論文 / 著書情報
Article / Book Information

論題(和文)	複数種類の関連度を組合せたファイル分類手法
Title(English)	A File Grouping Method Combining Multiple Types of Interfile-Relationships
著者(和文)	渡辺陽介, 小田切健一, 横田治夫
Authors(English)	Yousuke WATANABE, Kenichi OTAGIRI, Haruo YOKOTA
掲載誌(和文)	DEIM2009論文集
Citation(English)	Proc. DEIM2009
Vol, no, pages	, ,
発行日 / Pub. date	2009, 3

複数種類の関連度を組合せたファイル分類手法

渡辺　陽介† 　小田切健一†† 　横田　治夫†,††

† 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
E-mail: †{watanabe,otagiri}@de.cs.titech.ac.jp, ††yokota@cs.titech.ac.jp

あらまし　近年，個人が管理しなければならないファイルの数と種類が膨大となっており，大量のファイルを分類・整理し，目的のファイル群を効率的に探すための新たな情報管理技術が求められている．我々の研究グループでは，大量のファイルから関連するファイル同士を発見し，仮想的なディレクトリとして提示するシステムの開発を行っている．ファイル間の関連度の計算法として，単語出現頻度に基づく尺度や，アクセスの共起関係に基づく尺度など，様々なものが提案されている．より現実的には，要求に応じてこれらを使い分け，組合せることで必要なファイルのグループが生成できることが望ましいが，異なる関連度尺度を統一的に用いるための枠組みについては，これまであまり提案がされていない．本稿では，複数種類の関連度を利用可能なデータキューブ型ファイルグループ分析ツールについて提案する．本ツールでは，複数の関連度から求めたクラスタリング結果等を多次元のキューブとして統合・操作することができる．利用者はスライスやドリルダウン・ロールアップなどのキューブ操作を行って要求に合ったグループを生成し，特定のファイルに関連するファイル群を見つけ出したり，大量のファイルを整理したりすることができる．
キーワード　ファイル間関連度，クラスタリング，データキューブ

A File Grouping Method Combining Multiple Types of Interfile-Relationships

Yousuke WATANABE†, Kenichi OTAGIRI††, and Haruo YOKOTA†,††

† Global Scientific Information and Computing Center, Tokyo Institute of Technology
†† Graduate School of Information Science and Engineering, Tokyo Institute of Technology
E-mail: †{watanabe,otagiri}@de.cs.titech.ac.jp, ††yokota@cs.titech.ac.jp

Abstract　Since the number of files in PCs is increasing, we require file management tools to find target files and to classify large amount of files. Our research group has been developing a system which provides virtual directories consisting of related files. There are many types of methods to extract inter-file relationships, such as word frequency, access co-occurrence, and so on. In practical, users need to select and combine multiple types of inter-file relationships to form groups of files they want. For this purpose, we propose a tool for file group analysis by introducing data-cube concept. This tool provides cube-like operations for a set of files, and helps users to find relevant file groups.
Key words　Inter-file relationships, clustering, data cube

1. はじめに

近年，ネットワーク上で交換されるデータ量が増加し，メール，文書，画像，動画など，個人の PC に蓄積されるファイルの数と種類は膨大なものとなっている [1]．利用者の記憶だけを頼りに，大量のファイルの中から目的のファイル群を探し出すことは難しくなっており，デスクトップ検索ツール [11], [15] などの支援を得ながらファイルを探索することが行われている．

デスクトップ検索ツールは，ファイルから，パス名，ファイルに付加されたタグ，ファイル内の文字列等に含まれるキーワードを抽出し，索引付けを行っており，利用者が入力したキーワードに合致したファイル群をリストとして提示する．そのため，デスクトップ検索ツールは，特定キーワードを含むファイルをピンポイントで取得する目的に適している．だが，利用者が必要とするファイルが複数ある場合には，必ずしもそれら全てが特定キーワードに合致するとは限らないため，一般には複

数回の検索行動が必要となってしまう．

論理的な関係のある複数のファイルを扱う際には，一部のファイルから残りを芋づる式に取得できるように事前に整理されている状態が望ましい．そのためのアプローチとして，共通の特徴をもったファイル同士をグループ化し，必要に応じてグループ単位で提示できるようにしておくことが考えられる．それは例えば，あるファイルと共通なキーワードを含むファイル群も一緒に提示するということであるが，ファイル同士を関連付ける特徴量はキーワード情報だけではない．最終更新時刻の近さや，パス名の類似性，内部的な参照関係など，様々な観点から関連付けが可能である．我々の研究グループでも，ファイルアクセスログからアクセス共起関係を抽出し，それらをファイル間の関連度として用いる研究を行っている [8]．

これらのような関連度を個別に用いてファイル同士をクラスタリングする研究は従来から行われてきたが，より高度で柔軟なファイル利用を促進するためには，どれかの尺度を単体で用いるだけでなく，複数の尺度を用途に応じて切り替え，あるいは組合せていくことが必要となる．例えば，共通のキーワードを含むという特徴だけではグループ分けが十分にできなかったファイル群を，さらにアクセス共起関係でサブグループに分けることは有用であると考えられる．しかし，これら複数種類のファイル間関連度を統一的に扱うファイルグループ化の枠組みについては，まだ十分な検討がなされていない．

