カーネル法を用いた意味的類似度の定義とブートストラップの一般化

小町守
奈良先端大
mamoru-k@is.naist.jp

工藤拓
グーグル株式会社
taku@google.com

新保仁
奈良先端大
shimbo@is.naist.jp

松本裕治
奈良先端大
matsu@is.naist.jp

1 はじめに

言語における語義の曖昧性は自然言語処理における重要な問題である．語義曖昧性解消タスクにおいては，SemCor などの語義タグつきコーパスや WordNet といった整備された辞書を用いて学習を行う教師ありの手法が主流であるが，リソースを整備するために大きなコストがかかる欠点がある．

そこで，ブートストラップなどの半教師あり学習や，教師なしに語義曖昧性を解消する手法の研究が注目されている．ブートストラップは少量のシードインスタンスから他のインスタンスを得るためのパターンを抽出し，反復的にインスタンスを増やしていく手法であり，人手の介在を最小限に留めながら学習を行うことができる．しかしながらブートストラップには，反復の際にいったん多数のインスタンス集合と共起するパターン（**ジェネリックパターン** generic pattern）を抽出すると，それ以降シードと関連性の低いインスタンスを獲得してしまう問題（**意味ドリフト** semantic drift）がある．たとえば，「熱海」「下呂」というシード単語で Web コーパスから温泉地を得たいとき，これらの単語が「写真」と共起していると，「木村拓哉」などの用語を獲得してしまう，ということが起こる．そのため，*Espresso* [8] に代表されるブートストラップ手法においてはジェネリックパターンへの対処が必要であった．

また，ブートストラップ手法は経験的にはうまく動くことが知られているが，理論的な背景に乏しく，なぜうまく動くのかに関する考察もなされていない．特にシードインスタンスの数，反復回数，毎回の反復で選択するインスタンス・パターンの数など，さまざまなパラメータがあり，調整が難しい．これらのパラメータはタスク・ドメイン依存であり，現実的には経験的に決められることが多く，見通しがよくない．

本稿では *Espresso* を含むブートストラップをグラフ解析として定式化し，グラフ解析手法 HITS[5] におけるトピックドリフト現象と，ブートストラップにおける意味ドリフトとの関連性について指摘する．そして意味ドリフトを防ぐための手法として，グラフ解析で相対的重要度の計算に用いられる 2 つのカーネル法の適用を提案し，語義曖昧性解消タスクで実際に意味ドリフトが防げることを検証する．提案手法はブートストラップと比較してパラメータの数が少なく，理論的に意味ドリフトを防ぐことが保証されており，性能も優れていることを示す．

2 関連研究

完全な教師なし語義曖昧性解消手法としては Purandare ら [10] の研究がある．彼らは単語と文脈からなる共起行列を用い，1 次と 2 次の文脈ベクトル [11] を素性としてクラスタリングを行った．2 次の文脈ベクトルとはその文脈で共起する単語の 1 次文脈ベクトルの平均を用いて計算する文脈ベクトルであり，直接共起しない単語同士でも関連度を計算することができるが，彼らの手法では 3 次以上の文脈ベクトルは考慮されていない．

3 次以上の文脈ベクトルを考慮するグラフベースの手法としては，HyperLex [13] がある．HyperLex では単語をノード，単語間の共起の相対頻度をエッジとしたグラフを作り，クラスタリングを行うことによって語義曖昧性解消を行う．Agirre ら [1] は HyperLex に基づく手法と PageRank [2] を用いた手法の 2 つの手法を比較し，いずれの手法も最頻出語義を用いるベースラインを大きく上回ったと報告しているが，これらの手法は設定しなければならないパラメータ数が多く（7 個），最適化が難しい．

ブートストラップ手法を用いて入力インスタンスに類似したインスタンスを獲得する手法としては Pantel らの提案した *Espresso* [8] がある．*Espresso* は自己相互情報量に基づいた再帰的な計算でインスタンスとパターンをスコアリングし，ジェネリックパターンも用いる手法である．3 節で述べるように，彼らの手法はグラフ解析として一般化することができるが，彼らはグラフ解析との関係性について触れていない．

