修　士　論　文

語の出現予測による
テキスト分類のための
有用な複合素性の獲得

平成18年2月3日提出

指導教官　石塚 満 教授

東京大学大学院 情報理工学系研究科

66409 岡嶋 穣

内容梗概

ウェブなどの発達により、利用可能な電子的なテキストの量は近年ますます増大しており、同時に膨大な量のテキストを扱うための機械的な処理技術の需要が年々高まっている。テキスト分類は、その中でも特に基本的で重要なタスクであり、文書の集合が与えられた時に、文書を分析して定められたカテゴリに自動的に分類するものである。テキスト分類の技術はニュース記事の分類や電子メールのスパム判定など現実の至るところで用いられている。

テキスト分類においては、どんな素性を使って文書を表現するかが、分類の精度を左右する重大な決め手となる。文書中に出現する個々の語が文書を表現する最も基本的な素性であるが、複数の語を組み合わせることでさらに高度な素性を作る手法が様々に研究されている。語の出現の間の統計的な類似性を利用して類似した語同士をまとめてひとつの素性にする手法や、複数のタスクに共通して役立つ構造を抽出し素性とする手法などがある。

しかし、これらの手法は、素性を作る基準としてはいくつかの問題点を抱えている。語の出現の統計的な類似性を基準とする手法は、出現が類似しているから複合させることが文書の識別の役に立つとは限らない。複数のタスクに共通して役立つ構造を素性とする手法は、普遍的に役立つ素性を高く評価するために、実際の目的となるタスクにだけ役立つような素性が無視されてしまう。

本研究では、このような問題点を克服するために、語の出現の予測に基づいて素性を作る新しい手法を提案する。トピックは通常そのトピックに特に関係して出現する様々な語が存在する。もし、そのような語が文書中に出現するかどうかを予測するのに役立つ素性があるならば、その素性はトピックを予測する時にも同様に役に立つ素性であると考えられる。この語の出現の有無を予測できるかどうかという基準を用いれば、文書の識別力を持ち、また特定のタスクにのみ役立つ普遍的ではない複合素性も作り出すことができる。

提案手法では、ある語の出現の有無を他の語の出現を手がかりに予測する問題を作り、分類器で学習する。そして学習された分類器の出力を文書の新しい素性とする。この素性は、ある語が出現すべき状況であるかを表わす素性となり、語の出現の実際の観測とは異なる情報を表現する。

提案手法による複合素性の評価を行うために、Reuters-21578 と 20-newsgroups という二つの代表的なテキスト分類のデータセットで実験を行い、分類精度を向上させることを確かめた。特に用例数が少ない場合に分類精度が高く向上し、提案手法の半教師つき学習としての有用性が示された。

目 次

図 目 次

表 目 次

第1章　序論

1.1　テキスト分類と文書の表現

ウェブなどの発達により、利用可能な電子的なテキストの量は近年ますます増大しており、同時に膨大な量のテキストを扱うための機械的な処理技術の需要が年々高まっている。テキスト分類は、その中でも特に基本的であり重要な技術である。文書の集合が与えられた時に、文書を分析して、定められたカテゴリに自動的に分類する。この技術は、例えば配信されたニュース記事をそれぞれのカテゴリに分類すること、あるいは電子メールがスパムかどうか判定するなど、現実の至るところで使われている。

このテキスト分類においては、文書を数値的にどう表現するかが重要な課題となる。文書はそのままの形では機械的に処理できず、文書の特徴を表わすいくつかの数値の組み合わせで表現されて初めて、分類器で分類を学習できる。この文書を表現するために用いられる特徴を**素性 (feature)** と呼ぶ。テキストがうまく分類されるかは、この素性の選び方に深く依存しており、テキストの分類の手がかりになるようなより良い素性を設定して文書を表現する手法が様々に提案されている。

最も基本的な文書の表現としてよく用いられるのは、文書中に個々の語が出現しているかを素性とする方法である。文書は、素性とする語の集合と同じ数の要素のベクトルで表わされ、各要素がそれぞれひとつの語に対応する。各要素の値は、その語がその文書に出現しているかを表わし、語の有無を表わすバイナリ値や、文書中の語の頻度などが用いられる。この表現形式は”bag of words”と呼ばれる。単純ながら高い分類精度を発揮し、今なおテキスト分類における代表的な素性として用いられている。

個々の語を独立に扱う”bag of words” は、文書を極度に単純化しているために、文書が本来持っていた情報を大きく減じていると考えられる。文書の本来持っていた情報を表現し、よりテキスト分類の精度を高めるために、複数の語の情報を組み合わせ、新しい素性を生成する方法が様々に研究されてきている。

複数の語を組み合わせ素性にする手法としては、語の出現の間の統計的類似性を手がかりとするものがよく知られている。例えば、Latent Semantic Indexing[1] は語同士の相関関係を元に素性を生成する。独立成分分析は互いに独立になるように語を組み合わせ素性を生成する。

また、複数のタスクに同時に共通して役に立つ素性を生成する手法が知られて

いる。Ando と Zhang の研究 [2] や Amit らによる研究 [3] などがある。

1.2 従来手法の問題点

これらの従来手法は、素性を作る基準としてはいくつかの問題点を抱えている。語の出現の統計的な類似性を基準とする手法の第一の問題点は、複合する語の選び方である。互いに強く相関する語の組み合わせだけが、カテゴリの予測に役立つ素性であるとは限らない。第二に、語の組み合わせ方が予測の精度を上げるために適切なものとは限らない。複数のタスクに共通して役立つ構造を素性とする手法は、普遍的に役立つ素性を高く評価するために、実際の目的となるタスクにだけ役立つような素性が無視されてしまう。

1.3 語の出現の予測

本研究では、このような問題点を克服するために、語の出現の予測に基づいて素性を作る新しい手法を提案する。トピックは通常そのトピックに特に関係して出現する様々な語が存在する。もし、そのような語が出現するかを予測できるかどうかを予測できる素性が存在すれば、そのような素性はそのトピックを予測するのにも役に立つ素性であると考えられる。この語の出現の有無を予測できるかどうかという基準を用いれば、文書の識別力を持ち、また特定のタスクにのみ役立つ普遍的ではない複合素性も作り出すことができる。

提案手法では、ある語の出現の有無を他の語の出現を手がかりに予測する問題を作り、分類器で学習する。文書中の各語の出現の有無を正例と負例として、語の出現を予測する二値の分類問題を作成する。これを分類器を用いて学習することにより、ある語の出現を予測するような、それ以外の語から生成された出力値を得る。この出力の値を新たな素性とみなし、カテゴリの分類の素性として用いる。この素性は、ある語が出現すべき状況であるかを表わす素性となり、語の出現の実際の観測とは異なる情報を表現する。

語の出現予測に基づく複合素性を用いてテキスト分類実験を行った。実験セットとして、テキスト分類における最も代表的な実験セットである新聞記事コーパス Reuters-21578 と 20-newsgroups を用いた。特にラベルつき用例が少ない場合について実験し、提案手法の半教師つき学習としての効果を検討した。

1.4 論文の構成

第 2 章で、複数の語から素性を生成するための従来研究について概観する。第 3 章では、素性の生成の基準として、語の出現の予測を用いることを提案する。第 4 章で、語の出現予測を用いて素性を生成する具体的な手順について説明する。第 5

章で、テキスト分類コーパスについて実験を行い、提案手法による素性の有効性を調べる。第 6 章で、本研究で得られた成果をまとめる。

第2章　関連研究

本章では、テキスト分類の精度を高めるために、複数の語を組み合わせ新しい素性を用いる先行研究について概観する。

2.1　フレーズ素性

”bag of words”で用いられる個々の語の出現よりも高度な素性として初期に考えられたのはフレーズである [4, 5, 6]。ここでフレーズとは、文書中に連続して出現する複数の語のかたまりを指す。フレーズには、文を言語的に解析して得られた文法的なものや、あるいは統計的に意味のある語の連続として抽出された統計的なものの二種類がある。

フレーズ素性は連続した複数の語を扱うことで、個々の語を独立に扱うよりも、より文書で表現された意味を適切に反映し、テキスト分類に寄与することが期待された。しかし、実際の実験では、フレーズ素性は単純な”bag of words”の場合の精度を大して改善できないことが報告されている [7]。この原因として、フレーズ素性は統計的に用例が取りづらいことが挙げられている。フレーズは、語の連続を単位とすることで、個々の語よりもさらに頻度が低くなり、十分な用例数が確保できない。また同じ内容の文書であっても、全く同じフレーズが出現するとは限らず、結果的に個々の語だけを扱う場合よりも統計的に質が低く、素性として役に立たないものとなる。

2.2　Term Clustering

テキスト分類を行う上で問題になるのは、類義語や多義語の問題である。類義語の問題を解決する方法のひとつとして、Term Clustering という手法が用いられている [7, 8, 9, 10]。互いに類似度の高い語同士をクラスタリングすることで、類義語をひとつの素性としてまとめることで、分類精度を上げることを試みている。出来るクラスターが理想的な類義語に対応していないことなどの問題があり、なかなか精度上昇につながらないことが指摘された [7]。近年では、Information Bottleneck (IB) clustering [11] を用いた手法 [12, 13] や、クラスタリングによる相互情報量の減少を用いた手法 [14, 15]、ナイーブベイズの精度改善に用いた研究 [16] などがある。

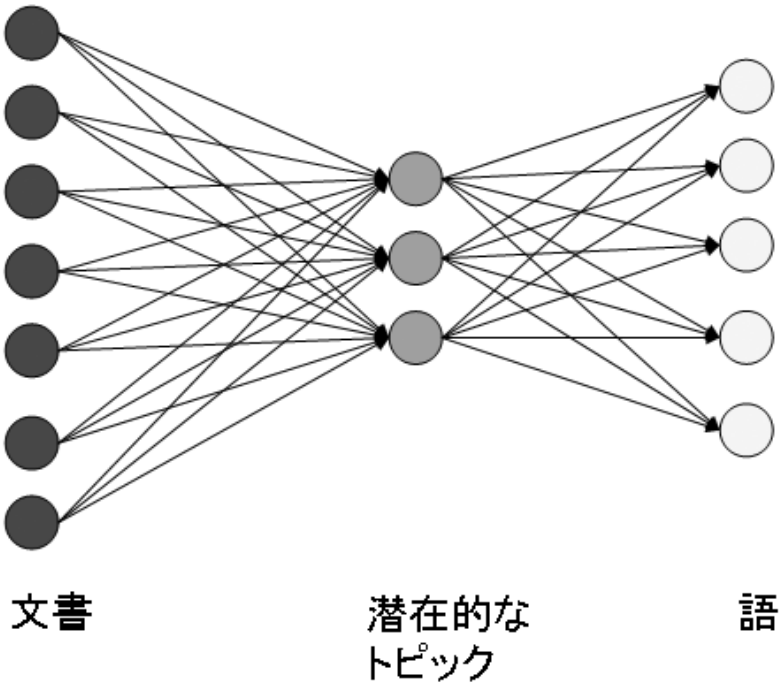

図 2.1: LSI と pLSI の概念図

2.3 Latent Semantic Indexing

Latent Semantic Indexing (LSI)[1] は、情報検索の分野で発達した手法であり、類義語や多義語の問題を解決するため、語と文書のあいだに存在する潜在的トピックを統計的に抽出する。

