機械学習フレームワークによる共起関係を利用したテキスト分類

福元　伸也† 　淵田　孝康†

† 鹿児島大学学術研究院理工学域　〒 890–0065 鹿児島県鹿児島市郡元 1–21–40
E-mail: †{fukumoto,fuchida}@ibe.kagoshima-u.ac.jp

あらまし　インターネットの普及や計算機環境の発達により，膨大なデータを処理するための研究が脚光を浴びている．近年では，ビッグデータ呼ばれる大規模データから有益な情報を取り出そうとする研究も広く行われており，テキストデータの解析に関する研究も盛んである．本研究では，語の意味まで考慮した特徴ベクトルの生成を実現するため，分類語彙表を利用した共起行列生成手法を提案する．単語同士による共起行列を，語の意味を考慮した分類語に置き換えることにより，共起行列の大きさの増大を抑えることができる．得られた共起行列を用いた分類には，学習器としてアンサンブル学習の 1 つである Random Forest と，それを改良した Global Refinement Random Forest (GRRF) を用いた．また，機械学習フレームワークの Jubatus を用いて識別を行った．実験では，ニュース記事のカテゴリ分類を行い，いくつかの学習器で識別精度を比較し検証を行った．

キーワード　テキスト分類, 機械学習, 共起行列

1. はじめに

近年，さまざまな分野で膨大な量のデータが生成され，大量のデータを扱う機会が増大している．また，情報機器の発達やクラウド環境の充実も著しく，テキストデータの解析に関する研究も盛んに行われている．テキストデータをグループ分けするテキスト分類に関する研究も広く行われており，大量の文書データを効率よく分類する手法も多く提案されている [1]．テキスト分類においては，文章を構成する語の重みを特徴ベクトルとして表現して分類を行う．このため，テキストデータが，単なる文字データとして扱われ，その語の意味までは考慮されておらず，人間が持つ自然な言葉の印象とは異なる結果を生じることがある．

本研究では，単語の特徴ベクトル生成において，分類語彙表 [2] を利用する．分類語彙表は，人手により語を意味によって分類・整理した類義語集であり，語の持つ意味をうまく反映させることで，我々人間の感覚に近い特徴ベクトルの生成が期待できる．

文書内には，似たような意味を持つ単語が複数存在するため，単語同士の共起行列を用いて特徴ベクトルを生成すると，本来似ている意味の単語が，距離の離れた特徴ベクトルとして表現されてしまい，精度の低下が生じてしまう問題がある [3]．

本研究では，語を意味により分類したシソーラスである分類語彙表を用いることで，単語の持つ意味を考慮した共起行列を作成する．その共起行列を学習データとして，分類のための学習器に与える．学習器には，アンサンブル学習の 1 つであるランダムフォレスト（Random Forest）[4] を利用し，また，Random Forest を改良した Global Refinement Random Forest [5] を用いて，文書分類を行った．また，アンサンブル学習とは別の学習器による分類も試みた．大規模データに対し，高度な分析が可能な機械学習フレームワークである Jubatus [6] を用い，Jubatus 上で動くいくつかの学習アルゴリズムを試してみた．Jubatus は，大規模データのさまざまなデータ分析に優れた性能を示している．実験では，ニュース記事のカテゴリ分類を行い，Random Forest と Global Refinement Random Forest との比較や，Jubatus における複数の学習アルゴリズムでの識別率の比較を行った．

2. 関連研究

テキスト解析は，非常に重要な技術である．これまで，テキスト解析に関する多くの研究が，分類精度の向上にチャレンジしており，分類に関するさまざまな学習法を提案している．文書分類の研究として，グラフ構造を学習に利用する方法や単語の係り受け関係を用いて分類を行う研究や文書中に現れる語の共起関係を用いたものなどがある [7]．江里口らは，2 種類の情報に基づいた類似度をグラフの辺の重みとしたグラフ構成法を提案し，文書分類の精度を向上させた [8]．花井らは，確率モデルのナイーブベイズ法を利用して，2 単語間の依存関係を考慮して，より正確な確率を計算する手法を提案し，文書分類の精度向上を図った [9]．Wang らは，語の重要度の決定において，PageRank アルゴリズムを用いることが，分類に有効であることを示した [10]．

