

## 年代順を考慮した工業製品の進化系統図作成法

太田 章悟† 武藤 敦子† 犬塚 信博†

†名古屋工業大学大学院 工学研究科 情報工学専攻

### 1 はじめに

工業製品は複雑に影響しあい発展してきた。この複雑な影響関係を可視化し、製品から製品への系統を知ることで、企業研究などに役立てられる。これまでに文化の発展を生物の進化と捉え、これを文化に適用する研究がされてきた [1]。そこでは矢じりや写本においてもミームという文化における遺伝情報が変異しているという考えに基づき、与えられた対象集合の枝分かれの流れを推定 (図 1a) する。しかしここでは対象の年代順が無く、したがってどの対象からどの対象へ系統が受け継がれていったかを分析できない。本研究では、年代順を考慮して、系統関係の可視化 (図 1b) を行うアルゴリズムを提案する。

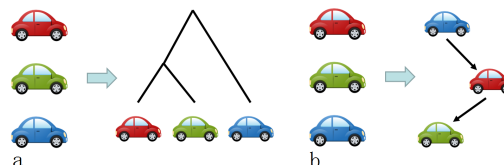


図 1: 既存手法の系統樹 (a) と提案する進化系統図 (b)

### 2 進化系統図アルゴリズムの提案

本研究では、既存研究 [1] と同様、対象に対し、その属性の有無を二値で表現した属性ベクトルを扱う。そして、年代を扱うためこれに製造年を付加する。

本研究では二つのアルゴリズムを示す。これらは、系統関係が存在すると思われる、異なる年代の製品間に有向辺をひいていくアルゴリズムである。この考えに基づいて、まず単純に属性が似ている製品同士に有向辺をひくナイーブアルゴリズムを示し、その考えを拡張して属性の継承に着目した進化系統図アルゴリズムを提案する。

#### ナイーブアルゴリズムを用いた進化系統図

ナイーブアルゴリズム (図 2) は、ある製品について、それよりも製造年が前であるすべての製品について、共通している属性の個数が十分に大きい場合に有向辺をひく。このとき、製造年で降順に並んだ 3 つの製品  $i, j, k$  について、 $k$  から  $j$ 、 $j$  から  $i$  への有向辺がともにある場合、 $k$  から  $i$  への有向辺はひかないという制約を設ける。 $k$  から  $i$  への属性の継承は間接的に表現されているためである。

#### 製品属性の継承に着目した進化系統図

ナイーブアルゴリズムでは、2 製品間の属性の類似しか考慮していないため、似ているだけで本来は系統関係のない製品間に有向辺がひかれやすいと考えられる。そこで、製品属性の継承に着目した提案アルゴリズム (図 3) では有向辺をひく基準をナイーブアルゴリズムよりも厳しくし、属性が 3 世代に渡って継承されている場合に有向辺をひく方針にすることでこの点を改善する。製品  $j$  から  $i$  に有向辺をひく際に、 $j$  に有向辺をひいている製品  $k$  があつた場合、 $j$  と  $i$  に加え  $k$  の 3 製品すべての共通している属性の個数が閾値を超えてい

**INPUT**  $A_1, \dots, A_n$  :  $A_i$  は製品  $i$  の属性ベクトル (ベクトル長:  $l$ );  
 $t$  : 閾値 ( $0 \leq t \leq 1$ );  
**OUTPUT**  $G$  : 作成する進化系統図のグラフ;

1.  $G :=$  (製品番号  $1, \dots, n$  を節点にもつ空グラフ)
2. **For**  $i := 1$  to  $n$
3.   **For**  $j := i - 1$  down to 1
4.     **If** 節点  $j$  と  $i$  の年代が異なる
5.      $\wedge G$  において、節点  $j$  から  $i$  への経路が存在しない
6.      $\wedge (A_j$  と  $A_i$  の共通属性数) /  $l \geq t$  **Then**
7.        $G$  の節点  $j$  から  $i$  に有向辺をひく