文書とそこに登場する語の対応を考えたとき、対応関係を表わす行列 X を考えることができる。語の総数を m、文書の総数を n としたとき、この行列の大きさ $m * n$ であり、各要素はその文書に登場する語の頻度となる。

$$X = \begin{pmatrix} w_{11} & w_{12} & w_{13} & \cdots & w_{1n} \\ w_{21} & w_{22} & & & \vdots \\ \vdots & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ w_{m1} & \cdots & \cdots & \cdots & w_{mn} \end{pmatrix} \tag{2.1}$$

この行列に対し次の特異値分解 (SVD) を行う。

$$X = U\Sigma V \tag{2.2}$$

X の階数を r としたとき、Σ は X の特異値を対角要素に持つ対角行列である。U は特異値に対応する左特異ベクトルからなる $m * r$ の行列、V は特異値に対応す

る左特異ベクトルからなる $r * n$ の行列である。このとき、Σ の対角要素である特異値を、値の大きいほうから $k(< r)$ 個選択することで、階数 k の対角行列 Σ_k を考えることができる。これに対応する特異ベクトルの行列 U_k、V_k を用いて、次の行列 X の近似 X_k を得る。

$$X_k = U_k \Sigma_k V_k \tag{2.3}$$

この X_k は語と文書の対応関係を、より小さな次元で説明していることになる。この Σ_k の要素を LSI では潜在的なトピックと捉える。このとき出現のしかたが似ている語、同じような語を含む文書は同じトピックへと写像されるため、語や文書の類似度を、元の X よりも適切に捉えることができることが期待される。このトピックを語からなる複合素性と見なせば、テキスト分類の素性としても用いることができる。すなわち、上で得られた左特異ベクトルの行列 U_k を用いて、語のベクトルをトピックのベクトルへ変換する。このトピックのベクトルは複数の語の線形結合からなる素性のベクトルであり、これをテキスト分類の素性として用いる。

ただしテキスト分類に用いられた場合、LSI で生成される素性にいくつかの問題点が指摘されている [17]。類義語をまとめて作った素性よりも、元のひとつの語のほうがカテゴリを識別するためには有効なこともあることや、あるいは、全体に対する用例数の少ないカテゴリに関する語がノイズとして無視されるために、用例数の少ないカテゴリほど分類精度が悪化することなどである。

この問題を解決するために、文書を各カテゴリに応じて分け、カテゴリごとに LSI を施し素性を生成する手法が提案されている。初めは各カテゴリの正例だけを集めて LSI を行う方法が提案されたが [18]、正例だけでは識別的な情報が得られないという欠点があった。近年は、正例だけでなく、正例とよく似た負例を混ぜて LSI を施すことで、カテゴリ特有のまた識別力のある複合素性を生成する方法が研究されている [17]。

2.4 Probabilistic Latent Semantic Indexing

pLSI[19, 20] は確率的な解釈に基づく LSI の発展形である。ある文書 d において語 w が出現しているときを考える。このとき、トピックを表わす潜在クラス $z \in Z = z_1, ..., z_K$ を設定し、文書と語がこのトピックに関して次のように生成するとモデル化する。

- 確率 $P(d)$ で文書 d を選択
- 確率 $P(z|d)$ で潜在クラス z を選択
- 確率 $P(w|z)$ で語 w を生成

これを結合確率で書き以下の式を得る。

$$P(d_i, w_j) = \sum_{k=1}^{K} P(z_k)P(d_i|z_k)P(w_j|z_k) \tag{2.4}$$

$$= P(d_i) \sum_{k=1}^{K} P(z_k|d_i)P(w_j|z_k) \tag{2.5}$$

このとき、文書と語はトピックに関して互いに条件つき独立であるという仮定がなされている。

トピックに関する語と文書の生起確率が pLSI のモデルにおけるパラメータである。このパラメータを求めるために、最尤推定を行う。すなわち、以下の式で表わされる log-likelyhood 関数を最大化することを試みる。

$$L = \sum_{d \in D} \sum_{w \in W} n(d, w) log P(d, w) \tag{2.6}$$

ただし、$n(d, w)$ は文書における語の頻度を表わす。

最尤推定を用いてこのパラメータを最適化するために EM アルゴリズム [21] が用いられる。E ステップにおいて、トピックの事後確率が計算される。

$$P(z|d, w) = \frac{P(z)P(d|z)P(w|z)}{\sum_{z'} P(z')P(d|z')P(w|z')} = \frac{P(z)P(d|z)P(w|z)}{P(d, w)} \tag{2.7}$$

M ステップにおいては次のように再計算される。

$$P(w|z) = \frac{\sum_d n(d, w)P(z|d, w)}{\sum_{d,w'} n(d, w')P(z|d, w')} \tag{2.8}$$

$$P(d|z) = \frac{\sum_w n(d, w)P(z|d, w)}{\sum_{d',w} n(d', w)P(z|d', w)} \tag{2.9}$$

$$P(Z) = \frac{1}{\sum_{d,w} n(d, w)} \sum_{d,w} n(d, w) \tag{2.10}$$

pLSI は LSI と同様にテキスト分類への応用が研究されている。Cai ら [22] は pLSI とブースティングを組み合わせる手法を提案した。ブースティングとは分類精度のあまり良くない複数の分類器を結合させることにより高い分類精度を達成する手法であり、テキスト分類の分野でも高い成果を挙げている [23]。Cai らは pLSI のパラメータ $P(z_k|d_i)$ を文書のセマンティックなレベルでの表現と捉え、このパラメータのそれぞれの値に基づく分類と、個々の語の出現に基づく分類を弱い仮説と捉え、代表的なブースティング手法である AdaBoost[24] を用いてブースティングを行うことで、テキスト分類の精度を改善している。

2.5 Independent Component Analysis

Independent Component Analysis (ICA、独立成分解析) は、信号処理分野で生まれ、後にテキスト処理に応用された手法である。複数の互いに独立な信号源から信号が送られていて、これらが入り混じった観測信号が複数のマイクで同時に受信されたものとする。この観測信号から、元の互いに独立な信号を復元しようとするのが独立性文化遺跡の目的である。テキスト処理においては、個々の語がマイクに当たり、ひとつの文書で見られた語の集まりが、同時に観測された信号と見なされる。このとき、個々の語の背後に互いに独立な潜在的な因子が隠れていると仮定して、その因子が個々の語という結果として観測されると仮定する。この因子を独立成分解析で導くことで、文書を意味的な因子で表現することができる。

近年では、カテゴリ割り当ての尺度を改善する研究 [25] や、元の語素性と ICA による素性を同時に用いる研究 [26] などがある。

2.6 複数のタスクの学習による共有構造の生成

以上に上げた手法は、主に語のあいだの出現の類似性を手がかりとして語を複合させる手法であった。このような手法の問題点として、出現の仕方が類似する語をひとつの素性にまとめることが、必ずしもタスクの精度上昇につながることが保証されていないことがある。これに対し、タスクの精度が上昇することを基準にして新たな複合素性を生成する研究も行われている。その中でひとつの潮流をなすのが、複数のタスクの学習において、各タスクに共有される構造を見出し素性化する研究であり、テキスト分類に限らず、画像認識などの分野などにも用いられる汎用的な機械学習手法として提案されている。

共有構造を見つけ出す手法の根底にあるアイディアは、あるタスクに役に立つ素性は、他の類似したタスクにも役に立つはずであるという考え方である。例えば、ある人間の顔画像を認識したいときに役に立つ、男か女か、あるいは黒い髪かそうでないか、というような個々の判断要素は、他の人間の顔画像を認識する際にも役に立つはずである。そこで、複数の人間の顔画像の認識に役立つような素性を生成すると、他の未知の問題に対しても役立つ素性が生成されていることが期待できる。これはテキスト分類の文脈で言えば、あるトピックかどうかだけを学習するのではなく、複数のトピックの分類に同時に役に立つような複合素性を生成するということになる。共有構造を見つけ出す手法は、多数の観測を説明するような少数の潜在的因子を見出そうとする点で、これまでに述べた LSI などと同様の意味があり、特異値分解など同じような手法が用いられる。

共有構造を見つけ出す最新の研究の一例として Amit ら [3] による研究を説明する。

2.6.1 Amit らによる複数タスクの共有構造の生成手法

マルチクラス分類の目的を、用例 X からラベル $Y = 1, \ldots, k$ への写像 $H : X \to Y$ を学習することとする。さらに線形の分類器を仮定する。すなわち、各クラス $y \in Y$ について重みベクトル W_y をパラメータとして、写像が次の形で表わされることを仮定する。

$$H_w(x) = argmax_{y \in Y} W_y^t \cdot x \tag{2.11}$$

ここで Crammer ら [27] の方法に基づき目的関数を定義する。すなわち、empirical loss と正規化項のトレードオフを最小化することで重みを学習する。正規化項は字式で表わされる。

$$\sum_y ||W_y||^2 = ||W||_F^2 \tag{2.12}$$

ただしここで $||W||_F$ は行列 W のフロベニウスノルムを表わす。損失関数は正しいクラスと間違ったクラスを比較したときのヒンジ損失となる。

$$l(W; (x, y)) = \max_{y' \neq y} [1 + W_y^t \cdot x - W_y^t \cdot x]_+ \tag{2.13}$$

