社団法人 情報処理学会　研究報告
IPSJ SIG Technical Report

2004－ＦＩ－ 76　(8)
2004－ＮＬ－163　(8)
2004／9／16

最大マージン原理にもとづく多重トピック文書の自動分類

賀沢 秀人　　泉谷 知範　　平 博順　　前田 英作
NTT コミュニケーション科学基礎研究所
〒619-0237 京都府相楽郡精華町光台 2–4
{kazawa,izumi,taira,maeda}@cslab.kecl.ntt.co.jp

概要

本論文では，与えられたトピック集合の中から文書が該当するトピックを全て選びだす多重トピック文書の自動分類にたいして，最大マージンラベリング法と呼ぶ新しい学習手法を提案する．文書多重ラベリングにおいては，トピックの任意の組合せ（ラベル）を独立したクラスとみなした多クラス分類学習を行うことにより，より精度の高いラベリングが実現できると期待される．しかし，文書分類に代表される多重ラベリングの実問題においては，ラベルあたりのサンプル数の減少にともなう過学習が問題となり，こうした試みは実際にはなされてこなかった．提案手法では，各ラベルを高次元空間に埋め込んだ後にその空間でのマージンを最大化することにより，過学習を押え精度の良い多重ラベリングを実現する．実際に，Web 文書の文書多重ラベリングを対象として，Parametric Mixture Model[1], BoosTexter[2], SVM[3], 最近傍法といった様々な種類の従来手法との比較実験をおこない，提案手法がより高精度なラベリングをより少ない訓練データで実現できることを実証した．

キーワード： 多重ラベリング，マージン，カーネル，多クラス分類，文書分類

Maximum Margin Labeling for Multi-Topic Text Categorization

Hideto Kazawa, Tomonori Izumitani, Hirotoshi Taira and Eisaku Maeda
NTT Communication Science Laboratories
2-4, Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0237 Japan
{kazawa,arrigan,hirao,maeda}@cslab.kecl.ntt.co.jp

Abstract

In this paper, we address the problem of learining in multi-category document labeling. The goal of multi-category document labeling is to assign a document all the relevant categories from a given category set. The proposed learning method, Maximal Margin Labeling (MML), treats multi-category labels, as well as single-category labels, as independent classes and learns a kind of multi-class classifier on the multi-class problem. Since the number of multi-category labels are quite large in general, data sparseness becomes a serious challenge to MML. Thus we utilize a maximal margin principle in a high-dimensional space, into which all possible labels are embedded, to avoid over-fitting. MML is compared with other labeling methods, Parametric Mixture Model, BoosTexter, Support Vector Machines, and k nearest neighbors, using a collection of multi-category labeled Web pages. The results show that MML outperforms other methods and its high performace is apparent even with a small number of training samples.

Keywords: multi-category labeling, multi-label, margin, kernel, multi-class categorization, document classification

1　多重トピック文書の自動分類

本論文では，与えられたトピック集合の中から文書が該当するトピックを全て選びだす**多重トピック文書の自動分類**にたいして，新しい学習法を提案する．以下では，一つの文書に複数のトピックが付与される場合も考慮するため，ある文書 D に該当するトピックの集合を D の**ラベル**と呼び，トピックと区別して扱う．また，ラベルを付与することを**ラベリング**と呼ぶ．

従来ラベリングの学習にあたっては，文書があるトピックに属するか否かという二値分類問題に分解したのち，個々の分類器を学習するという方法が取られることが多かった．しかし，この方法では個々の**トピック**に固有の素性のみが分類に利用され，複数のトピックが組み合わさった**ラベル**に固有の素性は分類に利用されない可能性がある．