3 ブートストラップのグラフ解析としての解釈

ブートストラップはシードインスタンス集合から分類器を作成し，分類された全インスタンスのうち確信度の高いインスタンスを新たなシードインスタンス集合として反復を繰り返す．

最終的に出力するインスタンス集合としては，累積的に毎回の反復ステップで得たインスタンスを順に出力する方法と，最終的な分類器の上位 n 個の出力をもって最終出力とする方法があるが，今回は語義曖昧性解消を目的としたブートストラップを対象としているため，Yarowsky [14] にならって後者の解釈を取る．[1]

[1] 後者の利点は間違えて分類されたインスタンスが反復を繰り返すことによって正しく分類される可能性があることで，前者の利点は反復初期に獲得されたシードインスタンスと関連性の高いインスタンスを保持できることである．固有表現抽出など意味ドリフトが起きやすいタスク・ドメインの場合は前者の解釈を取ることが多い．

3.1 ブートストラップの定式化

Espresso [8] に代表されるブートストラップのアルゴリズムは以下のように書き下せる．

1. シードインスタンスのスコアベクトル $\mathbf{i}_0$ を与える．
2. パターン-インスタンス共起行列を P とし（パターン p とインスタンス i の共起は行列の (p,i) 要素で与えられる），パターンのスコアベクトル $\mathbf{p}$ を次式により計算する．

$$\mathbf{p}_n = P\mathbf{i}_n \tag{1}$$

3. 次の反復に用いるインスタンスのスコアベクトルを

$$\mathbf{i}_{n+1} = P^T\mathbf{p}_n \tag{2}$$

により求める．

4. 停止条件が満たされるまでステップ 2 と 3 を繰り返す．

最終的なインスタンスのリストはシードインスタンスに関連度が高い順に並んでいることが期待される．

式 (1) において，共起行列 P の (p,i) 要素 $P(p,i)$ を

$$P(p,i) = \frac{1}{|I||P|} \times \frac{pmi(i,p)}{\max pmi} \tag{3}$$

$$pmi = \log_2 \frac{|i,p|}{|i,*||*,p|} \tag{4}$$

と置くと *Espresso* と同じ式になり，*Espresso* は上記の特殊系となっている．ただし，$\max pmi$ は全インスタンス・全パターン中での自己相互情報量 pmi の最大値であり，$|I|, |P|$ は全インスタンス・全パターン数である．

Espresso などのブートストラップアルゴリズムでは，毎ステップで獲得されたインスタンス集合を用いてそれにマッチする文脈パターンを新たに抽出する，というステップ（**パターン抽出**）が含まれている．しかしながら，新たなパターンの抽出はブートストラップにとって必須の処理ではないことが [6] で示されているため，パターン抽出ステップを省略し，上記のようにパターン-インスタンス共起行列を作成してよいと考えられる．

さて，インスタンスの類似度行列を $A = P^TP$ と置くと，n 回の反復後のインスタンスのスコアベクトルは式 (1), (2) の反復ステップを再帰的に行うことにより，

$$\mathbf{i}_n = A^n\mathbf{i}_0 \tag{5}$$

と書ける．

ここで A を隣接行列とするグラフを考える．このグラフが強連結のとき，各反復ごとに $\mathbf{i}_n$ を正規化しながら $n \to \infty$ とするとシードベクトル $\mathbf{i}_0$ によらず $\mathbf{i}_n \to$（A の主固有ベクトル）となる．$A = P^TP$ なので，このベクトルは，P を隣接行列とする，パターンとインスタンスの二部グラフに対して HITS が返す権威度ベクトルと一致する [5]．このベクトルが与えるランキングはシードインスタンスによらず，一定である．このことは，上のブートストラップの反復を繰り返していくと，必ず意味ドリフトが避けられないことを示唆している．同様の現象は HITS によるグラフ解析でも観察されており，トピックドリフトとして知られている．これは [9] では述べられていない事実であり，3.2 節でそれを検証する．また，ブートストラップではグラフ解析とは異なり反復の際にスコア上位のパターン・インスタンスのみを用いて計算を行うヒューリスティクスが使われる．この足切り操作の有無と意味ドリフトとの関連についても調査する．

