HOKKAIDO UNIVERSITY

Title	Webタグの情報粒度を考慮したwebネットワーク構造の階層クラスタリング
Author(s)	竹内, 尚; 鈴木, 育男; 山本, 雅人 他
Citation	2010年度精密工学会北海道支部学術講演会講演論文集, 2010, 79-80
Issue Date	2010-09-04
Doc URL	https://hdl.handle.net/2115/51319
Type	journal article
File Information	ho20107980.pdf

Hokkaido University Collection of Scholarly and Academic Papers : HUSCAP

409

Webタグの情報粒度を考慮したwebネットワーク構造の階層クラスタリング

北海道大学情報科学研究科 ○竹内尚　鈴木 育男　山本 雅人　古川 正志

要旨:情報には粒度が低い上位概念と粒度が高い下位概念の関係性が存在する. 例えばプログラミング言語を上位概念としてjavaやc,rubyなどが下位概念とするのが自然といえる. この関係性を考慮し協同的タギングシステムのwebページとwebタグとの関係性を用いてボトムアップ的にクラスタリングし, 階層的な木構造を生成する階層クラスタリング法を提案し, 実用性を検証する.

1 はじめに

Web上のテキストや画像などのコンテンツに対して, エンドユーザがタグと呼ばれる分類情報を付加可能なソーシャルタギングを用いたサービスが増加している. サービスの例として動画共有サイトのニコニコ動画や, はてなブックマークなどが挙げられる.

フォークソノミのタグには上位概念と下位概念の関係が見られるが, ユーザーはディレクトリ検索のような複雑な階層構造を意識してタグを付けることがなく, また, メタノイズが存在することから容易にタグの階層構造を構築することは難しい. 本稿のタグの階層構造とは, タグに用いられている単語の抽象の度合いによる階層構造を指し, 抽象性が高く具体性の低い上位概念は幅広いWebページにタグが用いられていると考えられる. 逆に, 抽象性が低く具体性の高い下位概念はごく一部のWebページのタグに用いられると考えられる. また, タグの階層構造を構築したタクソノミはオントロジとして見る事ができるため, 自動的にオントロジを構築する研究やディレクトリ検索システムの構築など に利用する研究も考えられる. しかし, タグの分類を行う上で問題となることは, 多義語や同義語によってタグそのものを分類しても違う場所に同じ名前のタグが存在する構造を得る事ができない.

そこで本稿では,Webページ群に階層クラスタリングを施し, 分類結果からタグの配置を行うボトムアップ的なアプローチ提案し, 実データに提案手法を適用する. また, 分類結果からタグの配置を行うためにタグ間の関係を調査する.

2 ネスター学習システム

ネスター学習システム[1]は, クラス情報の持つ教師ベクトルを入力し, 入力したベクトルを中心とした超球を拡張する事でクラス情報を持つ識別領域を生成する. 教師データの入力が終了し, クラス情報の判らないベクトルを入力する事で入力されたベクトルと識別領域を比較してクラス領域内に存在するベクトルであれば, クラス情報を返す事でクラス判別する事ができる識別学習システムである. このシステムを改良し, 多分木を構築する階層クラスタリング法を提案する.

3 提案手法

提案手法には, 併合するベクトル集合の決定とベクトルの併合の大きく2つのプロセスがある. 併合するベクトル集合の決定では, ベクトルを中心とした超球を拡張していく事でを距離が近いベクトル集合の抽出を行う. ベクトルの併合では決定したベクトル集合に重み付けを行い, ベクトルの併合を行う. この手順の繰り返しにより, 多分木構造の階層クラスタリングの生成を行う.

3.1 アルゴリズム

以下の手順によってクラスタリングを行う. URLベクトル $u_i = \{t_{i1}, t_{i2}, \ldots, t_{iK}\}$,$t$ はタグの付けられた回数,K はタグの種類,URLベクトル集合 $\mathrm{U}^h = \{\mathbf{u_1}, \mathbf{u_2}, \ldots, \mathbf{u_L}\}$,$h$ は階層数(初期値は0), ベクトル u_i を中心とする半径の集合

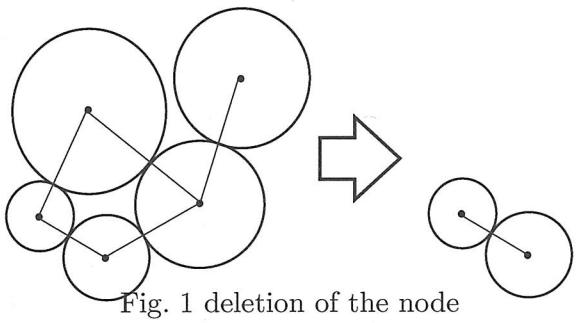


Fig. 1 deletion of the node

$\mathrm{R}_i = \{r_1, r_2, \ldots, r_L\}$, 半径を増加分 $r_{in} = \frac{1}{arg\ \max_i \sum_{j=0}^{L} t_{ij}}$ としたとき以下の手順で階層クラスタリングを行う.