そこで本研究では，複数種類のファイル間関連度を統合可能なファイルグループの分析ツールを提案する．本ツールでは，利用者の要求に合わせて，グループ分けに用いる尺度の選択，グループ化の粒度の調節，条件に合ったグループの選別の操作等を行えるようにするため，それらの操作を実現可能なモデルとして，OLTP で用いられているデータキューブ [4] を用いている．本ツールにおけるキューブの 1 つの次元（ディメンジョン）は，1 種類の関連度に基づいてファイルをクラスタリングした結果を表している．利用者は通常のデータキューブとほぼ同様な操作を行って，必要な関連度において適切な粒度でグループ化の行われた部分キューブを生成する．キューブ操作として，特定の種類の関連度で生成されたグループを条件指定により選別し，部分キューブを切り出す操作であるスライス・ダイスや，各関連度における階層的クラスタリングの結果を用いて，必要に応じてグループ化の粒度を調整する操作であるロールアップ・ドリルダウンを扱う．最終的に生成されたグループは，ファイルの整理やデスクトップ検索の検索結果の拡張に利用される．本稿では，提案ツールの実装とそれを用いた評価実験についても述べる．

本稿の構成は以下のようになっている．まず，2. では関連研究との関係について述べる．次に，3. では本研究で用いるファイル間関連度について紹介する．そして，4. で本研究で提案するデータキューブ型ファイルグループ分析ツールについて説明し，5. 節では実装システムについて述べる．6. では，評価実験の結果を示す．最後に 7. で，まとめと今後の課題を述べる．

2. 関 連 研 究

まず，キーワード検索とキーワード以外の関連情報を組合せた検索システムについて述べる．Google デスクトップサーチ [11] のタイムライン表示機能では，キーワード検索の結果ファイルから，同じ時間帯に更新されたファイル群を時間軸上で探すことが可能である．また，FRIDAL [9] はファイルアクセスログ中の共起関係を用いて，キーワード検索にマッチするファイルだけでなく，共起したファイルをも含めてランキング表示することができる．これらは単純なキーワード検索よりも多くのファイルを探すことができるが，ファイルのグループ化を目的としているわけではない．

ファイルに付与された複数の属性情報に基づいて，ファイルのグループを提示するシステムとしては Semantic File Systems [3] がある．Semantic File Systems では，各ファイルは実体のあるディレクトリに属す代わりに，属性に関する問合せで表現される仮想的なディレクトリに属する．また，ファイルシステムとは異なるが，Windows Vista ではファイルのタグ情報などを検索でき，検索結果を検索フォルダとして保存して再利用する機能が提供されている．これらはファイル自身の属性情報を用いたグループ化であり，ファイル間の関連度は用いてられていない．

ファセット探索（ファセット検索）[5], [6] は，複数の属性情報を持つオブジェクトに対して，属性情報の値を順次指定していくことで対象を絞り込む探索手法である．探索の過程では，選別されたオブジェクトの集合と，さらに絞り込むための属性値の候補が提示される．ファセット探索をファイルのグループを探す目的で利用することは可能である．ただし，ファセット探索においては，ファイル名やタグ情報などのファイル自身の属性情報を用いたグループ化は容易だが，アクセス共起のようなファイル間の関係としてしか表現できないものは扱いが難しい．

参照関係や依存関係，コピーとオリジナルなどのファイル同士の関係を扱うシステムとして，Linking File Systems (LiFS) [2] や InfoSpaceGoverner [7] などがある．これらのシステムでは，ファイル間の関係を説明するリンクを張ることができ，リンクを用いた検索や探索，依存関係に基づいた一貫性のチェックなどの機能を提供している．本研究は，数値化されたファイル同士の関連度からファイルのグループを生成することを目的としており，グラフ構造の扱い等については考慮していない．

3. ファイル間関連度

本研究で用いるファイル間関連度の尺度について紹介する．ここで述べる以外にも様々な尺度は存在するが，ファイル間の関係を数値として表現できるものであれば，提案ツールに取り入れることは可能である．以下では，ファイル x とファイル y の関連度 $Sim(x, y)$ を [0,1] の区間に正規化するようにしている．1 が最も関連が強く，0 は全く関連がないことを意味する．

3.1 テキスト類似度

従来からテキストデータの検索やクラスタリングなどでよく使われている尺度である [10]．共通のキーワードが多く含ま

れているファイル同士には強い関連があるとみなす．ここではファイル x, y に含まれるキーワードを抽出して単語出現頻度による文書ベクトル v_x, v_y として表現し，ベクトル同士の成す角の余弦をテキスト類似度とする．

$$Sim_{text}(x,y) = \frac{v_x \cdot v_y}{||v_x||||v_y||}$$

今回はファイルに付与されたタグ情報やファイル名自体は使用しないものとする．画像などの非テキストファイルの場合は，自身との類似度だけが 1 で，それ以外は 0 とする．

テキスト類似度を用いる利点は，文字列データ限定ではあるが，ファイルの中身に基づいて類似度が算出できることである．利用者が探したいファイル群に共通のキーワードが含まれている場合に有効である．欠点として，当然ではあるが，画像や動画などのテキスト情報が少ないファイル同士の関連度を正確に計算することはできない．また，書類ファイルが定型の書式をベースに作成されている場合などには，書式自身に含まれる単語が共通キーワードとなってファイル群がグループ化されてしまうような事態が起こりうる．

3.2 更新時刻類似度

ファイルの更新時刻同士の近さに応じて決まる尺度である．更新時刻が近いファイルは，一連の作業において作成・編集された可能性が高く，時間が近いほど強い関連があるとみなせる．以下の式で計算する．

$$Sim_{mtime}(x,y) = 1 - \frac{|mtime(x) - mtime(y)|}{max_{a,b \in F}(|mtime(a) - mtime(b)|)}$$

ただし，$mtime(f)$ はファイル f における最終更新時刻（基準時からの経過秒数）を表すものとする．分母は [0,1] 区間に正規化するために入れられている．

