M-038

ファイル更新履歴に着目した作業を代表するフォルダの推定手法

A Method for Discovering Representative Folders of Tasks Using File Update History

西 良太 † Ryota Nishi　　**乃村 能成** † Yoshinari Nomura

1. はじめに

計算機を利用して作業を行う際，以前の作業内容の確認とファイルの再利用を目的に過去のファイルを利用することが多い．たとえば，文書作成の際に過去の同様の文書の一部を変更して作成する場合がある．また，計算機内には作業を代表するフォルダである**ワーキングディレクトリ**が存在する．過去のファイルを利用する際は，同様の作業のワーキングディレクトリを確認することで，目的のファイルの発見が容易になると考える．

しかし，日々の作業で多くのフォルダを様々な階層に作成しており，その中から目的のワーキングディレクトリを探すことは手間となる．**表 1** に著者の 1 人が日常的に大学の講義や研究活動等で使用する計算機を用いて，約 3ヶ月間収集したファイルアクセス履歴を示す．この期間に 4,185 個のフォルダを作成・更新しているが，その中で利用者が意識するワーキングディレクトリは 39 個である．このため，計算機内のワーキングディレクトリを自動的に推定したいという要求がある．

ワーキングディレクトリを推定する手法として，ファイル更新履歴を用いた手法が提案されている [1]．本稿では，既存手法を改良したワーキングディレクトリ推定手法を述べ，推定精度の評価を行う．

2. ワーキングディレクトリの推定

2.1 ワーキングディレクトリとは

我々は，日々の作業で計算機内に作成・更新するファイルをフォルダ分けにより整理している．これらのフォルダの中には，作業を代表する主要なフォルダが存在する．フォルダ構造の例を**図 1** に示す．図 1 において，「第 1 回」と「レポート 1」は「TEX による資料作成」という作業を代表するフォルダである．このように，ユーザが行った作業を代表する主要なフォルダをワーキングディレクトリと呼ぶ．

2.2 ワーキングディレクトリ推定手法

本章では，既存手法を改良したワーキングディレクトリ推定手法を述べる．推定には，ファイル更新履歴を用いる．ファイル更新履歴とは，ユーザがファイルを更新した際に，その更新時刻とファイルパスを記録したものである．以下に推定の手順を示す．

(手順 1) ファイル更新履歴を作業ごとに分割

† 岡山大学大学院自然科学研究科，Graduate School of Natural Science and Technology, Okayama University

表 1　著者の計算機で収集したファイルアクセス履歴

項目	
収集期間	2019 年 1 月 6 日〜3 月 29 日
履歴数	168,509
履歴内のフォルダ数	4,185
履歴内のワーキングディレクトリ数	39

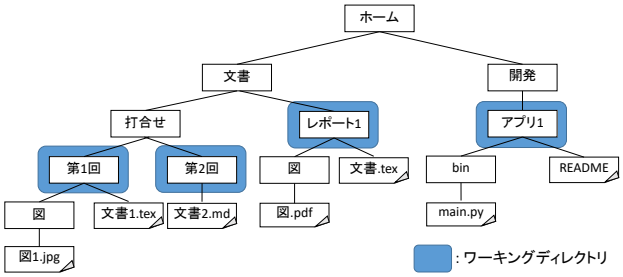

図 1　フォルダ構造の例

(手順 2) 機械的なファイル生成により作成された更新履歴の削除

(手順 3) 残った更新履歴内のファイルを包含するフォルダをワーキングディレクトリと推定

既存手法との相違点として，(手順 1) における分割の条件が異なる．また，新たに（手順 2）を追加した．以降で，(手順 1) と（手順 2）について詳細に説明する．

2.3 （手順 1）ファイル更新履歴を作業ごとに分割

2.3.1 ファイルパス相違度による分割

ユーザが作業を切り替えた際には，作業を行うフォルダの階層が大きく変わる可能性が高い．このため，ファイル更新履歴内で隣接する履歴のファイルパス相違度を定義し，相違度が閾値を超えた場合に更新履歴を分割する．ファイルパス相違度は以下の 2 つの観点から算出する．

(1) ファイルパス間の移動コスト

あるファイルから別のファイルに移動するときの工数をファイルパス間の移動コストとする．

(2) 階層の深さによる重み付け

ファイルパスの深さを (1) の移動コストに重み付けとして与える．本稿では，1–2 層目間の重みを 7，2–3 層目間の重みを 5，3–4 層目間の重みを 3，それ以降を 1 とする．

