素性の分割利用による識別性能の向上とスプログへの応用

有久　亘†　　佐藤　一誠†　　中川　裕志††

† 東京大学大学院 情報理工学系研究科　〒 113–0033 東京都文京区区本郷 7–3–1
†† 東京大学 情報基盤センター
E-mail: †{arihisa,sato}@r.dl.itc.u-tokyo.ac.jp, ††nakagawa@dl.itc.u-tokyo.ac.jp

あらまし　近年、ブログの増加とともにスパムブログ（以下、スプログとする）と呼ばれる広告収入を得ることを目的として機械的に生成されたブログが増えている．機械学習を用いたスプログの識別は有効であり、文書内の単語やアウトリンク数など様々な素性が提案されている．本研究ではこれらの素性に加えて文書内の潜在トピックを用いて識別性能の向上を目指す．しかし、TFIDF など従来の素性と潜在トピック素性とを組み合わせた識別は個々の素性集合での識別に比べ、精度が悪くなることがある．この問題に対し、我々は素性集合ごとに学習し、その結果を組み合わせることで精度が良くなる識別手法を提案する．この手法は各素性集合において SVM で学習した結果を確率値に変換しオッズ比を計算して組み合わせるものである．ラベル付きブログをデータセットとして用いた実験により，この手法が有効であることを示した．

キーワード　スパムフィルタリング、LDA、サポートベクターマシン、素性分割

Improving discriminative performance by partitioning feature spaces and its application to Spam Blog detection

Wataru ARIHISA†, Issei SATO†, and Hiroshi NAKAGAWA††

† Graduate School of Information Science and Technology, The University of Tokyo.　Hongo 7–3–1, Bunkyo-ku, Tokyo, 113–0033 Japan
†† Information Technology Center, The University of Tokyo
E-mail: †{arihisa,sato}@r.dl.itc.u-tokyo.ac.jp, ††nakagawa@dl.itc.u-tokyo.ac.jp

Abstract　In recent years, as the number of blogs increases, the number or Spam blogs which is generated automatically to obtain advertisement income are also increasing. The effectiveness of Spam blog filtering using machine learning approach was shown, and the various features, the words in the documents, the number of out-link was proposed. In addition to such features, we use latent topics in the documents to improve spam detection. However, performance of Spam blog detection is deteriorated by the simple combination of the latent topic feature and other features. We propose a novel method of the combination of the latent topic feature and other features. This method is that we transform the results of learning by SVM to the probability, calcurate the odds ratio, and produt them. In experiments using the actual spam blog dataset, the result of our method is promising.

Key words　Spam Filtering, Latent Dirichlet Allocation, Support Vector Machine, partition of feature space

1. はじめに

近年、ブログの増加に伴い、広告の表示や特定のサイトへ誘導すること等を目的として自動生成されているスパムブログ（スプログ）の数も増加している．機械的に大量生成されるスプログに対して、人手による識別は効率的ではなく、そのため機械学習による自動識別を考えるのは自然である．スパムの識別は与えられた文書をスパムと非スパムに分ける二値分類であり、文書分類において効果を発揮し、スプログ識別においても成果をあげている Support Vector Machine（以下、SVM とする）が広く使われている [5] [6]．ブログ以外のウェブページやメールのスパム対しても機械学習を用いた識別手法が有効であることが示されており、そこでは様々な素性が用いられている．記事内容の表層的素性として単語頻度やアウトリンク数、メタタグ、アンカーテキスト、タイトル内の単語、URL 内の単語などが提案されているが、本論文では tf-idf を改善した pivoted

tf-idf を考える．我々はスプログ識別の性能を向上させるため、表層的素性に対し、文書の潜在的な素性としてトピックを導入することを試みる．

訓練データ数の違いによって潜在トピック素性と tf-idf 素性の有効性は異なり、両素性の単純な組み合わせによる識別は精度の悪化を招く．そこで、各素性集合の良さを生かした識別手法を考えたい．本論文ではそのような識別手法として各素性集合の学習結果を組み合わせる Partitioned Logistic Regression(PLR) [2] を SVM に拡張した方法として Partitioned SVM(PSVM) を提案する．

PLR は各素性集合においてロジスティック回帰で学習した結果からオッズ比を計算し、それらを掛け合わせたものを識別に用いる．提案手法の PSVM では，まず SVM により tf-idf 素性と潜在トピック素性でそれぞれ学習した結果を重みを考慮して確率値に変換する．次に、その確率値からオッズ比を計算し、各素性集合ごとのオッズ比を掛け合わせることで全体のオッズ比を算出し識別に用いる．評価実験の結果，この手法は各素性集合をそのまま組み合わせる手法に比べ，性能の向上が見られた．

本論文の構成は以下のとおりである．まず，2 節においてスプログついて概観を述べ、3 節では識別に使用する tf-idf 素性と潜在トピック素性について説明する．続いて，4 節では，学習結果の組わせ方法として PLR について詳細に説明を行った後、我々の提案手法である PSVM について述べる．5 節ではその性能評価のための実験と結果を紹介する．6 節にまとめを述べる．