更新時刻は，ファイル中のテキスト情報と違って，全てのファイルにおいて利用可能な情報であるという利点がある．ただし，閲覧中心のファイルと編集中心のファイルがあるような場合には，閲覧中心のファイルは更新時刻が変化しないため，編集作業が進めば進むほど，編集中心のファイルとのグループ化が難しくなってしまうという問題点がある．

3.3 パス類似度

ファイルのパス名の近さに応じて決まる尺度である．同一フォルダや親子兄弟関係にあるファイルには関連があると考える．本研究では編集距離に基づき，パスの類似度を算出する．編集距離は文字の挿入・削除・置換によって同じ文字列を得るまでに必要な操作の回数によって表わされる．距離尺度であるので，これを類似度に変換するために，[0,1] 区間に正規化した後，1 との差を取っている．

$$Sim_{path}(x,y) = 1 - \frac{editDistance(x,y)}{max_{a,b \in F}(editDistance(a,b))}$$

パス類似度は，利用者がフォルダ名やファイル名を適切な名前を付与する場合においては有効である．逆に，デスクトップ上などに無秩序にファイルを配置し，「新規テキストドキュメント.txt」といったデフォルトのファイル名のまま使用してしまうような利用者の場合には，パス類似度が役に立たないこともある．

3.4 アクセス共起度

2 つのファイルを同時にアクセスしていた場合，その 2 ファイルは同一の作業に利用していた可能性が高い．そのような共起が長期間や複数回に渡って発生していれば，より強い関係であるとみなすことができる．ファイル使用の共起は，ファイルのオープンとクローズのログを記録することで知ることができる．[9] では，ファイルサーバの Samba [14] のログを用いており，ファイルオープン時刻の近さや共起間隔などの要素も考慮されているが，ここでは最もシンプルに共起時間の累計を用いる．

$$Sim_{cooccur}(x,y) = \frac{\Sigma_{0 \leqq i \leqq n} t_i(x,y)}{max_{a,b \in F}(\Sigma_{1 \leqq j \leqq m} t_j(a,b))}$$

ただし，$t_i(x,y)$ はファイル x, y の i 番目の共起における共起時間を表す．

アクセス共起度ではファイルオープンが共起してさえいればよく，また更新時刻類似度とは異なって閲覧中心のファイルと編集中心のファイルを関連付けることが可能である．ただし，滅多に使用しないファイルや作成直後のファイルの場合には，共起関係がほとんど抽出できないため，グループを作ることが難しい．

以上で述べた通り，ファイル間関連度の尺度は複数あるが，どれにも長所・短所があり，利用者の習慣やファイルの種類等によって左右される．そのため，用途に応じて使い分け，組合せる必要がある．

4. データキューブ型ファイルグループ分析ツール

本節では，データキューブ型ファイルグループ分析ツールについて述べる．本ツールは，複数種類のファイル間関連度の情報をもとに，様々な切り口のファイルのグループを提供する（図 1）．データキューブの考え方に基づいているため，同様に次元や階層といった概念が存在する．また，通常のデータキューブと同様のスライスやドリルダウン等の操作を提供する．ただし，通常のデータキューブが個々のデータやグループ自体よりも集約処理結果を提示することを目的とするのに対して，本ツールにおけるキューブはファイルの集合そのものの発見・生成を支援することを目的とする．

4.1 次　　元

本ツールが提供するデータキューブの各次元（Dimension）は，特定の関連度尺度に基づいてファイル群をクラスタリングした結果のクラスタの集合である．クラスタリングに用いる関連度は 3. で述べたものを対象とする．本研究で用いるクラスタリング手法については 4.1.1 で述べる．図 1 は 3 つの関連度尺度のクラスタリング結果を使ったキューブを表している．例えば，ファイル dews.ppt はテキスト類似度のクラスタリング結果ではクラスタ t3 に属し，更新時刻類似度ではクラスタ m1 に属し，アクセス共起度ではクラスタ c1 に属するファイルであることを表している．

また，補助的な目的として，ファイル名（辞書順），ファイルサイズ，ファイルタイプ等のファイル自身の属性情報も次元として用いる．複数種類の尺度によるクラスタリング結果とファ

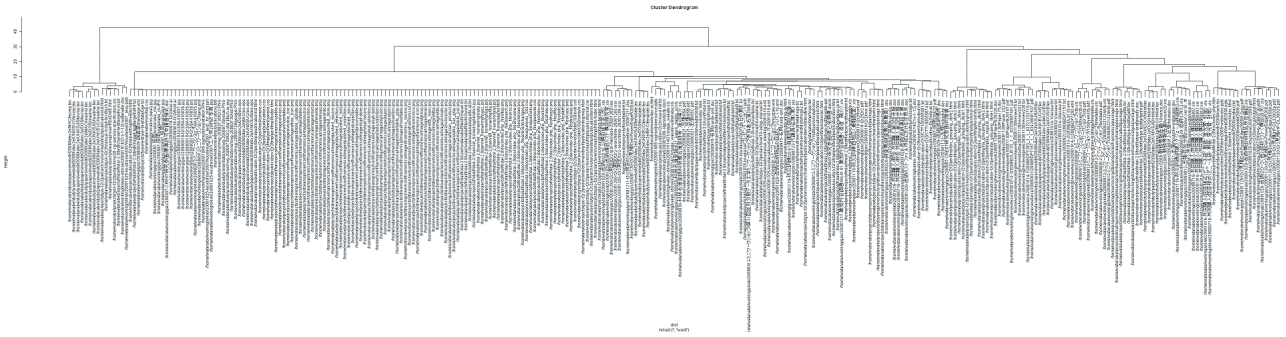

図 2　階層的クラスタリングによるデンドログラム（テキスト類似度）
Fig. 2　Dendrogram geenerated by hierarchical clustering

