編集距離を用いた順序木とタグ木パタンのマッチング

久保山 哲二† 宮原 哲浩†† 安田 浩†
† 東京大学国際・産学共同研究センター †† 広島市立大学情報科学部

1 はじめに

木構造データを比較照合する問題は、計算生物学、構造化テキストデータベース、画像解析、自動推論などの分野で広く研究されている。近年、半構造データからの知識発見の分野においても、木構造データの比較参照問題は、重要性を増している。

半構造データからの構造的特徴の発見手法として、タグ木パタン (tag tree pattern) を用いた手法が提案されている [1]。タグ木パタンは、木構造を埋め込める変数を持つ根付き順序木であり、半構造データを特徴付けるのに適した木構造パタンである。一方、木の編集距離を用いた木の近似マッチングは、文字列の編集距離の概念を木構造データに拡張したものであり、10 年来、計算生物学、自然言語解析、画像解析をはじめとするさまざまな分野で研究が進められている [3]。

タグ木パタンと順序木のマッチングについては、効率的な判定アルゴリズムが知られている [2]。半構造データからの情報抽出のためには、マッチング後に変数に対応する木を抽出することが不可欠である。本稿では、タグ木パタンの変数の表現力に制約を加え、木の編集距離を用いた効率のよいマッチングと情報抽出を実現することを目的とする。そのために見通しのよい木の編集距離の定式化を行い、タグ木パタンにあわせた制約を付加したマッチング手法を提案する。

2 タグ木パタン

木構造データの特徴を表現するためのタグ木パタンは木のノードにラベルを持つ順序木である。変数は、ノードにつけられた特殊なラベルである。Zhang らは、2 種類の変数 (VLDC, Variable Length Don't Care) として path-VLDC(◊ 変数) と umbrella-VLDC(∧ 変数) を木パタンの記述のために導入した [5]。本稿では、情報抽出のために必要と考えられる新たな変数 △(anchor-VLDC) を導入する。簡単のため、ラベルはノードに付けられるものとし、変数はノードの特殊なラベルとして定義する。本来の定義 [1][2] では辺にラベルを持つ木である。なお、ここで扱うタグ木パタン中の変数ラベルは変数の種類を表すために用いる。同じラベルの変数でも別の変数として扱う。図 1(a),(b),(c) に、それぞれの変数について順序木とタグ木パタンのマッチングの例を示す。

3 木の編集距離を用いたマッチング

Zhang らによる木の編集距離の計算方法 [4] を、新たに下記のように定式化する。

定義 3.1. **木**(tree) は根ノードと互いに素な木の順序列から構成される。このような木の順序列を**森**(forest) という。木の列 $T_1, \ldots, T_n$ から構成される森を $F = (T_1 \circ \cdots \circ T_n)$ と表記し、根ノード v と森 F から構成される木を $v(F)$ と表記する。

ここで、木は順序木であり、ノードはラベル付けされているものとする。また、木はただ 1 つの要素からなる森であるとし、空の森は $\emptyset$ で表す。

定義 3.2. F と F' をそれぞれ森とする。森 F と F' の**編集距離**(edit distance) は、F を F' に変換するために必要な編集操作に要するコストの最小値であり、$\delta(F, F')$ と表記する。

編集距離の計算をする際には、通常次の 3 つの編集操作を用いる。(1) **代入**(substitution):ノード v のラベルを別のノード v' のラベルに置き換える $(v \to v')$。(2) **挿入**(insertion):ノード v を挿入する $(\lambda \to v)$。(3) **削除**(deletion):ノード v を削除する $(v \to \lambda)$。また、それぞれのコストを、$\gamma(v \to v')$, $\gamma(\lambda \to v)$, $\gamma(v \to \lambda)$ と表記する。

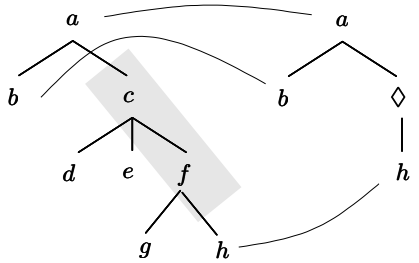

図 1(a) マッチングの例 (path-VLDC)

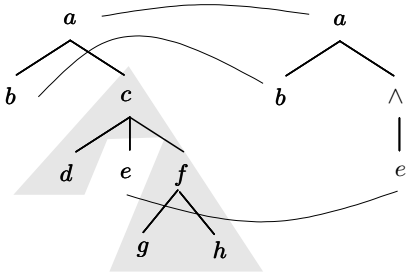

図 1(b) マッチングの例 (umbrella-VLDC)

