ファイル検索に向けたアクセスログからのファイル間関連度の導出

Inter-file Relationship Derivation from Access Logs to Search for Files

渡部 徹太郎 ♥　小林 隆志 ♦
横田 治夫 ♠

Tetsutaro WATANABE　Takashi KOBAYASHI
Haruo YOKOTA

急速に増え続けるファイルに対して，デスクトップサーチなどのキーワード検索型の検索手段が用いられるようになってきているが，ファイルにキーワードを含まない画像ファイルやデータファイルなどを見つけ出すことはできない．本稿では，キーワードを含まないファイルも検索できるように，同時アクセスされるファイル間の関連性を利用することを想定し，ファイルサーバのアクセスログから数値化した関連度を導出する方法を提案する．さらに，被験者実験により導出された関連度が，もともとのファイル間の関連性を反映していることを確認する．

Keyword based search functions like the desktop search have been used for files increased rapidly. However, these functions are ineffective for files including no keyword, such as picture files or data files. In this paper, we propose a method of deriving the relationship value from access logs of a file server to enable search for files including no keyword using the relationship. We also evaluate how the derived values reflect the original relationship between files.

1. はじめに

近年，ファイルシステム内のファイルの数，サイズ共に爆発的に増加している [1]．増え続けるファイルに対して既存のファイルシステムではディレクトリ階層構造が提供されているが，膨大なファイル群に対して適切にディレクトリ階層を構成することは難しい．また，全てのファイルが適切にディレクトリ階層に格納されていたとしても，ファイル名が不適切であったり，ファイルが深い階層に格納されていた場合，ディレクトリ構造を辿るだけでは欲しいファイルを探すのは困難である．

そのような背景から，ファイルシステムに対して全文検索を提供する，Google Desktop[2]， Windows Desktop[3]，Spotlight[4]，Namazu[5] を利用したファイル検索などが用いられてきた．しかし，全文検索ではキーワードを含まないファイルは，キーワードと関連があっても検索できないという問題点がある．例えば論文ファイルは，その論文のタイトルや提案している手法名で検索できるが，その論文中の図の画像ファイルや実験で用いたデータファイルは，その論文と関連があるが，テキストファイルではないため検索できない．しかしこのようなファイルを検索したいという要求は高い．この要求に対し，画像に対するキーワード検索では Google Image Search [6] が提案されているが，ファイルシステム内のファイルには画像参照情報が無い場合が多く適用できない．

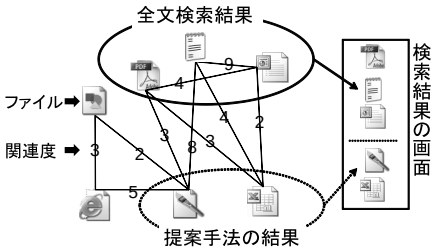

図 1　本研究で提案するファイル検索の方法の例
Fig. 1　An example of our search method

本稿では全文検索できないファイル群に対して，キーワード検索を行うために必要なファイル間関連度を算出する手法を提案する．また，関連度算出の実装を行い，被験者実験により導出された関連度が，もともとのファイル間の関連性を反映していることを確認する．

2. 提案するファイル検索方法の概要

本研究では全文検索できないファイル群に対してキーワード検索を行う為に，準備としてファイルシステム内のファイルに対してファイル間関連度を数値化し格納する．キーワード検索要求が来た際は，まずキーワードで全文検索を行い，その結果と関連度の高いファイル群を検索結果に添えて提示する．これにより，全文検索できないファイル群に対してもキーワード検索を実現できる (図 1 参照)．

3. アクセスログを利用した関連度の抽出

前節で説明したファイル検索を実現するためのファイル間関連度として，本研究では頻繁に同時に使うファイル群に着目した．コンピュータ上である作業をする際，関連するファイルを同時に開いて，参照しながら，あるいは必要な部分をカットアンドペーストしながら進めることが多い．このような場合，作業毎に使用するファイル群は類似するため，頻繁に同時に使われるファイル群は関連が高いと考えられる．本研究ではこのようなファイル群を求めるために，ファイルサーバのアクセスログからユーザのファイル使用情報を抽出する．

3.1 ファイル使用情報の抽出

本研究ではファイルサーバのアクセスログからそれぞれのユーザ毎にファイルのオープン／クローズの情報を取得し，ファイル毎の使用時間情報を抽出する．基本的にはファイルのオープン時刻からクローズ時刻までをファイル使用時間とするのだが，対象