ただし、

$$[z]_+ = max(0, z) \tag{2.14}$$

とする。以上より、トレードオフのパラメータを C として、重み W を求めるための学習ルールは次の式で表わされることになる。

$$\min_W \frac{1}{2} ||W||_F^2 + C \sum_{i=1}^{m} l(W; (x_i, y_i)) \tag{2.15}$$

この式は 2 値の分類を仮定すれば SVM での学習と一致する。

$$W_1 = -W_2 = \frac{1}{2} w_{svm} \tag{2.16}$$

ここまではマルチクラス分類の設定を行っただけであるが、ここで目的である共有構造を導入する。共有構造を見つける手法には非線形な共有構造を仮定するものが多いが、非線形な共有構造を最適化するには非凸な問題を解かなければならないため、ここでは線形な共有構造を仮定する。すなわち、共有される構造 r は入力ベクトル x の線形関数 $F_r^t x$ で表わされるものとする。そして、最終的なラベル y は、この共有構造のさらなる線形変換、$G_y^t(F^t x)$ で与えられるものとする。つまり、LSI や pLSI が語と文書のあいだに潜在トピックを仮定したのと同じように、入力と出力のあいだに共有構造の介在を仮定するのである。このとき目的関数は次のようになる。

$$H_G, F(x) = argmax_{y \in Y} G_y^t \cdot (F^t x) \tag{2.17}$$
$$= argmax_{y \in Y} (FG)_y^t \cdot x \tag{2.18}$$

何も制約をつけなければ、線形の変換をつなげただけであり、この目的関数を最大化したときの結果に変化はない。しかし、ここで F と G のノルムの大きさに制限を加えることで、共通構造を抽出する。

$$\min_{F,G} \frac{1}{2}||F||_F^2 + \frac{1}{2}||G||_F^2 + C \sum_{i=1}^{m} l(FG; (x_i, y_i)) \tag{2.19}$$

このように、共通構造を表現する線形変換のノルムの大きさに制限を加えることで、線形変換の各要素はできるだけ小さな値を取るようになる。このことは、LSI などで特異値分解を行い少数の潜在トピックで語と文書の対応関係を説明しようとしたのと同様であり、複数のタスクを学習する際にできるだけ少数の要素を用いて分類の精度を上げることで、複数のタスクに共通する複合素性を取り出しているのである。式 2.19 は半正定値計画問題に帰着することで解くことができる。

2.6.2 AndoとZhangによる複数のタスクを用いた半教師つき学習

複数のタスクを用いて共有構造となる複合素性を導く方法の中でも特に注目すべき研究は Ando と Zhang[2] のものである。共有構造の見つけ方は特異値分解を用いており先行研究と類似した手法である。注目すべき点は、この複数のタスクを用いて共有構造を用いる手法を、半教師つき学習に応用していることである。半教師つき学習とは、正解ラベルつきの学習データから学習するいわゆる教師つき学習と、ラベルなしのデータから学習する教師なし学習の中間に当たるものである。すなわち、少数の正解ラベルつきの学習データと、多数のラベルなし学習データを併用して、少数の正解ラベルだけを用いて学習するよりも高い精度を得ようとするものである。

まず、Ando と Zhan による複数タスクの共有構造の学習方法について説明し、後にそれを半教師つき学習に応用する方法について述べる。

複数タスクの共有構造の学習方法

共有構造が入力ベクトルの線形変換によって得られる低次元の部分空間であるという仮定を置く。すなわち、入力ベクトルを $x \in R^p$ としたとき、共有構造は $h \times p (h < p)$ 次元の行列 Θ による線形変換で表わされるものとする。

$$\Theta x \tag{2.20}$$

この共有構造素性と元の入力素性を併用して、次の分類器を作成するものとする。

$$f_\Theta(w, v; x) = w^T x + v^T \Theta x \tag{2.21}$$

このとき、w は入力素性にかかる係数、v は共有構造素性にかかる係数である。この分類器の分類エラーが最小になるように、Θ を定めることが、共有構造素性の学習に相当する。

複数のタスクが与えられたときの、パラメータ θ に関する分類エラーの最小化は次の式で書くことができる。

$$\hat{\theta} = arg \min_{\theta \in \Gamma} \left[r(\theta) + \sum_{l=1}^{m} \mathrm{O}_l(S_l, T_l, \theta) \right] \tag{2.22}$$

ただし、$r(\theta)$ は θ の値に関する正則化パラメータであり、$\mathrm{O}_l(S_l, T_l, \theta)$ は、学習用例 S_l とその他の追加情報 T_l と θ から導かれる学習結果のパフォーマンスである。このパフォーマンスを empirical risk minimization (ERM) を用いて測ることにすれば、目的とする最適化は次の式で表現できる。

$$[\{\hat{\boldsymbol{w}}_l, \hat{\boldsymbol{v}}_l\}, \hat{\Theta}] = arg \min_{\{\boldsymbol{w}_l, \boldsymbol{v}_l\}, \Theta} \left[r(\Theta) + \sum_{l=1}^{m} \left(g(\boldsymbol{w}_l, \boldsymbol{v}_l) + \frac{1}{n_l} \sum_{i=1}^{n_l} L(f_\Theta(\boldsymbol{w}_l, \boldsymbol{v}_l; \boldsymbol{X}_i^l)), Y_i^L) \right) \right] \tag{2.23}$$

$g(\boldsymbol{w}, \boldsymbol{v})$ は重みベクトル $(\boldsymbol{w}, \boldsymbol{v})$ に関する近似的な正則化条件を、$r(\Theta)$ はパラメータ Θ に関する近似的な正則化条件を表わす。$f_\Theta(\boldsymbol{w}_l, \boldsymbol{v}_l; \boldsymbol{X}_i^l)$ は $\boldsymbol{w}_l, \boldsymbol{v}_l$ と入力データ $\boldsymbol{X}_i^l$ から導かれる予測器であり、$L(f_\Theta(\boldsymbol{w}_l, \boldsymbol{v}_l; \boldsymbol{X}_i^l)), Y_i^L)$ はその予測器とラベルつきデータ Y_i^L との不一致を測る損失関数である。

2 乗正則化を用いることとし式 2.21 を代入すれば、この式はさらに次のように書ける。

$$[\{\hat{\boldsymbol{w}}_l, \hat{\boldsymbol{v}}_l\}, \hat{\Theta}] = arg \min_{\{\boldsymbol{w}_l, \boldsymbol{v}_l\}, \Theta} \sum_{l=1}^{m} \left(\frac{1}{n_l} \sum_{i=1}^{n_l} L((\boldsymbol{w}_l, \boldsymbol{v}_l)^T \boldsymbol{X}_i^l, Y_i^L) + \lambda_l ||\boldsymbol{w}_l||_2^2 \right) \tag{2.24}$$
$$s.t. \Theta\Theta^T = \boldsymbol{I}_{h \times h}$$

λ_l は定数である。さらに個々の問題 l に関して以下の補助変数 $\boldsymbol{u}_l$ を導入する。

$$\boldsymbol{u}_l = \boldsymbol{w}_l + \Theta^T \boldsymbol{v}_l \tag{2.25}$$

これにより $\boldsymbol{w}$ を式から取り除くことができる。

$$[\{\hat{\boldsymbol{u}}_l, \hat{\boldsymbol{v}}_l\}, \hat{\Theta}] = arg \min_{\{\boldsymbol{u}_l, \boldsymbol{v}_l\}, \Theta} \sum_{l=1}^{m} \left(\frac{1}{n_l} \sum_{i=1}^{n_l} L(\boldsymbol{u}_l \boldsymbol{X}_i^l, Y_i^L) + \lambda_l ||\boldsymbol{u}_l - \Theta^T \boldsymbol{v}_l||_2^2 \right) \tag{2.26}$$
$$s.t. \quad \Theta\Theta^T = \boldsymbol{I}_{h \times h}$$

これが最終的な最適化のための式である。
最適化の手順は次のように行う。

- $(\Theta, \boldsymbol{v})$ を固定して、$\boldsymbol{u}$ に関して (2.26) を最適化する
- $\boldsymbol{u}$ を固定して、$(\Theta, \boldsymbol{v})$ に関して (2.26) を最適化する
- 収束するまで繰り返す

このうち第 1 ステップは凸な問題なので既知の様々な最適化手法で解ける。以下では第 2 ステップについて詳細に説明する。$\boldsymbol{u}$ を固定するために、式 (2.26) は次のように書ける。

$$[\{\hat{\boldsymbol{v}}_l\}, \hat{\Theta}] = arg \min_{\{\boldsymbol{v}_l\}, \Theta} \sum_{l=1}^{m} \lambda_l ||\boldsymbol{u}_l - \Theta^T \boldsymbol{v}_l||_2^2 \quad s.t. \quad \Theta\Theta^T = \boldsymbol{I}_{h \times h} \tag{2.27}$$

線型代数によれば、Θ が固定されたとき

$$\min_{\boldsymbol{v}_l} ||\hat{\boldsymbol{u}}_l - \Theta^T \hat{\boldsymbol{v}}_l||_2^2 = ||\hat{\boldsymbol{u}}_l - \Theta^T \hat{\boldsymbol{u}}_l||_2^2 - ||\Theta^T \hat{\boldsymbol{u}}_l||_2^2 \tag{2.28}$$

が成り立ち、最適な値のとき $\hat{\boldsymbol{v}}_l = \Theta \hat{\boldsymbol{u}}_l$ となる。$\boldsymbol{v}_l$ を排除してこの式を適用することにより、式 (2.27) を次のように書き直すことができる。

$$\hat{\Theta} = arg \max_{\Theta} \sum_{l=1}^{m} \lambda_l ||\Theta \hat{\boldsymbol{u}}_l||_2^2 \quad s.t. \quad \Theta\Theta^T = \boldsymbol{I}_{h \times h} \tag{2.29}$$

ここで $p \times m$ の大きさの行列 U を次のように定義する。

$$U = [\sqrt{\lambda_1} \hat{\boldsymbol{u}}_1, \ldots, \sqrt{\lambda_m} \hat{\boldsymbol{u}}_m] \tag{2.30}$$

これを代入して

$$\hat{\Theta} = arg \max_{\Theta} tr(\Theta U U^T \Theta^T) \quad s.t. \quad \Theta\Theta^T = \boldsymbol{I}_{h \times h} \tag{2.31}$$

を得る。ただし $tr(A)$ は行列のトレース (trace) である。この式の形を見れば分かるように、この式の解は U の特異値分解で得ることができる。すなわち、U の特異値分解を $U = V_1 D V_2^T$ の形で表わすとき (ただし、D の対角要素は降順に並べ