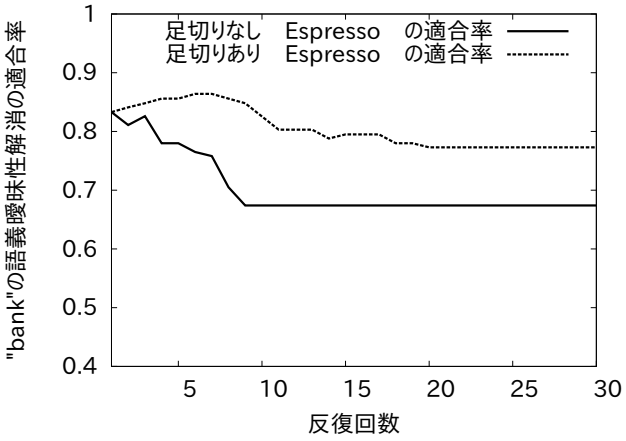

図 1: *Espresso* を足切りあり・なしで用いたときの結果

3.2 *Espresso* を用いた実験

我々は *Espresso* [8] を Senseval-3 Lexical Sample Task のデータ [2] に適用し，アルゴリズムの収束解析を行った．このデータは名詞・動詞・形容詞の 57 単語を含む文で，対象となる単語のうち名詞と形容詞については WordNet の，動詞については Wordsmyth の語義に従って単語の語義が付与されている．

実験におけるパターン-インスタンス共起行列 P は Senseval-3 Lexical Sample Task のトレーニングデータとテストデータを用いて作成した．インスタンスは各用例（語義が不明の単語）であり，文脈パターンとして，パラグラフ内の bag-of-words をグローバル文脈として，インスタンスの前後 n 単語（$n = 3$）から作成した単語列パターンをローカル文脈として用いた．単語は Porter's Stemmer によってステミングを行い，単語の表記のみを用いた．トレーニングデータに付与されている語義の情報は評価時以外用いていない．

評価の際には語義を当てる対象の用例をシードとし，計算したスコアが最も高いインスタンス 3 個のうち，多数を占める語義をシステムの出力とした．語義が同数の場合はスコアの一番高い語義を選択した．[3]

対象の単語としては *bank* を使用した．*bank* は訓練事例 262 個，評価事例 132 個で，訓練・評価事例中の最頻出語義は「土手」の意味の 86 個（F 値 0.674）である．*Espresso* で用いられるパラメータとして，初期パターン数 k は 200 個を用いて足切りし [4]，反復ごとに k を 1 つずつ増加させた．毎回の反復で足切りするインスタンス数 m は 100 個を用いた．

図 1 はインスタンス・パターンの足切りの有無と意味ドリフトの関連を調べたものである．横軸は *Espresso* の各ステップの反復回数であり，縦軸は適合率（正しく語義を当てられた比率）である．今回はどの単語の語義

[2] http://www.senseval.org/senseval3/data.html

[3] Senseval-3 Lexical Sample のデータでは 1 単語あたり平均 6.47 語義である．

[4] パターン数を多くしているのは関係抽出や固有表現抽出と違い，語義曖昧性解消タスクではデータが非常にスパースなためである．

を当てるのかは全て事前に与え，必ず語義を 1 つ出力する設定のため，再現率と適合率は一致する．

足切りなし *Espresso* は反復ごとに頻出語義を選択する傾向が強まり，9 回目の反復で全て語義を「土手」（最頻出語義）と返すようになった．つまり，最初はシードインスタンスに関連性の高いインスタンスが高いスコアを付与されていたが，予想したように徐々にジェネリックパターンに高いスコアが割り振られ，意味ドリフトが起きている．ブートストラップで得られたインスタンスを HITS の重要度に従ったランクと比較したところ，全て一致しており，3.2 節で述べたように意味ドリフトと HITS におけるトピックドリフトは同じ原因で起きていると考えられる．