♥ 学生会員　東京工業大学大学院情報理工学研究科計算工学専攻　tetsu@de.cs.titech.ac.jp
♦ 正会員　名古屋大学大学院情報科学研究科　tkobaya@is.nagoya-u.ac.jp
♠ 正会員　東京工業大学学術国際情報センター, 東京工業大学大学院情報理工学研究科計算工学専攻　yokota@cs.titech.ac.jp

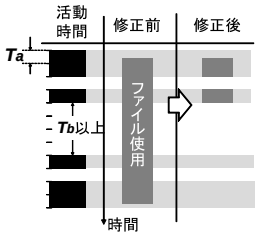

図 2 問題 (B) への対処
Fig. 2 Access duration modification dealing with the problem (B)

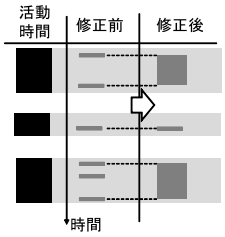

図 3 問題 (C) への対処
Fig. 3 Access duration modification dealing with the problem (C)

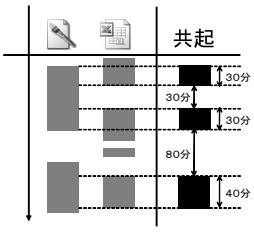

図 4 共起の例
Fig. 4 An example of the duration of the simultaneous accesses

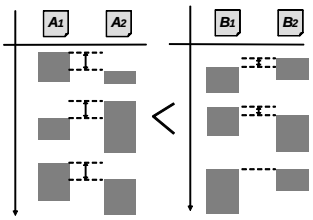

図 5 使用開始時間の類似度の考慮
Fig. 5 Consideration of open-timing similarity

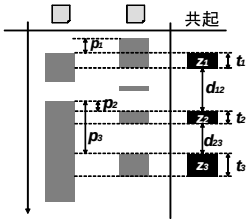

図 6 関連度計算で用いる変数
Fig. 6 Variables for calculating the relationship degree

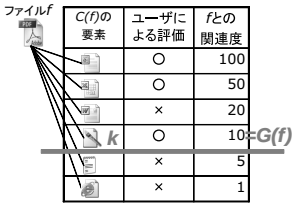

図 7 ユーザによる採点の例
Fig. 7 An example of ratings by a user

を Windows に限定し，Samba を利用したファイルサーバのアクセスログを解析した結果，オープン／クローズと実際のファイル使用が異なる以下の問題があることが分かった．

A) いくつかの理由で，あるオープンに対するクローズが取れず，正確な使用時間が計算できない場合がある．
B) ユーザはファイルを開いたまま席を立つ場合がある．
C) ファイルをロックをするアプリケーションとメモリに読み込んで，ファイルのロックを開放してしまうアプリケーションがあり，後者の場合はオープン／クローズが実際のファイルの使用時間と一致しない．

本研究では，これらの問題に対し，以下のように対処することで，実際のファイル使用時間に近づける方法を提案する．

(A) への対処は，ログを先頭から探索していき，オープンを見つけたら，対応するクローズを再帰的に探す．対象のログの末尾まで対応するクローズが発見できない場合は，そのオープンの直後にクローズを補完する．(B)，(C) への対処のために，まず以下の情報をアクセスログから抽出する．

ア) ユーザの活動時間: ユーザのアクセスログから T_a の間隔でログの有無を調べ，ログがあれば活動していたと見做し，活動の有無を時系列でリストにしたもの．

イ) 直ぐに閉じてしまうファイルタイプのリスト: アクセスログから拡張子別にファイル使用時間の平均を調べ，平均が T_c 以下のファイルタイプをリストにしたもの．

(B) のログへの対処法は，まず (ア) より活動時間外のファイル使用を消す．加えて，T_b 以上活動時間が空いた場合は，ファイルを閉め忘れて帰宅したと見做して，それ以降のログを消す (図 2 参照)．(C) のログへの対処法は，(イ) の対象となるファイルタイプに対して，(ア) の活動時間内で最初のファイルアクセスから最後のファイルアクセスまでをずっと使用していたと見做して，使用時間を拡大する (図 3 参照)．

3.2 関連度の算出

前節で作成したファイル使用情報を基にファイル間の関連度を計算する．関連度の算出方法の説明の前に，関連度の大小関係を定義する．

3.2.1 関連度の大小関係

本研究では，関連度の高いファイル同士は長い期間，各作業の度に，同じタイミングで利用されるものと仮定する．そのため，関連度の大小関係を考慮する上で以下の 4 つのルールを導入した．