また，単語の特徴ベクトル生成において，単語の共起行列を作成するために，文書に現れた単語間の共起頻度を利用する手法が提案されている．単語同士の共起頻度を取るだけでは，意味的に近い単語であっても，別の共起頻度としてカウントされてしまい，それらの単語の特徴ベクトルが離れてしまう問題があった [11]．別所らは，単語間の共起頻度ではなく，単語とコーパスにおける単語に付随する意味属性との共起頻度を取る手法を提案している [12]．

3. 共起行列生成

3.1 共起行列における問題点

文書中に出現する単語の共起関係に基づき，ある単語と別の

分類語
意味属性　家具

出現語＼共起語	---	a: たんす	b: ケース	---
---	---	---	---	---
A: 衣装	---	12	65	---
B: 衣服	---	43	9	---
---	---	---	---	---

図 1　共 起 行 列

単語の共起関係の頻度を成分にした行列が共起行列である．単語間の共起頻度を利用した共起行列では，対象となったすべての単語が含まれてしまうため，行列の大きさが巨大になってしまう．行列の次元数が大きくなると次のような問題がある．
1) 次元数の増大に伴い，計算コストが増大する，2) スパースな行列になる，3) 本来，近い関係にあるべき特徴ベクトルが離れた状態になってしまう，などが挙げられる．そこで，単語行列を属性行列に変換する手法が提案されており，笠原らは，国語辞典を用いる手法を提案している [13].

3.2　単語による特徴ベクトル

全テキストデータに含まれる単語を w_i とし，N 個の単語が含まれているとすると，単語 w_i の特徴ベクトルは次のように表される．

$$w_i = (v_{i1}, v_{i2}, \cdots, v_{iN}) \tag{1}$$

ただし，v_{ij} は w_i における重みである．単語特徴ベクトル w_i を要素とした列ベクトルは，次のような行列で表される．

$$F_w = \begin{pmatrix} w_1 \\ \vdots \\ w_N \end{pmatrix} = \begin{pmatrix} v_{11} & \cdots & v_{1N} \\ \vdots & \ddots & \vdots \\ v_{N1} & \cdots & v_{NN} \end{pmatrix} \tag{2}$$

この単語特徴行列から属性行列を生成する．属性数（行列の列数）を m とすると，単語の属性ベクトル $\acute{w}_1, \cdots, \acute{w}_N$ および，その列ベクトルは次の行列で表される．

$$F_p = \begin{pmatrix} \acute{w}_1 \\ \vdots \\ \acute{w}_N \end{pmatrix} = \begin{pmatrix} \acute{v}_{11} & \cdots & \acute{v}_{1m} \\ \vdots & \ddots & \vdots \\ \acute{v}_{N1} & \cdots & \acute{v}_{Nm} \end{pmatrix} \tag{3}$$

ただし，$\acute{v}_{ij}$ は $\acute{w}_i$ における重みである．

3.3　分類語による共起行列の生成

単語間の共起頻度を利用した共起行列の作成において，単語同士の共起をどのように取るかにより，単語間の共起ベクトルの距離が離れてしまう問題がある [12].

図 1 の表は，左端の列が記事中に現れた単語の例を表しており，上段の共起語は，同一文章中に現れた共起語を表している．また，表中の数字は，その出現頻度を表している．単語間の共起に基づいた共起行列の作成手法では，出現語 A と B が，意味の近い単語であったとして，その共起語 a と b が別々にカウントされる．(a と b も意味の近い単語）そうなると，A と B のベクトルは，意味の近い単語であるにもかかわらず離れてしまう．そこで，共起語に現れた a と b を同じ意味を表す 1 つの単語にまとめることが出来れば，A と B のベクトルは離れない．

図 1 で見てみると,「衣装」と「衣服」は，意味の近い単語である．その共起行列は,「衣装」は「ケース」の出現頻度が高く,「衣服」は「たんす」の出現頻度が高い．そうすると,「衣装」と「衣服」それぞれの特徴ベクトルは，離れた状態となる．これを「たんす」と「ケース」の分類語である「家具」にまとめることができれば，それぞれの特徴ベクトルの向きは離れずに済むことになる．

