社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

複数タスク環境におけるユーザ操作に基づくファイル間関連度の導出

定免　睦昌†　　國島　丈生†　　横田　一正†

† 岡山県立大学大学院情報系工学研究科　〒 719–1197 岡山県総社市窪木 111
E-mail: †{joumen,kunishi,yokota}@c.oka-pu.ac.jp

あらまし　ユーザは複数のファイルを使用してタスク（作業）を行うことが多い．これらタスクで使用するファイルの多くはフォルダによって分類されているが，複数のタスクで共通して使用するファイルが存在する場合，フォルダに分類することが難しくなるという問題がある．また，複数のタスクを実行している状況下では，ファイルが起動している時間だけでタスク毎にファイルを分類することは不可能である．そこで本研究では，ユーザの操作に着目してファイル間関連度を抽出し，関連度の高いファイル群をタスク毎にグループとしてユーザに提示することで，タスク毎に使用された可能性の高いファイルの発見を可能にするファイル管理システムを提案する．

キーワード　ファイル間関連度，クラスタリング，ファイル管理

Derivation of relationships between files based on user's desktop operations in multiple tasks environment

Yoshiaki JOMEN†, Takeo KUNISHIMA†, and Kazumasa YOKOTA†

† Graduate School of Systems Engineering，Okayama Prefectural University
E-mail: †{joumen,kunishi,yokota}@c.oka-pu.ac.jp

Abstract　When a user performs some tasks on computers, many files related to the task are used during it. As users tend to maintain these files by storing them into a single folder, it is difficult to maintain files related to multiple tasks by just using folder categorization. Moreover, in the recent multi-process desktop environment, it is not possible to assure that the files processed simultaneously are always related to the same task. Based on these considerations, we propose a file clustering method based on user tasks in multi-process desktop environment. Relationships between files are calculated from the user's desktop operations, considering changeovers between multiple tasks. Files are categorized into some groups, reflecting the user's tasks. The groups are helpful for the user to find files related to the specified task.

Key words　relationships between files，clustering，file management

1. ま え が き

ユーザが PC 上でタスクを行う場合，複数のファイルを参照しながらタスクを進めることが多い．1 つのタスクに専念するだけでなく，同時に並行して複数のタスクを進めることは珍しくない．同時に複数のファイルを参照しながらタスクを進めることが多いので，ファイルを同時刻に開いている間の時間は，1 つのタスクで使用されている可能性が高い．

ファイル間の関連度を扱う研究は以前から進められており，ファイルの共起時間に着目している研究も多く存在する [1] [2] [3] [4]．ファイル間の関連性を算出する上で，ファイルの共起時間は有用であるが，複数のタスクを同時並行向的に進行していったときに，別タスク間で使用するファイルにも関連付けが発生するという問題がある．そこで，本研究では，ファイルの共起時間だけでなく，ユーザの GUI 操作などに着目し，複数タスク環境でのタスクに基づくファイル間関連度を算出する手法，および，これを用いてファイル群をクラスタリングする手法を提案する．

ユーザの GUI 操作では，ユーザが着目しているウインドウのアクティブ時間や，ウインドウフォーカスの遷移を考慮することで，タスク毎に使用するファイルを分配することが可能になり，複数タスクで複数のファイルを使用する場合でも，タスク毎にファイル間関連度を算出することが可能になる．例えば，ユーザがパワーポイントを作成するために，PDF ファイルや過去に作成したパワーポイントを参照するタスクと，報告用のレポートを作成するために，グラフや数値データのテキストファ

イルを参照するといった 2 つのタスクを考える．ユーザはパワーポイントのウインドウをアクティブ状態にして作業を進め，参考となる PDF や過去に作成したのパワーポイントのウインドウに遷移しながら 1 つのタスクを進める．そして，ある程度までタスクを進めて中断し，報告用のレポートを作成するファイルのウインドウをアクティブにし，参考となるグラフや数値データのウインドウに遷移するといった行動は珍しくない．この場合，ウインドウがアクティブになっている時間は，ファイルの着目度を表しており，ウインドウフォーカスの遷移はファイル間の繋がりを表していると考えることができる．

以下に本稿の構成を述べる．始めに，2 章では関連研究と本研究との関係を述べる．次に，3 章ではファイル間の関連度の尺度と算出方法について述べ，4 章ではクラスタリング手法について説明する．5 章ではファイル間関連度の要素を抽出する方法を紹介し，最後に，6 章でまとめと今後の課題を述べる．

