高さ制約付き無順序木の高速類似検索アルゴリズムについて

深川大路† 阿久津達也‡ 高須淳宏† 安達淳†
†国立情報学研究所　‡京都大学化学研究所

1 概要

複雑な構造を持つデータに対する類似検索は大きな計算量を必要とする．近年の情報爆発によるデータの大量化，多様化にともない，複雑なデータを高速に処理する必要が生じている．一方，現在の計算機による類似検索技術は主に文字列や数値ベクトルを対象としており，複雑な構造を持つデータについては個々のデータの性質に応じたアルゴリズムを選択し利用する必要がある．したがって，複雑な構造を持つアルゴリズムの改良や解析は重要な課題である．

構造を持つデータの中で代表的なものに木がある．木は，階層構造を持つデータであり，画像処理，生命情報学などでも利用されている表現力と柔軟性に富んだデータ表現方法である．なかでも，XML は記述力が高く，インターネットで用いられる技術との親和性などから利用が増加している．

木の照合アルゴリズムについては，文字列と同様に広く研究されているが，順序木の場合でも $O(n^3)$ 時間アルゴリズムが最良の成果である [3]．一方，より難しい無順序木の場合には Max SNP 困難であることが知られており [6]，厳密な解を効率的に計算することは難しいとされている．

最近，我々は高さ h の無順序木の編集距離が $2h+2$ 倍の誤差で近似可能である事を証明した．また，大規模データベースを扱うための実装上の工夫についても考察を行った．

2 定義

2.1 木，森

本稿では，根付き木 (rooted tree) を扱う．複数の根付き木からなるグラフ構造を根付き森 (rooted forest) とよぶ．以下では，根付き木 (森) を単に木 (森) とかく．

木 T の頂点の集合を $\mathcal{V}(T)$ とかく．根付き木は根 (root) とよばれる特殊なの頂点をただ一つ持つ．根でない頂点は親 (parent) とよばれる頂点を持ち，その頂点は親頂点の子 (child) とよばれる．複数の頂点が同じ親をもつとき，それらは兄弟 (sibling) とよばれる．各頂点 $v \in \mathcal{V}(T)$ はラベル $\mathsf{label}(v) \in \Sigma$ を持つ．アルファベット Σ はラベルが取りうる値の集合である．簡単のため，本稿では Σ は有限集合と仮定する．ある頂点 v から親を辿って到達できるような頂点を v の祖先 (ancestor) とよび，v のすべての祖先からなる集合を $\mathsf{anc}(v)$ とかく．ある頂点 v を祖先に持つ頂点を子孫 (descendant) とよぶ．

森 F が木 T の部分森 (subforest) であるとは，$\mathcal{V}(F) \subseteq \mathcal{V}(T)$ かつ F が T の祖先子孫関係を保存することをいう．同様に部分木も定義される．木 T の部分森 F が完全 (complete) であるとは，T の頂点 x を F が含むならば，F は x の子孫もすべて含むことをいう．

2.2 木の編集距離

二つの木が与えられたとき，一方の木に対して適切な編集操作 (edit operation) の列を適用することによって他方の木へ変換できる．ある二つの木の間に必要な編集操作の回数の下限を，単位コスト編集距離 (unit cost edit distance)，あるいは単に 編集距離 (edit distance) という．代表的な編集操作の種類は，削除 (deletion)，挿入 (insertion)，置換 (substitution) の三つであり，本稿でもこれらを用いる．

3 完全部分木に基づく木の特徴ベクトル

本節では，木を完全部分木に基づく特徴ベクトルに変換する方法と，その性能について述べる．

編集距離とは異なり，数値ベクトルの距離は効率的に計算可能である．このため，既存の応用研究において木を何らかの特徴量をもとに数値化するという手法が用いられる場合が多い [1, 5]．木の特徴量としては，部分構造などがよく利用される．

A fast similarity search algortihm for unordered trees of bounded height

Daiji FUKAGAWA†, Tatsuya AKUTSU‡, Atsuhiro TAKASU†, and Jun ADACHI†
†National Institute of Informatics
101-8430, Tokyo, Japan
‡Institute for Chemical Research, Kyoto University
611-0011, Kyoto, Japan
{takutsu}@kuicr.kyoto-u.ac.jp