本研究では，意味の似ている語をまとめると共起ベクトルの距離は近くなるという点に着目し，単語同士の共起頻度を用いるのではなく，単語に付随する意味属性を利用する．単語の意味属性には，単語を意味によって分類整理したシソーラスである分類語彙表 [2] を利用し分類語に適用する．分類語彙表を構成する項目は，図 2 のようになっており，共起行列に用いる意味属性には，その中の「分類項目」を用いた．共起行列の 1 列目には，形態素解析の結果得られた単語のうち，名詞のみを取り出し入力し，数字の部分は，1 文中に共起する頻度をカウントした数が入った行列となっている．また，1 行目には，意味属性として分類語彙表の分類項目の語が入る [14].

このようにして得られた共起行列は，式 (3) に相当し，単語間の共起行列である式 (2) から式 (3) を導き出す作業は，次式で表される変換行列 K を求めることに等しい [15].

$$F_p = F_w K \tag{4}$$

ただし，K は，N 行 m 列の行列である．

4.　識別のための学習手法

4.1　Random Forest による学習

アンサンブル学習では，決して精度の高くない複数の学習器を用いて，それらの結果を統合して，精度の向上を図ろうとする．一般に，アンサンブル学習では，異なるデータサンプルから，比較的単純な学習器を複数組み合わせることにより，全体として高い精度を実現するモデルの構築が行われる．アンサンブル学習の 1 つに，データセットからブートストラップによって，複数の学習用データセットをサンプルとして生成し，分類を行うランダムフォレスト（Random Forest）[4] と呼ばれる学習法がある．(以下，RF と表す.）RF は，複数の木（tree）に

レコード ID 番号／見出し番号／レコード種別／
類／部門／中項目／分類項目／分類番号／
段落番号／小段落番号／語番号／見出し／
見出し本体／読み／逆読み

図 2　分類語彙表の項目

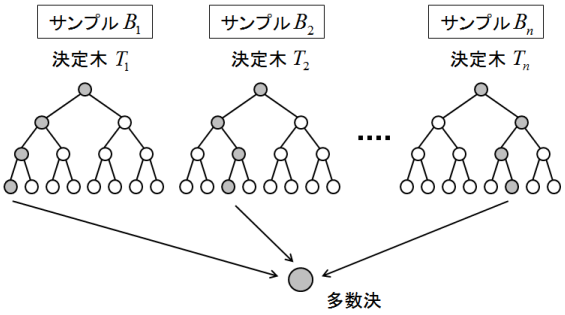

図 3　Random Forest

よって構成される機械学習アルゴリズムである．ここでの木は，決定木のことで，それぞれの決定木の性能はあまり高くない．それらを複数組み合わせることにより，高い予測精度を持つ学習器となる．RF では，決定木として二分決定木が主に用いられ，個々の決定木がアンサンブル学習における弱学習器となる．RF のアルゴリズムは，以下のようになる [16]．(図 3 参照)

（1） 与えられたデータセットから n 個のブートストラップ・サンプル $B_1, B_2, \cdots, B_n$ を作成する．ただし，構築したモデルを評価するために 1/3 のデータを除いてサンプリングする．除いたデータを OOB（out-of-bag）データと呼ぶ．

（2） $B_k(k = 1, 2, \cdots, n)$ における M 個の変数の中から m 個の変数をランダムサンプリングする．M は，データセットの中の変数の数を表し，m は，$m = \sqrt{M}$ が多く用いられる．