一方，足切りあり *Espresso* は反復回数を繰り返しても意味ドリフトは起きず，常に関連性の高いインスタンスを選択している．20 回目の反復で収束しているが，このときの適合率は最頻出語義のベースラインを約 10 ポイント上回っている（0.773）．これにより，ブートストラップにおける足切りヒューリスティックは意味ドリフトを抑えるために必須の処理であることが判明した．

4 グラフ解析を用いた意味ドリフトの解決

前節で見たように，足切りなし *Espresso* はグラフベースの手法と考えることができるが，意味ドリフトの影響を受け，大域的重要度の高いインスタンスを盲目的に選択する傾向がある．一方，足切りあり *Espresso* にはこの問題がないものの，毎回選択されるパターンとインスタンスをシードインスタンスに対して計算しなければならないという欠点がある．そこで，足切りなし *Espresso* の利点を保ったまま意味ドリフトを解消するために，2 つの手法を提案する．

4.1 ノイマンカーネル

Kandola ら [4] は文書間の類似性の計算に単語を用いた手法として**ノイマンカーネル**を提案した．このカーネルを文書-単語行列に対して用いるのではなく，パターン-インスタンス共起行列に対して用いることで，シードインスタンスに対する相対的重要度が計算できる．

P をパターン-インスタンス行列とし，$A = P^T P$，その主固有値 λ とすると，拡散係数 $\beta(0 \leq \beta < \lambda)$ の**ノイマンカーネル**行列，K_β は以下のとおり定義される．

$$K_\beta \quad = \quad A(I - \beta A)^{-1} = A \sum_{n=0}^{\infty} \beta^n A^n \qquad (6)$$

インスタンス i, j 間の類似度は K_β の (i, j) 要素で与えられる．

伊藤ら [15] はこのノイマンカーネルが共引用関連度と HITS 重要度との混合を表現していることを示したが，ノイマンカーネルは $P^T P$ が表わす共引用グラフ上での各ノード間の全ての経路の数の重みつき和を求めていることに相当し，PageRank 同様 n 次の文脈ベクトルを考慮に入れていることになる．n が小さいときの $(P^T P)^n$ の各行は各ノード間の関連度を示し，n が大きいときの $(P^T P)^n$ の各行は HITS 重要度ランキングベクトルに近づく．ノイマンカーネルは $n = 1$ から ∞ までの $(P^T P)^n$ の重みつき和であり，拡散係数 β が小さい場合には関連度に偏った，大きい場合には HITS の重要度に偏った順位づけを返す．従って，β を小さめに設定することで，意味ドリフトを抑制し，かつ高次の文脈ベクトルを考慮できると期待できる．

表 1: グラフベースの語義曖昧性解消手法

アルゴリズム	適合率	再現率	F 値
HyperLex	64.6	64.6	64.6
PageRank	64.5	64.5	64.6
Espresso（足切りなし）	47.0	47.0	47.0
Espresso（足切りあり）	66.5	66.5	66.5
ノイマンカーネル	67.2	67.2	67.2
正則化ラプラシアン	67.1	67.1	67.1

4.2 正則化ラプラシアンカーネル

ノイマンカーネルは関連度と重要度の混合を行うことができる定式化となっているが，ノイマンカーネルの相対的重要度計算ではジェネリックパターンに強い重みがつくという問題点がある．つまり，拡散係数 β を大きくして大域的重要度を重視すると，どんな単語とも共起するようなジェネリックパターンに重みがつき，HITS 同様意味ドリフトが起きてしまう．一方，拡散係数 β を小さくして関連度を重視すると，高次の文脈パターンを十分考慮することができない可能性がある．そこで，この問題点をグラフラプラシアンを用いて解決する．

