DEIM Forum 2012 E10-4

確率的文書生成モデルを用いた多重トピック文書分類手法に関する考察

丹羽　歩美†　　深川　大路†

† 同志社大学文化情報学部文化情報学科　〒 610–0394 京都府京田辺市多々羅都谷 1–3
E-mail: †bii0155@mail4.doshisha.ac.jp, ††dfukagaw@mail.doshisha.ac.jp

あらまし　本研究では，各文書が複数のトピックを有することを許容するような確率的文書モデルを扱う．近年，各文書が複数のトピックを持つことを許容するさまざまな確率モデルが提案されてきた．それらのうち特にナイーブベイズ分類器 (NB) を実装し，Web データに対して適用しデータの処理においてさまざまな重み付けの手法を試すことにより，より適切なトピック分類の手法を探り，トピック分類における有効性を検証した．
キーワード　確率的文書生成モデル，テキストマイニング，トピック分類

A Study on Multi-topic Document Classification Using a Generative Probabilistic Model

Ayumi NIWA† and Daiji FUKAGAWA†

† Faculty of Culuture and Information Science, Doshisha University
1–3 Tatara Miyakodani, Kyotanabe, Kyoto 610–0394, Japan
E-mail: †bii0155@mail4.doshisha.ac.jp, ††dfukagaw@mail.doshisha.ac.jp

1. はじめに

まず，現代においては日常的に莫大な数の文書に触れることがもはや珍しいことではなく，むしろ日常茶飯事となっている．そのため，自分が求める文書を正確に手早く見つけ出す手法が必要となっているという現状が存在する．また，デジタルデバイドという言葉に表されているように個々人の情報検索能力には大きな差が存在している．情報検索能力の差によって自分の求める情報を的確に見つけ出せる人と見つけ出せない人が存在しており，自分の求める情報を的確に見つけ出せない人は日常的に不利益を被っていると考えられる．それに加え，情報を探し出す側面ではなく情報を整理する側面においては，従来人手でも行うことが十分可能であった文書のトピック分類が文書の母体数の爆発的な増加により不可能になっており，より効率的な分類手法の必要性が叫ばれている．なお、本研究におけるトピックとは文書において話題になる事柄を指す．

さて，一般的に文書というものは所属トピックが単一ではなく多重であると言われている．ゆえに文書分類問題とは，ある文書がどのトピックに所属するかを所属トピックを複数許容して分類することであると定義することができる．また，ここにおいて所属トピックの種類やその総数は予め定められている場合と定められていない場合が存在している．

本研究の目的は効率的に文書を分類する手法を提案し，従来手法の分類性能・分類時間の改善を図ることである．また，本研究においては文書というものが所属するトピックは単一ではなく複数を許容し，研究を行った．

なぜトピックを単一に限定するのではなく多重を許容されなくてはならないかは次の文書を例に使用して述べる．

> 6 日午前 10 時 10 分ごろ、沖縄県・久米島の北北東約 102 キロの日本の排他的経済水域（EEZ）で、中国の海洋調査船「科学 1 号」が船尾からロープとワイヤのようなものを垂らしながら漂泊しているのを、警戒中の海上保安庁の航空機が確認した。

上記の文書は 2011 年 12 月 6 日の Yahoo!JAPAN ニュースにおけるある記事(注1)の概要である．この文書は「国際」「政治」「経済」といったトピックに所属することが考えられるが，ただひとつだけのトピックに絞り込むことは適切ではない．この問題を解決するためには，文書をトピックごとに分類する際において文書に対して付与するトピックをただひとつに限定するのではなく，複数のトピックに重複して所属することを認める必要がある．上記の例のように，一つの文書が複数のトピックに重複して所属することができるような文書を多重トピック文書と定義する．