2. 関 連 研 究

文献 [1] [2] [3] では，ファイルアクセスログからファイルのオープン・クローズ時間を抽出し，ファイル間関連度を算出する．2 つのファイルを同時にアクセスしていた場合，その 2 つは同一の作業に利用していた可能性が高いと報告している．共起時間や，ファイルの使用時刻の近さから，ファイル間関連度を算出する点では本研究と一致するが，複数のタスクを同時に平行して行う場合については考慮されておらず，ファイルサーバのアクセスログを用いてファイルの使用履歴を抽出してる点で本研究とは異なる．

文献 [4] では，デスクトップ上のファイルにアクセスした時刻の差やファイルにアクセスした順序，一定時間内にアクセスしたファイル数の密度等によって，ファイル間の関連度を算出する．デスクトップ上のファイルを対象としてファイル間関連度を算出する点では本研究と一致するが，具体的な関連度の要素を抽出する方法について述べられておらず，複数のタスクを同時に平行して行う場合についても考慮されていない点で本研究とは異なる．

文献 [5] では，OS のイベントとプラグインを組み込んだアプリケーションから動作履歴を抽出し，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，関連度を算出する．そして，関連度を基にしたデータによる関連検索と，着目度を基にした参照していたデータを時系列に表示するビューアを提供するシステム，俺デスクを実装している．OS のイベントを用いて，動作履歴を抽出する点では本研究と一致するが，複数のタスクを同時に平行して行う場合について考慮されていない点で，本研究とは異なる．

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説明する．ファイル間関連度とは，ファイル間における関連性を数値化して表したものである．

抽出した複数の要素から，2 つのファイル間における関連度を算出して，同一作業に属する可能性を計算する．以下では，ファイル x とファイル y の間における関連度を算出する方法について述べる．

3.1 共起時間 T

2 つのファイルの使用時間が重複していた合計時間に着目した尺度である．また，ユーザが退席した場合を考慮して，キーボードとマウスの入力がない時間が閾値 $t_{inactive}$ より大きかった場合，次にキーボードかマウスの入力があるまでの時間を共起時間から引いたものを，実際の共起時間と考えるものとする．合計時間が大きいほど，2 つのファイル間において関連性が高いものとする．

例えば，ファイル x とファイル y を同時に使用していた時間が n 回あり，それぞれの時間が $t_i(1 \leqq i \leqq n)$ であるとき，ファイル x とファイル y の共起時間 $T(x,y)$ は次の式で表される．

$$T(x,y) = \sum_{i=1}^{n} t_i \tag{1}$$

3.2 ウインドウフォーカスの移動回数 F

2 つのファイル間でウインドウフォーカスの移動回数に着目した尺度である．操作の誤りでファイルを参照してしまう可能性を考慮して，ウインドウアクティブ時間の要素 t_j が閾値 t_{attend} 以下のものは考えない．合計回数が多いほど，2 つのファイル間において関連性が高いものとする．

ファイル x のウインドウからファイル y のウインドウにフォーカスが移動する場合を考える．このとき，フォーカスの移動後ファイル y のウインドウがアクティブになった時間を $t_j(1 \leqq j \leqq l)$ とすると，ファイル x とファイル y のフォーカス移動回数 $F(x,y)$ は以下の式で表される．

$$\begin{cases} n_k = 0 & (0 \leqq t_j \leqq t_{attend}) \\ n_k = 1 & (t_{attend} < t_j) \end{cases} \tag{2}$$

$$F(x,y) = \sum_{k=1}^{l} n_k \tag{3}$$

3.3 ウインドウアクティブ時間 A

ファイルが起動している間にそのファイルを参照した合計時間に着目した尺度である．操作の誤りでファイルを参照してしまう可能性を考慮して，参照時間が閾値 t_{attend} 以下のものは考えない．また，ユーザが退席した場合を考慮して，キーボードとマウスの入力がない時間が閾値 $t_{inactive}$ より大きかった場合，次にキーボードかマウスの入力があるまでの時間をウインドウのアクティブ時間から引いたものを，実際のアクティブ時間と考えるものとする．合計時間が大きいほどそのファイルを軸に作業を行っていた可能性が高いものとする．

