4N-02

コンテンツ解析を含む大規模データ分析処理に対するトレーサビリティ

山田 真也 † 天笠 俊之 †† 北川 博之 ††

† 筑波大学 情報学群 情報科学類 †† 筑波大学 計算科学研究センター

1 はじめに

近年多様な目的に対応したデータ分析手法が開発され，意思決定における活用が進みつつある．その中で，分析結果に対するトレーサビリティの必要性が高まっている．データ分析における出力データがどのようにして導出されたかという情報は Lineage と呼ばれ，以前からデータベースの分野において研究が行われてきた．

以前は分析処理の対象となるデータは多くの場合数値データであったためそれらに対する分析処理は，リレーショナル演算のような比較的単純な演算子を組み合わせることで表現することが可能であった．しかしながら，科学技術の向上によって生活の様々な場面から多種多様なデータを集積することが可能になった現在，データ分析の対象となるデータは画像や文章のようなコンテンツデータにまで広がり，より高度な分析に対する Lineage が要求されるようになっている．そのため，リレーショナル演算においても UDF（利用者定義関数）を用いた分析が一般化している．

本稿は，UDF を含むリレーショナル演算で記述できるコンテンツ解析処理に対して，分析処理実行時のオーバヘッドが少ない導出手法を提案する．

2 先行研究

本研究に特に関連する 2 つの先行研究について述べる．Zheng らの研究 [1] では，コンテンツ解析を伴う分析処理の Lineage とは単にどの入力データが分析結果に貢献したかという情報（Derivation）だけでなく，抽出・利用されたデータが入力データのどの部分であるかという情報（Location Specifier）もまた必要であることに言及した．Zheng らは分析処理をリレーショナル演算に UDF を加えることでモデル化し，Derivation と Location Specifier を組み合わせたものを Lineage として計算する仕組みを提案した．この手法では Lineage 計算は分析処理自体と同時に行うため，最初に Lineage 計算を行えば以後 Lineage の計算の必要がないという長所の反面，本来の分析処理にオーバヘッドが発生するという短所がある．しかし，Lineage の参照頻度があまり高くない場合，Lineage が必要になった時に後から Lineage を計算する方がより適切であると考えられる．

Cui らの研究 [2] では，リレーショナル演算からなる処理に対して，Derivation の計算方法を示した．この計算方法では，Derivation の計算は本来の処理の後に実行するため実行時のオーバヘッドがかからないという長所がある．しかし Cui らの研究の枠組みには UDF が含まれていない．

本研究では，Cui らが提案した Derivation の計算方法を UDF によるコンテンツ解析を含む処理に対しても対応できるようにするために枠組みの拡張を行う．

3 UDF を含むリレーショナル演算

本稿では Set semantics に基づくリレーショナル演算を対象とする．コンテンツ解析処理は UDF としてモデル化し，UDF の処理を表現するために新たなオペレータを定義する．そのオペレータをリレーショナル演算のオペレータと組み合わせ，コンテンツ解析処理をリレーショナルモデルのビューとして表現する．

ビューを構成するオペレータ

ビュー V はリレーショナル演算の 6 つのオペレータ（Projection, Selection, Join, Aggregation, Union, Difference）と，コンテンツ解析を UDF を用いてモデル化するオペレータ Function（ϕ）で構成する．Function オペレータは，元のタプルに UDF の処理結果（Value）と処理結果が元データのどの部分から導出されたかという情報（Locator）の 2 つの属性を追加するオペレータであるとする．Function オペレータの定義を以下に示す．

定義 (*Function* オペレータ)**.** タプル $t \in V$ の属性 E を入力としてコンテンツ解析を行う UDF $udf(t.E)$ の処理結果の Value を $udf(t.E).Value$ と表記し，その Locator を $udf(t.E).Locator$ と表記すると，Function オペレータは以下の処理を行う．

$$udf\colon Domain(E) \rightarrow 2^{Value \times Locator}$$

Traceability for big data processing including contents analysis

Masaya Yamada†

Toshiyuki Amagasa†† and Hiroyuki Kitagawa††

†College of Information Science, University of Tsukuba

††Center for Computational Sciences, University of Tsukuba

$\phi_{udf(E)}(V) = \{\langle t,\ v,\ l\rangle \mid t \in V,\ v \in udf(t.E).Value,\ l \in udf(t.E).Locator\}$

4 Lineage

本稿における Lineage とは，1) どの入力タプルが出力データに貢献しているかという情報（Derivation）と，2) Derivation として計算されたタプルに含まれるコンテンツデータのどの部分が解析処理で使われたかという情報（Locator）の 2 つを組み合わせたものであるとする．