入力: 学習データ $(\boldsymbol{X}_i^L, Y_i^l)(l = 1, \ldots, m)$
パラメータ: h と $\lambda_1, \ldots, \lambda_m$
出力: $h \times p$ 次元の行列 Θ
初期化: $\boldsymbol{u}_l = 0(l = 1 \ldots m)$、および恣意的な Θ
iterate
  for $l = 1$ to m **do**
    Θ を固定し $\boldsymbol{v}_l = \Theta \boldsymbol{u}_l$ として、$\hat{\boldsymbol{w}}_l$ について解く:
      $\hat{\boldsymbol{w}}_l = arg\min_{\boldsymbol{w}_l} \left[\frac{1}{n_l} \sum_{i=1}^{n_l} L(\boldsymbol{w}_l^T \boldsymbol{X}_1^l + (\boldsymbol{v}_l^T \Theta) \boldsymbol{X}_1^l, Y_i^l) + \lambda_l |\boldsymbol{w}_l|||_2^2 \right]$
    $\boldsymbol{u}_l = \hat{\boldsymbol{w}}_l + \Theta^T \boldsymbol{v}_l$
  endfor
  U の特異値分解を計算
    $U = V_1 D V_2^T$
  V_1^T の最初の h 行を Θ の行にする
until converge

図 2.2: Ando と Zhang による複数タスクの共通構造の最適化アルゴリズム

られているものとする)、Θ の行は V_1^T の最初の h 行となる。これは U の特異値から大きい順に h 個を取った場合の対応する左特異ベクトルである。

以上から、マルチタスクの共通構造を抽出するアルゴリズムは表2.2のように書ける。

厳密に言えば収束するまで反復することができるが、実際は最初の反復だけで十分である。Θ が多少厳密でなくともタスクの正確さに影響は少ないからである。

この手法は特異値分解を用いているため、主成分分析（LSI）と似た手法に見える。しかし、本質的に異なるのは、主成分分析はデータに対して次元削減を行っているのに対し、この手法は予測器の予測結果に対して次元削減を行っている点である。このため、主成分分析と異なり強力な識別性能を持っている。

半教師つき学習における応用

本項では、ここまで説明したマルチタスクにおける共通構造の生成手法を半教師つき学習に応用する方法について説明する。肝となるアイディアは、ラベルなしのデータから**補助問題**を複数作成してそれを同時に解くことで、マルチタスクの場合に与えられる複数のタスクの代わりにし、共通構造を作成するというものである。以下では区別のため、最終的な目標である元々のタスクを**目的問題**と呼ぶ。手順は次のようになる。

- 元のラベルなしデータに自動的に補助的なクラスラベルが貼って補助問題用のラベルありデータとし、得られた補助問題を複数同時に解くことで、最適

なパラメータ Θ を得る

- Θ を固定して $\boldsymbol{w}$ と $\boldsymbol{v}$ に関して最適化することで、**目的問題**を解く

補助問題としては、元々の目的問題と類似した問題を選ぶ。そのことで、補助問題を解くのに役立った共通構造が、最終的に目的問題を解くのにも役に立つことを目指す。

２つのステップを分けて学習するよりも、補助問題と目的問題を同時に合わせて解いて共通構造を得ることもできる。ただし、一般に半教師つき学習では、ラベルありデータよりラベルなしデータのほうが遥かに多いので、目的問題を合わせても効果はあまり関係がない。

補助問題の性質としては次の２つが求められる。

- 自動的なラベル付与: 補助問題のラベルは自動的に付与できなければならない
- 関連性: 補助問題は目的問題と関係していなければならない

補助問題の作成方法としては、おおまかに次の二つの道筋が考えられる。ひとつは、完全に教師なしの学習、もうひとつは部分的に教師つきの学習である。二つの方法について以下に詳しく説明する。

第一の方法は、データの副次的構造の有無をラベルとして予測することで、補助問題とするものである。入力データの素性表現が与えられたとする。このとき、この素性表現の一部をマスクし（隠し)、他のマスクされていない素性を手がかりとしてこの隠された素性を予測する。実装的には、マスクするとはその素性の値を０で置き換えることになる。

例えば語のタギングの問題なら、それぞれの語の位置ごとに、周囲の語を手がかりとして、現在の語を予測するという補助問題を作成することができる。

この方法は上で挙げた補助問題に要求される二つの条件の前者を満たす。データの素性表現は与えられるはずであるから自動的にラベルを付与して補助問題を生成することは容易である。一方、後者はどの副次的構造の予測を補助問題として選ぶかが重要である。もし目的問題と全く無関係な素性を選んで予測対象として補助問題を作ってしまった場合、得られる共通構造も目的問題と無縁な無駄な素性が多く得られることになる。ただし ERM 学習器はあまり無駄な素性によって予測を阻害されることはない。その一方で、目的問題と密接に関係した素性を予測対象として選ぶことができれば、補助問題を解くことで得られる共通構造は、目的問題を解くために非常に有用なものになるだろう。

幸い、ラベルなしデータはラベルありデータに比べて非常に多くの量を得られることがしばしばであり、補助問題として用いる素性には教師つき学習の場合よりも広い選択肢が考えられる。例えば、教師つき学習においては、高次の素性はデータのスパースネスの問題があり用いるのに不便があることが多い。しかし、補

助問題は大量のラベルなしデータから生成することができるためにこのようなスパースネスの問題を回避することができ、登場することの少ない高次の素性でも補助問題のラベルとして用いることができる。このため、元々の目的問題に深く関連するような高度な素性も補助問題のラベルとして考えることができる。また、このような高次の素性を予測対象とした補助問題から得られる素性は、低次元の共通構造であるから、教師つき学習である目的問題に適用する際にはデータスパースネスの問題を避けることができる。これは、そのような低次元の共通構造においては、頻度が稀な素性が行列 Θ によって結合されるためであり、結果として低次元の共通構造の出現頻度はより大きくなるためである。

テキスト分類問題における補助問題を考える。テキスト分類と類似した補助問題をラベルなしの文書から作成するのに考えられる方法は、文書中で代表的な語を文書のラベル代わりに使う方法である。この文書中で代表的な語としては、文書中で最も頻度の高い内容語を用いる。内容語とは、be 動詞や冠詞などのどの文書にも普遍的に使われるような機能語以外の語のことである。

実際の補助タスクは次のようになる。内容語を W_1 と W_2 の 2 つの集合に分割するものとする。文書 x が与えられたとき、学習器は W_2 に含まれる語だけを用いて、W_1 に含まれる語がその文書で最も頻度が高い語になるか予測する。すなわち補助タスクは $|W_1|$ 個の 2 値の予測問題になる。例えば、

$W_1 =$ ”stadium”, ”scientist”,”stock”,

$W_2 =$ ”baseball”,”basketball”, ”physics”, ”market”,

と分割されたとする。このとき W_1 に含まれる語を全て観測されていないものとみなし、語”stadium”が他の”scientist”や”stock”よりも多く出現しているか否かを、W_2 に含まれる”baseball”、”basketball”、”physics”、”market”を手がかりとして予測することになる。同様に”scientist”や”stock”が最も頻度に高い語かどうかを予測する問題が作成できるので、補助タスクの数は $|W_1|$ 個となる。

テキスト分類と類似した補助問題を作成する第二の手法は、”co-training”[28] から着想を得たものである。”co-training”はいくつかの自然言語処理分野で用いられたブートストラップ法 [29] や EM[30] に深く関係する手法である。高い precision を持つ分類器を用いて、ラベルなしのデータにラベルを付与してラベルありのデータを増強することで学習の精度を上昇させる手法である。ただしこのことでノイズが増えてしまう危険性もある。この”co-training”を参考にして、第二の手法では、他の素性を用いたときの目的問題の分類器の分類結果を補助問題として用いる。

2 つの異なる素性空間への写像 Φ_1 と Φ_2 を考える。まず片方の写像を用いて、ラベルつきデータから目的問題の分類を学習する。補助問題は、この分類結果を予測する学習器を、別の素性空間への写像 Φ_2 を使って学習するものとなる。この手法の良いところは、分類器の分類結果をあくまで補助問題のためだけに用いるというところである。”co-training”は、元々のラベルに不確実な分類結果によるラベルを混ぜることでノイズを発生してしまう問題があったが、この手法では元の目的問題を解く際にはそのようなラベルの混入は起こらない。目的問題を解くと

きにはあくまで元々の正しいラベルだけを用い、補助問題を使って共通構造を作成するときにのみ不確実な分類結果をラベルとして用いるのである。

手順は以下のようにまとめることができる。

1. 素性空間への写像 Φ_1 を用いて、目的問題のラベルつきデータ Z で分類器 T_1 を作成する
2. T_1 を使ってラベルなしデータにラベルを付与することで補助問題を作成する
3. 素性空間への写像 Φ_2 だけを用いて、パラメータ θ を学習する
4. 上で学習したパラメータ Θ と適当な素性空間への写像 Ψ を用いて、目的問題の分類器を学習する

この手法を用いた学習の例として、次の 3 つが挙げられている。

例１．T_1 の予測結果を予測する　最も単純な考えとしては、分類器 T_1 の予測結果をそのまま予測することが考えられる。元の目的問題が c 個の分類を扱うものであれば、この方法で c 個の 2 値分類問題が補助問題として得られることになる。テキスト分類タスクにおいては、以前の例と同様に、語の集合を W_1 と W_2 の二つに分割し、片方の語 W_1 だけを用いて文書のカテゴリを学習する。そしてその分類器でラベルなしデータのラベルを予測して補助問題の学習データを生成する。その上で、残り半分の語 W_2 だけを用いて、この付与されたラベルを学習する。つまり、このときは素性空間への写像 Φ_1 と Phi_2 とは、入力データの文書を構成する語のうち W_1 に含まれる語だけを素性にする、W_2 に含まれる語を素性とする、ことを表わす。

例２．分類器の付与したラベルの上位 k 個を予測する　他に考えられるのは、分類器 T_1 が最も高いラベルの度合いを与えた小数の k 個のラベルの結合を予測するというものである。この時ラベルの度合いと言っているのは、$confidencevalue$ などと呼ばれる、分類器がそれぞれのデータに付与する、あるラベルらしさの度合いである。確率的な原理に基づく分類器であれば、それぞれのデータについて判断されるそのラベルである確率がこの度合いとなる。もし c 個の分類問題であれば、$c!/(c-k)!$ 個の問題が生成できる。テキスト分類タスクにおいては、例えば元々SPORTS や ECONOMY などのカテゴリの分類を行う目的問題があったとする。このとき $k = 2$ ならば、補助問題のひとつとして、分類器 T_1 がある文書に対して SPORTS と ECONOMY というラベルをこの順番に高い度合いで付与しているかどうかを予測する問題が生成される。