例えば，ファイル x を開いているウインドウが m 回アクティブになったとき，アクティブになった時間をそれぞれ $t_j(1 \leqq j \leqq m)$ とすると，ファイル x のウインドウアクティブ時間 $A(x)$ は次のように表される．

$$A(x) = \sum_{j=1}^{m} t_j \qquad (t_{attend} < t_j) \tag{4}$$

3.4 正　規　化

本研究では，複数の尺度を組み合わせてクラスタリングすることを考えているので，どの尺度も同等に扱う必要がある．以下では，ファイル x とファイル y の関連度を [0,1] の区間に正規化を行う．1 が最も関連性が高く，0 は全く関連性がないことを意味するが，クラスタリング手法では距離の概念を用いているため，値が小さいほど関連性が高い必要がある．そのため，必要に応じて 1 と正規化した値との差分をとることで，大小関係を逆転させる．

$$R_T(x,y) = 1 - \frac{T(x,y)}{max_{a,b \in AllFiles}(T(a,b))} \tag{5}$$

$$R_F(x,y) = 1 - \frac{F(x,y)}{max_{a,b \in AllFiles}(F(a,b))} \tag{6}$$

3.5 ファイル間の距離決定方法

正規化したファイル間関連度の尺度を組み合わせて，最終的なファイル間の距離を決定する手法について説明する．ウインドウ共起時間とウインドウフォーカスの遷移は，2 つのファイル間における関連度を表す尺度であり，ファイル間の距離を決定するのに有用だと言える．ファイル x とファイル y の距離 $D_R(x,y)$ は以下の式で表される．

$$D_R(x,y) = R_F(x,y) + R_T(x,y) \tag{7}$$

4. クラスタリング手法

クラスタリングは，大きな集合を共通の特徴を持つ部分集合へと分ける手法のことである．要素そのものに特徴があり，特徴が似ている要素を切り分ける手法と，要素自体には特徴がなく，要素間の関係のみを用いて関係の高い集合を発見する手法がある．本研究では，ファイル間の関係のみを用いるため，前者の手法は用いることができず，後者のクラスタリング手法の 1 つである階層的クラスタリングを用いた．

4.1 断層的クラスタリング

階層的クラスタリングは，最初，各対象の要素を 1 つのクラスタとみなし，距離が最小のクラスタを統合して，親クラスタを形成するという作業を繰り返すことで，クラスタの階層構造を作り出す手法のことである．その際，クラスタ間の距離を測る必要があるが，要素間の距離しか与えられていないため，クラスタに含まれる要素間の距離を元にクラスタ同士の距離を定義する．距離の定義によってクラスタ間の距離が変化するため，距離の定義の仕方で要素が統合されていく順番や最終的なクラスタの大きさが異なる．

4.1.1 ウォード法

クラスタ内の平方和の増加分が最小のクラスタ同士を統合する手法．対象ファイル x，y の間の距離 $D(x,y)$ から，クラスタ C_1，C_2 間の距離を $D(C_1,C_2)$ と定義する．

$$D(C_1,C_2) = E(C_1 \cup C_2) - E(C_1) - E(C_2) \tag{8}$$

$$\text{ただし, } E(C_i) = \sum_{x \in C_i} (D(x,C_i))^2 \tag{9}$$

一般に階層的クラスタリングで最も分類感度が高いといわれているので，本研究で用いることにした．

4.1.2 アクティブ時間を考慮したクラスタリング手法

ウォード法の他に，ウインドウのアクティブ時間をクラスタリングに組み込んだ手法を提案する．クラスタ内で，ウインドウのアクティブ時間が一番大きい要素を選択し，選択した要素間の距離が最小値であるクラスタを統合する手法である．ウインドウのアクティブ時間は，ファイルに対する着目度を表しており，着目度が大きいほどそのファイルを軸に作業を行っていた可能性が高いので，クラスタを統合する際に有用だといえる．

クラスタ C に属するファイル x の中で，アクティブ時間 $A(x)$ が最も大きいファイルを選択し，選択したファイル間の距離 $D(C_i,C_j)(i \neq j)$ が一番短いクラスタを統合する方法でクラスタリングを行う．選択するファイル $S(C)$ と，選択したファイル間の距離の最小値 $D(C_i,C_j)$ は以下の式で表される．

$$S(C) = \{x | max_{x \in C}(A(x))\} \tag{10}$$

$$D(C_i,C_j) = min(D(S(C_i),S(C_j))) \tag{11}$$

4.2 デンドログラム

階層的クラスタリングの結果を樹形図で表したものをデンドログラムという．縦軸にクラスタ間の距離をとり，横軸に対象クラスタを適宜並べ，統合したときの距離の高さでクラスタ間を結ぶことで，2 分木を形成するように描かれる．階層的クラスタリングの結果生成されたデンドログラムを，任意の高さの水平線で切断することで，切断された高さ以下の親クラスタ単位で，最終的に任意のクラスタを得ることができる．