(注1)：http://headlines.yahoo.co.jp/hl?a=20111206-00000117-jij-soci

そこで本研究では文書をさまざまな手法により複数トピックに所属することを許容して多重トピック分類する方法について考察する．

2. 関 連 研 究

本章では確率的文書生成モデルと実験において実際に使用した確率モデルであるナイーブベイズ分類器について説明をする．

2.1 確率的文書生成モデル

確率的文書生成モデルにおいて代表的なものとしては，ナイーブベイズ分類器やサポートベクトルマシンといった二値分類器がよく知られている．なお，二値分類器はそのトピックに当てはまるか当てはまらないかを元に分類を行うため，文書分類を行う際に複数トピックに対してそれぞれトピックごとに二値分類器を適用し，しきい値を設定することで複数トピックに所属することを許容した文書分類が可能になる．

2.2 ナイーブベイズ分類器

ナイーブベイズ分類器 [5] はメールのスパムフィルタをはじめとして日常的に多く使用されている確率に基づいた分類器である．事例 d に対して，事後確率 $P(c \mid d)$ が最大となるクラス $c \in C$ を出力する．なお，C はクラスの集合である．基本的にはベイズの定理

$$p(c \mid d) = \frac{P(c)P(d \mid c)}{P(d)}$$

の性質を利用しながら右辺が最大となるクラス c を出力するという単純な仕組みに基づいた分類器である．また，この計算において，分母 $P(d)$ はクラスに依存しないため，最大となるクラスを決定する際には不要である．つまり，ナイーブベイズ分類器においては分子 $P(c)P(d \mid c)$ を最大にする c_{max} を出力する問題のみを考えれば良い．そのため，

$$c_{max} = \underset{c}{\operatorname{argmax}} \frac{P(c)P(d \mid c)}{P(d)} = \underset{c}{\operatorname{argmax}} P(c)P(d \mid c) \qquad (1)$$

となる．しかし，$P(d \mid c)$ を計算することは文書における単語の種類数とその組み合わせを網羅しなくてはならない．つまり，すべての d において単語の種類数と組み合わせを何回起こるか調べ，$P(d \mid c)$ を最尤推定にてもとめることは非現実的である．よって，ナイーブベイズ分類器は文書である d に簡略化したモデルを仮定して $P(d \mid c)$ を求める．

ここでは，多項モデルを用いて $P(d \mid c)$ を求める式の導出について紹介する．多項モデルにおいては文書中の各位置についてどんな単語が起こるかをモデル化する．ここで，文書 d 内において単語 w がそれぞれ $n_{w,d}$ 回起こる確率は多項分布を用いて文書の長さを加味すると

$$P(d \mid c) = P(K = \sum_w n_{w,d}) \frac{(\sum_w n_{w,d})!}{\Pi_w n_{w,d}!} \Pi_w q_{w,c}^{n_{w,d}}$$

ここで K は文書の長さを表す確率変数であり，$P(K = \sum_w n_{w,d})$ は長さ $\sum_w n_{w,d}$ である文書が起こる確率である．また，

$$q_{w,c} = \frac{\text{クラス } c \text{ に所属する訓練文書全体での } w \text{ の出現回数}}{\text{クラス } c \text{ に所属する訓練文書全体での全単語の出現回数}}$$

である．また，ここにおいては文書の長さはクラスに依存しないことを仮定している．以上より多項モデルを利用したナイーブベイズ分類器は

$$P(c)P(d \mid c) = P(c)P(\sum_w n_{w,d}) \frac{(\sum_w n_{w,d})!}{\Pi_w n_{w,d}!} \Pi_w q_{w,c}^{n_{w,d}}$$

を最大化するようなクラス c を出力することを考えれば良い．

ここで，

$$\begin{aligned} \underset{c}{\operatorname{argmax}} P(c)P(d \mid c) &= \underset{c}{\operatorname{argmax}} P(c)P(\sum_w n_{w,d}) \frac{(\sum_w n_{w,d})!}{\Pi_w n_{w,d}!} \Pi_w q_{w,c}^{n_{w,d}} \\ &= \underset{c}{\operatorname{argmax}} P(c) \Pi_w q_{w,c}^{n_{w,d}} \end{aligned} \qquad (2)$$