2. スプログとは

スプログとは広告の表示や特定のサイトへ誘導すること等を目的として自動生成されているブログのことである．スパム一般の特徴としては次の三つが考えられる [7]．(1) 機械的に生成された内容：スパムブログは他のウェブサイトやブログから取ってきたテキストをコピーしたり繋ぎ合わせたりすることで機械的に生成されている (2) 付加価値の無さ：機械的に寄せ集められた他サイトからのテキストなので付加価値が乏しい (3) 経済的な動機：スパムブログはアフィリエイト広告を表示したり、アフィリエイトサイトに誘導したりするためにある．スパムブログには上記のスパムが有する特徴に加えて次の特徴がある．(1) 頻繁に更新される記事：ブログの読み手は新しい記事に興味があり、なるべく多く彼らの目に触れるためスパムブログは絶えず新しい記事を生成する (2) トラックバックやコメントリンクの存在：ブログでは読み手が自由にリンクを張れるので、リンクをサイトへのレコメンドと捉えることはできない．

日本国内のスプログに関してはニフティが国内の約 9 割強のブログ、4.5 億記事（2008 年 3 月現在）を分析対象として、その中のブログ記事中に占める、スプログ率、並びにその種類等を調査している [8]．ニフティは 2007 年 10 月〜2008 年 2 月の各月ごとにそれぞれ約 10 万記事をサンプリングして、スパムブログの割合を調査した結果、5ヶ月間の平均で、約 40%がスプログということが分かったとしている．以下は各月でのスパム率である．

- 2007 年 10 月: 39.3%
- 2007 年 11 月: 40.1%
- 2007 年 12 月: 39.7%
- 2008 年 1 月: 39.9%
- 2008 年 2 月: 40.5%

ブログサービスを提供する事業者にとっては，サーバーや回線の容量がスプログによって圧迫されてしまい，サービスに支障が出る危険性がある．また、企業にとって，ブログの口コミ情報は人気のバロメータとして貴重な情報になるが、スプログが多くなると，自社の製品がどれだけ支持されているかを見極めるのが難しくなる．一般ユーザーにとっては大量に生成されたスプログによって有益なコンテンツが埋もれてしまい，検索サービスなどから欲しい情報にたどり着けなくなってしまう．故に、スプログの識別は現代的課題であり、多くの研究がなされている．スプログは機械的に生成されており、作られる目的が経済的動機ということもあって、いくつかの統計量において人手で作られた一般的なブログと異なる点が見られる．続く章ではその違いを捉えたスプログ識別に有用な素性について見ていく．

3. 各素性の概観

本節では識別に用いる潜在トピック素性と tf-idf 素性について説明する．

3.1 潜在トピック素性

これまでいくつかのトピックに基づいたテキスト分析の手法が提案されてきた．Hofman [4] は意味解析のために生成モデルである probabilistic latent semantic indexing(PLSI) を導入したが、PLSI には未知の文書のトピックの推定が不可能であり、パラメータの数が文書の数に比例して増加してしまうため、訓練データに過学習してしまうという問題があった．この問題を克服したのが Blei [1] らの提案した Latent Dirichlet Allocation(LDA) である．LDA はトピックを語彙上の分布として、文書をトピック上の分布としてモデル化しているが、これは文章ごとに異なり、一つの文書の中に混ざり合って存在しているトピックの自然なモデル化といえる．以下に、その概要を示す．まず、V 個の語彙と T 個のトピック、n 個の文書があるとする．各トピック z に対して V 上の分布ϕ_zが $Dir(\beta)$ からサンプリングされる．同様にして各文書 d に対して T 上の分布θ_dが $Dir(\alpha)$ からサンプリングされる．つまり、θ_dからトピック z が抽出され、ϕ_zから単語が抽出される．そのグラフィカルモデルを図 1 に示す．

LDA の推定方法の一つとしてギブズサンプリングがある [3]．ギブズサンプリングは $p(x_i|x_{-i})$(ここで、$x_{-i} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$) がわかっているときに、同時分布 $p(x), x \in \boldsymbol{R}^n$ からサンプリングを行うための MCMC アルゴリズムである．マルコフ連鎖の k 番目の遷移 $x^{(k)} \to x^{(k+1)}$ は次のように生成される．インデックス $i, 1 \leqq i \leqq n$ を選び、i を除くすべての x に対して $x^{(k+1)} = x^{(k)}$とし、$x_i^{(k+1)}$は $p(x_i|x_i^{(k)})$ からサンプリングする．LDA の目標は単語 w が与えられたときのトピック z の分布 $p(z|w)$ を推定することである．それ故、ギブズサンプリングでは $p(z_i|z_{-i}, w)$ を計算しなければならな

