2E1-1

再帰的なグラフクラスタリングを利用した言語連想データの処理について

Associative Language Data Processing by Recurrent Graph Clustering Methods

鄭 在玲 Jaeyoung Jung　三宅真紀 Maki Miyake　赤間啓之 Hiroyuki Akama

東京工業大学大学院社会理工学研究科
Department of Human System Science, Tokyo Institute of Technology

Words and their relations can be represented by a graph to make a semantic network. Markov Clustering (MCL) proposed by Van Dongen in 2000 is one of the useful graph clustering methods, which makes words on the semantic network be grouped according to their concepts. On this process, however, each of the words gets to belong to only one cluster by the method of hard clustering as a main feature of MCL. After all, all clusters come to lose possible conceptual overlapping among them since they are separated superficially by having no common word. In this sense, MCL might be considered inappropriate for analysis of language data. In this study, we propose a new clustering method called Recurrent MCL, which recovers hidden connections among MCL clusters. We show that the problem of MCL is overcome by applying MCL again to the MCL clusters processed by this RMCL.

1. はじめに

単語と単語の連想関係をグラフ表現化した意味ネットワークにおいて、類似した単語を概念という形でクラスター化する場合、Van Dongen(2000)の Markov Clustering(MCL)は、その精度の高さゆえきわめて有効である。しかし、クラスター間の重複という意味的曖昧さを許さないハードクラスタリングの一義性など、言語データに適用した場合、その出力結果にはいくつか不自然な点が見出される。

本研究では、言語データに適用する際の本質的な問題を解消するため、クラスター結果を再入力し、クラスター間の関係を復元したうえで、MCL を反復しておこなうという、独自なクラスタリング手法 RMCL（Recurrent Markov Clustering）を構想する。これにより、自動的にクラスターの概念名を決定しながら、概念カテゴリー間の相互関係を示すリンク情報を（再）構築することが可能になり、人間の多様な視点に合った形で概念の抽出が行われ、言語資料のダイナミックで深く多相的な体系化がもたらされると考えられる。

2. RMCL とそのアルゴリズム

Markov Clustering (MCL) とは、グラフ上でマルコフ過程に従う酔歩を繰り返す時、遷移行列自体を漸次修正することで、次第に酔歩するエージェントがグラフの密なエリアに捉えられ、抜け出せなくなるよう仕向け、結果としてグラフ自体を非連結のサブグラフに分割させるという手法である。

MCL に言語データを適用した場合、分類不足の問題が起きることが知られている。すなわち、言語データ特有の偏った次数分布により、少数だが、極端に大きなサイズのクラスターが生成されるということである。ただし、三宅ら(2006)は、この問題を BMCL（Branching Markov Clustering）という手法を用いて部分的に解決している。一方過分類のケースも生じやすく、不必要に細分化されすぎたクラスターの一括化、再統合が必要になる。そのための解決策はいくつか考えられるが、ここで提案する RMCL は、以下に述べるような本質的利点をもっている。

すなわちもともと MCL は、ノード間のオーバーラップがないハードクラスターをもって収束結果を出力しているので、収束へ向けての途中のソフトクラスタリングがもつ意味世界の広がりの可能性を、そのままではうまく活用できない。また中間のソフトクラスターを単にマージするだけでは、MCL のもつ非曖昧化の効果を減殺するだけである。RMCL は、両者の限界を媒介しつつ意味の様々な広がりを厳密に計算確定できる手法である。すなわち収束以前のクラスタリングステップにおけるノード間のオーバーラップデータをもとにして、収束状態におけるハードクラスター間に対し、自動的に潜在的な隣接関係を復元できるので、さらに MCL クラスターを新たなノードとして再度 MCL に投入し、グラフをダウンサイジングすることが可能になる。

RMCL には、飛び石アルゴリズムとアリバイ崩しアルゴリズムのふたつがある。飛び石アルゴリズムは、以下のステップに従う。

1） 収束クラスターステージ $ClusterStage_k$ とそれ以前の選んだクラスターステージ $ClusterStage_i$ = {$C_i(1)$, $C_i(2)$, ..., $C_i(r)$}(1,2,..,r はクラスター番号)との間で、クラスターステージ間行列 $ClusterStage_k$-$ClusterStage_i$ Matrix = Cluster-Word $Matrix_k$ × Tr (Cluster-Word $Matrix_i$)を計算する。