よって，c の最大化について考えるためには $P(c)\Pi_w q_{w,c}^{n_{w,d}}$ について考えればよく，処理は学習とテストの 2 段階に分かれている．まず，学習段階で教師あり学習を行う．その結果，各トピックごとの単語出現確率分布を推定することが可能となる．テスト段階では学習した確率分布に基づいてテストデータの所属トピックを推定する．

3. 提 案 手 法

本章においては提案手法と提案手法に関連する知識について具体的に説明をする．

3.1 概　　要

本研究における最大の研究目的はナイーブベイズ分類器を使用した際の最も適切な多重トピック分類を行うことである．

そのため，本研究においてはナイーブベイズ分類器を用いて文書のトピック分類を行う際に，各トピックにおける特徴的な使用語を考慮して文書を分類する際の分類器に重み付けに利用した．各トピックにおける特徴的な使用語の重み付けには TF-IDF 値を利用した．また，各トピックにおける特徴語をより適切に表現するために中頻度語のみを抽出したデータを作成し，実験を行った．

3.2 TF-IDF 値

TF-IDF 値とは索引語の重み付けを行う際によく使われる手法である．情報検索においては，索引語に対して重みをつけることによって，その索引語の各文書中での重要度を考慮することができる．その結果，検索質問に対する検索結果文書群の検索質問に対する適合度が計算可能になり，同じ索引語を含む文書であっても文書中の使用方法から検索質問に合致するか否か判断できる．つまり，検索結果の文書群を順序付けし，より検索結果に合致するものを上位の検索結果として提示することができる．まず，TF(Term Frequency) とは各単語の出現頻度の総数である．ここで，$n_{i,j}$ はある文書 j 中に出現する単語 i の出現頻度である単語頻度を表している．この単語 i の出現頻度は文書 j における単語 i の重み w_i^j と考えることができる．

$$\begin{aligned} tf_{i,j} &= w_i^j \\ &= n_{i,j} \end{aligned} \qquad (3)$$

上記の式 (3) により TF 値を算出することができる．しかし，式 (3) は文書の長さによる影響を考慮していない．そのため，文

書の長さを考慮するためにと以下のような式が算出できる．

$$tf_{i,j} = w_i^j = \frac{n_{i,j}}{\sum_k n_{k,j}} \quad (4)$$

よって，文書の長さを考慮した TF 値は上記式 (4) により算出することが可能になる．

次に，IDF(Inverse Document Frequency) 値について説明する．TF 値は各文書における各単語の出現頻度を表していた．特定の文書において出現回数が多い単語は一見その文書における重要語のように見える．しかし，その単語がどの文書においてもまんべんなく高頻度で出現していた場合，文書の特徴をはかる上ではあまり役に立たず，むしろ他の重要語を隠してしまう可能性がある．そのため，文書ごとの特徴単語を調べるためにはある特定の単語が全文書中のどれくらいの数の文書に出現するかを表す IDF 値を考慮する必要がある．$|D|$ を総文書数，$|\{d : i \in d\}|$ は単語 i を含む文書数であるとする．

$$idf_i = \log \frac{|D|}{|\{d : i \in d\}|} \quad (5)$$

上記の式 (5) により IDF 値を算出することができる．この式 (5) では IDF 値はある単語が少ない文書中にしか登場しない場合は値が大きくなり，多くの文書中に登場する場合は逆に値が小さくなる．これにより，IDF 値を単語の重みとして用いると少数の文書にしか登場しない単語に大きな重みを付与することができる．

これらの式 (3,5) をもとに，TF-IDF 値を算出する．文書 j における特定の単語 t の頻度を表す TF 値と，特定の単語 i が現れる総文書頻度の逆数の対数である IDF 値を元に算出する．

$$tf\text{-}idf_{i,j} = tf_{i,j} \times idf_i = n_{i,j} \times \log \frac{|D|}{|\{d : i \in d\}|} \quad (6)$$