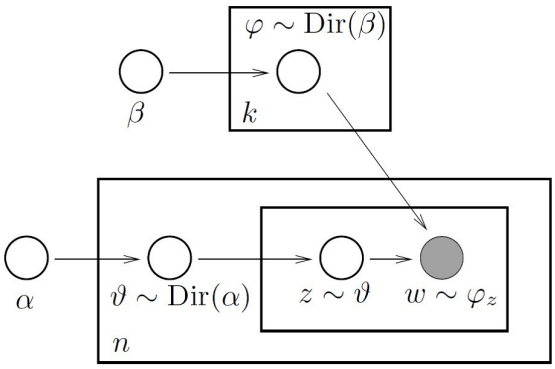

図 1　LDA のグラフィカルモデル

いが、これは以下のように閉じた形で計算が可能である.

$$p(z_i|z_{-i}, w) = \frac{n_{z_i}^{t_i} - 1 + \beta_{t_i}}{n_{z_i} - 1 + \sum_t \beta_t} \cdot \frac{n_d^{z_i} - 1 + \alpha_{z_i}}{n_d - 1 + \sum_z \alpha_z} \quad (1)$$

ここで、d は i 番目の文書、t_iは i 番目の単語、$n_{z_i}^{t_i}$はトピック z_i と単語 ti がともに現れる数、n^{z_i}はトピック z_iの数、$n_d^{z_i}$は文書 d 内の z_iの数、n_dは文書 d の長さである. 十分な繰り返しの後、トピック z が割り当てられる. z がわかっていれば、ϕ と θ の推定ができて

$$\phi_{z,t} = \frac{n_z^t + \beta_t}{n_z + \sum_t \beta_t} \theta_{d,z} = \frac{n_d^z + \alpha_z}{n_d + \sum_z \alpha_z} \quad (2)$$

ギブズサンプリングによって得られたサンプル平均から $p(z_{ji} = t|w_{ji})$ を計算し $z_{ji} = \mathrm{argmax}_t p(z_{ji} = t|w_{ji})$ として、文書 j の i 番目の単語 w_{ji}の潜在トピック z_{ji} を推定して素性とした. よって、文書 j を表す素性集合は z_{ji} となる.

3.2　tf-idf 素性

潜在トピックに対して、テキスト内の単語を使った素性として pivoted tf-idf [12] を考える. pivoted tf-idf は文書の長さを考慮に入れた tf-idf である. 文書 d と単語 w に対して pivoted tf-idf は以下のように定義される.

$$tf \cdot idf(w, d) = \frac{1 + \ln(tf_d(w))}{(1 - s) + s \cdot \frac{dl(d)}{avgdl}} \cdot \left(\frac{N + 1}{df(w)}\right) \quad (3)$$

ここで、$tf_d(w)$ は文書 d における単語 w の頻度、s はパラメータ、$avgdl$ は文書の平均長、$dl(d)$ は文書の長さ、N は文書数、$df(w)$ は w が出現する文書数である. つまり、文書 d の長さが全文書の平均長に対して大きければ、その分だけ tf-idf 値は割り引かれることになる. そして、その度合いを決めているのがパラメータ s である. $s = 2$ として pivoted tf-idf を計算し、この値を素性とした.

4.　提 案 手 法

ここでは 3 節で説明した素性を組み合わせる手法として従来研究である PLR について述べた後、提案手法である PSVM について述べる.

4.1　識別精度の悪化

tf-idf の素性集合を $\boldsymbol{F}^{\mathrm{TFIDF}}$、潜在トピック素性の集合を $\boldsymbol{F}^{\mathrm{LDA}}$ とし、各素性を用いた場合の SVM の学習結果を SVM($\boldsymbol{F}^{\mathrm{TFIDF}}$)、SVM($\boldsymbol{F}^{\mathrm{LDA}}$) とする. tf-idf 素性と潜在トピック素性を用いた SVM による識別結果を図 2 に示す. TFIDF が SVM($\boldsymbol{F}^{\mathrm{TFIDF}}$)、LDA が SVM($\boldsymbol{F}^{\mathrm{LDA}}$)、TFIDF+LDA が SVM($\boldsymbol{F}^{\mathrm{TFIDF}} \cup \boldsymbol{F}^{\mathrm{LDA}}$) を表している. 横軸は訓練データ数、縦軸は accuracy である. accuracy は識別精度を表す指標であり、正解したテストデータ数を $n_{correct}$、全テストデータ数を n_{total} とすると以下のように定義される.

$$accuracy = \frac{n_{correct}}{n_{total}} \quad (4)$$

正解したテストデータ数とはスパムであるものをスパムと非スパムであるものを非スパムと正しく識別できたブログサイトの数である. 潜在トピック素性を用いた識別と tf-idf の値を素性として用いた識別を比べると、前者は訓練サンプル数が少ない場合に後者より精度が高いが、訓練サンプル数が多くなるにつれ、後者の精度のほうが高くなる. また、両素性を組み合わせた場合の識別精度は tf-idf 素性のみを使った場合に比べて低くなっている.