本稿では，ファイルパス相違度の閾値は 19 とした．

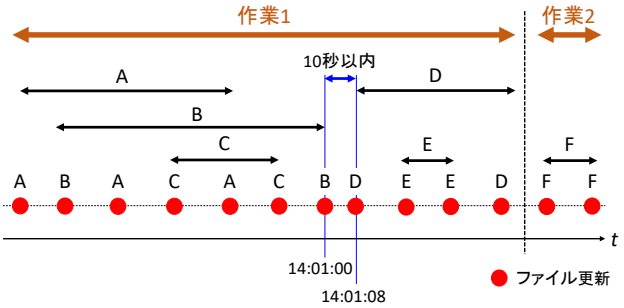

図 2　更新時刻の交錯によるファイル更新履歴の分割

2.3.2　更新時刻の交錯による分割

2.3.1 項のファイルパス相違度により分割された各ファイル更新履歴について，さらに更新時刻の交錯により分割する．分割の例を**図 2** に示す．たとえば，図 2 では，ファイル A，B，および C の更新時刻が交錯している．また，ファイル B とファイル D のように更新時刻の差が 10 秒以内の場合は同じ作業とみなし分割しない．図 2 では，ファイル F のみ更新時刻が交錯していないため作業 2 として分割している．

2.4　（手順 2）機械的なファイル生成により作成された更新履歴の削除

ファイル更新履歴の中には，ユーザが行った作業により記録された更新履歴の他に，ダウンロード等により機械的に生成されたファイルの更新履歴が存在する．これらの更新履歴により推定されるワーキングディレクトリは，ユーザが意識するワーキングディレクトリとは異なる可能性が高い．このため，機械的なファイル生成により作成されたファイル更新履歴を削除する．

本稿では，分割されたファイル更新履歴に含まれる履歴について，すべての隣接する更新履歴の時刻の間隔が 1 秒以内であれば機械的なファイル生成とみなす．

3.　評価

3.1　評価方法

ワーキングディレクトリの推定結果を適合率と再現率により評価する．適合率 Precision と再現率 Recall は以下の式 (1) と式 (2) により定義される．

$$\mathrm{Precision} = \frac{|\mathrm{WD}_{\mathrm{correct}} \cap \mathrm{WD}_{\mathrm{discovered}}|}{|\mathrm{WD}_{\mathrm{discovered}}|} \quad (1)$$

$$\mathrm{Recall} = \frac{|\mathrm{WD}_{\mathrm{correct}} \cap \mathrm{WD}_{\mathrm{discovered}}|}{|\mathrm{WD}_{\mathrm{correct}}|} \quad (2)$$

ここで，$\mathrm{WD}_{\mathrm{correct}}$ は正解ワーキングディレクトリ集合で，$\mathrm{WD}_{\mathrm{discovered}}$ はワーキングディレクトリと推定されたフォルダ集合である．

3.2　評価環境

評価に用いる実験データについて述べる．実験には，表 1 に示すファイルアクセス履歴を用いた．履歴には，計算機の利用者が意識するワーキングディレクトリが 39 個含まれていた．

表 2　評価結果

項目	値
正解ワーキングディレクトリ数	39
推定されたフォルダ数	71
正しく推定されたフォルダ数	37
Precision	0.5211
Recall	0.9487

表 3 （手順 2）適用前後のファイル更新履歴数と履歴内のフォルダ数の比較

	（手順 2）適用前	（手順 2）適用後
履歴数	168,509	107,357
履歴内のフォルダ数	4,185	2,902

3.3　評価結果

評価結果を**表 2** に示す．評価の結果，Recall は 0.9487 となった．このことから，ユーザが意識するワーキングディレクトリのほとんどを正しく推定できたことが分かる．しかし，Precision は 0.5211 となり，推定されたフォルダの約半分はユーザがワーキングディレクトリと意識していないフォルダであった．

3.4　分析

2.2 節の（手順 2）において，どの程度機械的なファイル生成により作成された更新履歴を削除できたかを分析する．**表 3** に，（手順 2）適用前後の履歴数とフォルダ数を示す．（手順 2）の適用後は履歴内のフォルダ数が約 30.7%減少しており，ある程度効果があることが分かる．しかし，適用後の履歴内のフォルダ数は 2,902 個であり，ユーザが手作業でファイル更新を行ったフォルダ数としては多い．このため，機械的なファイル生成とみなす条件について再検討する必要がある．

また，実際の利用において 71 個のフォルダをユーザに提示するのは多すぎる．このため，短期間のファイル更新履歴を用いて推定することが考えられる．表 1 の履歴から 1 週間分を抽出して推定を行ったところ，1 週間に平均 10 個程度のフォルダがワーキングディレクトリと推定されることが分かった．たとえば，ファイルブラウザの拡張機能として，過去 1 週間の履歴を用いて推定したフォルダをユーザに提示することを想定すると 10 個程度であれば利用可能であると考える．また，この程度の数であれば，ワーキングディレクトリでないフォルダが含まれていても許容できると考える．

4.　おわりに

本稿では，ファイル更新履歴に着目したワーキングディレクトリ推定手法を述べ，推定手法の評価を行った．今後は，機械的なファイル生成による更新履歴の削除手法と推定されたフォルダを提示するユーザインタフェースについて検討する必要がある．

参考文献

[1] 池田ゆう子，乃村能成：Inbox による文書整理システム，情報処理学会研究報告，Vol. 2016-DPS-167, No. 30, pp. 1–7 (2016).