隣接行列が対称行列 A で与えられる重みつき無向グラフ G を考える．G のラプラシアン L は以下で与えられる．

$$L = D - A \qquad (7)$$

ここで D は次数対角行列であり，i 番目の対角要素は

$$D(i, i) = \sum_j A(i, j) \qquad (8)$$

で与えられる[5]．ここで $A(i, j)$ は A の (i, j) 要素である．式 (6) において A の代わりに $-L$ を使用し，右辺第 1 項の A を削除すると**正則化ラプラシアンカーネル** [12] を得る．ここで，$\beta(\geq 0)$ を拡散係数とすると，行列

$$R_\beta = \sum_{n=0}^{\infty} \beta^n (-L)^n = (I + \beta L)^{-1} \qquad (9)$$

は**正則化ラプラシアン行列**と呼ばれる．

正則化ラプラシアンもノイマンカーネル同様グラフ内の全経路を計算するモデルとなっているため，高次の文脈パターンを考慮に入れることができる．また，式 (7),(8) が示すように，負のラプラシアン $-L$ はグラフ G の自己ループの重みを変更したものと考えられる．すると，あるインスタンスが他の多くのインスタンスと共に共起するほど，より強い負の重みがそのノードの自己ループに付与される．つまり，ジェネリックパターンを介して共起しているインスタンス間の関連度はノイマンカーネルの場合と比較して低く見積もられる．

5 実験

実験設定は 3.2 節と基本的に同じであるが，単語 *bank* だけではなく (i) 全名詞を対象にした比較 (ii) 全単語を対

[5] グラフラプラシアンでは A を正規化すると性能が向上するという報告 [3] があるので $D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ によって正規化した．

表 2: 教師なし語義曖昧性解消手法

アルゴリズム	適合率	再現率	F 値
Espresso（足切りなし）	42.8	42.8	42.8
Espresso（足切りあり）	63.6	63.6	63.6
ノイマンカーネル	64.9	64.9	64.9
正則化ラプラシアン	65.4	65.4	65.4
ベースライン（最頻出語義）	55.2	55.2	55.2
Cymfony	57.9	57.9	57.9
Prob0	54.7	54.7	54.7
clr04	45.0	45.0	45.0
Duluth-SenseRelate	40.3	38.5	39.4

表 3: β によるノイマンカーネルの適合率の変化

アルゴリズム	適合率	再現率	F 値
Espresso（足切りなし）	42.8	42.8	42.8
$\beta = 1 \cdot 10^{-4}$	44.1	44.9	44.5
$\beta = 5 \cdot 10^{-5}$	55.2	55.2	55.2
$\beta = 1 \cdot 10^{-5}$	64.9	64.9	64.9
$\beta = 1 \cdot 10^{-6}$	64.9	64.9	64.9

象にした比較，の二つを行った．両カーネルにおけるパターン-インスタンス共起行列 P の (i, j) 要素は *Espresso* にならって *pmi* を用いた．評価には適合率と再現率，そして F 値（適合率と再現率の調和平均）を使用した．

表 1 はグラフベースの語義曖昧性解消手法を比較するために掲げた．Agirre らは名詞のみを対象にしているため，全名詞中正しく語義を当てられた数から算出した適合率と再現率を示した．HyperLex や PageRank よりも *Espresso* や提案手法のほうが数ポイント高い適合率であった．

表 2 は Senseval-3 Lexical Sample Task での教師なし語義曖昧性解消手法を比較するために掲げた．比較対象としては WordNet などの外部リソースを用いないシステム [7] である．提案手法は教師なしの他の語義曖昧性解消手法と比較して高い適合率を保っている．

表 3 は Senseval-3 Lexical Sample Task においてノイマンカーネルの拡散係数 β を変化させたときの性能の変化を示している．実際，β が小さいときは関連度に，大きいときは大域的重要度に偏った結果となっている．