このことを説明するために，ある科学文献の分類システムにおいて，量子計算に関する文献は「量子物理」と「計算科学」の二つのトピックがラベリングされるように学習をおこなうことを考える．一般に，量子計算の文献においては qbit という量子情報の単位をあらわす単語が必ずと言って良いほど出現するが，それ以外の分野の文献で見かけることは稀である．したがって，理想的な文献分類システムにおいては，「qbit が出現したら量子物理と計算科学の二つをラベリングする」というルールが学習されると期待される．しかし，従来のトピックごとに問題を分解する方法では，このように複数のトピックを一度に付与するルールが獲得されることはない．また，「qbit」は量子物理や計算科学の各**トピック**を区別する特徴としては一般性に欠けるため[*1]，「qbit が出現したら量子物理のトピックが該当する」というルールも学習される可能性は低い．

以上の例からわかるように，従来のトピックごとに分割する方法には，複数のトピックが組み合わさった場合に有効な特徴を見落としてしまい，結果として分類精度が低下

[*1] 量子計算の文献数は量子物理や計算科学のそれと比べると非常に少ない．

記号	説明
$\mathbf{x}$ $(\in \mathbb{R}^N)$	文書ベクトル
Γ $(= \{c_1, \ldots, c_l\})$	全てのトピックからなる集合
L $(\subset \Gamma)$	ラベル
Λ $(= 2^\Gamma)$	全ての可能なラベルからなる集合
$L[j]$	$c_j \in L$ なら 1，それ以外 0

表 1: 本論文で使用する記号

する可能性がある．

本論文では，この問題を克服するために，ラベルを一つのクラスとみなして一種の多クラス分類学習をおこなう**最大マージンラベリング**を提案する．提案手法では，ラベルに含まれるトピック数とは無関係に，それぞれのラベルが独立の分布を形成すると仮定しているため，複数のトピックからなるラベルに固有の素性も含めて学習されるという長所がある．

本論文の構成は次の通りである．まず 2 で，本論文で用いる用語と記号の定義をおこなう．次に 3 で，従来のラベリング学習方法の問題点を指摘する．4 では，最大マージンラベリングについて説明したのち，実装上の問題点を議論する．5 では，インターネットから収集した多重トピック Web ページ [1] を用いて，最大マージンラベリングと従来手法の比較をおこなう．最後に，6 で本論文のまとめをおこなう．

2 用語および記号の定義

本論文で使用する用語について定義する（表 1）．

文書の素性ベクトルを $\mathbf{x} \in \mathbb{R}^N$ で表わす．各文書にラベリングする対象を**トピック**と呼ぶ．本論文では $\{c_1, c_2, \ldots, c_l\}$ の l 個のトピックをラベリングに用いるとする．全トピックの集合を Γ で表わす．

ラベルとは，文書に付与されたトピックの集合を指す．以下では，ラベルを L であらわし，全ての可能なラベルからなる集合を Λ であらわす．すなわち $\Lambda = 2^\Gamma$（Γ の巾集合）．また，ラベルに含まれるトピックの数を**ラベルサイズ**と呼ぶ．最後に，ラベル L にあるトピックが含まれるかを表わす記号として $L[j]$ $(0 \leq j \leq l)$ を導入する．$L[j]$ は，$c_j \in L$ のとき $L[j] = 1$，それ以外のとき $L[j] = 0$ と定義する．

3 従来手法の問題点

ここでは既存のラベリング学習の問題点を指摘する．

3.1 2 クラス分類器の組合せ

既存の学習手法で最も頻繁に用いられるのが，文書があるトピックに属するか否かの 2 クラス分類を学習する手法で，文書分類 [4] で多く用いられている．一つの分類器で識別できるのは一つのトピックのみなので，トピック数だけ分類器を学習する必要がある．

このアプローチでは，分類に寄与するのは個々の**トピック**に固有の素性である．そのため，複数のトピックが組み合わさった**ラベル**に固有な素性が存在しても，それが分類に利用されることはない．しかし，ラベル固有の素性が存在する場合には，当然，それを利用した方がラベリングの精度は高くなることが期待される．以上の議論より，2 クラス分類器を組み合わせる方法は学習の効率が悪いと考えられる．

3.2 BoosTexter[2]

BoosTexter はブースティング・アルゴリズム [5] の一種であり，一つの素性の有無だけにもとづいてラベリングをおこなう弱学習器を組み合わせて，より精度の高いラベリングをおこなう手法である．