（3） ブートストラップ・サンプル B_k の m 個の変数を用いて，未剪定の最大の決定木 T_k を生成する．

（4） n 個のブートストラップ・サンプル B_k の決定木 T_k について，OOB データを用いてテストを行い，推測誤差を求める．

（5） その結果を統合し，新たに分類器を構築する．分類の問題では多数決をとる．

4.2 RF の大域的改良（Global Refinement）

RF において，複雑な問題を扱う場合，学習データをうまく適合させるためには，深い木構造が必要になる．このことは，メモリコストの増大などの問題を生じている．Ren らは，RF の学習と予測に食い違いがあるのではないかと考えた．そこで，RF の構造はそのままで，重みを変えることにより精度が向上するのではないかと考え，RF の性能を改善する新たな手法として，Global Refinement Random Forest を提案した [5]．(以下，GRRF と示す.) GRRF は，リーフ（葉）部分の重みを再学習させるパートと重要性の低い部分の枝刈りのパートからなり，これら 2 つのパートを交互に実行しながら学習を行う．

4.2.1 RF の再学習

RF におけるリーフベクトルの学習は，次式に従い行われる．

$$\min_{w} \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} l(y_i^t, \hat{y}_i) \qquad (5)$$
$$s.t.\ y_i^t = w_t\ \phi_t(x_i),\ \ \forall i \in [1, N],\ \forall t \in [1, T],$$

ただし，T は木の数，$\phi_t(x)$ は指標ベクトル，w_t は t 番目の木のリーフマトリクス，y_i^t は t 番目の木の予測を表す．

表 1　学習アルゴリズム

アルゴリズム	特徴
Perceptron	・線形分離可能である場合、有限回数で分離可能 ・分類器で正しく分類出来なかった場合に、重みを更新する
PA	・Perceptron よりも学習効率が高い ・学習データが正しく分類できたら、重みを更新しない
CW	・Perceptron, PA と比べて学習効率は高いが、計算量は大きい ・出現頻度を考慮して、重みベクトルにガウス分布を導入して更新する
AROW	・学習データにノイズが含まれた場合意に他の手法と比べ優れた学習効率を示す ・計算量は、CW と同程度 ・CW と同様の手法を実現しつつ、複数の条件を同時に考慮しながら最適化する
NHERD	・特徴ベクトルが正規分布に従って生成されているモデルを利用し学習を行う

RF の大域的改良は，次式により行われる．

$$\min_{W} \frac{1}{2}\|W\|_2^2 + \frac{C}{N} \sum_{i=1}^{N} l(y_i, \hat{y}_i) \qquad (6)$$
$$s.t.\ y_i = W\ \Phi(x_i),\ \ \forall i \in [1, N].$$

この式には，リーフベクトルが過学習になるのを避けるため，L2 正則化項が入っている．また，C は，正則化と学習データの損失のバランスを取るためのパラメータである．式 (6) で表される目的関数は，サポートベクタマシン (SVM) のそれと同じになる．すなわち，大域的最適化による凸最適化により解決できる．大域的改良において，木の構造は変えずに，リーフのベクトルのみを変更する．

4.2.2 大域的枝刈り（Global Pruning）

大域的な枝刈りでは，大域的最適化を用いて，以下に示す手順により，あまり重要でないリーフの統合を繰り返す．

1. 式 (6) に従いリーフベクトルを最適化する．
2. リーフベクトルのノルムがゼロに近づいたら，隣接したリーフを 1 つの新しいリーフに統合する．
3. 統合は，隣接したリーフで，リーフベクトルの L2 ノルムを計算し，その値が小さい方から一定の割合で，新しいリーフに統合する．
4. 新たな木構造に従って，指標ベクトルを更新する．枝刈りされたリーフの指標値のみを削除し，新しいリーフに指標値を追加する．

学習過程では，精度がしきい値以下になるなどの特定の条件を満たすまで，1 ～ 4 のステップを繰り返す．

4.3 Jubatus による学習

ビッグデータのような大量のデータを処理するための機械学習として，Hadoop [17] が提供されている．また，大量のストリームデータを処理する機械学習として，Jubatus が提供されている [6]．Jubatus は，リアルタイム処理，分散並列処理，深