2） $ClusterStage_i$ のオーバーラップ情報を使って、$ClusterStage_k$ のハードクラスターを以下の式で再連結する。Cluster $Matrix_i$ = $ClusterStage_k$-$ClusterStage_i$ Matrix × Tr ($ClusterStage_k$-$ClusterStage_i$ $Matrix_i$)

3） Cluster $Matrix_i$ の対角成分を 0、非対角成分のうち 0 でないものを 1 に置換し、隣接行列 Adjacency $Matrix_i$ を生成する。

4） Adjacency $Matrix_i$ から不要な過剰接続を排除する。各収束クラスター$C_k(j)$の要素数を d として、

```
for n=1,...,d-1 {
  for m=n+1,...,d {
   # C_k (n) connects to C_k (m)
     If (Adjacency Matrix_i (n)(m) = = 1)

     Then, {

     C'_k (n) = { C_i (p)| C_k (n) ∩ C_i (p) ≠ φ, 1 ≤ p ≤ r};
     C'_k (m)={ C_i (q)| C_k (m) ∩ C_i (q) ≠ φ, 1 ≤q ≤ r};
```

連絡先：鄭在玲，東京工業大学，東京都目黒区大岡山 2-12-1，(T/F)03-5734-3254，catherina@dp.hum.titech.ac.jp

If (Intersection (C'_k (n), C'_k (m)) = =
{C'_k (s) | s ≠ n, m , 1≤s≤r})

Then, Adjacency Matrix$_i$ (n)(m) = 0;
}
}
}

飛び石アルゴリズムの名は、収束段階のクラスターとまだ未収束の各段階のクラスターの間で、その間のクラスター変化を一切スキップする形で照合を行うことから来ている。

一方、アリバイ崩しアルゴリズムは、以下のステップに従う。

1） クラスターステージのリストを作る。
ClusterStagesList=
{$ClusterStage_1$,$ClusterStage_2$,...,$ClusterStage_k$}

2） 各 $ClusterStage_i$ で複数のクラスターに多重帰属するすべてのノード oln(p)を挙げる。
OverlappingNodes($ClusterStage_i$)=
{oln(1),oln(2),...,oln(p),...,oln(m)};

3） 各 oln(p)を含むすべてのソフトクラスターの和集合を olc(p)として取る。
OverlappingClusters(oln(p))=olc(p)=
$\bigcup_j$ ($ClusterStage_i$ (j) ⊃ oln(p))

4） 各 oln(p)について、過去に olc(p)で共起した他のノード conodes(p)を挙げ、conodes(p)を含むクラスターを探索し、その間に新たな隣接関係を設定する。
For each oln(p){conodes(p)=olc(p) ∩¬ {oln(p)} = {con(1),con(2),...,con(q),…,con(n)}};
MakeAdjacency($ClusterStage_k$ (j) ⊃ conodes(p))

アリバイ崩しアルゴリズムの名の由来は、推理小説からのアナロジーによる。すなわち、このアルゴリズムにおいては、異なる MCL クラスターに属する要素が、未収束段階のいずれかで、ともに同一クラスターに帰属していたという、過去の“implication（連累、含み）”の証拠を丹念に取り上げていくからである。

3. 成果

MCL を単語の連想関係を収集したコーパスである石崎連想概念辞書（慶応大学）に施した。石崎連想概念辞書とは、小学生の学習基本語彙中の名詞を刺激語とし、10 人の被験者の連想に基づいて構成された、 33,018 語、240,093 の単語対からなる電子コーパスである。この辞書から次数(degree)1 と「次数(degree)2 カヴァチャー1」の希少語を除く全 9373 語を取り上げ、MCL の計算を行ったところ、1408 個の類似・同系列の概念クラスターに自動分類された。さらにそれら 1408 個の各クラスターの内で次数最大の単語ノードを代表となる概念クラスター名とし、RMCL により隣接関係を設定してさらに MCL 計算を反復して行ったところ、たとえば、クラスターステージ 2 との間の飛び石アルゴリズム計算でクラスター数は 349 に再統合された。クラスターステージ 2 との飛び石計算の結果が最もバランスの取れたものであり、クラスターサイズの平均が 4.24、分散が 50.23、そして、分類が適当かどうかの主観的な判断をそれぞれのクラスターに対して 1 人の被験者に行わせたところ、誤分類率は 3.98%であった。