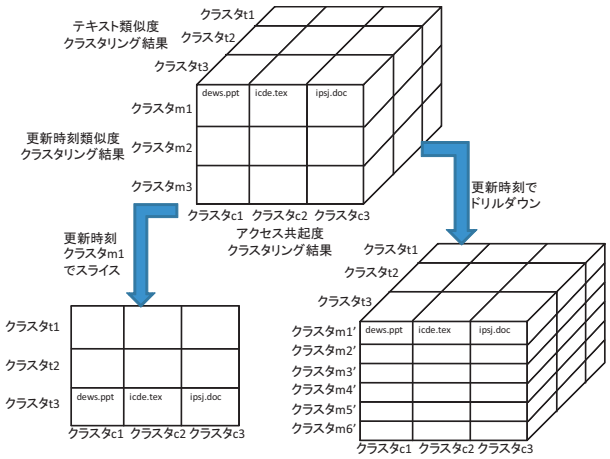

図 1　データキューブ
Fig. 1　Data cube

イル自身の属性情報とを組合せ，ファイル集合に対するグループ化を行っていく．

4.1.1　クラスタリング手法

3. のファイル間関連度の計算式を使って，N 個のファイルに対して関連度を計算すると $N \times N$ の関連度行列を生成できる．この生成した関連度行列に対してクラスタリング手法を適用する．本研究では，ファイルのクラスタリングには凝集型階層的クラスタリング手法 [4] を用いる．これは，各要素だけで成る最小のクラスタ集合からスタートして，関連度の強いクラスタを順に併合していき，最後に全体を統合したクラスタができるまで併合処理を繰り返すというものである．クラスタ同士の関連度を求める方法には，クラスタ内の要素間の関連度の最小値を用いるもの（単連結法），最大値を用いるもの（完全連結法），平均値を用いるもの（群平均法）などがある．

クラスタを併合する過程は，デンドログラムと呼ばれるツリーであらわすことができる．図 2 は著者のホームディレクトリに置かれたファイル 300 個に対して，テキスト類似度による階層的クラスタリングを行った様子である．根に近い方で併合されているクラスタは，関連度が低いとみなせる．階層的クラスタリングでは，このツリーをある閾値の高さで部分木に切断することで，その閾値よりも強い関連度をもったファイル同士のクラスタが生成できる．より根に近い断面で分割すると粒度の荒いクラスタが生成され，逆により葉に近い断面で分割すると粒度の細かいクラスタが生成される．本研究では，このデンドログラムにおける併合関係の階層をデータキューブにおける概念階層としてグループの粒度調整に用いる．

4.2　階　　層

通常のデータキューブでは，各次元（Dimension）をより抽象的な単位で集約化するために，概念階層を用いる．本ツールでも同様に階層情報を用いたファイルのグループ化を行う．

ファイル自身の属性情報を扱う次元の場合，階層関係の情報は事前に用意したものを用いるものとする．例えばファイルタイプであれば，.doc や.png をより抽象化した上位の概念として，文書や画像といった階層関係を用いる．

クラスタの場合は，概念階層の代わりに階層的クラスタリングの出力であるデンドログラムの情報を用いる．4.1.1 で述べた通り，デンドログラムはクラスタ同士の併合順序を表わしており，根に近いほど荒い粒度でのグループ化の結果を表わしている．これらの階層情報を用いて，キューブ操作のロールアップ・ドリルダウンの操作が行われる．

4.3　キューブ操作

4.3.1　スライス・ダイス

スライスはキューブの 1 つの次元における条件を指定して，部分キューブを取り出す操作である．2 つ以上の次元において条件を指定し，部分キューブを切りだす操作はダイスと呼ばれる．本ツールにおいても，利用者はキューブから自分の必要な部分キューブを取り出すことができる．クラスタに対するスライスの条件としては，クラスタラベルを直接指定する方法だけでなく,「ファイル A を含むクラスタ」といったクラスタ内要素に基づく条件も指定可能とする．図 1 左下は,「更新時刻類似度のクラスタラベルがクラスタ m1」であるもの，という条件でスライスを行った場合である．

4.3.2　ロールアップ・ドリルダウン

データキューブにおいてロールアップは,「月単位」のグループを「年単位」にするなど，概念階層におけるより上位の単位でデータをまとめ直す操作である．逆にドリルダウンはより詳細な単位でデータをグループ化し直す操作である．

本ツールにおいて，ファイルの属性情報の次元については，通常のロールアップ，ドリルダウン操作と同様の動作である．

例えばファイルタイプでは，.doc などの拡張子によるグループ分けをロールアップして，文書や画像などのより大きなグループに変える．

クラスタに対するロールアップ・ドリルダウンは，クラスタリングの粒度を変える操作として用いる．デンドログラムのツリーを，より高い位置で分割することで粒度の荒いクラスタを生成し，逆に低い位置で分割することで粒度の細かいクラスタを生成する．図 1 右下は，更新時刻類似度についてドリルダウンを行い，クラスタをより細かい単位にした例である．

4.3.3 ピ　ボット

ピボットはキューブを視覚化するときの軸の方向や表示順序等をかえる操作である．これについては，従来のデータキューブとほぼ同様である．

5. プロトタイプシステム

本節では，現在開発中のデータキューブ型ファイル分析ツールの実装について述べる．本システムは Ruby(ActiveScriptRuby) および R [13] で実装されている．主要部分は Ruby だが，クラスタリングの処理に関しては R を用いている．システムアーキテクチャを図 3 に示す．本システムは，関連度抽出部，クラスタ生成部，データキューブ制御部，データキューブ表示部からなる．

5.1 関連度抽出部

関連度抽出部はファイルシステム中のファイル群から関連度の情報を取得するためのモジュールである．現状では，3. で述べたテキスト類似度，更新時刻類似度，パス類似度，アクセス共起度が抽出可能である．