[2] では，BoosTexter が，正解ラベル L と予測ラベル $\hat{L}$ の間の Hamming loss

$$\frac{|L \cap \hat{L}^c| + |L^c \cap \hat{L}|}{l},$$

もしくは ranking loss

$$\frac{|\{(c, c') \in L \times L^c : \mathrm{scr}(c) \leq \mathrm{scr}(c')\}|}{|L||L^c|},$$

を最小化する学習であることが示されている．ここで A^c は集合 A の補集合，scr は BoosTexter による各トピックの信頼度（confidence score）である．

しかし，[1] で指摘されているように，文書多重ラベリングにおいて通常成り立つように $|L| \ll l$ の場合，$\hat{L} = \{\}$（空集合）とすることで，容易に小さい Hamming loss を得ることができる．そのため，Hamming loss は学習基準として議論の余地がある．また，ranking loss を基準とした場合は，信頼度の閾値を別手段で決定する必要があり，実用上問題がある．

3.3 Parametric Mixture Model[1]

Parametric Mixture Model（以下，PMM）は，ナイーブ・ベイズ法 [6] をラベリング学習に拡張したもので，次のような確率モデルにもとづいている．まず，文書の各素性はラベル固有の多項分布にしたがって出現する．そして，各ラベルの多項分布は，ラベルを構成するトピック固有の多項分布を，等確率で混合することで得られる．

したがって，PMM では，あるラベルで頻出する素性はそのラベルに含まれるトピックのいずれかでも良く出現すると仮定されている．そのため，複数のトピックからなるラベルに固有の素性が存在した場合，適切なモデル推定ができない．

4 最大マージンラベリング

本節では，新しいラベリング学習手法として**最大マージンラベリング（Maximal Margin Labeling）**を提案する．（以下では，MML と略記する．）

最初に，MML の概要を説明する．MML は，各ラベルを独立したクラスとみなして，一種の多クラス分類学習をおこなう．そのため，従来手法ではうまく取扱えなかったラベル固有の素性をラベリングに反映させることができる．一方，現実の文書多重ラベリングでは，クラス数（ラベル数）が非常に多くなるため，クラスあたりの訓練データ数が少なくなり過学習の危険性が高くなる．そこで，MML では Suppor Vector Machines (SVM)[3] などで用いられている，**最大マージン原理**を応用し，過学習を回避する．

次に具体的に MML の説明をおこなう．MML は次の 3 つのステップによりラベリングをおこなう．以下では訓練データを $\{(\mathbf{x}_i, L_i)\}_{i=1}^m$ で表わす．

【ステップ 1（ラベルの埋め込み）】 以下で説明する埋め込み関数 $\phi : \Lambda \mapsto \mathbb{R}^M$ を用いて，訓練データの全てのラベル $\{L_i\}_{i=1}^m$ を $\mathbb{R}^M$ 中のベクトルに写像する（図 1）．ϕ は次の条件を満たすように構成する[*2]．

$$\langle \phi(L_1), \phi(L_2) \rangle = F_L(L_1, L_2). \tag{1}$$

ここで，$\langle , \rangle$ は $\mathbb{R}^M$ における内積である．一方，F_L は次で定義される関数で，以下では**ラベル F 値**と呼ぶ[*3]．

$$F_L(L_1, L_2) = \begin{cases} 1 & \text{if } |L_1| = |L_2| = 0 \\ \frac{2|L_1 \cap L_2|}{|L_1| + |L_2|} & \text{otherwise.} \end{cases} \tag{2}$$

ラベル F 値はラベルの近さを表現する量で，ラベルに含まれるトピックが完全に一致したときに 1，一つも一致しないときに 0 となる．また，Hamming loss とは異なり，ラベル F 値を大きくするような自明なラベルは存在しない．