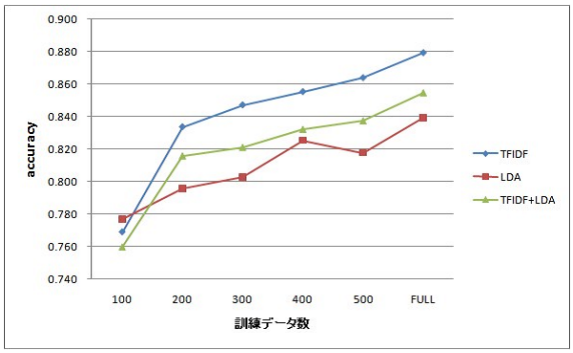

図 2　tf-idf 素性と潜在トピック素性を使った実験結果

上記の問題の原因を探るため、各素性における SVM による出力値 $\mathbf{W} \cdot \mathbf{X} + b$ の分布を比較する. ここで、係数 $\mathbf{W}$ は線形識別器の重みベクトルであり、非負値である b はバイアス項と呼ばれるパラメータである. 各出力値を 0.1 の幅でヒストグラムにし、その頂点を結んだものを図 3 に図示する.

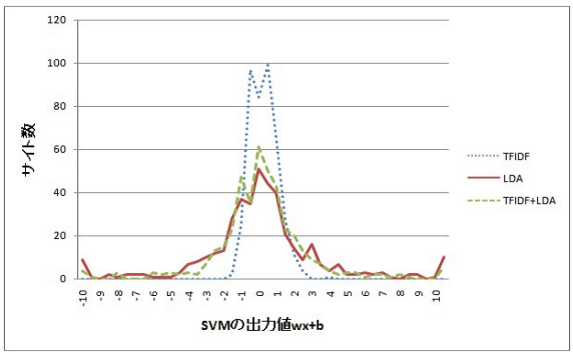

図 3　SVM の出力値の分布

TFIDF が SVM($\boldsymbol{F}^{\mathrm{TFIDF}}$)、LDA が SVM($\boldsymbol{F}^{\mathrm{LDA}}$)、TFIDF+LDA が SVM($\boldsymbol{F}^{\mathrm{TFIDF}} \cup \boldsymbol{F}^{\mathrm{LDA}}$) の出力値を表している. 横軸は出力値、縦軸はヒストグラムの幅に含まれる出力値のサイト数である. 訓練データ数によって分布の形は多少異なるが、両素性を

足し合わせた上で学習したときの SVM による出力値の分布は潜在トピック素性のみを使用した場合の出力値の分布に近い形の分布となる.

また、学習結果の出力値の相関行列は表 1 のようになっている. ここから SVM($\boldsymbol{F}^{\mathrm{TFIDF}} \cup \boldsymbol{F}^{\mathrm{LDA}}$) が SVM($\boldsymbol{F}^{\mathrm{LDA}}$) に近い

表 1　SVM による出力値 $\mathbf{W} \cdot \mathbf{X} + b$ の相関行列

	TFIDF	LDA	TFIDF+LDA
TFIDF	1		
LDA	0.368	1	
TFIDF+LDA	0.403	0.996	1

ものとなることが説明できる. つまり、tf-idf 素性と潜在トピック素性の単純な組み合わせによる識別精度の劣化は両素性の良さを有効活用できていないことにあるといえる.

4.2　素性分割型ロジスティック回帰

両素性の良さを活かす方法としては各素性で学習した結果を組み合わせる方法が考えられる. このような方法の研究はいくつか存在し、Raina [11] らは最初に各素性集合ごとにナイーブベイズで学習し、その結果の重みをロジスティック回帰で学習させている. その他の学習結果の組み合わせの方法として Lin らの素性分割型ロジスティック回帰 (Partitioned Logistic Regression,PLR) [2] がある.

$Y \in \{0,1\}$ を n 個の素性によって構成されるインスタンス$\boldsymbol{X} \in R^n$ のラベルとする. 特徴空間は事前に決められた k 個の集合に分割できると仮定する. つまり、$\boldsymbol{X} = (x_1^1, \ldots, x_{n_1}^1, x_1^2, \ldots, x_{n_2}^2, \ldots, x_1^k, \ldots, x_{n_k}^k)$ とできることとする. ここで、x_i^jは j 番目の集合の i 番目の特徴量を表しており、$n = \sum_{j=1}^{k} n_j$. 故に、j 番目の特徴量の集合を$\boldsymbol{X}_j$とすると元のインスタンスは$\boldsymbol{X} = (\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_k)$ と表現できる. j 番目の重みベクトルとバイアス項をそれぞれ$\boldsymbol{W}_j$と b_j とすると各モデルで推定される事後分布は

$$P(Y=1|\boldsymbol{X}_j) = \frac{\exp(\boldsymbol{W}_j \cdot \boldsymbol{X}_j + b_j)}{1+\exp(\boldsymbol{W}_j \cdot \boldsymbol{X}_j + b_j)} \tag{5}$$