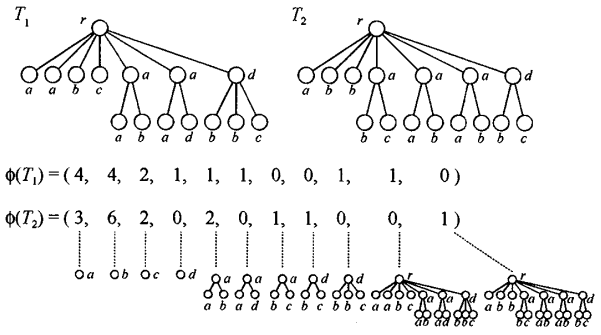

図 1: 木の特徴ベクトル

我々は，完全部分木の頻度を特徴量として用いるアルゴリズムの性能を理論的に解析し，近似アルゴリズムとしての近似性能を導いた [4].

3.1 完全部分木の頻度ベクトル

与えられた木 T に対して，特徴ベクトル $\phi(T)$ は，木全体の集合から自然数への写像 $\mathbb{N}^{\mathcal{T}}$ である．木 t が T の部分木であるとき，特徴ベクトルの対応する要素 $\phi_t(T)$ の値は $\#(T,t)$, すなわち，木 t が T に完全部分木として出現する頻度である．木 t が木 T に出現しない場合，$\phi_t(T)$ の値は 0 とする．図 1 はこの手法によって計算された特徴ベクトルの例である．

木 T はちょうど $|\mathcal{V}(T)|$ 個の完全部分木を含むため，特徴ベクトル $\phi(T)$ の非零要素の個数は高々 $|\mathcal{V}(T)|$ 個である．したがって，無限次元をもつ特徴ベクトル $\phi(T)$ は単純なデータ構造で簡潔に表現可能である．

3.2 完全部分木に基づく木の近似埋め込み

木と編集距離がなす距離空間 $(\mathcal{T}, \mathsf{dist}(\cdot,\cdot))$ から，木の特徴ベクトルとその L_1 ノルムがなす距離空間への埋め込みを考える．この埋め込みによって，木 T_1 と T_2 との距離（あるいは類似度）は元の距離とは異なる値をとるが，その誤差は一定の範囲におさまるため，距離は近似的に保存される．我々は，木の高さが高々 h であるとき，この誤差が高々 $2h+2$ 以下に抑えられることを証明した [4]．より具体的には，以下の不等式が成立する事を証明した．

$$\frac{1}{2h+2} d_\phi(T_1,T_2) \le \mathsf{dist}(T_1,T_2) \le d_\phi(T_1,T_2). \quad (1)$$

ただし，$d_\phi(T_1,T_2)$ は特徴ベクトル $\phi(T_1), \phi(T_2)$ 間の L_1 距離である．

4 今後の課題

一般に，データベースで用いられる XML は高さが低い木として考えることができる場合が多く，そのため，本稿で述べた近似アルゴリズムは比較的良い性能となることが予想される．高さの大きな木に対する近似アルゴリズムの改良は今後の課題である．

参考文献

[1] N. Augsten, M. Böhlen, J. Gamper: Approximate matching of hierarchical data using pq-grams, *VLDB'05* (2005) 301–312.

[2] P. Bille: A survey on tree edit distance and related problems, *Theor. Comput. Sci.* **337** (2005) 217–239.

[3] E. D. Demaine, S. Mozes, B. Rossman, O. Weimann: An optimal decomposition algorithm for tree edit distance, in: *ICALP'07*.

[4] D. Fukagawa, T. Akutsu, A. Takasu: Constant factor approximation of edit distance of bounded height unordered trees, *SPIRE'09*.

[5] A. J. Müller-Molina, K. Hirata, T. Shinohara, A tree distance function based on multi-sets, *ALSIP'08*.

[6] K. Zhang, T. Jiang, Some MAX SNP-hard results concerning unordered labeled trees, *Inf. Proc. Lett.* **49** (1994) 249–254.