上記式 (6) により，各単語に対する重要度を重み付ける値である TF-IDF 値が算出可能となる．

3.3 中頻度問題

中頻度単語とは文字通り，出現頻度が高度でも低度でもない中頻度の単語のことを指す．なぜ，中頻度の単語を使用するかというと，まず文書において出現頻度の高い単語というものはその文書の特徴を端的に表現していると一般にはいうことができる．しかし，特定の文書においてのみ高頻度で出現する単語とすべての文書に対してまんべんなく高頻度で出現する単語では同じ高頻度単語であっても文書における重要度が全く変わってしまう．そのため，高頻度の単語を一律で削除することにより，特別文書内容に影響を与えると思われない高頻度に出現する単語を文書から削り，文書モデルがより端的に文書の内容を表すことができるようになる．また，登場頻度が低頻度の単語はもともと文書の内容に対して大きな影響を与えるとは考えづらい．つまり，削除しても文書自体の内容に関しては何ら問題はない，むしろ，登場頻度が低頻度の単語を削除することにより，単語全体の母数が少なくなるため，システムにおける計算時間や記憶容量といった部分に関しては好都合である．したがって，しきい値の上限と下限を定め，高頻度語と低頻度語を削除することにより，確率的文書生成モデルを中頻度語のみで表現することとした．

4. 評価実験

4.1 実験概要

本節においては評価実験に利用したデータセットの内容と実験概要，評価方法について説明した後，実際に実験を行った後の実験結果について述べる．

4.1.1 データセット

実験には SBM サービスである livedoor クリップ(注2)のブックマークデータを利用した．2010 年 6 月現在のブックマーク URL のデータ総数は 3,005,129 件であり，このうち重複した URL を削除した結果データ数は 409,996 件となった．実験に使用するコーパスデータを Web 上の文書に限定するため更にデータの厳選を行い，Web ページの内容を表している HTML データ 170,825 件を実験に利用することとした．なお，このデータは 2010 年 6 月現在のブックマークデータである．これらをインターネットロボット GNU Wget(注3)利用して 2011 年 10 月に Web ページのデータを取得した．データはブックマークされた Web ページの HTML データの中から HTML ヘッダーなどの Web ページの内容に影響を与えない部分と Javascript のような組み込みのコンテンツをテキストブラウザ w3m(注4)を用いて削除し、Web ページの本文のみを抽出した．

分類性能をはかる上で必要な正解例は各ブックマークデータに付与されているタグを元に文書の分類名を付与する．データセットにおいて付与されているタグの総数は 96536 件であった．このうち出現頻度上位 20 件のタグを正解トピック名として使用した．正解トピック名となったタグとそのタグが付与された web ページの数を表 1 にて示す．

トピック名	web ページ数	トピック名	web ページ数
レビュー	1054	口コミ	854
感想	873	web	379
javascript	1988	効果	854
google	1625	評価	483
求人	1403	アルバイト	1187
方法	397	ブログ	1175
css	1231	あとで読む	1744
ネタ	2428	2ch	2082
tips	2939	評判	529
求人広告	1087	firefox	902

表 1　対象データ

正解トピック名として使用したタグは レビュー/口コミ/感想/web/javascript/効果/google/評価/求人/アルバイト/方法/

(注2)：livedoor クリップ：http://clip.livedoor.com/
(注3)：GNU Wget：http://www.gnu.org/software/wget/
(注4)：w3m：http://w3m.sourceforge.net/

	ナイーブベイズ	名詞の出現頻度が中頻度のものを利用したナイーブベイズ	文書長を考慮せず TF-IDF 値を求めたものを利用したナイーブベイズ	文書長を考慮して TF-IDF 値を求めたものを利用したナイーブベイズ
適合率:P	0.1095	0.1090	0.0210	0.0210
再現率:R	0.2026	0.2030	0.0389	0.0389
F-measure	0.1422	0.1418	0.0273	0.0273

