンジンにおける検索結果のランキング, bit 2000 年 8 月号 (Vol.32), pp.8-14

[12] 徳永 健伸, 情報検索と言語処理 言語と計算, 東京大学出版会 (1999 年 11 月)

[13] 成凱, 平野真太郎, 上林彌彦, プロキシログ解析に基づくトップページの抽出と検索, 第 14 回データ工学ワークショップ, 2003.

[14] B.U. Oztekin, G. Karypis , V. Kumar, Expert Agreement and Content Based Reranking in a Meta Search Environment using Mearf,

[15] 向亨, 成凱, 上林彌彦, 利用履歴に基づく PageRank アルゴリズムの改良, 第 13 回データ工学ワークショップ, 2002.

[16] T. H. Haveliwala, Topic–Sensitive PageRank, In proceedings of the 11th International World Wide Web Cpnference,2002.

[17] J. C. Borda, Mémoire sur les élections au scrutin, Histoire de l'Académie Royale des Sciences, 1981.

[18] C. Cortes and D. Pregibon, Giga–Mining, Knowledge Discovery and Data Mining, pp174-178, 1998.