図 2: ナイーブアルゴリズム

た場合に有向辺をひく。 $j$  に有向辺をひいている製品が複数ある場合は、それぞれその製品と  $j, i$  の共通している属性の個数のうち最大のを閾値と比較する。それ以外はナイーブアルゴリズムと同様である。

### 3 ゲームソフトへの適用実験

先に示したナイーブアルゴリズムと提案した進化系統図アルゴリズムを用いて、比較的短期間に多数の製品が製造され、多くの進化系統の存在が期待できる 1983~1992 年発売のファミリーコンピュータ用ゲームソフトを対象に、閾値を 0.80 ~ 0.95 まで変えて様々な進化系統図を得た。その一例を図 4 に示す。なお、ジャンルはアクション、発売元は多くの人気アクションゲームを発売した 3 社に絞り、58 製品を対象とした。キャラクターがジャンプするか、敵を攻撃して倒すことができるかなど、アクションゲームを構成する重要な要素の有無を属性とした。

得た進化系統図を評価するため、二点に着目する。

#### (1) シリーズ再現度

第一の観点での評価のため、シリーズ再現度を次の式で定義する。この評価値は、同一のシリーズが一つの経路の中に含まれる割合である。

$$eval1 = \frac{1}{|F|} \sum_{f \in F} maxseries(f)$$

ここで  $F$  は少なくとも 1 つのシリーズものを含む極大経路の集合、 $maxseries(f)$  は  $f$  の頂点のうち、最大頻度のシリー

A method constructing phylogenetic networks of industrial products with chronological order

†Shogo OHTA †Atsuko MUTOH †Nobuhiro INUZUKA

†Department of Computer Science and Engineering, Graduate School of Engineering, Nagoya Institute of Technology

**INPUT**  $A_1, \dots, A_n$  :  $A_i$  は製品  $i$  の属性ベクトル  
(ベクトル長:  $l$ );  
 $t$  : 閾値 ( $0 \leq t \leq 1$ );  
**OUTPUT**  $G$  : 作成する進化系統図のグラフ;

1.  $G :=$  (製品番号  $1, \dots, n$  を節点にもつ空グラフ)
2. **For**  $i := 1$  to  $n$
3.   **For**  $j := i - 1$  down to  $1$
4.     **If** 節点  $j$  と節点  $i$  の年代が異なる
5.      $\wedge G$  において、 $j$  から  $i$  への経路が存在しない **Then**
6.       **If**  $j$  に有向辺をひいている親節点が存在する **Then**
7.          $P := \{p_1, \dots, p_k\}$  :  $G$  における  $j$  の親節点の集合
8.          $m := \arg \max(A_i, A_j, A_p \text{ の共通属性数})$
9.         **If**  $(A_i, A_j, A_m \text{ の共通属性数}) / l \geq t$  **Then**
10.          $G$  の節点  $j$  から  $i$  に有向辺をひく
11.       **Else**
12.         **If**  $(A_i \text{ と } A_j \text{ の共通属性数}) / l \geq t$  **Then**
13.          $G$  の節点  $j$  から  $i$  に有向辺をひく