テキスト類似度抽出部では，全文検索システム Hyperestraier [12] の類似検索機能の出力を用いている．利用者はテキスト類似度の抽出対象にしたいファイル群をあらかじめ Hyperestraier に登録しておく必要がある．Hyperestraier に付属するファイルコンバーター xdoc2txt の機能により，プレーンテキストだけでなく Office 文書や一部の PDF からのテキスト抽出が可能である．

パス類似度および更新時刻類似度の抽出部は，ファイルシステムをスキャンして，直接パスやファイル更新時刻の値を取得している．

アクセス共起度抽出部は，ファイルサーバ上の Samba [14] が記録したログ情報を用いて関連度を算出する．Samba のログには，ファイルのオープンまたはクローズに関する情報が時刻，ユーザ名，ファイル名等と共に記録されている．オープンのみで対応するクローズが存在しないような壊れたログも一部記録されるが，FRIDAL [9] と同様のログクリーニング手法を用いて補間を行っている．

5.2 クラスタ生成部

クラスタ生成部では，各関連度に対して階層的クラスタリングを実行し，それぞれの関連度に基づくクラスタ階層を得る．クラスタの併合過程を表わすツリーを生成するところまでが，このモジュールの役目である．この部分は完全に既存手法であるので，本システムでは R が提供する hclust 関数を用いている．hclust 関数では，クラスタ同士の関連度を計算する方法（単連結法，完全連結法，群平均法等）を引数で切り替えることができる．

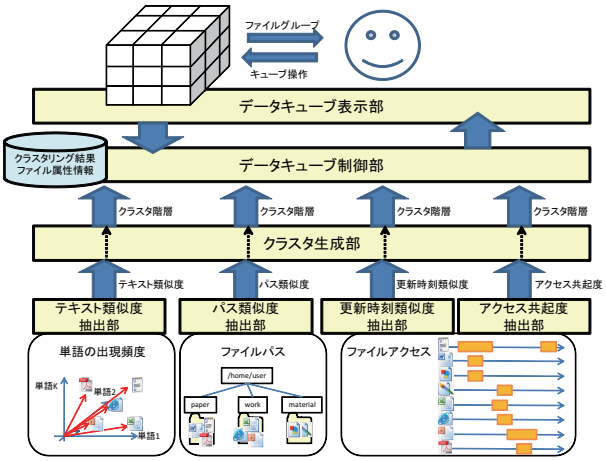

図 3　システムアーキテクチャ
Fig. 3　System architecture

5.3 データキューブ制御部

データキューブ制御部では，クラスタ生成部が出力したクラスタ階層の情報を統合してキューブとして扱えるようにし，データキューブ表示部へ必要な情報を提供する．表示部からのロールアップ，ドリルダウン命令に基づいて，対応する類似度のクラスタ階層を閾値で切断し，適切な粒度のクラスタを生成する処理も行う．

5.4 データキューブ表示部

利用者がファイルのグループを閲覧し，またキューブ操作をするためのインターフェースとしての機能を提供する．表示部は，Microsoft Excel をベースにしたインターフェースとなっている．もともと Excel にはピボットテーブルと呼ばれるキューブに近い機能があったため，現実装の表示部ではピボットテーブルを含んだ Excel ファイルとして結果を出力するようになっている．図 4 はテキスト類似度とパス類似度のクラスタリング結果を 2 次元の表に出した様子である．Excel のピボットテーブルでは，画面右側のメニューで，どの次元を列と行に表示するかを自由に選ぶことができる．ドリルダウン，ロールアップの操作については Excel から実行できないため，別ウインドウからの操作となる．

5.5 利　用　例

実際に著者のファイルをグループ化した場合の操作例を紹介する（図 5）．まず要求 1 として，「テキスト類似度のクラスタをアクセス共起度で細分化して見たい」を考える．

（1） 図中の要求 1(a) のキューブは，テキスト類似度の次元のみ 10 分割し，それ以外は分割なしにした状態である．このとき，対象クラスタ「text06」は共通キーワードを含む 7 個のファイルから構成されている．

（2） 図中の要求 1(b) は，(a) のキューブに対してドリルダウンを行い，アクセス共起度の次元を 10 分割にした状態である．この操作により，対象クラスタ「text06」はアクセス共起

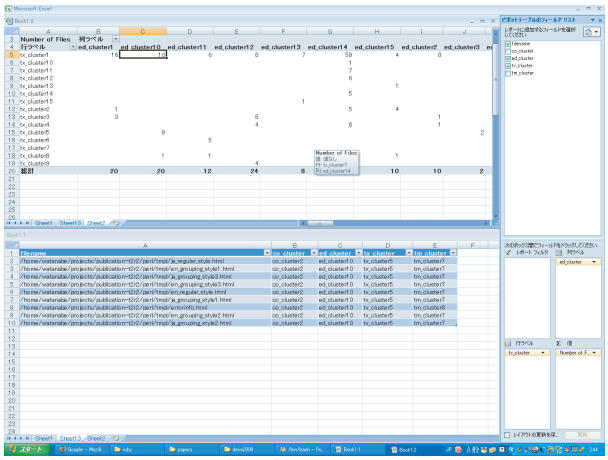

図 4　ファイルグループ出力 (行：テキスト類似度クラスタ，列：パス類似度クラスタ)

Fig. 4　Output of file groups (row: clusters based on text similarity, column: clusters based on path similarity

度により 2 つのグループへ細分化されている．

（3） 図中の要求 1(c) は，(b) のキューブに対してさらにドリルダウンを行い，アクセス共起度のグループ化の粒度を 20 分割に状態である．この操作により，対象クラスタ「text06」は，3 ファイル，2 ファイル，2 ファイルのグループへと分けられ，より共起関係の強いものだけのグループを生成することができる．