4.3 クラスタとタスク

ユーザの操作に基づいて，ファイル間関連度やクラスタ間の距離を決定しているので，タスク毎に使用されている可能性の高いファイルが関連付いている．デンドログラムを用いて，ある閾値の距離でクラスタを切断するこにより，タスク毎に関係のある集合を得ることができる．閾値の取り方については，実際に実装を進めた上で決定したい．

5. イベントの取得

ファイル間の関連度の尺度となる要素は，常駐プログラムでファイルの使用やユーザの GUI 操作を監視させ，OS のイベントをフックすることで抽出する．具体的な監視対象は，ウインドウのフォーカス，キーボード，マウス，ファイルアクセスである．マウスとキーボードの監視は主に，ユーザの退席を監視する目的で行う．なお、OS のイベント処理は OS ごとに異なるため、本研究では Windows を対象として具体的な手法を検討した．

5.1 Win32API

Win32API とは，Microsoft Windows の API のなかで，特に 32 ビットプロセッサで動作する OS で利用できる API の名称である．Win32API では，.NetFramework で対応できないウインドウのアクティブ情報や，キーボードの入力情報等を取得するために用いることにした．

定期的にタイマーでウインドウのアクティブ情報を取得することで，ウインドウ固有の情報を得ることができる．例えば，ウインドウがアクティブ状態になっている時間や，ウインドウ

がアクティブ状態になる遷移情報，ウインドウで開いているファイルの絶対パス等を得ることが可能になる．

これら抽出した要素をある一定の間隔で DB に保存し，ユーザの要求が発生したときに，DB から必要な要素を呼び出してファイル間関連度を算出する．

5.2 問 題 点

ウインドウで開いているファイルの絶対パスに関しては，取得できないファイルも存在しており，現在，その原因と解決策を調査中である．また，最近ではファイルをタブで開くアプリケーションの開発が流行っており，タブで開いたファイルについては情報を取得できないといった問題がある．前者の暫定的な解決策として，ウインドウのタイトルにファイル名が表示されるものであれば，ファイル名の取得後デスクトップ検索を行い，絶対パスを取得する方法が挙げられる．しかし，デスクトップ上に同じファイルが存在することが考えられ，目的とするファイルの絶対パスを特定することが不可能である．

6. まとめと今後の課題

本稿では，複数タスクにおいて複数のファイルを参照する場合に，ユーザの GUI 操作やファイルの共起時間に着目することで，タスクに基づいたファイル間関連度を算出し，クラスタリングを行う手法を提案した．

タスクで使用するファイルはタスク毎にまとめて分類されることが多いが，タスク間で共通に使用するファイルは，フォルダ構造による分類が難しいという問題がある．既存のクラスタリング手法では要素を排他的に統合するため，タスク間で共通して使用するファイルが存在する場合でも，複数のクラスタに属することができず，この問題を解決できない．今後の課題としては，複数のタスクにおける複数のファイルを参照する場合，別タスク間で共通して使用するファイルを，タスク毎に属するようにできるクラスタリング手法を考えることが挙げられる．また，実装を行い，本稿で提案した手法の評価や考察を行う必要がある．

文 献

[1] 小田切健一，渡辺陽介，横田治夫，アクセス履歴を用いたユーザの作業に対応する仮想ディレクトリの生成，2009．第 8 回日本データベース学会年次大会 (DEIM2009)．

[2] 渡部徹太郎，小林隆志，横田治夫，ファイル検索におけるアクセスログから抽出した関連度の利用 (情報抽出，夏のデータベースワークショップ 2007(データ工学，一般))．電子情報通信学会技術研究報告．DE，データ工学，Vol．107．

[3] 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)．

[4] 井ノ口伸人，吉川正俊，アクセス履歴を考慮したファイル間の関連度を用いたデスクトップ検索 (履歴応用，夏のデータベースワークショップ dbws 2006)．情報処理学会研究報告．データベース・システム研究報告，Vol．2006，No．77．

[5] 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，2006．第 47 回プログラミング・シンポジウム報告集．