式（1）と式（2）から，類似したラベルは $\mathbb{R}^M$ において近傍の点に写像されることがわかる．この性質により，一般的な多クラス分類学習とは異なり，MML ではあらかじめ重要なクラスを絞りこんで計算時間を削減することが可能となる．（詳細は 4.1.2 を参照.）

【ステップ 2（写像の学習）】 以下に示す最適化問題を解いて，文書・ベクトル空間 $\mathbb{R}^N$ から，ラベル埋め込み空間 $\mathbb{R}^M$ への線形写像 W（$M \times N$ 行列）を求める．

$$\begin{aligned} \min_W \quad & \frac{1}{2}\|W\|^2 + C\sum_{i=1}^m \sum_{\lambda \in \Lambda, \lambda \neq L_i} \xi_i^\lambda \\ \text{s.t.} \quad & \left\langle W\mathbf{x}_i, \frac{\phi(L_i) - \phi(\lambda)}{\|\phi(L_i) - \phi(\lambda)\|} \right\rangle \geq 1 - \xi_i^\lambda \\ & \xi_i^\lambda \geq 0 \\ \text{for} \quad & 1 \leq i \leq m, \forall \lambda \in \Lambda, \lambda \neq L_i, \end{aligned} \tag{3}$$

ここで $\|W\|$ は W のフロベニウス・ノルムであり，次の式で定義される．

$$\|W\| = \sqrt{\frac{1}{MN}\sum_{i=1}^M \sum_{j=1}^N W_{ij}^2}.$$

図 1 を用いて，式 (3) の意味を説明する．まず単純化のため ξ_i^λ を無視して考える．すると，求める W は，

$$\min_{i, \lambda \neq L_i} \left\langle W\mathbf{x}_i, \frac{\phi(L_i) - \phi(\lambda)}{\|\phi(L_i) - \phi(\lambda)\|} \right\rangle \geq 1 \tag{4}$$

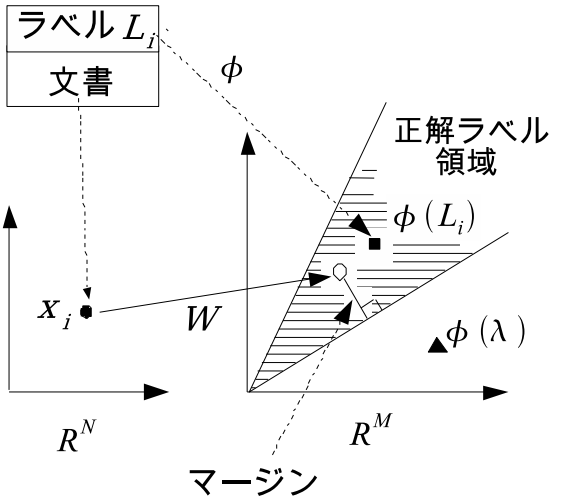

図 1: 最大マージンラベリング

という条件のもとで，$\|W\|$ を最小化する写像となる．式（4）左辺の内積は，W が訓練データ中の文書 $\mathbf{x}_i$ を写像した先 $W\mathbf{x}_i$ と，誤ったラベル $\lambda\,(\neq L_i)$ の領域との距離（**マージン**）を表わす．ここでラベル L の領域 $D(L)$ は，次の式で定義される[*4]．

$$\begin{aligned} D(L) &= \{\mathbf{v} \in \mathbb{R}^M | \arg\max_{\lambda \in \Lambda} \langle \mathbf{v}, \phi(\lambda) \rangle = L\} \\ &= \{\mathbf{v} \in \mathbb{R}^M | \langle \mathbf{v}, \phi(L) \rangle - \langle \mathbf{v}, \phi(\lambda) \rangle \geq 0 \\ &\qquad \text{for } \forall \lambda \in \Lambda, \lambda \neq L\}. \end{aligned}$$