ルール 1 共起時間の累計が長いほど関連度が高い．
ルール 2 共起回数が多いほど関連度が高い．
ルール 3 共起の間隔が離れているほど関連度が高い．
ルール 4 ファイル使用開始パターンが類似しているほど関連度が高い．

これらのルールについて例を挙げて説明する．ルール 1，ルール 2，ルール 3 に関して，ファイル使用時間表が図 4 の様になっている場合，ファイル使用時間表の重なっている部分を「共起」と呼び，この例では共起時間の累計が 100 分であり共起回数が 3 回である．これらの値が大きいほど関連度が高いとする．また共起の間隔時間の合計は 110 分であるが，この値が大きいほど長期間経過後も二つのファイルを一緒に使っていたことになるので，値が大きいほど関連度が高いとする．ルール 4 に関して，図 5 の A_1, B_2 の方が B_1, B_2 よりファイル使用開始時間が近いものが多く，一緒に使い始めることが多いと考えられるので，より関連度が高いとする．

3.2.2 関連度の算出

上述の大小関係を考慮した関連度の算出方法を説明する．二つのファイルの共起時間の合計を T，共起回数を C，共起の間隔度を D，ファイル使用開始パターンの類似度を P としたとき，関連度 R は $\alpha, \beta, \gamma, \delta (0 \leq \alpha, \beta, \gamma, \delta \leq 1)$ を定数として

$$R = T^{\alpha} \cdot C^{\beta} \cdot D^{\gamma} \cdot P^{\delta}$$

となる．また同時に使用したことがない二つのファイルの関連度は 0 とする．加えて二つのファイルが同じディレクトリに属して

いる場合，すでに使用者によって関連があると分類されているので，検索対象から除くために関連度を 0 とする．

各値の算出方法を説明する．図 6 を参照されたい．二つのファイルの各共起を $\{z_1, z_2, \cdots, z_n\}$ とし，その時間を $\{t_1, t_2, \cdots, t_n\}$，共起の間の時間を $\{d_{12}, d_{23}, \cdots, d_{(n-1)n}\}$，さらに各共起の基となっている二つのファイル使用の使用開始時間の差を $\{p_1, p_2, \cdots, p_n\}$ として，各値は下記のように計算する．

$$T = \sum_{i=1}^{n} t_i \qquad C = n$$

$$D = \begin{cases} 1 & n = 0 \\ \sum_{i=1}^{n-1} d_{i(i+1)} & o.w. \end{cases} \qquad P = \begin{cases} 1 & \forall i \;\; p_i = 0 \\ \left(\sum_{i=1}^{n} p_i\right)^{-1} & o.w. \end{cases}$$

3.2.3 関連度のパラメータの決定

関連度 R のパラメータ $(\alpha, \beta, \gamma, \delta)$ の決定法を説明する．本研究ではパラメータの決定のために，ユーザのファイル関連に対する採点を用いる．

■手順 1 採点の対象となるファイル集合を H とする．各 $f \in H$ に対して $C(f)$ をファイル f と共起しているファイルの集合とする．$C(f)$ の各要素に対してファイルの使用ユーザが見て関連があるものを○，関連がないものを×の二値に分類する (図 7 参照)．

■手順 2 各 $f \in H$ に対して，ファイル別閾値 $G(f)$ を求め，その平均を関連度閾値 $\overline{G}$ とする．$G(f)$ の求め方は，f に対して $C(f)$ の中で「○の評価が付いている」かつ「最も f との関連度が最も低い」要素 k を求め，$G(f) = R(f, k)$ とする．言い換えると，$C(f)$ 内の関連度 $G(f)$ 以上の要素の集合が○の再現率を 100 %とするような $G(f)$ を求めている．

■手順 3 各 $f \in H$ に対して，ファイル別再現率 $REC(f)$ を求め，その平均を再現率 $\overline{REC}$ とする．$REC(f)$ の求め方は，まず f に対して $\hat{C}(f) = \{c | c \in C(f), R(f, c) \geq \overline{G}\}$ とする．つまり $\hat{C}(f)$ は $C(f)$ の中で関連度が $\overline{G}$ 以上の要素の集合である．次に各 f に対してファイル別再現率 $REC(f)$ を以下のように計算する．

$$REC(f) = \frac{\hat{C}(f) \text{ 中の○の数}}{C(f) \text{ 中の○の数}}$$

■手順 4 $\overline{REC}$ はあるパラメータの基で求まるので $\overline{REC}(\alpha, \beta, \gamma, \delta)$ と書き直す．このとき最適パラメータ $(\alpha^*, \beta^*, \gamma^*, \delta^*)$ を以下のように計算する．

$$(\alpha^*, \beta^*, \gamma^*, \delta^*) = argmax(\overline{REC}(\alpha, \beta, \gamma, \delta))$$