次に要求 2 として,「要求 1 で見つけたグループの 1 つと同時期に更新したファイルを知りたい」を考える．

（1） 図中の要求 2(a) は，要求 1(c) と同じキューブである．ここで対象とするグループは「text06 および cooccur02 に属すセル」に対応するグループとする．

（2） 図中の要求 2(b) は，(a) のキューブに対してドリルダウンを行い，更新時刻類似度の次元を 10 分割に変更した状態である．この時，対象グループは更新時刻類似度のクラスタ「mtime01」に属しており，要求 2 を満たすには mtime01 に属すグループだけに条件を絞ればよいことがわかる．

（3） 図中の要求 2(c) は，(b) のキューブに対してスライスを行い，更新時刻類似度のクラスタ「mtime01」で切断した状態である．表示されている各セルは，対象グループと同時期に更新されたファイルのグループに対応する．

以上のように，本ツールによって容易に複数の関連度尺度を用いたファイルの分類を行うことができる．

6. 評価実験

複数のファイル間関連度を組合せることで，ファイルのグループ化をよりきめ細かく行えることを確認するため，プロトタイプシステムを使って評価実験を行った．

6.1 実験データ

今回は実験データとして，著者のホームディレクトリに置かれたファイルのうち，表 1 の拡張子を持つもの 300 個を用いた．300 個のうち，Hyperestraier でキーワードが抽出できたのは 200 ファイルであった．また，実験で用いた 2008 年 8 月

表 1　実験ファイル拡張子

Table 1　Extensions of experiment files

.JPG .PNG .ai .bib .css .csv .doc .docm .docx .eml .eps .htm .html .ico .jpg .pdf .png .ppt .pptx .rtf .tex .txt .xls .xlsx

表 2　人手で分類された正解集合

Table 2　Answer set created by human

研究発表 1(1 件) 研究発表 2(29 件)
イベント 1(3 件) イベント 2(8 件) イベント 3(13 件)
ソフト開発 1(16 件) ソフト開発 2(24 件) ソフト開発 3(5 件) ソフト開発 4(49 件) ソフト開発 5(54 件)
会議資料 1(6 件) 会議資料 2(9 件) 会議資料 3(4 件) 会議資料 4(3 件)
備品管理 1(4 件) 備品管理 2(3 件) 備品管理 3(9 件) 備品管理 4(6 件)
レビュー 1(2 件) レビュー 2(2 件) レビュー 3(1 件) レビュー 4(2 件) レビュー 5(3 件) レビュー 6(3 件) レビュー 7(3 件)
事務書類 1(3 件) 事務書類 2(4 件) 事務書類 3(2 件) 事務書類 4(2 件) 事務書類 5(5 件) 事務書類 6(3 件) 事務書類 7(4 件) 事務書類 8(1 件) 事務書類 9(1 件) 事務書類 10(1 件) 事務書類 11(2 件) 事務書類 12(3 件)
その他 1(5 件) その他 2(2 件)

7 日〜12 月 24 日までの Samba ログには，300 個のうちの 135 個のファイルについての共起関係が 834 件記録されていた．

正解セットを作るため，これらのファイルに対して人手によって分類を行って，表 2 のような 39 種類のラベルを付与した.「研究発表」は論文の Tex や発表 ppt ファイル,「イベント」は会議の参加案内や名簿ファイル,「ソフト開発」ではマニュアルや設定ファイル，アイコン画像などを含む.「事務書類」は出張などの申請書で，同型のテンプレートのものを多く含む．

各ファイル間関連度を用いて生成されるクラスタ数はすべて 20 個とした．クラスタリングにおけるクラスタ間関連度の計算法には群平均法を用いている．キーワードが抽出できなかったり，共起関係が存在しないファイル群は未分類グループとしてまとめるものとした．

6.2 実験結果

1 つ以上の関連度尺度を用いたグループ化の実験結果は表 3 である．表の 1 列目は関連度の組合せを表し，2 列目はその時に生成されたグループ数，3 列目はグループ内の平均ファイル数，4 列目は全グループのエントロピーの平均値/最大値を表す．1 つのグループが全て同じラベルを持つファイルから構成される場合は，エントロピーが 0 となる．まず，各関連度尺度を単体で用いて生成されたクラスタの特徴について説明し，次に 2 つ以上のファイル間関連度を組合せた場について述べる．

6.2.1 各関連度尺度単体での特徴

テキスト類似度は，平均エントロピーが 4 つの中で最も高く，様々なラベルの混ざったクラスタが出る結果となった．特徴的なクラスタとして，関連する研究内容を扱った「研究発表 1」と「研究発表 2」をまとめたクラスタや,「備品管理 1」と「備品管理 2」の物品リストや見積書をまとめたクラスタ等が生成された．また，出張届など同型のテンプレートに記入して作成された事務書類をまとめたクラスタや，複製されたファイル同士のクラスタなども確認された．当然ながらキーワードの抽出できなかったファイルはグループ化できていない．

更新時刻類似度でのクラスタでは，大まかな時期ごとには別れたが，まだ様々な仕事で作成したファイルが混ざった状態であった．更新時刻類似度の場合には 20 の分割ではまだ粗すぎ