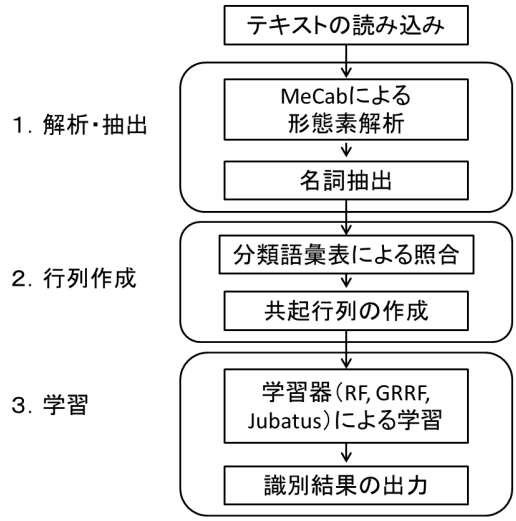

図 4　処理の流れ

い解析などの特徴を持っている．

ここでは，アンサンブル学習器とは別の学習器として，Jubatus について，その Jubatus が持つ複数の学習アルゴリズムについて説明する．文書分類に必要な操作は，多クラス分類であり，Jubatus は，線形識別器を用いて，これを実現している．Jubatus の多クラス分類において利用できる学習アルゴリズムには次のものがある．(i) Perceptron [19], (ii) Passive Aggressive (PA) [20], (iii) Confidence Weighted Learning (CW) [21], (iv) Adaptive Regularization Of Weight vectors (AROW) [22], (v) Normal Herd (NHERD) [23] である．各学習アルゴリズムの特徴を表 1 に示す．今回我々は，NHERD を除いた 4 つの学習アルゴリズムを用いて文書分類を試みた．

共起行列の生成と RF, GRRF および Jubatus の学習器による文書識別までの処理の流れを図 4 に示す．

5. 実　　験

3 節で説明した共起行列作成法を利用して，単語の特徴ベクトルを生成し，それをいくつかの学習器に与えて，識別精度を調べる実験を行った．実験では，毎日新聞社のサイト [24] よりニュース記事を収集し，記事に現れる単語のカテゴリ分類を行った．分類に用いた記事の数は，1,800 で，政治，経済，社会，スポーツ，エンターテイメントの 5 つのカテゴリに分類する．読み込んだテキストデータは，形態素解析器の MeCab [25] を用いて解析し，その中から名詞の単語を取り出し，共起行列作成のための出現単語として使用した．抽出された単語の数は，18,761 個であった．この出現語を用いて，共起行列を生成すると行列の大きさは，$18,761 \times 18,761$ となる．提案手法では，分類語彙表を用いて共起行列を生成するため，共起行列の大きさを小さくすることができる．分類語彙表を用いた場合，単語を分類項目でまとめることができ，その数は，510 個となった．ただし，単語の数も，分類語彙表に掲載されている単語にとどまることとなり，その数は，10,645 個となり，すなわち，共起行列の大きさは，$10,645 \times 510$ となった．

表 2　識 別 結 果

	RF	GRRF
正識別率 (%)	87.6	88.9

表 3　学習アルゴリズムによる比較

	Perceptron	PA	CW	AROW
regularization weight	-	-	1.0	1.0
正識別率 (%)	82.1	84.5	83.8	85.4

生成された共起行列を学習データとして学習器に与える．まず，RF と GRRF を用いて精度を比較した．GRRF におけるリーフの統合は，低い方から 10% とした．結果を表 2 に示す．その結果，RF と GRRF では，GRRF の識別率が高い結果となった．

次に，学習器に Jubatus を用いて実験を行った．学習アルゴリズムとして，Perceptron，PA，CW，AROW の 4 つの学習アルゴリズムを用いて識別を行った．表 3 に各学習アルゴリズムごとの識別結果を示す．比較した 4 つの学習アルゴリズムの中では，AROW による識別率が最も高い結果となった．

6. おわりに

本研究では，テキストデータ中に現れる単語の特徴ベクトル生成において，単語の共起頻度から分類語彙表を利用して，共起行列を作成する手法を提案した。単語の特徴ベクトルを用いたクラスタリングにおいて，アンサンブル学習の 1 つであるランダムフォレスト (RF) と RF を改良した GRRF を用いて比較した．また，機械学習フレームワークである Jubatus を使用し，Jubatus に対応した複数の学習アルゴリズムで比較を行った．実験では，ニュース記事のカテゴリ分類を行い，GRRF と AROW を用いた場合，精度が高かった．