本研究では，この最適パラメータを関連度算出に用いるとともに，最適パラメータを与える関連度閾値を最適閾値 G^* とし，ファイル検索の際に関連度が G^* 以上の関連のみを考慮することによって，検索結果のフィルタリングに用いる．

4. 評価実験

本研究では，節 3. で提案した関連度が，もともとのファイル間の関連性を反映しているかどうかを，被験者実験により確認する．

4.1 実験環境

実験では，現在広く用いられている Windows 互換ファイルサーバの Samba2.2.3a をログレベル 2 で実行し，それを WindowsXP から二人の被験者に約 4ヶ月間使ってもらい，アクセスログを採取した．システムファイル等のアクセスは無視するように解析対象の拡張子は bib，doc，gif，htm，html，jpg，mpg，mpeg，pdf，ppt，tex，txt，xls とした．また 3.1 節の各定数は $T_a = 30$[分] , $T_b = 5$[時間] , $T_c = 10$[秒] とした．

4.2 実験

まず，それぞれの被験者のログからファイル使用時間表を作成した．次に全ファイルの中からランダムに選び，被験者に二つのファイルの関連に対して○×採点を行ってもらった．その中からランダムに半分のファイルを選択し学習用セットとし，そこから計算された最適パラメータと最適閾値は以下のようになった．

$(\alpha^*, \beta^*, \gamma^*, \delta^*)$	(0.2, 0.3 , 0.1 ,0.3)	(0.1 , 0.6 , 0.2, 0.8)
G^*	0.628	0.150

これらの値を用いて，評価セットの各ファイル f に対して再現率 $REC(f)$ と適合率 $PRE(f)$ を求め，それぞれの値を評価セット内のファイルで平均したものが以下のようになった．

	被験者A	被験者B
再現率の平均値	81.0 %	82.6 %
適合率の平均値	66.2 %	45.0 %

4.3 考察

関連度のパラメータについては，共起時間の累計だけを考える (1,0,0,0) の組み合わせや，共起回数だけを考慮する (0,1,0,0) の組み合わせよりも，回数，使用開始パターンの類似，共起の間隔を被験者に合わせて組み合わせた方が，より有効な関連度を計算できることがわかった．

再現率，適合率に関しては，本研究では全文検索で見つからないファイルを探すのが目的であるため，多少不適合ファイルがあっても再現率の向上を優先する．そこで，被験者 A，B 共に再現率の平均が 80 %以上であった結果より，提案する検索手法を実現する際に，この関連度を用いることが可能であると考える．

ログのクリーニングに関しては評価実験は行っていないが，具体的な事例として，(A)(B) の問題を修正したことによって本来関連の無い二つのファイルの共起が除去されたことを確認している．また，(C) の問題を修正したことにより，ある論文の Tex ファイルのログが修正され，論文の図の画像ファイルや，グラフの基となっている実験データファイルと論文の Tex ファイルの共起が抽出できた．このような事例が観測されたため，ログのクリーニングは有効であったと考える．

5. 関連研究

関連研究について三つの観点から議論する．

5.1 階層構造の問題点への解決

Grifford らの Semantic File Systems[7] では，各ファイルはディレクトリに所属しない代わりに複数の属性を持ち，この属性を用いてファイルを管理するファイルシステムであり，情報共有等に役立つと報告している．また，石川らの研究 [8] では属性の表現に RDF[9] を用いることにより，セマンティックウェブ関連の技術の応用を容易にし，ファイルの整理や一貫性の管理を実現している．これらの研究は階層構造の問題点への解決という点では本研究と一致するが，キーワードに関連するものを探すことは

していないので，その点で本研究と異なる．

5.2 ユーザの動作履歴利用

俺デスク [10] はイベントログと組み込んだプラグインから動作履歴を抽出し，それを基にウェブページやファイル (以降まとめてデータと呼ぶ) の着目度，データ間の関連度を算出する．そして，関連度を基にしたデータ名による関連検索と，時系列のビューアを提供するシステムである．俺デスクでは，動作履歴を抽出するためユーザ環境に手を加える必要があるが，本研究ではアクセスログを利用するので，手を加えること無くシステムを実現できる．また，俺デスクの検索ではデータ名を入力し関連しているデータを提示するものであって，本研究で提案するキーワードによるファイル検索とは異なる．関連度の算出においても，俺デスクではデータアクセス開始時間のみを考慮しているのに対し，本研究では共起時間の累計，共起回数，間隔等を考慮しており，関連ファイル間の識別能力に違いがある．