1. URLベクトル集合 U^h を選択し, 最近傍の超球を探索する.

$$u_j = arg\ \min_j \{\|\mathbf{ui} - \mathbf{uj}\| - r_i - r_j\} \tag{1}$$

2. 以下の条件で選択されたベクトル $u_i(u_i \in \mathrm{U}^h)$ を中心とする超球の拡張を行う.

$$r_i \Rightarrow \begin{cases} r_i + r_{in} & if\ r < \|\mathbf{u_i} - \mathbf{u_j}\| - r_i - r_j \\ \|\mathbf{u_i} - \mathbf{u_{i+1}}\| - r_j & if\ r > \|\mathbf{u_i} - \mathbf{u_j}\| - r_i - r_j \\ r_i & otherwise \end{cases} \tag{2}$$

条件2が適用された場合はベクトルをノードとして u_i, u_j にエッジを張り, グラフを構築する.

3. 全ての超球が他の超球と接した状態になり, 全てのベクトル対して式2の条件3が適用されるまで手順1を繰り返す. 終了の場合手順3へ.

4. 生成されたグラフから r_i が大きい順に $\frac{|\mathrm{U}^h|}{2} - 1$ 個のノードを図1の様に削除する.

5. 残ったノードの中でエッジの張られているノードと対応するベクトル集合 $\mathrm{U}_l^m = \{\mathbf{u_{l1}^m}, \mathbf{u_{l2}^m}, \ldots, \mathbf{u_{ln}^m}\}(l = 1, 2, \ldots, N)n$ は, 部分グラフに含まれるノードの数. 部分グラフに対応する集合 $\mathrm{C}^m = \{\mathrm{U}_1^m, \mathrm{U}_2^m, \ldots, \mathrm{U}_N^m\}N$ は部分グラフの数. ベクトル集合 U_l^m に対応する重み集合 $\mathrm{P}_l^h = \{p_{l1}, p_{l2}, \ldots, p_{ln}\}(l = 1, 2, \ldots, N)$,$p_{ln}$ は併合されたURLの数 ($h = 0$ のとき全て1) を定義し, 以下の式を用いてベクトルを併合し, ベクトル集合を $\mathrm{M}^m = \{\mathbf{u_1^e}, \mathbf{u_2^e}, \ldots, \mathbf{u_N^e}\}$ を作成する.

$$\mathbf{u_l^e} = \frac{p_{l1}\mathbf{u_{l1}^m} + p_{l2}\mathbf{u_{l2}^m} \cdots + p_{ln}\mathbf{u_{ln}^m}}{p_{l1} + p_{l2} \cdots + p_{ln}} \tag{3}$$

また, 生成されたベクトル u_l^e に対応する重み p_l^e を以下の式で更新する.

$$p_l^e = p_{l1} + p_{l2} \cdots + p_{ln} \tag{4}$$

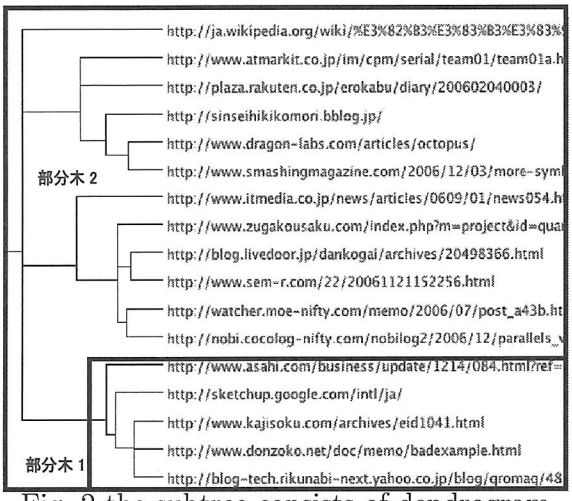


Fig. 2 the subtree consists of dendrogram

併合後のベクトル集合を M^m とすると U^{h+1} を以下のように計算する.

$$\mathrm{U}^{h+1} = \{\mathrm{U}^h \cap \bar{\mathrm{C}}^m\} \cup \mathrm{M}^m \tag{5}$$

6. U^{h+1} を U^h として手順 1 に戻る.$|\mathrm{U}^{h+1}| = 1$ になり, 併合によって 1 つのベクトルとなった場合は終了.

3.2 タグのネットワーク化

クラスタリング結果をデンドログラムした結果図 3 から部分木 w_1 と w_1 を含む上位階層の部分木 $w_2(w_1 \in w_2)$ それぞれをコサイン距離でネットワークを作成し, 結果を比較する. ネットワーク作成のアルゴリズムを下記に示す.