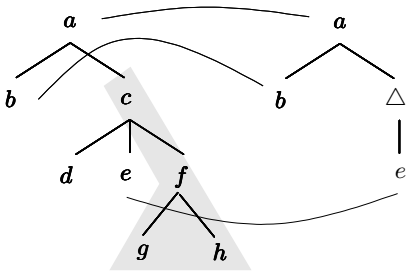

図 1(c) マッチングの例 (anchor-VLDC)

編集操作の列 $S = s_1, \ldots, s_n$ が与えられたとき、その全コストは $\gamma(S) = \sum_{i=1}^{n} \gamma(s_i)$ とする。よって、木 T と T' の編集距離は次のように定義できる。

$\delta(T, T') = \min\{\gamma(S) | S$ は T を T' に変換する編集操作列 $\}$

Applying Tree Edit Distance to the Matching of Ordered Trees and Tag Tree Patterns

† Tetsuji KUBOYAMA (kuboyama@ccr.u-tokyo.ac.jp)
†† Tetsuhiro MIYAHARA (miyahara@its.hiroshima-cu.ac.jp)
† Hiroshi YASUDA (yasuda@ccr.u-tokyo.ac.jp)
Center for Collaborative Research, The University of Tokyo(†)
Faculty of Information Sciences, Hiroshima City University(††)

木の編集距離の計算のため、動的計画法などを用いたさまざまなアルゴリズムが提案されている [3]。本稿でも、動的計画法による定式化を行う。下記のように、編集距離の計算は、森を部分問題に分解してゆく過程である。

$$\delta(\emptyset,\emptyset)=0,$$
$$\delta(F_1\circ v(F_2),\emptyset)=\delta(F_1\circ F_2,\emptyset)+\gamma(v\to\lambda),$$
$$\delta(\emptyset,F_1\circ v(F_2))=\delta(\emptyset,F_1\circ F_2)+\gamma(\lambda\to v),$$
$$\delta(F_1\circ v(F_2),F_1'\circ v'(F_2'))=$$
$$\min\left\{\begin{array}{lcl}\delta(F_1\circ F_2,F_1'\circ v'(F_2')) & + & \gamma(v\to\lambda),\\ \delta(F_1\circ v(F_2),F_1'\circ F_2') & + & \gamma(\lambda\to v'),\\ \delta(F_1\circ F_2,F_1'\circ F_2') & + & \gamma(v\to v')\end{array}\right\}$$

F_1, F_1' は各々値として $\emptyset$ をとりうることに注意。

4 タグ木パタンと順序木のマッチング

Zhang らにより VLDC をノードのラベルに含む木を用いたマッチング手法が提案されている [5]。本稿では、この手法をベースに第 3 節で述べた式に以下の式を追加することによりタグ木パタンと変数を含まない順序木からのマッチングと情報抽出の計算方法を示す。ここでは、マッチングを、最小コストの編集操作列 S から得られる木 T と T' 間のノードの対応として定義する。また、情報抽出を、VLDC に代入される木構造の抽出として定義する。任意のノード v について、$\gamma(\Diamond\to v)=0$, $\gamma(\wedge\to v)=0$, $\gamma(\wedge\to v)=0$ とする。

- 根のラベルが $\Diamond$ の場合:

$$\delta(\Diamond(F),v(F'))=$$
$$\min\left\{\begin{array}{lcl}\delta(F,v(F')) & + & \gamma(\Diamond\to\lambda),\\ \delta(\Diamond(F),F') & + & \gamma(\lambda\to v),\\ \delta(F,F') & + & \gamma(\Diamond\to v),\\ \delta(\emptyset,F') & + & \\ & \min\limits_{1\le k\le n}\{\delta(\Diamond(F),T_k')-\delta(\emptyset,T_k')\} & \end{array}\right\}$$
ここで、$F'=(T_1'\circ\cdots\circ T_n')$ とする

情報抽出の際には、下から 1 行目、2 行目の式のいずれかの値が最小であったときに限りラベル v の情報を保存する (他の変数の場合も同様に計算できるので以下省略)。

- 根のラベルが $\wedge$ の場合:

$$\delta(\wedge(F),v(F'))=$$
$$\min\left\{\begin{array}{lcl}\delta(F,v(F')) & + & \gamma(\wedge\to\lambda),\\ \delta(\wedge(F),F') & + & \gamma(\lambda\to v),\\ \delta(F,F') & + & \gamma(\wedge\to v),\\ \min\limits_{1\le k\le n}\{\delta(\wedge(F),T_k')\}, & & \\ \min\limits_{1\le k\le n}\{\sigma(F,T_1'\circ\cdots\circ T_k')\} & & \end{array}\right\}$$
ここで、$F'=(T_1'\circ\cdots\circ T_n')$ とする