例３．分類の度合いの範囲を予測する　また、ラベルの度合いの範囲を指定して、その範囲に収まっているかどうかを予測する問題も考えられる。例えば、SPORTS

と判断される度合いが 0.5 を超えているかどうか、などである。

以上、補助問題を作成するための方法として、データの副次的な構造を予測する教師なし学習の方法と、他の分類器による分類結果を予測する部分的に教師つきな方法の 2 種類を説明した。しかし、補助問題の作成方法はここで挙げた方法に限らない、この手法の強みのひとつは、いくらでもまだこの先に補助問題を生成して試すことができるという点である。

Ando と Zhang は以上に説明した手法を用いて、20-newsgroup コーパスと Reuters-RCV1 コーパス (”new Reuters”) に対してテキスト分類実験を行った。補助問題の作成方法としては、前述の、最も頻度の高い語をラベル代わりにする方法や、他の分類器の予測結果を上位 k カテゴリまで予測する方法などを用いた。

実験した結果、ラベルつき用例数がごく少数のとき、教師あり学習に比べて、提案する半教師つき学習手法の精度が大きく上回った。ただし、ラベルつき用例数が一定数に達すると精度は教師あり学習とあまり差がなくなった。

また、補助問題の生成方法同士で比較するとき、ラベルつき用例数が特に少ないときは分類器の予測したラベルの上位 k 個を予測する補助問題よりも、テキスト中で最も頻度の高い語を予測する補助問題を用いたときのほうが高い精度を示した。このことは、前者がラベルつき用例を用いて学習した分類器 T_1 の予測結果を用いて学習していることから理解される。ラベルつき用例数が少ないとき、T_1 の分類の信頼性もまた低いので、前者の補助問題で学習される共通構造もまた信頼性の低いものとなる。その一方、ラベルつき用例数が比較的大きくなるとこの関係は逆転し、前者の上位 k 個を用いる方法が、後者の最も頻度の高い語を予測する方法の精度を上回ることになる。これは、後者の方法は、登場する語を予測するだけの完全に教師なしの学習方法であり、ラベルつき用例数が増えても学習される共通構造に変化がないのに対し、前者の方法は、ラベルつき用例を用いて学習された分類器 T_1 の予測結果を補助問題の用例データとして用いるために、ラベルつき用例の量が増えれば増えるほど、分類器 T_1 の予測精度が上がり、そのため共通構造の学習の用例データもより良質なものとなり、結果として得られる共通構造もより有効なものが得られるようになるからである。そして、最も高い精度を達成したのは、この両方の補助問題を併用して共通構造を学習したときであった。

また co-training との比較も行っており、両方のコーパスで提案手法は co-training を上回る成果を得ている。

以上はテキスト分類における実験について述べたが、その他、Named Entity Chunking や手書き文字の判別についても実験を行い、高い精度を得ている。

2.7　おわりに

本章では、複数の語を用いて新たな素性を生成する従来手法について概観した。連続した語を句として素性とする古典的な手法のほか、語の出現の統計的類似性を用いた手法や、複数のタスクに同時に役に立つ素性を生成する手法などを紹介した。

第3章　語の出現の予測

本章では、素性生成の新しい基準として、語の出現の予測性という基準を提案する。

3.1　これまでの素性生成の基準

複数の素性を用いて複合素性を生成するためには、生成される複合素性の良さの評価を行う尺度が必要になる。前章で見た、語の複合素性の生成方法の様々な先行研究は、フレーズの素性など近年あまり用いられない手法を除くと、主に以下の二種類に大別される。

- 共起や相関など語の出現の統計的類似性を用いたラベルなしの手法
- 複数タスクの予測を用いたラベルありの手法

前者はTerm clusteringやLSIや独立成分分析などを含み、テキスト分類のラベルなどは使わずに、語のあいだの統計的傾向から複合素性を決定する手法である。その根本には、「出現の傾向が似通っている語同士を組み合わせると良い素性ができる」という仮説がある。実際にこの手法はテキスト分類分野においていくつかの大きな成果を示している。これは、主に類義語の問題の解決の視点から説明される。文書中には”car”や”automobile”など、ほとんど類似した意味を指していながら、語としては異なるものが使われている場合がある。これらの語は同じような文書、同じようなトピックのもとで使われるはずであり、何らかの統計的な偏りを示すと考えられる。そのような統計的な偏りを表わす指標としては、相互情報量やχ^2値などが考えられ、Term clusteringはこのような値に基づいて語のクラスタリングを行い複合素性を生成する。LSIや独立成分解析も手法的には異なりながら、出現が類似した語同士を組み合わせ同じ素性にし、出現が類似しない語同士は異なる素性にするという点で共通の発想に基づいている。

語の出現の統計的類似性に基づく手法の欠点は、出現が統計的に類似した語同士を複合させることが、必ずしもテキスト分類の精度を上げるわけではないということである。複合した新しい素性よりも、複合する前の個々の語のほうがテキスト分類においては有効であるという結果も、前章で触れたようにしばしば報告されている。これは、出現の傾向が類似しているからといって必ずしも同じ意味

の語であるわけではなく、それらをまとめてしまうことで、実はテキスト分類に有効だった個々の語の特徴が失われてしまうことがある。語の統計的類似性に基づく方法は、生成される複合素性が将来の分類において識別力を有するかどうかを、複合素性の有効性を測る指標として持っていない。これは、テキスト分類のラベルを用いずに、統計的類似性だけを用いて語を複合させる手法の限界だと言える。

複数のタスクを用いて共通する構造を複合素性として取り出す手法はこのような欠点を克服するものと位置づけることができる。すなわち、複合素性の有効性の指標として、「複数のタスクの分類性能を上げる」という尺度を導入することによって、識別力を持った複合素性を生成することができる。

しかしこの複数のタスクを用いる方法にもいくつか問題点は存在する。第一の問題は、複合素性を生成するために多数の類似したタスクを用意しなければならないことである。このとき、タスク同士が類似していなければ、共通の構造として有効な複合素性を取り出すことができない。

第二の、そしてもっと根本的な問題点はそもそも複数のタスクに役立つことが、複合素性の生成の判断基準として最も適当なものかという問題がある。生成する複合素性は、必ずしも多くのタスクに普遍的に役立つものである必要はない。極端に言えば、目標とするひとつのタスク、前章で用いた用語を使えば目的問題ひとつに対して役に立つ素性であれば良いのであって、その他の多くの補助問題に対して有効である必要は必ずしもない。複数のタスクでの精度を同時に上げることを基準にして複合素性を生成する手法は、ある特定の少数のタスクにのみ非常に役立つ素性ではなく、多くの普遍的なタスクに少しだけ役立つ素性が優先して生成される。複数のタスクでの精度を同時に上げることは、必ずしも目的問題に対しての効果を最適化するものではなく、複合素性の生成基準としては最適なものではない。

Ando と Zhang の手法は、語の出現の統計的類似性を用いる前者の手法と、複数タスクの共通構造を用いる後者の手法、双方の利点を取ったものだと言える。複数タスクに役立つ素性が良い素性であるという原理に基づき最適化を行っているために、分類タスクを行う上で重要な識別力を保持した複合素性を生成することができる。このことは後者の利点である。また、ラベルなしのデータを用いる点では前者と利点を共有する。すなわち、ラベルなしデータから補助問題を生成するというアイディアによって、類似したタスクを多数用意しなければならないという、後者の第一の問題点を克服することに成功している。Ando と Zhang の論文においては、論文中で触れられた補助問題の生成方法だけでなく、もっと有益な複合素性を生成するための補助問題が存在することが示唆されており、今後も補助タスクの数は増大させることができると考えられる。

しかし Ando や Zhang らの問題は根本的に複数タスクの共通構造を手に入れるという手法に依っているために、上述の第二の根本的な問題点、複数タスクを同時に解決できることが必ずしも良い素性の絶対の判断基準とは限らないという点

については解決できていないままである。例えば、最も頻度の高い語を予測するという補助問題では、語を W_1 と W_2 の 2 つのグループに分け、W_1 に含まれる語全てについて、それぞれ最も頻度の高い語になっているかどうかを判断する補助問題が $|W_1|$ 個生成される。しかし、この補助問題の中に、実際の目的問題と無関係な問題が含まれていればいるほど、生成される共通構造は目的問題と乖離した、目的問題の分類に寄与しない素性となる。

我々は、上述の二つの複合素性の生成基準、すなわち、出現の統計的類似性と複数タスクへの共通構造という二つの基準を退ける。それではどのような基準が、素性生成のために適切なものとなるだろうか。次項ではこの点について論じ、新たな素性生成の基準として語の出現の予測性を用いることを提案する。

3.2 語の出現の予測性

直接的にトピックのラベルの予測が行えるかという以外に、素性の良さを評価できるような基準はあるだろうか。我々はここで、素性の良さの基準として、文書中の語の出現を予測できるかどうかを用いることを提案する。

3.2.1 トピック予測の類似タスクとしての語の出現予測

その最も直感的な理由の説明は、トピックには通常そのトピックに特に関係して出現する様々な語が存在する、ということである。例えば、スポーツというトピックに関する文書であれば、その中には野球、サッカー、バレーボール、などのスポーツの種目名や、ボールやゴールといったスポーツに関する文脈で良く登場する言葉がよく出現しているだろう。そして、スポーツというトピックに関連する文書でなければ、このような語はあまり出現していないだろう。また、料理というトピックに関する文書であれば、その中には、カレーライスや炒飯などの料理名や、ドレッシングやみじん切りといった料理に関する文脈で良く登場する言葉がよく出現しているだろう。そして、料理というトピックに関連する文書でなければ、このような語はあまり出現していないだろう。このようなトピックと特に関係して出現する語の性質を利用して、このような語を素性として用いるのが、そもそもの”bag of words”な手法の根本的なアイディアであった。