4.1 Derivation

Derivation は処理の出力タプルがどの入力タプルによって導出されたかという情報のことである．Derivation はタプルレベルで計算することができ，タプル t が導出された処理の入力テーブルを t の Derivation として導出されたタプル集合に置き換えて処理を再実行するとタプル t が再生成されるという特徴がある．

4.2 Locator

Locator は，コンテンツ解析処理において入力として与えたコンテンツデータのどの部分を用いて解析処理を行ったかを示すデータであり，引数に与えるデータや解析処理の内容によって異なるものである．Locator は，解析処理をモデル化した Function オペレータによってのみ導出されるとする．

例えば，分析対象のデータが画像であるときには UDF が画像のどの領域を用いたかを示す bounding box が Locator になる．

5 分析フローの例

画像分析ワークフローを例にとってどのように Lineage が計算されるかを以下に示す．

シナリオ (想定する分析処理)**.** コンテンツに対する分析処理として，様々な場所で撮られた画像の中から地域 A で撮られた画像に注目し，その画像に映っている人物を特定する処理を行うことを想定する． それぞれの画像にはその画像がどの地域で撮られたものかを示すメタデータが記録されており，それらはリレーション R(ID, Img)，S(ID, Region) を用いて $R\{\langle 001, I_1\rangle, \langle 002, I_2\rangle, \langle 003, I_3\rangle\}$, $S\{\langle 001, A\rangle, \langle 002, B\rangle, \langle 003, A\rangle\}$ のように表現されているとする．ID は画像に対して一意に割り振られた値であり，R.Img の $I_i (i = 1..3)$ には画像のバイナリデータが保存されているとする．

上記に述べた分析処理は，R，S を用いて 1)R と S を ID に基づいて Join し，2)S.Region に基づき Selection を行い，3) 得られたビューの R.Img に対して画像から顔認識を行う UDF を実行する Function オペレータを適用し，4) 属性 Name のみを Projection で選択することで実現することができる．

表 1 UDF 適用直後のビュー

ID	Img	Region	Name	Locator
001	I_1	A	Alice	$\{x_1, y_1, w_1, h_1\}$
003	I_3	A	Bob	$\{x_3, y_3, w_3, h_3\}$

Lineage 計算の例. シナリオで示した処理フローの結果として $T\{\langle Alice\rangle, \langle Bob\rangle\}$ を得られたとする．そのとき T に含まれるタプル $t = \langle Bob\rangle$ の Lineage を計算することを考える．Lineage 計算は 2 つの段階を経て行われる．

まずはじめにタプル t の Derivation の計算を行う．Derivation は 4.1 節で説明した通り，出力タプルがどの入力タプルによって導出されたかという情報であるから，この例においてタプル t の Derivation $R^* \subseteq R, S^* \subseteq S$ は $R^*\{\langle 003, I_3\rangle\}, S^*\{\langle 003, B\rangle\}$ である．

次に，Derivation として計算された各タプルに対して対応する Locator を付加する．画像分析をモデル化した Function オペレータは画像が含まれるタプルに対して，画像から認識された人の名前とそれが画像のどこに写っていたかを示す bounding box を付加する処理を行う．先のシナリオにおける Function オペレータの実行直後のビューを表 1 に示す．このとき Derivation と Locator を組み合わせて，Lineage は $R^*_{lin}(ID, Img, Locator) = \{\langle 003, I_3, \{x_3, y_3, w_3, h_3\}\rangle\}$, $S^*_{lin}(ID, Region, Locator) = \{\langle 003, A, \emptyset\rangle\}$ のようになる．

6 まとめ

本稿では，コンテンツ解析処理を行うワークフローに対する Lineage 計算を可能にするため，Cui らが提案した枠組み [2] を拡張する方針で Lineage 計算の枠組みを提案した．今後は実験を通して Lineage 計算の計算コストの評価等を行う予定である．

謝辞

本研究の一部は JSPS 科研費 JP19H04114 の助成を受けたものである．

参考文献

[1] Nan Zheng, Abdussalam Alawini, and Zachary G Ives. Fine-grained provenance for matching & etl. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pages 184–195. IEEE, 2019.

[2] Yingwei Cui, Jennifer Widom, and Janet L Wiener. Tracing the lineage of view data in a warehousing environment. *ACM Transactions on Database Systems (TODS)*, 25(2):179–227, 2000.