今後の課題として，他の学習アルゴリズムを使用した場合の識別精度について調べてみたい．

文　　献

[1] R. M. Samer Hassan and C. Banea: “Random-walk term weighting for improved text classification”, Proc. of the First Workshop on Graph Based Methods for Natural Language Processing (2006).
[2] 国立国語研究所：“分類語彙表 - 増補改訂版”, 大日本図書刊 (2004).
[3] 有村博紀：“テキストマイニング基盤技術”, 人工知能誌, **16**, 2, pp. 201–211 (2001).
[4] L. Breiman: “Random forests”, Machine Learning, **45**, pp. 5–32 (2001).
[5] Ren, Shaoqing, Cao Xudong, Wei Yichen and Sun Jian: “Global refinement of random forest”, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.723–730 (2015).
[6] Jubatus, http://jubat.us/ja/.
[7] 渡部広一, 奥村紀之, 河岡司：“概念の意味属性と共起情報を用いた関連度計算方式”, 自然言語処理, **13**, 1, pp. 53–74 (2006).
[8] 江里口瑛子, 小林一郎：“テキスト分類のための潜在トピックを考慮したグラフ構成”, 第 4 回 インタラクティブ情報アクセスと可視化マイニング研究会, **SIG-AM-04-04**, pp.23–28 (2013).
[9] 花井拓也, 山村 毅：“単語間の依存性を考慮したナイーブベイズ法によるテキスト分類”, 情報処理学会自然言語処理研究会報告, **NL-166**, pp.101–106 (2005).

[10] D. B. D. Wei Wang and X. Lin: “Term graph model for text classification”, Springer-Verlag Berlin Heidelberg 2005, pp. 19–30 (2005).
[11] 片岡 良治 : “単語と意味属性との共起に基づく概念ベクトル生成手法”, 人工知能学会第 20 会全国大会論文集, **3C3-1**, pp. 1–3 (2006).
[12] 別所克人, 内山俊郎, 内山匡, 片岡良治, 奥雅博 : “単語・意味属性間共起に基づくコーパス概念ベースの生成方式”, 情報処理学会論文誌, **49**, 12, pp. 3997–4006 (2008).
[13] 笠原要, 松澤和光, 石川勉 : “国語辞書を利用した日常語の類似性判別”, 情報処理学会論文誌, **38**, 7, pp. 1272–1283 (1997).
[14] 尾脇拓朗, 福元伸也 : “単語の意味を考慮した共起ベクトルによるテキスト分類”, DEIM Forum 2014, **C6-2**, (2014).
[15] 笠原要, 稲子希望, 加藤恒昭 : “単語の属性空間の表現方法”, 人工知能学会論文誌, **17**, pp. 539–547 (2002).
[16] 金明哲 : “統計的テキスト解析”, ESTRELA, 182 (2009).
[17] Hadoop, http://hadoop.apache.org/.
[18] 岡野原大輔 : “大規模データ分析基盤 jubatus によるリアルタイム機械学習”, 人工知能学会誌, **28**, 1, pp. 98–103 (2013).
[19] F. Rosenblatt: “The perception: a probabilistic model for information storage and organization in the brain”, Neurocomputing: foundations of research MIT Press, pp. 89–114 (1988).
[20] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz and Y. Singer: “Online passive-aggressive algorithms”, The Journal of Machine Learning Research, **7**, pp. 551–585 (2006).
[21] M. Dredze, K. Crammer and F. Pereira: “Confidence-weighted linear classification”, Proceedings of the 25th international conference on Machine learningACM, pp. 264–271 (2008).
[22] K. Crammer, A. Kulesza and M. Dredze: “Adaptive regularization of weight vectors”, Advances in Neural Information Processing Systems, pp. 414–422 (2009).
[23] K. Crammer and D. D. Lee: “Learning via gaussian herding”, Advances in neural information processing systems, pp. 451–459 (2010).
[24] 毎日新聞, http://mainichi.jp/.
[25] Mecab, http://mecab.googlecode.com/svn/trunk/mecab/doc/.