しかし、これらの語の利用方法はラベルを識別するための素性とするだけに限らない。逆に、良い素性を導出するためのガイドとなる分類のラベルとしても用いることができる。より具体的に言えば、あるトピックに関する文書を識別できることの代替的なタスクとして、そのトピックに関する語を含む文書を識別できるかどうかというタスクを考えることができる。もし文書に野球という語が出現するかどうかを正確に識別できるような素性があれば、そのような素性は、スポーツというトピックの分類タスクが与えられたときにも、文書がスポーツカテゴリ

に分類できるかを識別できるような良い素性になるであろう。これは、野球という言葉が含まれる文書と、スポーツカテゴリに分類される文書が似通った集合になるからである。

同様のアイディアは、前章で紹介したAndoとZhangによる手法[2]でも用いられている。AndoとZhangは、トピックのラベルを予測することの補助タスクの一例として、文書中で最も頻度の高い語を予測することを提案している。これは、ある文書があるトピックに属するかどうかを判断することと、その文書中である語が最も頻度の高い語であるかどうかを判断することが類似しているという考えに基づいている。

3.2.2 不完全なラベルとしての語の出現

トピックに関連する語の出現の予測は、類似したタスクとして解釈できる。その一方、語の出現をトピックに関する文書への不完全なラベルと見なすこともできる。例えば、スポーツというトピックに属する文書には、非常にしばしば野球という語が出現しているだろう。しかしその一方、スポーツというトピックに属する文書には、必ず野球という語が出現しているというわけではない。また、スポーツというトピックの文書でなくても、野球という語が文脈上出現することもあるだろう。ということは、野球という語が出現しているということを文書に対するラベルと考えると、これは、スポーツというトピックに関する文書への不完全なラベルであると考えることもできる。不完全なラベルであるとは、正(positive)である文書に対して正のラベルが割り振られていないこともあるし、また負(negative)である文書に対して正のラベルが割り振られてしまっていることもあるということである。このような不完全なラベルを扱う研究としては、前章で少し触れた”co-training”[28]などが存在する。

この二つの解釈、語の出現の予測をトピック予測とは別の、しかし類似したタスクと捉える解釈と、語の出現の予測をトピック予測と同一の、しかしラベルが不完全なタスクと捉える解釈は、語の出現ラベルとトピックのラベルの関係をどう捉えるかによっている。語の出現ラベルをトピックのラベルとは異なるラベルと考えれば前者の解釈になり、トピックのラベルの不完全なものと捉えれば後者の解釈ができる。

3.3 語の出現の予測値

語の出現の予測ができるかどうかを素性の基準にするとしても、その基準を用いてどのように素性を生成するかはまた様々な手法が考えられる。

AndoとZhangの手法は、語の出現の予測をトピック分類と類似したタスクと解釈し、これらのタスクには共通して役に立つ素性があるはずだという前提を立て、

少数の素性が多くの語の出現予測に役立つように、その少数の素性の重みを最適化する。

しかし、この手法は前章で指摘したように複数タスクの共通構造を用いる手法の問題点をそのまま引き継いでいる。複数タスクに共通して役に立つ素性を抽出するために、必ずしも目的タスクを解決するために最適な素性が抽出されているとは言えない。例えば、上記の例で言えば、野球という語の出現を予測するタスクと、カレーライスという語の出現を予測するタスクを立て、両方のタスクに共通して役に立つ素性を抽出しようとするとき、そのような素性は、スポーツというトピックの分類にも、料理というトピックの分類にも特化していないため、最適な素性の重みとならない可能性がある。

ここで分類器を用いて、以下のような語の出現の予測問題を解くことを考えよう。学習の用例としては文書の集合を用いる。これらの文書に語の出現の有無をラベルを貼る。そして、これらの文書を表現する素性としては、その他の語の出現を表わす素性を用いる。この用例を用いて、ある語の出現を他の語の出現から予測する問題を、分類器を用いて解くことができる。このとき分類器の出力は、ある語の出現がどれだけもっともらしいかを示す値となる。以下ではこの分類器の出力の値を**語の出現の予測値**と呼ぶ。

例として線形の分類器を考える。線形の分類器は、入力ベクトル $\boldsymbol{x}$ が与えられたとき、$\boldsymbol{x}$ に重みベクトル $\boldsymbol{w}$ をかけた値を出力として返す。この値が、線形の分類器を用いた場合の、語の出現の予測値である。

$$\boldsymbol{w} \cdot \boldsymbol{x} \tag{3.1}$$

語の出現の予測値は「ある語が出現しそうな状況」であるかを示すものだと言える。

我々はこの語の出現の予測値を素性として用いることを提案する。この予測値は、素性として用いた他の語を入力とする関数の値だと言え、つまり複数の語を組み合わせた新たな素性値である。また、この予測値は、語の出現予測をした結果の値そのままであるから、上で述べたような、語の出現予測に役立つ素性だという条件を当然満たしている。

この素性はその文書に「語 が実際に出現している文書か」ではなく「語 が出現していそうな文書であるか」を表わす。カテゴリ分類をする上では、前者よりも後者のほうがよりカテゴリという概念に接近した重要な情報となることが期待できる。ある文書があるカテゴリに属するとき、必ず同じ語が出るとは限らないが、読者はそのカテゴリに属する語が出現することを予測しながら読み進めるだろうからである。

3.4 従来の問題点の解決

語の出現の予測値を素性に用いる手法は、これまでの手法が用いていた基準の持つ問題点を解決している。統計的な類似性を用いる手法には文書の識別力が考慮されていない問題があったが、本手法は、トピックの文書を識別する代わりに、トピックと深く関係する語を含む文書を識別することで、文書の識別力を持った素性を選ぶことができる。また特定のタスクにのみ役立つような普遍的ではない複合素性も作り出すことができる。特定のタスクにのみ役立つ素性は、その特定のタスクにのみ偏って出現するような語の出現を予測することで得ることができる。この点で、本手法は Ando と Zhang の手法よりも、特定のタスクに役立つ専門的な素性を生成できるという長所を持っていると考えられる。

3.5 語の出現予測の学習の評価基準

ここで語の予測の学習の評価基準について考える。SVM などの学習器は、予測ができるだけ実測と一致するように、つまりエラー率を最小化するように学習する。しかし、本手法では、必ずしも予測と実測がよく一致するような式を得たいわけではない。実際には語が出現していない文書であっても、語が出現しそうな同じカテゴリに属する文書であれば正になるような素性が、実際の語の観測値よりもカテゴリ予測に役立つと考えられる。そこで本手法では Recall(再現率) を優先して最大化するように語の予測を学習する。すなわち、語が実際に出現する文書では必ず正になり、 語は出ていないが類似した文書でも正になるような判別式を学習し素性とする。

第4章　語の出現の予測による複合素性の生成

前章で、我々は語の出現の予測を基準とした複合素性の有効性について述べた。そして、語の出現の予測を学習したときの分類器の出力を素性として用いる手法を提案した。本章では、そのような複合素性を生成する手法の具体的な手順について説明する。

4.1　定義

まず取り扱うテキスト分類問題について定義する。それぞれの文書が入力ベクトル $\boldsymbol{x} \in X$ で表わされ、またカテゴリを表わすラベル $y \in Y$ が与えられているとする。このとき入力とラベルの組み合わせによって学習用例 $(\boldsymbol{X}, Y)$ が定義される。未知の用例 $\boldsymbol{x}$ を正しいラベル y へ写像する分類器を学習するのがテキスト分類問題の目的である。またテキスト中に出現しうる語の集合 W が与えられているものとする。

入力ベクトル $\boldsymbol{x}$ はどのような形態のものでも構わないが、ひとつだけ条件がある。提案手法において、出現を予測される語の集合を $W_p \subset W$ とする。このとき、各語 $w \in W_p$ が文書に出現しているかどうかを $\boldsymbol{x}$ から判断できるということである。通常用いられる”bag of words”な文書表現はこの条件を満たしている。それ以外の文書表現、例えば n-gram などの素性を含んでいても構わないが、それぞれの語 $w \in W_p$ の有無を表わす関数は何らかの形で定義できなければならない。語 $w \in W_p$ の有無を表わす $\boldsymbol{x}$ の関数を次のように表わす。

$$f_w(\boldsymbol{x}) = \begin{cases} +1 & (\text{語 } w \text{ が } \boldsymbol{x} \text{ の表わす文書に出現している}) \\ -1 & (\text{それ以外}) \end{cases} \tag{4.1}$$

この関数は元の入力ベクトル $\boldsymbol{x}$ として何を選ぶかに依存するが、手法の本質には影響しない。

4.2　複合素性の生成アルゴリズム

複合素性の生成のアルゴリズムを表 4.2 に示す。

アルゴリズム 複合素性の生成

1: 基本となる素性の集合 F、出現を予測するための語の集合 $W_p \subset W$ を定める
2: **for** $w_p \in W_p$ **do**
3: 　w_p の出現を予測するための素性の集合 $W_f \subset W$ を設定する
4: 　元となる学習セットから、w_p の出現を学習するための学習セットを生成する
5: 　分類器で学習し、w_p の出現を表わす式を得る
6: 　この式を表わす素性を新たに加える
7: **end for**

まず初めに、複合素性を生成するための種になる語の集合 $W_p \subset W$ を定めなければならない。与えられた語について全ての出現を予測する、つまり $W_p = W$ としても良いが、二つの問題が考えられる。一つ目は、用例数の問題である。極端に頻度の少ない語の出現を予測するとき、語が出現している文書、すなわち学習において正例として得られる文書の数は非常に限られたものとなる。そのとき学習された分類器の分類精度は信頼性の低いものとなり、引いては出現の予測を表わす複合素性も分類の役に立たない不確かな素性になってしまう恐れがある。また、第二の問題として、語の数 $|W_p|$ に等しい数の補助問題が解かれるために、W_p が大きくなるほど計算量が大きくなることがある。以上の理由から、W_p を定める際には、あまりに頻度の少ない語は予測の対象から外すなどの処理が必要になるかもしれない。

こうして定めた W_p に含まれる各語 w_p について、それぞれその出現を予測することで、ひとつの複合素性を生成する。この複合素性を c_p とする。すなわち、W_p に含まれる語と同数の複合素性が生成されることになる。