となる. j 番目のモデルによって推定された事後オッズ比を $o_j = P(Y=1|\boldsymbol{X}_j)/P(Y=0|\boldsymbol{X}_j)$ とし、訓練データから推定される事前オッズ比を $o = P(Y=1)/P(Y=0)$ とする.. テストデータ $\boldsymbol{X}$ が与えられると、推定された事後オッズ比は

$$\frac{P(Y=1|\boldsymbol{X})}{P(Y=0|\boldsymbol{X})} = o^{(1-k)} \prod_{j=1}^{k} o_j \tag{6}$$

と定義できる. この事後オッズ比が 1 より大きければ $Y=1$ とし、そうでなければ $Y=0$ とする.

PLR モデルはナイーブベイズの仮定の下でロジスティック回帰モデルを統合したモデルとみなすことができる. つまり、クラスラベル Y が与えられたときに各素性集合は独立であり、

$$P(\boldsymbol{X}|Y) = P(\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_k|Y) = \prod_{j=1}^{k} P(\boldsymbol{X}_j|Y) \tag{7}$$

と書ける.

推定されたオッズ比と事後分布が正しければ、この仮定の下で式 (6) が真の事後オッズ比であることを以下のように示すことができる.

$$\begin{aligned}
\frac{P(Y=1|\boldsymbol{X})}{P(Y=0|\boldsymbol{X})} &= \frac{P(Y=1)P(\boldsymbol{X}|Y=1)}{P(Y=0)P(\boldsymbol{X}|Y=0)} \\
&= o \cdot \frac{\prod_{j=1}^{k} P(\boldsymbol{X}_j|Y=1)}{\prod_{j=1}^{k} P(\boldsymbol{X}_j|Y=0)} \\
&= o \cdot \prod_{j=1}^{k} \frac{P(Y=0)}{P(Y=1)} \cdot \frac{P(Y=1|\boldsymbol{X}_j)P(\boldsymbol{X}_j)}{P(Y=0|\boldsymbol{X}_j)P(\boldsymbol{X}_j)} \\
&= o^{1-k} \prod_{j=1}^{k} o_j
\end{aligned} \tag{8}$$

式 (5.2) はナイーブベイズの条件付き独立の仮定とは少し異なる. 各素性集合は条件付き独立であるが、各素性集合内における素性は独立である必要がない. この様子を表しているのが以下の図である.

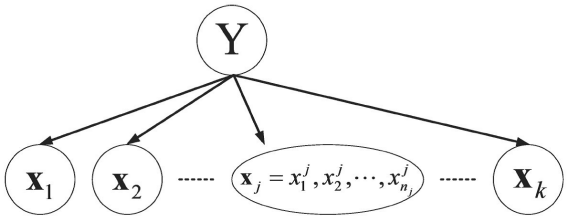

図 4　素性間の独立性の例

PLR モデルは k 個のロジスティック回帰によって構成されているとはいえ、全素性によって学習されるロジスティック回帰の形に容易に書き換えることができる.

$$\begin{aligned}
\log o(\boldsymbol{X}) &= \log\left(o^{1-k} \prod_{j=1}^{k} o_j\right) \\
&= (1-k)\log o + \sum_{j=1}^{k} \log o_j \\
&= (1-k)\log o + \sum_{j=1}^{k} (\boldsymbol{W}_j \cdot \boldsymbol{X}_j + b_j) \\
&= \sum_{j=1}^{k} (\boldsymbol{W}_j \cdot \boldsymbol{X}_j) + \left((1-k)\log o + \sum_{j=1}^{k} b_j\right)
\end{aligned} \tag{9}$$

つまり、k 個のモデルによって学習された重みを直接、全体のロジスティック関数の重みとし、新しいバイアス項 $(1-k)\log o$ と個々のモデルのバイアス項を足し合わせたものをバイアス項としている.

4.3　素性分割型サポートベクターマシン

PLR はナイーブベイズやロジスティック回帰よりも識別精度が高いことが経験的に知られている [2]. そこで、学習結果の組み合わせ方は PLR で用いられているオッズ比の掛け合わせを採用することとし、その方法を二値分類において高性能な SVM を用いて拡張する方法を考える. そのような方法を素性分割型サポートベクターマシン (Partitioned Support Vector

Machine, PSVM) として以下、詳細を述べる.

オッズ比の計算には確率値が必要だが、SVM は確率値を出力しない. そこで、シグモイド関数による SVM の出力値の確率値への変換を考える [10]. SVM による tf-idf 素性と潜在トピック素性の出力値の累積分布は図 5 のようになっている.