1. 部分木に含まれる URL のベクトル集合を転置し, タグベクトルに変換する.

2. タグの全組み合わせに対してコサイン類似度を計算し, 閾値 $T \leqq sim$ の場合は各々のタグをノードとしてエッジを張ることとする. タグベクトルを T_a, T_b とすると, コサイン類似度は以下の式で計算される.

$$sim = \frac{\mathbf{T_a} \cdot \mathbf{T_b}}{\|\mathbf{T_a}\|\|\mathbf{T_b}\|} \tag{6}$$

4 実データ適用実験

部分木の構造の変化による, タグ間のネットワークの変化を調査することで階層的に分類した結果から代表タグを配置することができるか調査する.

本稿ではデータセットに livedoor[2] から提供されている livedoor クリップのデータセットを用いた.

このデータセットからユーザがタグをつけた回数の出現頻度別に 50 個のタグを選出し, エントリを行, タグを列とするエントリの特徴ベクトルを作成した. データの削減のため, タグが 10 個以上付加されていないエントリを削除し,URL の数は 2554 である.

各手法における設定は閾値 $r = 1$, 閾値 $T = 0.5$ に設定し, 図 1 のように下の囲み部分を部分木 1, 図全体を部分木 2 として比較を行った.

5 考察

図 3 から部分木 1 ではネットワーク構造から"tips","software","web"の抽象性が高いタグの次数が高くどのタグにもエッジが張られる傾向があり, 図 4 から部分木 2 も同様

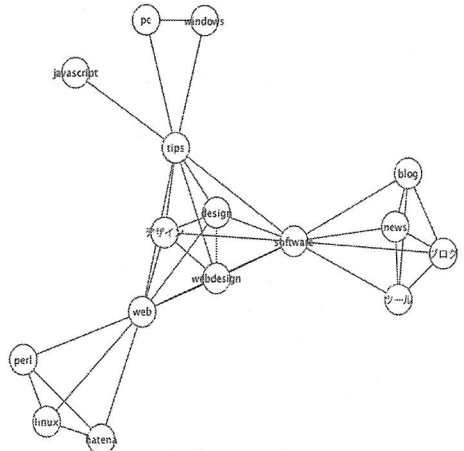


Fig. 3 部分木 1 から生成したネットワーク ($T = 0.5$)

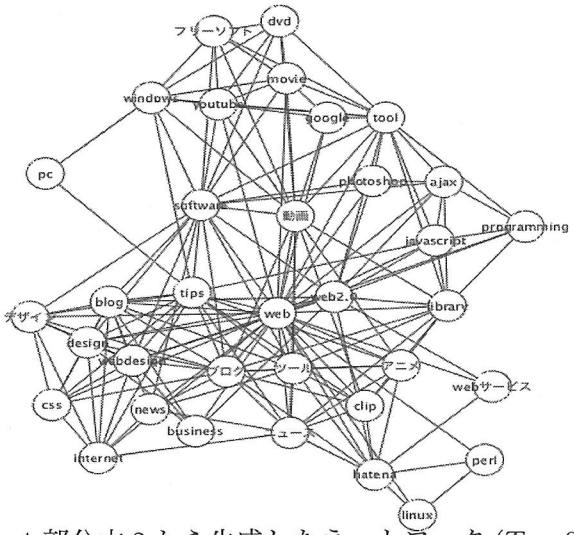


Fig. 4 部分木 2 から生成したネットワーク ($T = 0.5$)

の性質が見られた. このことから, 抽象性が高いタグは具体性の高いタグにも結びつきが高く, 圧倒的に抽象的なタグを付けるユーザが多いと考えられる. 一方, 具体性を持つ影響力の小さいタグでもネットワークとして存在していたことから, 次数が低くほかのタグから比較的孤立しているネットワークから代表的なタグを探索することができれば, 情報粒度に即したタグの配置が可能ではないかと考える.

しかし, 問題点としてベクトルモデルを使用しているため, タグの種類が 50 と少なく, 分類体系を表現するには多種多様なタグが必要となるため, 改善が必要である.

6 結論

本稿では web ページの階層的クラスタリングを生成法を提案し, その結果から部分木を取り出し, タグの関係性を調査を行った.

今後の展望として, 代表タグの配置方法の提案, 階層的クラスタリングの他手法との比較を行いたい.

参考文献

1) Cooper,L. N., C Elbaum, and D. L.Relly. Self Organizing Feneral Pattern Class Separator and Identifier, U.S. Patent 4,326,259. 1982.

2) livedoor Co.,Ltd. EDGE Datasets. http://labs.edge.jp/datasets/.