RMCL が MCL のおかした過分類を自動修正する効果について、「果物」を例に示す。まず、以下は「果物」と関係ある単語群が MCL によってクラスターされた結果である。

① { 柿, 汁, 房, 蔓, 李, レモン, アボガド, もぎ取る, ライチ, イチヂク, イチ ゴ, 果汁, 果糖, 果物, 果物屋, 果樹園, トマト, ドリアン, 食後, 南国, ナツミ カン, 搾る, 南の島, ナシ, 熱帯, 熟れる, 熟している, 熟す, 巨峰, グアバ, バ ナナ, グレープフルーツ, パパイヤ, リンゴ, マンゴー, フルーツ, フルーツパー ラー, ブドウ, マスカット, すもも, 成熟する }
② { 桃, プラム }
③ { 桃園, 白桃, ピーチ, 黄桃 }
④ { ポン柑, 甘夏, 果物園, 果実 }
⑤ { 葡萄畑, 青果店, ぶどう, グレープ }
⑥ { 実, 果実園, ザクロ, いちじく }
⑦ { 夏みかん, イヨカン, オレンジ色, ポンカン, ミカン, 柑橘類,ネーブル, ハッ サク, 愛媛 }
⑧ { 果樹, 梨, 洋ナシ, サクサク }
⑨ { 工場, プラント }
⑩ { オレンジ, ジューシー }
⑪ { キウイ }
⑫ { パイナップル, ココナッツ }
⑬ { マスクメロン }

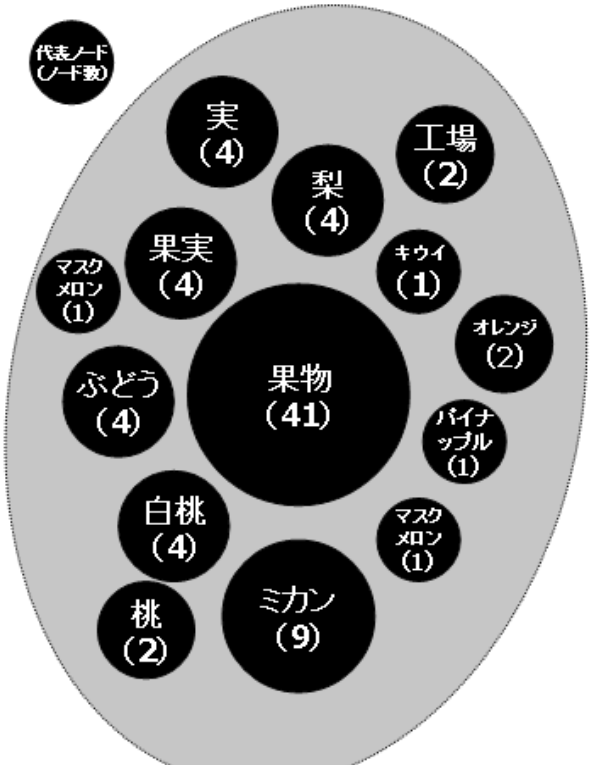

図1. MCL クラスターの RMCL 計算をもとにした再クラスター

上記の MCL クラスターでハイライトされている単語、すなわち、“果物”, “桃”, “白桃”, “果実”, “ぶどう”, “実”, “ミカン”, “梨（ナシ）”, “工場”, “オレンジ”, “キウイ”, “パイナップル”, “マスクメロン”は、それぞれ孤立した MCL クラスターの代表ノード（次数最大のものを選択）となる。このように「果物」に関するパラダイムは MCL によって不適切に細分化されてしまうことがわかる。そこで、飛び石アルゴリズムにより、MCL の第4クラスタリング段階との間で MCL クラスター間の関係を回復する RMCL 計算を行い、MCL 計算を反復すると、過分類されていたクラスターが“果実(fruit)”を代表ノードとする概念クラスターに統合され