結局，式（4）は誤ったラベル領域までのマージンの最小値がある定数（いまの場合，1）以上であるという条件を表わしている．

以上より，式（3）は（ξ_i^λ を除いて），次の最適化問題と同値である．

$$\max_W \frac{\text{誤ったラベル領域までの最小マージン}}{\|W\|}. \tag{5}$$

式（5）は，次のように解釈される．いま，いずれかの文書に微少な変化 δ が加わったとする．すると，その文書の写像先は $W\delta$ だけ変化する．このとき，誤った領域までの最小マージンが写像先の移動量 $\|W\delta\|$ よりも大きければ，その訓練データは正解ラベルの領域に留まることができる．したがって，$\|W\delta\| \leq \|W\|\|\delta\|$ であることに注意すると，式（5）は，訓練データ中の文書に微少な変化が生じても正解ラベルの領域に留まれるように，できるだけ大きなマージンを設けるという基準を表わしていることがわかる[*5]．（**最大マージン原理**）

実際のデータでは，式（4）が成り立たない場合もありうる．そのような場合に対処するために，式（3）では式（4）に違反した度合い ξ_i^λ に応じて重み C でペナルティを加えている（**ソフトマージン**）．

[*2] 4.1.1 で説明するように，MML を実行するには ϕ を陽に構成する必要はなく，式（1）を満たす ϕ の存在さえ保証できれば良い．紙面の都合上，証明は省略するが，F_L は $\Lambda \times \Lambda$ 上のカーネル関数 [7] であり，したがって，ϕ が存在することは数学的に保証される．

[*3] [1] では，式（2）を単に F 値（F-measure）と呼んでいる．しかし，情報検索などで通常用いられる F 値 [8] は，各トピックに属する文書をどれだけ正確に特定できたかを測る尺度であり，各文書のラベルをどれだけ正確に特定できたかを測るラベル F 値とは異なる量である．そこで，混乱を避けるため本論文では式（2）をラベル F 値と呼ぶ．

[*4] この定義より，任意のラベル領域は原点を頂点とする錐となる．

[*5] 文書分類のように有効な素性が非常に多く存在する問題 [9] では，文書の微少な変動に対して出力（ラベル）を一定に保つような不変性は，過学習を避けるために有効であることが知られている [10].

【ステップ３（ラベリング）】 MML によるテスト文書 $\mathbf{x}$ のラベリングは，式 (3) の解 W によって写像される領域のラベルを決定することでおこなう．すなわち，MML による $\mathbf{x}$ のラベルを $f(\mathbf{x})$ とすると，

$$f(\mathbf{x}) = \arg\max_{\lambda\in\Lambda} \langle W\mathbf{x}, \phi(\lambda)\rangle. \tag{6}$$

4.1 実装上の問題

4.1.1 ϕ の存在

式（3）は ϕ を陽に含んでいるため，実際の計算には適さない．しかし，式（3）の双対最適化問題 [11, 12] は，次の式になり ϕ を含まない．（導出は付録 A に示す．）

$$\begin{aligned}
&\max_{\alpha_i^\lambda} \sum_{i=1}^{m}\sum_{\lambda\neq L_i} \alpha_i^\lambda - \frac{1}{2}\sum_{i,j=1}^{m}\sum_{\lambda\neq L_i,\lambda'\neq L_j} \alpha_i^\lambda\alpha_j^{\lambda'}(\mathbf{x}_i\cdot\mathbf{x}_j)\\
&\times\frac{F_L(L_i,L_j)-F_L(L_i,\lambda')-F_L(\lambda,L_j)+F_L(\lambda,\lambda')}{2\sqrt{(1-F_L(L_i,\lambda))(1-F_L(L_j,\lambda'))}}\\
&\text{s.t.}\quad 0\le\alpha_i^\lambda\le C \quad \text{for } \forall i,\lambda\neq L_i.
\end{aligned} \tag{7}$$

ここで，ϕ に関する計算は全てラベル F 値 F_L に置き換えられている．また，$\mathbf{x}_i$ に関する計算も内積のみなので，任意のカーネル関数と置き換えることが可能である．式 (7) は α_i^λ に関する凸二次計画問題であり効率的に解くアルゴリズムが存在する [12].