表 2　実 験 結 果

ブログ/css/あとで読む/ネタ/2ch/tips/評判/求人広告/firefox の 20 件である．これらのタグが付与されている HTML データは 25214 件となった．また，データの多重度は 31%であった．なお，タグが複数付与されている場合のブックマークデータの HTML 文書は多重トピック文書となり，単一のタグのみ付与されている場合は単一トピック文書となる．

4.2 実　　験

実験はコーパスデータに対して処理を何も行なっていない文書に対するトピック分類，Bag-of-Words 表現に変換する際に算出した単語頻度をもとに，出現頻度 10 回未満の低頻度語と出現頻度上位 10 位の高頻度語を削除することで中頻度の出現語のみで構成されている文書に対するトピック分類，ナイーブベイズ分類器の学習段階において学習した各文書の単語に関して TF-ID 値 F で重み付けを行った分類器でのトピック分類，TF-IDF 値を算出する際に文書の長さを考慮した上で単語に対する重み付けを行った分類器でのトピック分類をそれぞれ行った．

なお，実験に使用した Web ページの所属トピックは表 (1) のようになっており，この中から訓練データを 3,000 件，テストデータを 2,000 件あわせて 5,000 件を無作為に抽出し，実験に使用した．

4.2.1 評 価 方 法

各手法の評価値を示す評価尺度には情報検索やテキスト分類問題で多く使用されている F-measure [7] を使用した．F-measure は P(precision:適合率) と R(recall:再現率) の一様重み付き調和平均で定義される. なお，TP を推定トピックの中で実際に正解トピックを推定できた数，N を推定トピックの数，C を実際の正解トピックの数とする．

本研究においての適合率は分類結果として得られた文書集合中にどれだけトピックに適合した文書を含んでいるかという正確性を表す指標であり，次の式 (7) で算出される．

$$P = \frac{\text{推定トピックの中で正解した数}}{\text{推定トピックの数}} = \frac{TP}{N} \tag{7}$$

また, 再現率は分類対象としている文書の中で分類結果として適合している正解文書のうちどれだけの文書を分類できているかという網羅性を表す指標である. 再現率は次の式 (8) により算出される．

$$R = \frac{\text{推定トピックの中で正解した数}}{\text{実際の正解トピックの数}} = \frac{TP}{C} \tag{8}$$

なお，これらの式において R は正しく分類された適合文書の数，N は分類結果の文書の総数，C は全対象文書中の正解文書の数にあたる．この二式から求めた P と R の一様重み付き調和平均が F-measure である．

$$\text{F-measure} = \frac{2PR}{P+R} \tag{9}$$

上記の式 (9) によって F-measure を求めることができる．

4.2.2 実 験 結 果

上記表 2 がコーパスデータに関して前処理を行わなかったナイーブベイズ分類、コーパスデータに関して名詞以外の形態素を削除した上で出現頻度が中頻度の単語集合によるコーパスデータによるナイーブベイズ分類，ナイーブベイズ分類器の学習段階において学習した各文書の単語に関して TF-ID 値 F で重み付けを行った分類器でのトピック分類，TF-IDF 値を算出する際に文書の長さを考慮した上で単語に対する重み付けを行った分類器でのトピック分類の適合率 P，再現率 R，F-measure である．

4.3 考　　察

表 2 で示されている実験結果においては，コーパスデータに対して全く前処理を行わずナイーブベイズ分類器を適用して分類を行ったトピック分類と文書中における名詞の出現頻度が中頻度のもののみを利用して生成したコーパスデータに対して適用したナイーブベイズ分類器との間には F-measure，トピック分類時間においてほぼ差が見られなかった．差が出なかったことについて考えられる原因としては，まず，文書が Web からとってきた生データであったため，それぞれの文書ごとに大きな違いが見られなかったということ．そして，正解トピック名としてソーシャルブックマークサービスにより付与されているタグの名前を用いたが，タグを付与するのはソーシャルブックマークサービスを利用しているユーザー自身であるということ．そのため，ユーザーごとにタグの付与に対するルール付けが異なるため同じタグを付与されている Web ページであっても内容が同じであるとは言いがたいということがいえる．つまり，この結果からはソーシャルブックマークサービスにおけるユーザーにより付与されるタグは，Web ページの所属トピック名としては不適切であるということがいえるだろう．また，学習段階において TF-IDF 値を求めたうえでトピック分類を行ったものに関しては適合率・再現率ともに文書に対して前処理をしなかった場合のナイーブベイズ分類器と比べとても低い結果が出てしまった．これは学習データの数が十分でなかったためにうまく重み付けを行うことが出来なかったのが原因ではない