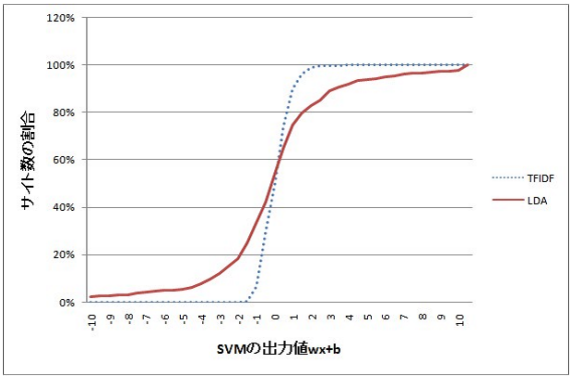

図 5　SVM の出力値の累積分布

TFIDF は tf-idf 値のみ、LDA は潜在トピック素性のみを使って学習した出力値の分布表している. この図からも SVM の出力値をシグモイド関数によって確率値に変換することは妥当であるといえる.

このように処理すれば、教師信号を $\{0,1\}$ から $\{-1,1\}$ へと変更することによって、PLR と同様に考えることができる. $Y \in \{-1,1\}$ を n 個の特徴量をもつインスタンス $\boldsymbol{X} \in \boldsymbol{R}^n$ のラベルとする. また、PLR のときと同様に特徴空間を事前に決めた k 個の集合に分割できるとする. つまり、$\boldsymbol{X} = (x_1^1, \ldots, x_{n_1}^1, x_1^2, \ldots, x_{n_2}^2, \ldots, x_1^k, \ldots, x_{n_k}^k)$ とできる. ここで、x_i^j は j 番目の集合の i 番目の特徴量を表しており、$n = \sum_{j=1}^{k} n_j$. ゆえに、j 番目の特徴量の集合を $\boldsymbol{X}_j$ とすると元のインスタンスは $\boldsymbol{X} = (\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_k)$ と表現できる. j 番目の素性集合で学習した SVM の識別関数は重みとバイアス項を $\boldsymbol{W}_j$ と b_j とすると、$\boldsymbol{W}_j \cdot \boldsymbol{X}_j + b_j$ となり、各モデルの事後分布は、シグモイド関数によって変換して

$$P(Y=1|\boldsymbol{X}_j) = \frac{1}{1+\exp(-\alpha_j(\boldsymbol{W}_j \cdot \boldsymbol{X}_j + b_j))} \tag{10}$$

とできる. ここで、α_j はシグモイド関数のパラメータであり、各素性集合ごとに適切な値を設定する必要がある. α が高いほど階段関数に近づき、学習結果である出力値を極端に変換する.

各モデルの事後分布から各モデルの事後オッズ比 $o_j = P(Y=1|\boldsymbol{X}_j)/P(Y=-1|\boldsymbol{X}_j)$ を計算し、訓練データから事前オッズ比 $o = P(Y=1)/P(Y=-1)$ を計算する. このとき、$\boldsymbol{X}$ が与えられたときの事後オッズ比は

$$\frac{P(Y=1|\boldsymbol{X})}{P(Y=-1|\boldsymbol{X})} = o^{1-k}\prod_{j=1}^{k} o_j \tag{11}$$

となる. この事後オッズ比が 1 より大きければ $Y=1$ とし、そうでなければ $Y=-1$ とする.

事後オッズ比のログをとると、

表 2　実験結果（訓練数ごとの accuracy）

訓練データ数	TFIDF	LDA	TFIDF+LDA	TFIDF+LDA(PSVM)
100	0.769	0.777	0.760	**0.790**
200	0.834	0.796	0.816	**0.838**
300	0.847	0.803	0.821	**0.859**
400	0.855	0.825	0.832	**0.862**
500	0.864	0.818	0.838	**0.864**
555	0.879	0.839	0.855	**0.881**

$$\begin{aligned}
\log o(\boldsymbol{X}) &= \log\left(o^{1-k}\prod_{j=1}^{k} o_j\right) \\
&= (1-k)\log o + \sum_{j=1}^{k}\log o_j \\
&= (1-k)\log o + \sum_{j=1}^{k}(\alpha_j(\boldsymbol{W}_j \cdot \boldsymbol{X}_j + b_j)) \\
&= \sum_{j=1}^{k}\boldsymbol{W}'_j \cdot \boldsymbol{X}_j + b' \qquad (12)
\end{aligned}$$

つまり、k 個の異なるモデルで学習された重み $\boldsymbol{W}'_j = \alpha_j \cdot \boldsymbol{W}_j$ は全体のロジスティック関数の重みとして用いることができ、$b' = (1-k)\log o + \sum_{j=1}^{k}\alpha_j \cdot b_j$ を新たなバイアス項と考えることができる. この式からも α が各素性集合ごとの学習結果を組み合わせる際の重みとなっていることがわかる.

5. 評価実験

本章では，我々が行ったこのシステムの性能評価とその結果について紹介する.

5.1 データセット

実験に使用するデータセットは Kolari [6] らが実験で使用したものと同様のブログデータを用いる. このデータは 1389 個のラベル付きのブログサイトから成っており、その内訳はスプログ 694 サイト、非スプログ 695 サイトとなっている. 各サイトに対し、html タグを除去した後、TreeTagger にかけ記号を含む単語を抽出した. 抽出した単語から URL や@を取り除き、tf-idf 値素性や潜在トピック素性の計算を行った.