図 3: 提案アルゴリズム

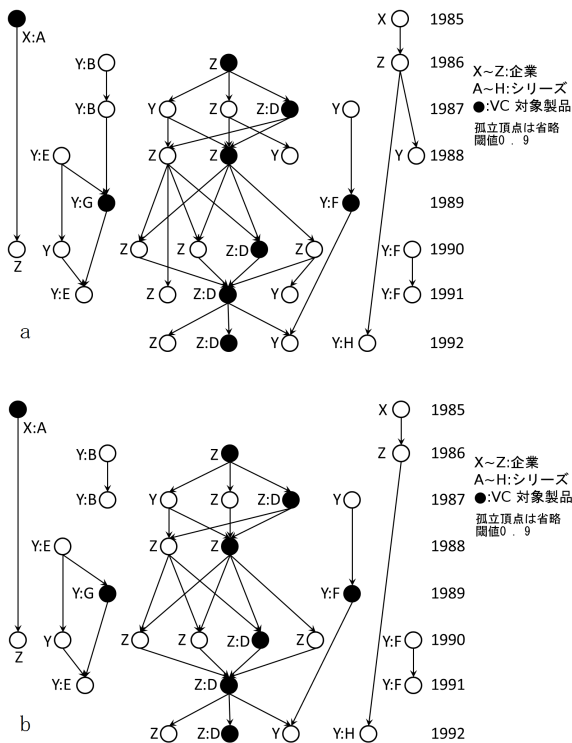


図 4: ナイープ法 (a) と提案法 (b) による進化系統図

ズの構成割合である。ただし、最大頻度が 1 のときこの値は 0 とする。極大経路とは、入次数 0 の節点を始点とし、出次数 0 の節点を終点とする経路である。

(2) 人気製品の影響度

第二の評価として、人気製品の影響度を次の式で定義する。この評価値は、かつて発売されていた一部のコンピュータゲームをダウンロード配信するサービスであるバーチャルコンソール (以下 VC) 対象製品を人気の高い製品とみなし、

表 1: 得た進化系統図の評価

	閾値	シリーズ再現度	人気影響度
ナイープ	0.80	-0.05	0.04
	0.85	-0.17	-0.10
	0.90	0.40	-0.01
	0.95	0.34	0.20
提案	0.80	0.26	0.30
	0.85	0.34	0.13
	0.90	0.42	0.12
	0.95	0.34	0.20

VC 製品と非 VC 製品の出次数の差に着目したものである。

$$eval2 = \frac{1}{|VC|} \sum_{s \in VC} d^-(s) - \frac{1}{|S - VC|} \sum_{s \in S - VC} d^-(s)$$

ここで  $S$  は製品全体の集合、 $VC$  はバーチャルコンソール対象製品の集合、 $d^-(s)$  は節点  $s$  の出次数である。

得た進化系統図を、各評価法で評価した (表 1)。全体としてナイープアルゴリズムより提案アルゴリズムの方が評価値が高い。また閾値別で考えた場合、基本的に閾値による制約が厳しいほど評価値が高くなるが、高すぎると一部のシリーズものを除いてシリーズが表現されず、シリーズ再現度は最終的に低くなっている。

他にもいくつか興味深い点がある。ナイープアルゴリズムの進化系統図は、全体としてあまり影響を与えていないと考えられる製品間に有向辺がある。これは、属性の類似のみで有向辺を引くためと考えられる。提案アルゴリズムは、この点を正しく修正している。

また、提案アルゴリズムにおいて、Z 社のシリーズ A は閾値による制約を厳しくしても進化系統図においてほぼ系統を成しているのに対し、X 社のシリーズ B は閾値による制約が緩くても系列を成さない。これは、シリーズ A が属性を多く継承しているのに対し、シリーズ B は続編になるにつれて新たな要素が次々に追加され、特徴が大きく変化しているためであると考えられ、この点において得た進化系統図の再現性は高いと考えられる。また、X 社が他の 2 社と比べると後の世代にあまり影響を残していないという結果も観察されるなど各社の特徴も観察される。

4 おわりに

年代情報を考慮した工業製品の進化系統図を得る手法を提案した。シリーズや人気製品の影響の観点から、ナイープ法よりも自然な系統関係が得られる可能性を示した。

提案した進化系統図アルゴリズムは、まだ系統関係を完全に再現できていない部分があるので、有向辺をひく制約を変える等の改善の余地があると考えられる。

参考文献

- [1] 中尾 央, 三中 信宏: “文化系統学への招待”, 勁草書房, 2012.
- [2] David Bryant and Vincent Moulton, NeighborNet: An agglomerative method for the construction of planar phylogenetic networks, Molecular Biology and Evolution, 21, 255-265, 2004.
- [3] 任天堂:”Wii バーチャルコンソール”, [www.nintendo.co.jp/wii/features/virtual\\_console.html](http://www.nintendo.co.jp/wii/features/virtual_console.html)