かと考えられる．

更に，表 1 からもわかるようにデータの数がトピックごとに均一ではなかったため，訓練データが少ない場合は多くの文書が合致してしまったり，逆に訓練データが多い場合は文書における特徴語が限られてしまうという問題点も推測できる．他にも，コーパスデータに対して前処理を行わなかった場合とコーパスデータの中の名詞のうち中頻度語のみを抽出したものとでトピック分類の性能に大きな差が生まれなかったことは，言い換えれば文書の内容を表す際に従来よりも少ない単語量で従来と同等の性能を残すことができるため，自然言語処理において文書の特徴を表すベクトルの次元を削っても文書内容は変化しないということが考えられる．

5. おわりに

本研究においてはナイーブベイズ分類器を利用して文書を適切な所属トピックに分類する手法を提案した．確率的文書生成モデルを用いて文書が所属するトピックを推定し，分類を行うためには文書ごとの各トピックに対する事後確率を考慮しなくてはならない．本研究においては使用したデータが Web ページの生データであり，確率的文書生成モデルの事後確率を推定する際に各トピックに対して生データの特性を生かしてより特徴づいた事後確率が推定されるとの仮定のもと，データの選定を行い，実験を実施したが，実際には生データであるがゆえに各トピックに対する文書生成の事後確率はあまり大きな差がでないという結果になってしまった．しかし，文書に対して何の処理もしていない場合のトピック分類と出現頻度が中頻度の名詞のみでのトピック分類の性能と分類にかかる時間においてほぼ遜色ない結果であった点から，文書の特徴を測る際出現頻度が中頻度の名詞のみで十分測れる知見を示した．

今後の課題としては複数トピックに分類を行う際の最適なパラメータの推定やしきい値の設定を行ったり，ナイーブベイズ分類器以外の確率モデルを利用してトピック分類を行い，各文書が所属するトピックの推定をより人間の判断による正解データに近づけることが挙げられる．また，今回は比較的小規模なデータによる実験であったため，大規模なデータを用いて同様の実験を行った際には現在の実験結果と比べ，実行時間の短縮や分類性能の上昇が見込まれると考えられる．これらを実験によって確かめることが今後の課題として挙げられる．

文　献

[1] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, Vol. 3, pp. 993–1022, March 2003.

[2] Thomas L. Griffiths and Mark Steyvers. Finding scientific topics. *PNAS*, Vol. 101, No. suppl. 1, pp. 5228–5235, 2004.

[3] 佐藤一誠, 中川裕志. 階層ベイズモデルによる多重トピック文書の確率的生成モデルの構築. 電子情報通信学会第 18 回データ工学ワークショップ &第 5 回日本データベース学会年次大会, 2007.

[4] 津田宏治. サポートベクターマシンとは何か. 電子情報通信学会誌, Vol. 83, No. 6, pp. 460–466, 2000-06-25.

[5] 高村大也. 言語処理のための機械学習入門. コロナ社, 2010.

[6] 上田修功, 斉藤和巳. 多重トピックテキストの確率モデル : パラメトリック混合モデル (バイオサイバネティックス, ニューロコンピューティング). 電子情報通信学会論文誌. D-II, 情報・システム, II-パターン処理, Vol. 87, No. 3, pp. 872–883, 2004-03-01.

[7] 徳永健伸. 情報検索と言語処理. 東京大学出版会, 1999.