SVM のソフトマージンを決める必要がある. そこで、データセットはパラメータチューニング用データ（SVM のスラック変数の係数調整用）、テストデータ、訓練データとして 1:3:6（それぞれ 139、416、834 サイト）に分けたものを 5 セット用意し実験を行った. 訓練データで学習した後、パラメータチューニング用データでスラック変数の係数を調整し、5 データセットの平均の accuracy が高かった値を使ってテストデータの識別を行う. テストデータを識別した結果も 5 データセットの accuracy の平均をとる. なお、SVM には SVM-Light [5] を用いた.

5.2 実験結果と考察

実験結果を図 6、図 7、表 2 に示す. 図 7 の横軸は訓練データの数、縦軸は各訓練データ数における accuracy を示す. TFIDF が SVM($\boldsymbol{F}^{\mathrm{TFIDF}}$)、LDA が SVM($\boldsymbol{F}^{\mathrm{LDA}}$)、TFIDF+LDA が

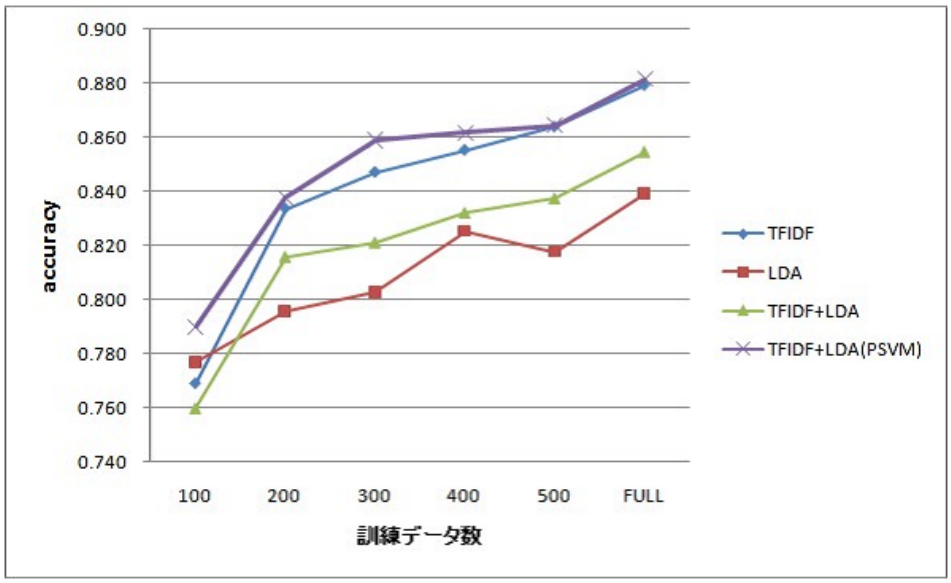

図 7　実 験 結 果

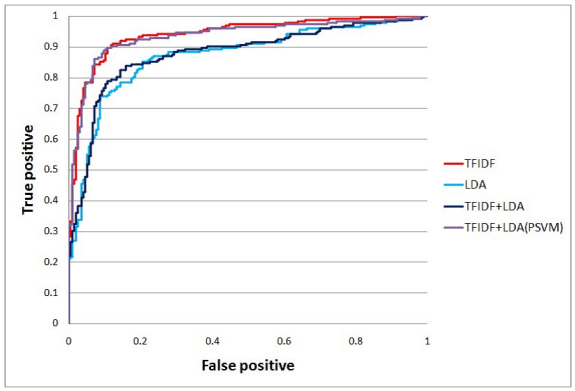

図 6　ROC 曲線

SVM($\boldsymbol{F}^{\mathrm{TFIDF}} \cup \boldsymbol{F}^{\mathrm{LDA}}$)、TFIDF+LDA(PSVM) が tf-idf 値と潜在トピックそれぞれで学習した結果を PSVM によって組み合わせた場合、つまり SVM($\boldsymbol{F}^{\mathrm{TFIDF}}$)·SVM($\boldsymbol{F}^{\mathrm{LDA}}$) を表している．また、図 6 は一つのテストセットに対する ROC 曲線である．ROC 曲線は横軸に非スパムをスパムと識別してしまう false positive の比率、縦軸にスパムをスパムと識別する true positive の比率としたときに学習器が描く曲線である．曲線の下の面積が大きい学習器ほど良い．非スパムをスパムと識別してしまうことは多大な損失であるので、accuracy 以外にもこのような指標も合わせて考える．実験において、潜在トピック素性のトピックの数は 200 としている．