次に、$w_p \in W_p$ の出現の予測から生成される複合素性を構成する要素となる素性の集合 $F_p \in F_x$ を定める。この素性集合 F_p は w_p の出現の予測を学習する際に用例を表現するための素性の集合であり、また w_p を種としてできる複合素性 c_p の構成要素でもある。この素性の集合は F_x 全体と同一でも良いし、相互情報量や χ^2 検定値など、なんらかの選択基準によって素性選択したものでも良い。ただし、語 w_p の出現を表わす素性は F からは除外する。あくまで、語 w_p の実際の出現の情報は用いず、他の語などの情報から w_p の出現を学習するのである。

w_p の出現を予測するための学習セットを生成する。元となる学習セットの全文書について、上で定義された語 w_p の有無を表わす関数 $f_{w_p}(\boldsymbol{x})$ の値 $\{+1, -1\}$ を、文書のラベルとする。各文書は、上で選ばれた F_p に含まれる素性を用いて表現する。これを w_p の出現予測のための学習セットとする。

この学習セットを用例として学習器に与える。学習器は w_p の出現の有無を予測するよう学習される。この学習器が出力する値が語 w_p の予測値であり、この予測値が新たな複合素性 c_p となる。この素性は語 w_p の出現の予測の正負を表現し、その値は語 w_p の出現の度合いを表わす。以上の手順を全ての語 $w_p \in W_p$ について

繰り返すことで、$|W_p|$ 個の複合素性を生成できる。

複合素性が生成できた後は、テキスト分類の学習を行う。全ての文書について、語の出現予測を行う分類器の出力値を計算することで、全ての文書に複合素性の値を付与する。その上で元の入力ベクトル $\boldsymbol{x}$ の素性と複合素性を合わせて文書の表現として分類器に入力して、テキスト分類を学習する。そして、未知の文書が与えられたときは、同じように複合素性の値を計算して付加し、分類器に入力することで、複合素性を用いた場合のテキスト分類を実行することができる。

第5章　実験

提案する複合素性の生成手法を用いて、テキスト分類の実験を行った。Reuters-21578 と 20-newsgroups という 2 つの代表的なテキスト分類コーパスについて実験した。

5.1　実験の設定

5.1.1　入力ベクトル

各文書は”bag of words”のベクトルとして表わされる。入力ベクトル $\boldsymbol{x}$ の各要素がそれぞれひとつの語に対応し、ベクトルの長さは語数 $|W|$ に一致する。各素性の値は tf*idf 値を用い、長さ 1 に正規化した。

5.1.2　評価基準

分類精度を評価するための基準としては、Precision と Recall から導かれる F 値を用いる。Precision(適合率) と Recall(再現率) は次のように定義される。

$$Recall = \frac{|\{\text{正に分類された文書}\} \cap \{\text{実際に正である文書}\}|}{|\{\text{実際に正である文書}\}|} \tag{5.1}$$

$$Precision = \frac{|\{\text{正に分類された文書}\} \cap \{\text{実際に正である文書}\}|}{|\{\text{正に分類された文書}\}|} \tag{5.2}$$

Recall は、実際に正である文書のうちどれだけの割合が正しく取り出せたか、Precision は、正に分類された文書のうちどれだけの割合が正しいものであったかを表わす。この両者はトレードオフの関係にあり、片方を上げようとすると片方が下がる関係にある。そこで、テキスト分類の評価においては次のように定義される F 値が評価基準として用いられることが多く、これをこの実験でも採用する。

$$F\text{値} = \frac{2 * Recall * Precision}{Recall + Precision} \tag{5.3}$$

5.1.3 学習器

学習器としては、語の出現予測とテキスト分類のどちらも Support Vector Machine (SVM) を用いるものとする。SVM は線形分類器であるので、その出力は素性の線形結合で表わされる。この式を計算した値が、我々が求める複合素性となる。ただし、ここでは素性は、線形結合の計算結果が正なら 1、負なら 0 を取る 2 値の素性とする。SVM は出力の値の符号が正であるか負であるかが重要であり、符号に対してその値の大小の差はそれほど重要な意味を持たないためである。また、値の離散化を行うことで、線形結合だけでは解けないタスクも解くことができるという利点がある。

Recall を評価基準とする SVM が Joachims ら [31] により提案されており、語の出現予測の学習にはこれを用いる。テキスト分類の学習には、エラー率を評価基準とする一般的なＳＶＭを用いた。

5.1.4 複合素性の生成

複合素性を実際に生成する際の設定を述べる。出現を予測する語の集合 W_p としては、最も頻度の高い 1000 語を用いる。これは、前章で述べたように、頻度の高い語ほど学習用例が多く、適切な複合素性が得られると期待できるからである。語の出現予測を行う際の、素性の選択の基準としては χ^2 検定値を用いる。W_p に含まれる各語 w_p について、χ^2 検定値に基づき、もっとも関連する語 25 個が素性として選ばれる。よって、得られる素性は 25 語の複合素性となる。あらゆる状況で使用されるような語の出現予測を Recall 基準で学習すると、どんな状況でも +1 になるような頻度の高すぎる複合素性が生成される。このような素性は分類に役に立たないので削除する事とする。予測素性の重みは全て 0.1 の固定値とした。また、予測素性を大量に加えることで全体の素性数が増えてしまうと、過学習の問題があり、正確に比較できない恐れがある。よって、予測素性を加えた後で、元の入力ベクトルと同じ素性数になるよう、chi 2 乗検定値を基準に素性選択することとする。

5.2 実験１：Reuters-21578 コーパス

初めにテキスト分類で用いられる最も代表的なコーパスである Reuters-21578 コーパスを用いて実験を行った。

5.2.1 Reuters-21578 コーパス

Reuters-21578 コーパスには 21578 個の記事と、135 のカテゴリが用意されている。実験では、典型的な実験設定である”ModApte” split を用いた。カテゴリが付

与された実験セットを、9603 個の学習記事と、3299 個のテスト記事に分割する。この学習記事に対しステミングとストップワード処理を行い、3 文書以上に登場する語を選び、8127 語を得た。これを基本となる入力 $\boldsymbol{x}$ を表わす素性とする。最も用例数の多いトピック 25 個を選び、予測するカテゴリとした。

ランダムに選んだ 4000 文を、複合素性を作成するための学習用例とする。

5.2.2 実験結果

生成された複合素性

提案手法で生成された複合素性の例をいくつか表 5.2.2 に示す。(他の実験は、25 語の複合素性を生成した場合の結果であるが、この項のみ、見易さを考慮して、15 語の場合の複合素性の例を示している。) 語"airlin(e)"の出現を予測すると、航空会社各社の社名の複合素性になっている。複合素性の中に含まれている"Carl Icahn"は航空会社の買収で有名な資本家である。"china"を予測すると、中国国内の地名や中国の人名など中国語固有名詞の複合素性となる。"dealer"を予測すると金融市場における専門用語が、"econom(ic)"を予測すると経済状況についての用語の複合素性が得られている。

得られた複合素性はそれぞれの語の出現する状況をよく表わしているものが得られている。

分類の結果

提案手法はラベルなしのデータを用いて新たな素性を生成する。よって、特にラベルつきのデータが少ない場合に有効に働くと考えられる。そこで、テキスト分類の学習に用いるラベルつき用例数を変化させて精度の上昇を調べた。また、特に正例数が少ないカテゴリについての精度を調べた。表 5.2.2 に結果を示す。これはマクロ平均の f 値である。用例数が少ないとき、また正例数が少ないカテゴリに対して精度が上昇している。

5.3 実験２：20-newsgroups コーパス

20-newsgroups コーパスについてもテキスト分類実験を行った。

5.3.1 20-newsgroups コーパス

20-newsgropus コーパスは、Reuters-21578 に次いでよく使われるテキスト分類用のデータセットである。20 のカテゴリからなる 19997 個の文書からなり、全ての

表 5.1: 提案手法で得られる複合素性の例

	airlin(e)		china		dealer	
b	-1.449		-0.02899		-0.1449	
	USAIR	7.659	chines(e)	8.68	interven(e)	3.749
	KLM	7.289	peke	5.623	euro-commerci(al)	3.332
	TWA	4.654	shanghai	3.749	quiet	2.707
	Icahn	3.124	yuan	3.332	euro-cp	1.874
	DOT	2.499	fujian	1.458	bundesbank	0.4641
	tran(s)	1.912	guangdong	1.447	sterl(ing)	0.1793
	flight	1.506	anhui	1.25	arrang(e)	0.09655
	air	1.258	shandong	0.8331	dollar	0.04776
	airway	0.8689	husbandri	0.8331	paper	0.02857
	plane	0.8172	zhejiang	0.8331	yen	0.0255
	Aviat	0.6751	yangtz	0.8331	interven(t)	0.02187
	pie	0.5975	hong	0.213	market	0.01477
	tex	0.5975	kong	0.1439	bank	0.00554
	piedmont	0.5543	daily	-0.08464	trade	-0.00309
	Carl	0.1449	provinc	-0.3899	currenc(y)	-0.04841

	econom(ic)		export	
b	-1.778		-0.7053	
	economy	3.978	coffe(e)	1.914
	monetary	1.644	shipment	1.202
	growth	1.323	quota	0.8673
	policy	0.7458	wheat	0.7366
	inflat(e)	0.6266	tonn	0.5605
	country	0.498	import	0.377
	domest(ic)	0.4638	country	0.3625
	grow	0.434	produc(e)	0.1594
	govern	0.2547	econom(ic)	0.159
	world	0.2437	offici	0.154
	foreign	0.2246	trade	0.1523
	told	0.1457	govern	0.08627
	minist(er)	0.07573	agricultur(e)	0.05011
	export	0.06388	world	0.001268
	trade	0.03989	lt	-0.1806

b は切片。() 内はステミングを行う前の語の元の形を示す。

表 5.2: テキスト分類の結果

カテゴリ	素性	用例数			
		2000	3000	5000	9603
上位 25 カテゴリ	個々の語の素性のみ	60.95	72.73	72.73	74.93
	複合素性を追加	61.73	74.54	73.91	75.82
上位 25 カテゴリの中の	個々の語の素性のみ	51.16	54.39	66.16	69.40
下位 15 カテゴリ	複合素性を追加	51.77	59.65	67.63	69.97

カテゴリにほぼ均等に文書が存在する。つまり、ひとつのカテゴリにつき約 1000 個の用例がある。

事前処理として全ての文書からヘッダーを削除した。ヘッダーには、subjects や送信者名などが含まれる。これらは一切実験には用いず、地の文だけを用いて分類の実験を行う。

コーパス中の全文書に対しステミングとストップワード処理を行い、20 文書以上に登場する語を選び、7810 語を得た。これを基本となる入力 $\boldsymbol{x}$ を表わす素性とする。