式（7）の解を用いると，式（6）は次のように書くことができる．

$$\begin{aligned}
f(\mathbf{x}) &= \arg\max_{L\in\Lambda} g(\mathbf{x},L)\\
g(\mathbf{x},L) &= \sum_{i,\lambda\neq L_i}\alpha_i^\lambda(\mathbf{x}_i\cdot\mathbf{x})\\
&\qquad\times\frac{F_L(L_i,L)-F_L(\lambda,L)}{\sqrt{2(1-F_L(L_i,\lambda))}}.
\end{aligned} \tag{8}$$

ここで，導出のため式（15）を用いた．式（7）同様，式 (8) も ϕ を陽に計算する必要はなく，$\mathbf{x}_i$ に関する計算もカーネルで置き換え可能である．

4.1.2 可能なラベルに関する和

式（7）(8) は，全てのラベルに関する和を含んでいる．一方，ラベル数はトピック数にたいして指数関数的に増えるので，トピック数が多い場合には式（7）(8) を正確に計算することは現実的ではない．そこで，α_i^λ の性質を考慮して，一部のラベルに関する和で近似する方法を提案する．

まず，α_i^λ は式（3）の 2 行目の不等式制約に対応する双対変数であることに注意する．（付録 A を参照のこと．）したがって，

$$\left\langle W\mathbf{x}_i, \frac{\phi(L_i)-\phi(\lambda)}{\|\phi(L_i)-\phi(\lambda)\|}\right\rangle \le 1$$

が成り立つとき，またそのときに限り，$\alpha_i^\lambda\neq 0$ となる．すなわち，$\mathbf{x}_i$ と誤ったラベル λ の領域とのマージンが小さいときに限り $\alpha_i^\lambda\neq 0$ となる．

直観的にはマージンが小さくなるのは，λ が正しいラベル L_i に類似している場合と考えられる．そこで，本論文では次のような近似法を提案する．

類似ラベルによる近似

L_i の類似ラベルの集合 Λ_i を L_i とある一つのトピックの有無だけ異なるラベルの集合，すなわち，$\Lambda_i = \{\lambda\in\Lambda \mid |(\lambda\cap L_i^c)\cup(\lambda^c\cap L_i)|=1\}$ と定義する．そして，$\lambda\notin\Lambda_i$ ならば $\alpha_i^\lambda = 0$ として，式（7）(8) を計算する．

この近似は $\sum_{\lambda\neq L_i}$ を $\sum_{\lambda\in\Lambda_i}$ に置き換えることに相当するため，計算量の大きな削減につながる．実際，前者は 2^l 個のラベルに関する和であるのにたいし，後者は l 個のラベルに関する和を取るだけで済む．（l はトピック数.）

4.1.3 可能なラベルに関する探索

式 (8) は，全ての可能なラベルの中から最適なものを探索するという形になっている．ところが，トピック数 l が多い場合，2^l 個存在するラベルを全て列挙することは現実的ではない．しかし，以下に示すように，l の多項式時間で探索する方法が存在する．

まず，式 (8) は次のようにラベルサイズごとの問題に分割できる．

$$\begin{aligned}
f(\mathbf{x}) &= \arg\max_{\lambda\in\{L_{(0)},L_{(1)},\ldots,L_{(l)}\}} g(\mathbf{x},\lambda),\\
L_{(n)} &= \arg\max_{L\in\Lambda,|L|=n} g(\mathbf{x},L).
\end{aligned} \tag{9}$$

一方，$g(\mathbf{x},L)$ は以下のように展開できる．

$$\begin{aligned}
g(\mathbf{x},L) &= \begin{cases} a_0 & \text{if } |L|=0\\ \sum_{k=1}^{l} a_{|L|}[k]L[k] & \text{otherwise,}\end{cases}\\
a_0 &= \sum_{i,\lambda\neq L_i}\frac{\alpha_i^\lambda(\mathbf{x}_i\cdot\mathbf{x})}{\sqrt{2(1-F_L(L_i,\lambda))}}\\
&\qquad\times 2(I[|L_i|=0]-I[|\lambda|=0]),\\
a_n[k] &= \sum_{i,\lambda\neq L_i}\frac{\alpha_i^\lambda(\mathbf{x}_i\cdot\mathbf{x})}{\sqrt{2(1-F_L(L_i,\lambda))}}\\
&\qquad\times\left(\frac{2L_i[k]}{|L_i|+n}-\frac{2\lambda[k]}{|\lambda|+n}\right).
\end{aligned} \tag{10}$$