5.3 データ間関連度

増田らの近傍検索システム [11] では，ファイルを含むディレクトリ構造，ファイルの更新時間，ファイルの内容の３つの情報を基に，ファイル間の関連度を算出し，ファイル間の連想アクセスシステムを提案している．しかし，ファイル内容の関連度計算は文書ファイルに限っている．これに対して，本研究では文書ファイル以外の画像ファイル，データファイル等のファイルに対しても等しく関連度を算出できる．

6. まとめと今後の課題

既存の全文検索ではキーワードを含んでいないファイルは，そのキーワードと関連があっても検索できないという問題点があった．本稿では全文検索できないファイルをキーワード検索する手法を実現するために必要なファイル間関連度を，同時に用いられるファイル間の関係に着目し，ファイルサーバへのアクセスログから抽出する手法を提案した．また関連度算出を実装し，その関連度を用いてあるファイルと関連度の高いファイル群を提示した所，そのファイル群の正解 (ユーザが関連があると判断した）の再現率が 80 %以上であった．この結果より，提案する検索手法を実現する際，この関連度を用いることが可能であると考える．

今後の課題としては以下の 4 点が挙げられる．

- 検索手法の実装を最後まで行い，その評価を行う．
- より多くのユーザを対象として評価実験を行う．
- 関連の方向や時間的減衰を考慮に入れた関連度算出手法の考案する．
- ファイルの使用時間の情報のみでなく，ファイルのコピーや移動をアクセスログから抽出し検索結果のフィルタリングに利用する．

[謝辞]

本研究の一部は，文部科学省科学研究費補助金特定領域研究 (19024028)，東京工業大学 21 世紀 COE プログラム「大規模知識資源の体系化と活用基盤構築」および独立行政法人科学技術振興機構戦略的創造研究推進事業 CREST の助成により行なわれた．

[文献]

[1] Nitin Agrawal, William J. Bolosky, John R. Douceur, Jacob R. Lorch: A Five-Year Study of File-System Metadata, 5th USENIX Conference on File and Storage Technologies(FAST'07), 2007.

[2] About Google Desktop, http://desktop.google.com/about.html

[3] Windows Desktop Search, http://www.microsoft.com/windows/desktopsearch/default.mspx

[4] Spotlight, http://www.apple.com/jp/macosx/features/spotlight/

[5] 全文検索システム Namazu, http://www.namazu.org/

[6] Google Image Search, http://images.google.com/

[7] D.K.Gifford, P.Jouvelot,M.A.Sheldoon, J.W.O'Toole,Jr: Semantic File Systems, 13th ACM Symposium on Operating Systems Principles, 1991.

[8] 石川憲一, 森嶋厚行, 田島敬史: 大規模ドキュメント空間管理のための意味ファイルシステムの構築, 信学技報 DE2006-115 pp.139-144,2006.

[9] W3C.Resource Description Framework(RDF), http://www.w3g.org/RDF

[10] 大澤亮, 高汐一紀, 徳田英幸: 俺デスク:ユーザ操作履歴に基づく情報想起支援ツール, 情報処理学会第 47 回プログラミング・シンポジウム, 2005.

[11] 増井俊之, 塚田浩二, 高林哲: 近傍関係にもとづく情報検索システム, 11th Workshop on InteractiveSystems and Software (WISS2003), 2003.

渡部 徹太郎 Tetsutaro WATANABE
平 18 東工大・工・情報工卒．同大大学院・情報理工・計算工・修士課程在学中．日本データベース学会学生会員．

小林 隆志 Takashi KOBAYASHI
平 9 東工大・工・情報工卒．平 11 同大大学院・情報理工・計算工・修士課程了．平 16 同大大学院・同専攻・博士課程了．平 14 同大学学術国際情報センター・助手．平 19 名大大学院・情報科学研究科・特任准教授．日本データベース学会，電子情報通信学会，日本ソフトウェア科学会，情報処理学会，ACM 各会員．

横田 治夫 Haruo YOKOTA
昭 55 東工大・工・電物卒．昭 57 同大大学院・情報・修士課程了．同年富士通（株）．同年 6 月（財）新世代コンピュータ開発機構研究所．昭 61（株）富士通研究所．平 4 北陸先端大学・情報・助教授．平 10 東工大・情報理工・助教授．平 13 東工大・学術国際情報センター・教授．工学博士．日本データベース学会理事．電子情報通信学会，情報処理学会，人工知能学会，IEEE，ACM 各会員．