19997 文のうち、1000 文をテストセットに、2000 文を用例セットに、また 5000 文を複合素性の学習用に、それぞれランダムに振り分けた。

5.3.2 学習用例数の設定

提案手法はラベルなしデータから語の出現予測を学習することで有効な複合素性を得ることができるために、半教師つき学習としての側面を持っている。この面での評価を行うために学習用例数が少ない場合について実験した。学習用例数を、100, 200, 500, 1000 と変化させ、学習用例数の変化が少ない場合について提案手法による複合素性がどのように影響を与えるか調べた。

5.3.3 実験結果

実験結果を図 5.1 に示す。提案手法による複合素性は、用例数が少ない場合に特に大きく分類に寄与している。

このことは、提案手法による複合素性の半教師つき学習としての有用性を示している。

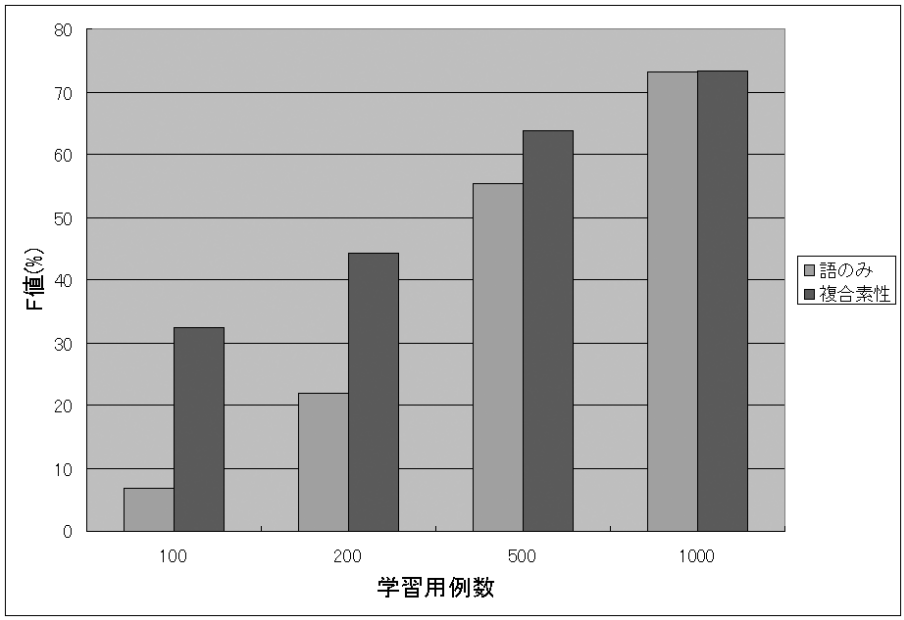

図 5.1: テキスト分類の結果：20 newsgroups

第6章　まとめ

本研究では、テキスト分類のための素性生成の従来研究を概観し、その問題点を克服するために、新たな素性生成の基準として語の出現の予測を用いることを提案した。また、語の出現の予測を用いた素性生成の手法として、語の出現の有無を予測するテキスト分類問題を分類器を用いて学習することで得られる分類器の出力の値を素性として用いる手法を提案した。特にこの手法はラベルなしのデータを用いて新しい素性を生成してその素性をラベルありデータを用いたテキスト分類に利用できることから、半教師つき学習の一種として考えることができる。この手法を用いて得られた素性はテキスト分類の精度を向上させることが確認された。特にラベルつきデータが少ないときに、単独の語のみの素性を用いる場合より大きく精度が向上し、提案手法の半教師つき学習手法としての有効性が示された。

謝辞

指導教員の石塚満教授には、修士の二年間に渡り多くの御指導御助言を頂きました。研究テーマが変わっていくことにもご容赦下さり、様々な関連研究についての知見をご教示下さいました。石塚教授の暖かな指導の御蔭でこうして研究を完成させることができました。深く感謝の意を申し上げます。

東京大学工学系研究科総合研究機構の松尾豊准教授には、研究のアイディアの段階から手法の検討や実験方法に至るまで丁寧で深い御指導を頂きました。本研究の理論的な構築や成果は松尾准教授のお力添えなしには有り得ないものでした。心より御礼申し上げます。

東京大学石塚研究室の皆様には、研究に関する議論や研究の環境設定などの面で大きなご助力を頂きました。ありがとうございます。

研究グループ「松尾ぐみ」では、論文の輪読を通じて研究分野への理解を深めることができ、またたびたび研究の発表をしては研究内容や発表方法へのご助言を頂きました。「松尾ぐみ」の皆様に感謝いたします。

参考文献

[1] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, Vol. 41, No. 6, pp. 391–407, 1990.

[2] Rie Kubota Ando and Tong Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *J. Mach. Learn. Res.*, Vol. 6, pp. 1817–1853, 2005.

[3] Yonatan Amit, Michael Fink, Nathan Srebro, and Shimon Ullman. Uncovering shared structures in multiclass classification. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pp. 17–24, New York, NY, USA, 2007. ACM.

[4] Norbert Fuhr and Chris Buckley. A probabilistic learning approach for document indexing. *Information Systems*, Vol. 9, No. 3, pp. 223–248, 1991.

[5] Hinrich Schutze, David A. Hull, and Jan O. Pedersen. A comparison of classifiers and document representations for the routing problem. In *Research and Development in Information Retrieval*, pp. 229–237, 1995.

[6] Konstadinos Tzeras and Stephan Hartmann. Automatic indexing based on Bayesian inference networks. In Robert Korfhage, Edie Rasmussen, and Peter Willett, editors, *Proceedings of SIGIR-93, 16th ACM International Conference on Research and Development in Information Retrieval*, pp. 22–34, Pittsburgh, US, 1993. ACM Press, New York, US.

[7] David D. Lewis. An evaluation of phrasal and clustered representations on a text categorization task. In *SIGIR '92: Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 37–50, New York, NY, USA, 1992. ACM.

[8] Y. H. Li and Anil K. Jain. Classification of text documents. *Comput. J.*, Vol. 41, No. 8, pp. 537–546, 1998.

[9] L. Douglas Baker and Andrew K. McCallum. Distributional clustering of words for text classification. In W. Bruce Croft, Alistair Moffat, Cornelis J. van Rijsbergen, Ross Wilkinson, and Justin Zobel, editors, *Proceedings of SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval*, pp. 96–103, Melbourne, AU, 1998. ACM Press, New York, US.

[10] N. Slonim and N. Tishby. The power of word clusters for text classification. In *23rd European Colloquium on Information Retrieval Research*, 2001.

[11] N. Tishby, F. Pereira, and W. Bialek. The information bottleneck method. In *Proceedings of the 37-th Annual Allerton Conference on Communication, Control and Computing*, pp. 368–377, 1999.

[12] Noam Slonim and Naftali Tishby. Document clustering using word clusters via the information bottleneck method. In *Research and Development in Information Retrieval*, pp. 208–215, 2000.

[13] Ron Bekkerman, Ran El-Yaniv, Naftali Tishby, and Yoad Winter. On feature distributional clustering for text categorization. In W. Bruce Croft, David J. Harper, Donald H. Kraft, and Justin Zobel, editors, *Proceedings of SIGIR-01, 24th ACM International Conference on Research and Development in Information Retrieval*, pp. 146–153, New Orleans, US, 2001. ACM Press, New York, US.

[14] Inderjit S. Dhillon, Subramanyam Mallela, and Rahul Kumar. Enhanced word clustering for hierarchical text classification. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 191–200, New York, NY, USA, 2002. ACM.

[15] I. Dhillon, S. Mallela, and R. Kumar. A divisive information-theoretic feature clustering algorithm for text classi cation, 2003.

[16] Karl-Michael Schneider. Techniques for improving the performance of naive bayes for text classification. In *CICLing*, pp. 682–693, 2005.

[17] Tao Liu, Zheng Chen, Benyu Zhang, Wei ying Ma, and Gongyi Wu. Improving text classification using local latent semantic indexing. In *ICDM '04: Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM'04)*, pp. 162–169, Washington, DC, USA, 2004. IEEE Computer Society.

[18] David Hull. Improving text retrieval for the routing problem using latent semantic indexing. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 282–291, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

[19] Thomas Hofmann. Probabilistic Latent Semantic Indexing. In *Proceedings of the 22nd Annual ACM Conference on Research and Development in Information Retrieval*, pp. 50–57, Berkeley, California, August 1999.

[20] Thomas Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Mach. Learn.*, Vol. 42, No. 1/2, pp. 177–196, 2001.

[21] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, Vol. 39, No. 1, pp. 1–38, 1977.

[22] L. Cai and T. Hofmann. Text categorization by boosting automatically extracted concepts, 2003.

[23] Robert E. Schapire and Yoram Singer. BoosTexter: A boosting-based system for text categorization. *Machine Learning*, Vol. 39, No. 2/3, pp. 135–168, 2000.

[24] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, Vol. 55, No. 1, pp. 119–139, 1997.

[25] Xavier Sevillano, Francesc Alias, and Joan Claudi Socoro. Reliability in ica-based text classification. In Carlos Garcia Puntonet and Alberto Prieto, editors, *ICA*, Vol. 3195 of *Lecture Notes in Computer Science*, pp. 1213–1220. Springer, 2004.

[26] Hiroya Takamura and Yuji Matsumoto. Feature space restructuring for svms with application to text categorization. In Lillian Lee and Donna Harman, editors, *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, pp. 51–57, 2001.

[27] K. Crammer and Y. Singer. the algorithmic implementation of multiclass kernel-based vector machines, 2001.

[28] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *COLT: Proceedings of the Workshop on Computational Learning Theory, Morgan Kaufmann Publishers*, pp. 92–100, 1998.

[29] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Meeting of the Association for Computational Linguistics*, pp. 189–196, 1995.

[30] Kamal Nigam, Andrew K. McCallum, Sebastian Thrun, and Tom M. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, Vol. 39, No. 2/3, pp. 103–134, 2000.

[31] T. Joachims. A support vector method for multivariate performance measures. In *International Conference on Machine Learning (ICML)*, pp. 377–384, 2005.

発表文献

- 岡嶋 穣, 松尾 豊, 石塚満, “専門用語の出現に基づく論文の重要度の分析.” 第 21 回人工知能学会全国大会.
- 岡嶋 穣, 松尾 豊, 石塚満, “語の出現予測を用いたテキスト分類.” 情報処理学会第 70 回全国大会. （発表予定）