6 議論

提案手法は既存のグラフベースの手法と比べて高い適合率を示し，グラフ解析の手法がブートストラップにおける意味ドリフトを防ぐために有効であることが分かった．名詞を対象にした語義曖昧性解消ではノイマンカーネルと正則化ラプラシアンは有意な差が見られなかったが，全単語を対象にした場合正則化ラプラシアンのほうが適合率が高かった．これは，動詞や形容詞の場合は名詞より周辺文脈に間するジェネリックパターンの影響が強いので，ラプラシアンによってそれを抑えることに成功しているためだと考えられる．グラフベースの手法はいずれも最頻出語義を用いるベースラインを大きく上回り，提案手法が優れていることを示している．

ブートストラップで用いられていたヒューリスティクスとしての足切りは，ジェネリックパターンを選択しないために省略できないステップであることが分かったが，そのためにはパターン数やインスタンス数といった多数のパラメータをタスク・ドメインに応じて調整する必要がある．提案手法はパラメータの数が *Espresso* などのブートストラップと比べて少なく，調整は容易である．

7 まとめ

本研究ではブートストラップを含めた再帰的パターン-インスタンス抽出アルゴリズムにグラフ解析としての解釈を与え，HITS におけるトピックドリフトとブートストラップにおける意味ドリフトの類似性について指摘した．ブートストラップでの意味ドリフトを防ぐ目的で，HITS との類似性からノイマンカーネルおよび正則化ラプラシアンが使えることを，語義曖昧性解消タスクにより示した．

今後，ブートストラップ手法で用いるヒューリスティックなパラメータに関してグラフ解析の観点から考察を加え，固有表現抽出タスクでも同様の手法が適用できるかどうか検証する予定である．また，ブートストラップ手法ではインスタンス獲得に用いたパターンは捨ててしまうことが多いが，[6] のようにスコアの高いパターンはそれ自体で獲得されたインスタンスの特徴づけに使えないかどうか，検討してみたい．

参考文献

[1] E. Agirre, D. Martínez, O. L. de Lacalle, and A. Soroa. Two graph-based algorithms for state-of-the-art WSD. In *Proc. of EMNLP*, pp. 585–593, 2006.

[2] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[3] R. Johnson and T. Zhang. On the Effectiveness of Laplacian Normalization for Graph Semi-supervised Learning. *JMLR*, 8:1489–1517, 2007.

[4] J. Kandola, J. Shawe-Taylor, and N. Cristianini. Learning Semantic Similarity. In *NIPS 15*, pp. 657–664, 2002.

[5] J. Kleinberg. Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM*, 46(5):604–632, 1999.

[6] M. Komachi and H. Suzuki. Minimally Supervised Learning of Semantic Knowledge from Query Logs. In *Proc. of IJCNLP*, pp. 358–365, 2008.

[7] R. Mihalcea, T. Chklovsky, and A. Kilgarriff. The Senseval-3 English lexical sample task. In *Proc. of Senseval-3*, pp. 25–28, 2004.

[8] P. Pantel and M. Pennacchiotti. Espresso: Leveraging Generic Patterns for Automatically Harvesting Semantic Relations. In *Proc. of COLING-ACL*, pp. 113–120, 2006.

[9] P. Pantel and M. Pennacchiotti. Espresso: Leveraging Generic Patterns for Automatically Harvesting Semantic Relations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, pp. 113–120, 2006.

[10] A. Purandare and T. Pedersen. Word Sense Discrimination by Clustering Contexts in Vector and Similarity Spaces. In *Proc. of CoNLL*, pp. 41–48, 2004.

[11] H. Schütze. Automatic Word Sense Discrimination. *Computational Linguistics*, 24(1):97–123, 1998.

[12] A. J. Smola and R. I. Kondor. Kernels and Regularization of Graphs. In *Proc. of COLT*, pp. 144–158, 2003.

[13] J. Véronis. HyperLex: Lexical Cartography for Information Retrieval. *Computer Speech & Language*, 18(3):223–252, 2004.

[14] D. Yarowsky. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In *Proc. of ACL*, pp. 189–196, 1995.

[15] 伊藤, 新保, 工藤, 松本. カーネル法による引用解析の統一的解釈. 人工知能学会論文誌, 19(6 SP-C):530–539, 2004.