関数 $\sigma(F,F')$ は森 F' の中で同一の親を持つ森を左側から削除して残った森と森 F との距離を返す関数 (suffix forest distance) である [5]。

$$\left(\begin{array}{l}\sigma(\emptyset,\emptyset)=0,\\ \sigma(F,\emptyset)=\delta(F,\emptyset),\\ \sigma(\emptyset,v(F))=0,\\ \sigma(\emptyset,T_1\circ\cdots\circ T_n)=0\ \ (T_i(1\le i\le n)\text{ が共通の親をもつ}),\\ \text{otherwise}\\ \sigma(\emptyset,F_1\circ v(F_2))=\sigma(\emptyset,F_1\circ F_2)+\gamma(\lambda\to v),\\ \text{otherwise}\\ \sigma(F,v(F'))=\min\{\delta(F,\emptyset),\delta(F,v(F'))\},\\ \text{otherwise}\\ \sigma(F_1\circ v(F_2),F_1'\circ v'(F_2'))=\\ \quad\min\left\{\begin{array}{lcl}\sigma(F_1\circ F_2,F_1'\circ v'(F_2')) & + & \gamma(v\to\lambda),\\ \sigma(F_1\circ v(F_2),F_1'\circ F_2') & + & \gamma(\lambda\to v'),\\ \sigma(F_1,F_1') & + & \delta(v(F_2),v'(F_2'))\end{array}\right\}\end{array}\right)$$

- 根のラベルが $\triangle$ の場合:

$$\delta(\triangle(F),v(F'))=$$
$$\min\left\{\begin{array}{lcl}\delta(F,v(F')) & + & \gamma(\triangle\to\lambda),\\ \delta(\triangle(F),F') & + & \gamma(\lambda\to v),\\ \delta(F,F') & + & \gamma(\triangle\to v),\\ \delta(\emptyset,F') & + & \\ & \min\limits_{1\le k\le n}\{\delta(\triangle(F),T_k')-\delta(\emptyset,T_k')\}, & \\ \min\limits_{1\le k\le n}\{\delta(F,F'-T_k')\} & & \end{array}\right\}$$
ここで、$F'=(T_1'\circ\cdots\circ T_n')$ とする
また、$F-T$ は、森 F から木 T を削除した森を表す

情報抽出のためのコスト関数設定: タグ木パタンについては、木構造データから、極小一般化タグ木パタンを発見する手法が提案されている [1]。この手法により得られるタグ木パタンの非変数部分は木の特徴づけのために必須と考えられる情報を含んでいる。そのため、ノードの削除とラベルの書き換えには相対的に高いコストを課す必要がある。(e.g.$\gamma(v\to\lambda)=5$, $\gamma(v\to v')=4$, $\gamma(\lambda\to v)=1$)。

計算のコスト: 新たに anchor 変数を追加し、情報抽出の計算を加えても、Zhang らと同等の計算量でマッチング計算を容易に行える。すなわち、タグ木パタン P と変数なし順序木 D について、下記の時間でマッチングを計算できる [5]。ここで、木 T のノードの総数を $|T|$、深さを $\mathrm{depth}(T)$、葉ノードの総数を $\mathrm{leaves}(T)$ と表記する。

$$O(|P|\times|D|\times\ \min(\mathrm{depth}(P),\mathrm{leaves}(P))\times \min(\mathrm{depth}(D),\mathrm{leaves}(D)))$$

5 おわりに

本稿では、タグ木パタンの変数の表現力に制限を加え、木の編集距離を用いた最適マッチングの手法を用いることにより、効率のよいマッチングおよび情報抽出を実現できることを示した。

参考文献

[1] Tetsuhiro Miyahara, Yusuke Suzuki, Takayoshi Shoudai, Tomoyuki Uchida, Sachio Hirokawa, Kenichi Takahashi, Hiroaki Ueda, "Extraction of Tag Tree Patterns with Contractible Variables from Irregular Semistructured Data," PAKDD 2003, Springer LNAI 2637, pp. 430-436.

[2] Yusuke Suzuki, Kohtaro Inomae, Takayoshi Shoudai, Tetsuhiro Miyahara, Tomoyuki Uchida, "A Polynomial Time Matching Algorithm of Structured Ordered Tree Patterns for Data Mining from Semistructured Data," ILP 2002, Springer LNAI 2583, pp. 270-284.

[3] Philip Bille, "Tree Edit Distance, Alignment Distance and Inclusion," Technical report TR-2003-23, IT University of Copenhagen, March 2003.

[4] Kaizhong Zhang, Dennis Shasha, "Simple Fast Algorithms for the Editing Distance Between Trees and Related Problems," SIAM J. Comput. 18(6), 1989, pp. 1245-1262.

[5] Kaizhong Zhang, Dennis Shasha and Jason T. L. Wang, "Approximate Tree Matching in the Presence of Variable Length Don't Cares," Journal of Algorithms, Vol. 16, No. 1, January 1994, pp. 33-66.