た (図1)。しかも、これらのクラスターは単純にマージされることなく独自性を保存していることに注目したい。

4. RMCL の利点

RMCL の利点は、むろん過分類の解消にとどまらない。グラフを言語の連想関係の分析に導入するメリットは、いうまでもなく、多変量解析の適用の場合と違い、言語に関する心的マップ上を移動できるルートの総検索が可能であるという点にある。また MCL のような高精度のグラフクラスタリングが導入され、さらに RMCL のように、最適な概念クラスターサイズを求める調整手法が実現されると、そのルートも単語を繋ぐばかりでなく(SP)、概念を繋ぐことができる(MCSP)など、ルーティングの必要性に応じ、種別化させることができる(鄭,2005)。

たしかに、多変量解析においても、クラスター分析の場合など、デンドログラムなどの形で、類似したものどうしを近傍に配し、全体を論理的に構造化してそこにノード間のパスを通すことはできる。しかし、そこでも場所、領域は示せても道はたどれず、エリアとエリアの間に沿って谷が横たわるだけである。RMCL はその谷の間にショートカットで道(パス)を通すことができる。

このように本稿では詳しく触れなかったが、グラフの特性であるデータ内の移動履歴の計算と結果保存に関して、さらに認知科学、教育工学などの様々な分野で応用が可能である。また BMCL と組み合わせたグラフの階層化、立体化により、人間の多様な視点に合った形で概念の抽出が行われ、言語資料のダイナミックで深く多相的な体系化がもたらされると考えられる。

5. まとめ

RMCL とは、Van Dongen[2000]が提案し、その後様々なカスタマイズが行われつつあるグラフクラスタリングのアルゴリズム MCL(Markov Clutering)に独自の改良を加え、言語データの大規模な意味ネットワークをコンパクトに洗練されたものに変えるためのアルゴリズム、およびそれを実装するシステムのことである。

実際の方法としては、MCL の収束状態におけるハードクラスター（ノード間のオーバーラップがないクラスター）間に、それ以前のクラスタリング段階におけるノード間のオーバーラップデータをもとにして、自動的に潜在的な隣接関係を復元し、さらにそれを再度 MCL に投入するというものである。すなわち、RMCL は、MCL→離接過程の逆流→隣接関係の復元→再 MCL の 1 サイクルと定義することができよう。

その際、あらたに収束クラスターをノードとするダウンサイジングされたグラフに関しては、さらに意味世界の一般性が高まったものとして、ノードの再命名を行うことが可能である。ここでは、同一ノードクラスター内で次数(degree)が最大なものを採用した。

今後の課題としては、MCL のアルゴリズムそのものを改良し、RMCL のような事後的修正ではなく、遷移行列のコントロールそのものによって、RMCL と同様の効果を作り出し、計算量を逓減することが挙げられる。

6. 謝辞

本研究は、21 世紀 COE プログラム（研究拠点形成補助金）「大規模知識資源の体系化と活用基盤構築」の言語・文献知識資源分野に関する研究の一環として行われたものである。データとして連想概念辞書の使用を許可してくださった石崎俊教授（慶応大学環境情報学部教授）に深く感謝の意を表したい。

参考文献

[鄭 2005] 鄭在玲, 三宅真紀, 畑中伸幸, 赤間啓之,:反復クラスタリングによる意味ネットワークに基づく作文支援システムの開発, 情報処理学会研究報告, Vol.2005, No123.

[三宅 2006] 三宅真紀, Jaeyoung Jung, 赤間啓之,:グラフクラスタリングとパターン分類を併用したストーリー・マップ生成の試み, 言語処理学会第 12 回年次大会(NLP2006).

[Okamoto 2001] Okamoto, J. & Ishizaki, S.: Associative Concept Dictionary and its Comparison Electronic Concept Dictionaries, PACLING2001.

[Van Dongen 2000] Van Dongen S.: Graph Clustering by Flow Simulation, PhD thesis, University of Utrecht, 2000.