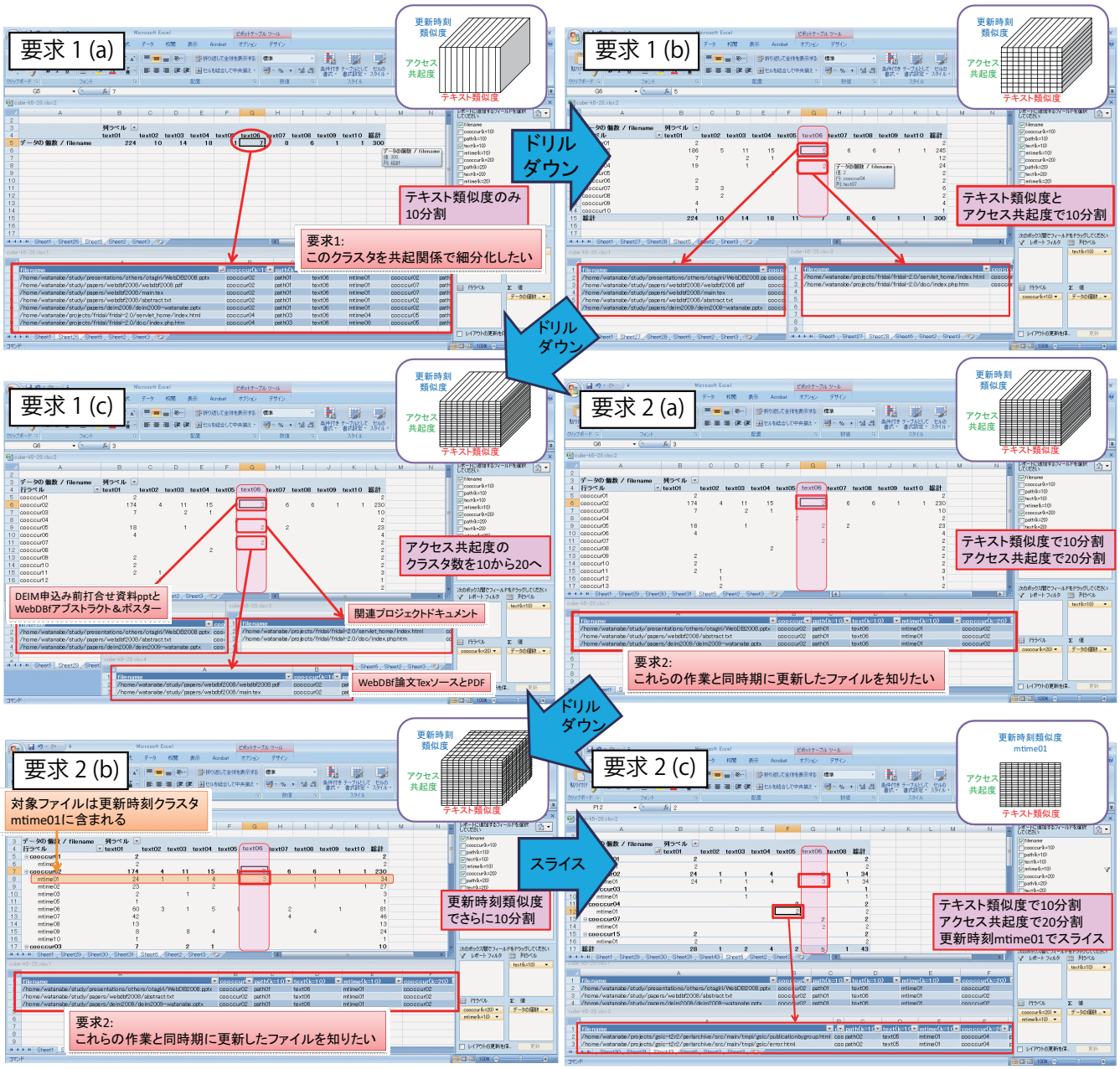

図 5　利　用　例

Fig. 5　Examples of file grouping

るということが考えられる．

パス類似度のクラスタでは，実際のフォルダ階層に近いクラスタが生成された．クラスタリングの粒度を細かくすると，階層の深いところと浅いところのファイルは別れたが，兄弟関係にあるフォルダにあるファイルはなかなか分離されなかった．著者は事務書類を兄弟関係にあるフォルダに置く習慣があるが，これらはクラスタ数 20 では分離できなかった．

アクセス共起度では,「ソフト開発 1」のソフトウェアのマニュアルとそのソフトの設定ファイルのクラスタや,「研究発表 2」の論文 Tex とコンパイル後の PDF のクラスタ,「レビュー 6」の論文とレビュー結果のようなクラスタが生成された．共起関係が抽出できたファイルのグループはかなり純度が高く，それが平均エントロピーの低さにつながっている．しかし，共起関係が記録されたファイルの組が多くないこともあり，半数以上のファイルがグループ化できなかった．

6.2.2　複数尺度の組合せ

基本的にどの組合せであってもより細かくファイルのグループを作ることができたが，特にテキスト類似度やアクセス共起度が単体では扱うことができなかったファイル群は，パス類似度や更新時刻類似度を組合せることで分割された．以下では，生成されたグループのうち特徴的だったものについて紹介する．

テキスト類似度で同じクラスタにされた同型テンプレートの事務書類は，更新時刻類似度と組合せることで時期ごとに分けることができた．それに対しテキスト類似度とパス類似度の組合せでは，同型テンプレートの事務書類を分けることができなかった．これは事務書類を兄弟関係にあるフォルダに置くとい

表 3　実験結果 (各次元クラスタ数=20)

Table 3　Experiment result (20 clusters in each dimension)