PSVM におけるパラメータ α は validation set による実験で決定し、その値は表 3 に示すように tf-idf 素性を重視するような結果となった．TFIDF+LDA は TFIDF に比べて精度が劣っている．これに対して、TFIDF+LDA(PSVM) は両素性の識別結果をうまく利用し、TFIDF を上回っている．特に訓練データが 100 と少ない場合、潜在トピック素性のみの使用でも TFIDF より精度が高いが、PSVM による識別のほうがより高精度となっている．その他の訓練データ数の場合は潜在トピック素性があまり有効でないので、tf-idf 素性のみを用いた場合に比べ目立った改善はないが、tf-idf 素性の精度を低下させることなく多少の改善が行われている．ただ、ROC 曲線で見ると PSVM は TFIDF+LDA と LDA を大幅に上回っているものの TFIDF とはあまり変わらない結果となっている．

表 3　α　の　値

訓練データ数	TFIDF	LDA
100	9	1
200	7	1
300	17	1
400	13	1
500	4	1
555	22	2

次に、tf-idf 素性と潜在トピック素性以外にスパム識別に有効であるような素性群 (アウトリンク数、総単語数、語彙数、タイトルの文字数、ドメイン名の文字数、平均単語長、アンカーテキストの量) を加えた場合の実験結果を表 4 に示す．表の縦軸は識別に用いた素性を、横軸は学習器となっている．

表 4　tf-idf 素性と潜在トピック素性とその他の素性を用いた場合の実験結果

素性	SVM	PSVM	PLR
TFIDF	0.8793		
TFIDF+LDA	0.8548	0.8813	0.8813
TFIDF+LDA+Contents	0.8524	**0.8822**	0.8798

ここでは分割型ロジスティック回帰 (PLR) との比較も行う．TFIDF+LDA+Contents は tf-idf 素性と潜在トピック素性、その他の素性を用いた場合の学習結果である．識別精度のあまり高くない素性を加えたために SVM、PLR の精度は下がっているが、PSVM は Contents による補正を行い精度を上げている．PSVM における各モデルの重みは上の場合と同様に validation set での実験に基づき、(TFIDF:25,LDA:2,Contents:1) とした．

以下で訓練データ数が 555 のときの識別の成功例と失敗例について具体的考えていく．

PSVM の結果には TFIDF の結果が反映されやすいが、ここでは LDA の結果によって失敗した例と成功した例について考える．まず、失敗例についてみると TFIDF は正しく識別できているが、LDA は誤って識別してしまい、その結果が最終結果に反映されてしまっている．TFIDF は頻出する poker といった単語からギャンブル＝スパムという判断を下しているものと思われるが、LDA ではギャンブルに関するトピックが識別に有効でなかったと考えられる．成功例においては TFIDF が誤った識別を行っているが、LDA が正しく識別しており、最終的に正しく識別できている．adult や chat といった単語から容易にスパムと識別できそうだが、これらの単語はスパム、非スパムともによく表れる単語であり、識別に有効でなかったといえる．

6. おわりに

我々はスプログ識別性能の向上を目的として、潜在トピック素性の導入を試みた．さらに、識別性能向上のため、素性集合ごとの学習結果を組み合わせる手法である PLR を SVM を使って拡張した PSVM を提案した．訓練データ数の違いによって両素性の識別精度は異なるが、PSVM はいかなる訓練データ数の場合にも両素性の良さを生かした識別を行い、高い識別性能を示すことがわかった．

文　献

[1] D. Blei, A. Ng and M. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3(5):993-1022, 2003.

[2] M. Chang, W. Yih, C. Meek, Partitioned logistic regression for spam filtering. *KDD 2008*: 97-105

[3] T. Griffiths. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(suppl_1):5228.5235, 2004.

[4] T. Hofman. Prbabilistic latent semantic indexing. *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50-57, 1999.

[5] T. Joachims. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. *Proceedings of the European Conference on Machine Learning*, 1998.

[6] P. Kolari, T. Finin and A. Jochi. SVMs for the blogosphere: Blog identification and splog detection. *AAAI Spring Symposium on Computational Approaches to Analyzing Weblogs*, 2006.

[7] Y. Lin, H.Sundaram, Y. Chi, J. Tatemura, B. L. Tseng. Splog Detection Using Selfsimilarity Analysis on Blog Temporal Dynamics. *AIRWEB 2007 Proceedings*, 2007.

[8] ニフティ．プレスリリース, March 2008 `http://www.nifty.co.jp/cs/07shimo/detail/080326003337/1.htm`.

[9] A. Ntoulas, M. Najork, M. Manasse and D. Fetterly. Detecting spam web pages through content analysis. *Proceedings of the 15th international conference on World Wide Web*, Edinburgh, Scotland, May 2006.

[10] John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized liklihood methods. In *Advances in Large Margin Classifiers*, pages 61-74. MIT Press, 1999.

[11] R. Raina, Y. Shen, A. Ng, and A. McCallum. Classification with hybrid generative/discriminative models. In *Proceedings of NIPS 16*, 2004.

[12] A. Singhal, G. Salton, M. Mitra, and C. Buckley. Document length normalization. *Information Processing and Management*, 32(5):619-633, 1996.