ここで $I[S]$ は S が真のとき 1，偽のとき 0 となる関数である．

式 (9) と式 (10) を組み合わせると，次のようなラベリング・アルゴリズムが構築できる．

1. $g_0 = a_0$，$K_0 = \{\}$（空集合）とする．
2. $n = 1$ とする．
3. $a_n[k]$ $(1\le k\le l)$ を式（10）から計算する．
4. $a_n[k]$ のうち大きいものから順に n 個のインデックス k を取り出し，その集合を K_n とする．
5. $g_n = \sum_{k\in K_n} a_n[k]$ とする．
6. $n < l$ であれば n を 1 増やして（3）へ戻る．
7. $n^* = \arg\max_n g_n$ として，$L = \{c_i | i\in K_{n^*}\}$ を出力して終了．

このアルゴリズムで最も時間がかかるのは（3）であり，$\alpha_i^\lambda\neq 0$ となる α_i^λ の数を n_α としたとき，$O(n_\alpha l^2)$ である[*6]．一方，4.1.2 の近似を用いると $n_\alpha \le ml$ であるか

[*6] l 個の $a_n[k]$ の計算で $O(n_\alpha l)$, それを l 回繰り返すので $O(n_\alpha l^2)$

ら，結局 $O(ml^3)$ となり，l の多項式アルゴリズムであることがわかる．

5 実験

MML の性能評価のため，[1] で使用された Web ページのデータを用いて，Parametric Mixture Model (PMM)[1]，BoosTexter (Boost)[2]，Support Vector Machines (SVM)[3]，最近傍法 (kNN) との比較をおこなった．これらの比較手法は，次のような基準で選択した．(1) 元来ラベリング学習法として提案されたもの（PMM, Boost），(2) 一般に文書分類において良い精度を示すことが知られているもの（SVM[9, 13]），(3) 実験で使用するデータで良い精度を示したもの（kNN[1]）．

なお, MML と SVM のカーネルとしては，ベクトル長で正規化した線形カーネル $\mathbf{x}\cdot\mathbf{x}'/\|\mathbf{x}\|\|\mathbf{x}'\|$ を用いた．MML の学習には 4.1.2 で説明した近似を適用した．また，kNN においても同様に，各素性ベクトルをベクトル長で正規化したのち距離計算をおこなった．

5.1 実験条件

5.1.1 Web ページデータ

本実験で使用した Web ページデータは，次のように作成されたものである．（詳細は [1] を参照.）まず，検索エンジン Yahoo! のトップページ（www.yahoo.com）に記載されている Business，Computers などのトップトピックからリンクを辿り，5 階層目までの Web ページを収集する．つぎに，各 Web ページがリンクされている全てのサブトピック（トップトピックの次の階層に記載されているトピック）を調べ，それをラベルとする[*7]．

実験においては，収集元となったトップトピックごとに Web ページを分割してデータセットを作成し，各データセットごとに Web ページのラベル（Yahoo!のサブトピックの集合）を予測する．表 2 に各データセットの概要を示す．

各データセットからランダムに 500 ページをパラメータ等の検定用データとして取り出し，さらに 3000 ページをテスト用データとして取り出した．また，残りのデータは訓練データを取り出すためのプーリングデータとした．