尺度	グループ数	平均ファイル数	エントロピー (平均/最大)	備考
Sim_{text}	20	15.0	0.369 / 1.105	分類不可 100 件
Sim_{mtime}	20	15.0	0.382 / 0.920	
Sim_{path}	20	15.0	0.174 / 1.291	
$Sim_{cooccur}$	20	15.0	0.064 / 1.270	分類不可 165 件
$Sim_{text} \times Sim_{cooccur}$	47	6.38	0.137 / 1.080	
$Sim_{mtime} \times Sim_{cooccur}$	48	6.25	0.112 / 1.270	
$Sim_{text} \times Sim_{path}$	57	5.26	0.112 / 1.045	
$Sim_{mtime} \times Sim_{path}$	63	4.76	0.092 / 0.700	
$Sim_{text} \times Sim_{mtime}$	75	4.00	0.091 / 0.678	
$Sim_{path} \times Sim_{cooccur}$	42	7.14	0.080 / 1.302	
$Sim_{text} \times Sim_{path} \times Sim_{cooccur}$	83	3.61	0.065 / 1.073	
$Sim_{mtime} \times Sim_{path} \times Sim_{cooccur}$	84	3.57	0.061 / 0.728	
$Sim_{text} \times Sim_{mtime} \times Sim_{cooccur}$	100	3.00	0.053 / 0.678	
$Sim_{text} \times Sim_{mtime} \times Sim_{path}$	107	2.80	0.036 / 0.602	
$Sim_{text} \times Sim_{mtime} \times Sim_{path} \times Sim_{cooccur}$	126	2.38	0.022 / 0.602	

う著者の習慣が効いてしまっているためと考えられる．

更新時刻類似度とパス類似度との組合せでは，ソフト開発で用いたアイコン画像など，同じフォルダに一括コピーして置いたままのファイル群がかなり明確にグループとして生成された．同じフォルダにあっても，何度も編集作業を加えたファイルは別のグループになっている．

アクセス共起度とパス類似度の組合せは，2 種類の組合せの中では平均エントロピーは最も低くなったが，最大エントロピーは最も高い．純度の高いグループとそうでないグループの偏りがあった．

3 種類以上の関連度の組合せでは，さらに細かくグループが生成されている．ただし，今回の実験データでは正解セットのラベルが 39 種類しかないため，過剰に分割されたグループのエントロピーを評価することはしない．

4 種類すべてを組合せた場合は 126 個のグループが生成されているが，ファイル数 10 未満のグループへ分離されないファイル群が 3 つ発見された．これらはいずれも同時期に同じフォルダに一括コピーした画像ファイルだった．今回は画像ファイルに対しては，更新時刻類似度とパス類似度しか手がかりがないため，このような結果になった．画像同士の類似度を求める手法も古くから研究されているので，それらの成果を取り入れることも今後の課題である．

7. まとめと今後の課題

本稿では，デスクトップ上のファイルの整理，検索のためのキューブ型ファイルグループ分析ツールを提案した．本ツールは，様々なファイル間関連度を用いて，ファイルグループの生成の支援を行う．最終的に生成されたグループは，ファイルの整理やデスクトップ検索の検索結果の拡張に利用可能である．

今後の課題としては，複数の利用者のファイル集合を対象にグループ化を行って，出力結果等についてのより詳細な評価を行うことがあげられる．また，ユーザーインターフェースとしての使い勝手の評価も行う必要があると考えられる．

謝辞　本研究の一部は，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，および文部科学省科学研究費補助金特定領域研究 (#19024028) の助成により行なわれた．

文　献

[1] Nitin Agrawal, William J. Bolosky, John R. Douceur and Jacob R. Lorch. “A Five-Year Study of File-System Metadata”, ACM Trans. Storage, Vol. 3, No. 3, p. 9, 2007.

[2] Alexander Ames, Carlos Maltzahn, Nikhil Bobb, Ethan L. Miller, Scott A. Brandt, Alisa Neeman, Adam Hiatt and Deepa Tuteja. “Richer File System Metadata Using Links and Attributes”, Proc. IEEE NASA Goddard Conference on Mass Storage Systems and Technologies, pp. 49–60, 2005.

[3] D.K.Gifford, P.Jouvelot, M.A.Sheldoon, J.W.O ’Toole,Jr. “Semantic File Systems”, Proc. ACM Symposium on Operating Systems Principles, pp. 16–25, 1991.

[4] Jiawei Han and Micheline Kamber. “Data Mining: Concepts and Techniques” Morgan Kaufmann, 2006.

[5] Greg Smith, Mary Czerwinski, Brian Meyers, Daniel C. Robbins, George G. Robertson and Desney S. Tan. “FacetMap: A Scalable Search and Browse Visualization Visualization and Computer Graphics”, IEEE Trans. Vis. Comput. Graph., Vol. 12, No. 5, pp. 797–804, 2006.

[6] Ka-Ping Yee, Kirsten Swearingen, Kevin Li and Marti Hearst. “Faceted Metadata for Image Search and Browsing,” Proc. CHI, pp. 401-408, 2003.

[7] 石川憲一, 森嶋厚行, 鈴木勇, 杉本重雄. 「共有ファイルサーバにおけるコミュニティ情報管理ツールの提案」第 17 回データ工学ワークショップ DEWS2006.

[8] 小田切健一, 渡辺陽介, 横田治夫. 「アクセス履歴に基づくファイル間関連度を用いたデスクトップ情報管理ツールの開発」WebDB forum 2008 (ポスター発表).

[9] 渡部徹太郎，小林隆志，横田治夫.「キーワード非含有ファイルを検索可能とするファイル間関連度を用いた検索手法の評価」第 19 回データ工学ワークショップ DEWS2008.

[10] 北研二, 津田和彦, 獅々堀正幹.「情報検索アルゴリズム」共立出版, 2002.

[11] Google デスクトップ. http://desktop.google.com/ja/features.html

[12] Hyper Estraier. http://hyperestraier.sourceforge.net/index.ja.html

[13] R Project. http://www.r-project.org/

[14] Samba, http://us3.samba.org/samba/

[15] Windows Search. http://www.microsoft.com/windows/products/winfamily/desktopsearch/default.mspx