5.1.2 素性抽出

各 Web ページは, 以下の 3 種類の方法で数値ベクトルに変換した [8]．(1) 出現単語ベクトル (Binary)．各単語の出現を 1，非出現を 0 であらわしたベクトル．(2) 単語頻度ベクトル (TF)．各単語の出現回数を要素とするベクトル．(3)単語頻度×逆文書頻度ベクトル(TF×IDF)．各単語の出現頻度と inverse document frequency $\log(N_d/N_w)$ と掛け合わせたものを要素とするベクトル．ここで N_d は文書の総数，N_w は該当単語の出現する文書数である．

なお，本実験では学習手法同士の比較が目的のため，[1] と同様，ストップワードの除去やステミングなどの処理はおこなわず，出現した単語を全てそのまま使用した．

学習手法	素性抽出方法	学習パラメータ
MML	TF, TF×IDF	C = 0.1, 1, 10
PMM	TF	Model1, Model2
Boost	Binary	R=$\{2, 4, 6, 8, 10\}\times 10^3$
SVM	TF, TF×IDF	C = 0.1, 1, 10
kNN	TF, TF×IDF	k = 1, 3, 5, 7

表 3: 素性抽出方法と学習パラメータ．下線のものをテストで使用.

5.1.3 評価尺度

ラベリング精度を評価するために，本論文では以下の三つの尺度を用いる．なお，以下では $\{L_1^{true}, L_2^{true}, \ldots, L_n^{true}\}$ をテストデータの真のラベル，$\{L_1^{pred}, L_2^{pred}, \ldots, L_n^{pred}\}$ をそれに対応する予測ラベルとする．

平均ラベル F 値 [1]　平均ラベル F 値 $\bar{F}_L$ は，真のラベルと予測ラベルの平均的な近さを評価する尺度である．

$$\bar{F}_L = \frac{1}{n}\sum_{i=1}^{n} F_L(L_i^{true}, L_i^{pred}) = \frac{1}{n}\sum_{i=1}^{n}\frac{2\sum_{j=1}^{l} L_i^{true}[j]L_i^{pred}[j]}{\sum_{j=1}^{l}(L_i^{true}[j] + L_i^{pred}[j])}. \quad (11)$$

平均検索 F 値　実用上の観点から言うと，検索精度（各トピックに属する文書を正確に特定する精度）を評価することも重要である．そこで，以下で定義する平均検索 F 値 $\bar{F}_R$ を用いて検索精度を評価する[*8]．

$$\bar{F}_R = \frac{1}{l}\sum_{j=1}^{l}\frac{2|R_j^{true}\cap R_j^{pred}|}{|R_j^{true}| + |R_j^{pred}|} = \frac{1}{l}\sum_{j=1}^{l}\frac{2\sum_{i=1}^{n} L_i^{true}[j]L_i^{pred}[j]}{\sum_{i=1}^{n}(L_i^{true}[j] + L_i^{pred}[j])}. \quad (12)$$

ここで，$R_j^{true}(R_j^{pred})$ はトピック c_j を含むラベル $L_i^{true}(L_i^{pred})$ の番号 i の集合である．

完全一致率　最後に，最も厳しい評価尺度として完全一致率 EX を用いる．完全一致率は，予測ラベルと真のラベルが完全に一致する割合である．

$$EX = \frac{1}{n}\sum_{i=1}^{n} I[L_i^{true} = L_i^{pred}]. \quad (13)$$

5.1.4 素性抽出方法と学習パラメータの選択

素性抽出方法 (Binary, TF, TF×IDF) と, 学習パラメータ（MML/SVM におけるペナルティ C，PMM のモデル [1]，BoosTexter における弱仮説数 R，kNN における最近傍数 k）の組合せの中で最適なものを選ぶために，(1) プーリングデータからランダムに取り出した 2000 ページを使用して，各組合せで学習し，(2) 検定用データにおける平均ラベル F 値を計算し，最も成績の良かった組合せをテストに用いた．

となる．

[*7] 一般に，Yahoo! に登録されている Web ページは，複数のサブトピックからリンクされうる．

[*8] 平均検索 F 値は情報検索でマクロ平均 F 値と呼ばれるものに等しい．本論文では，ラベル F 値と区別するために検索 F 値と呼ぶことにする．

データセット名（略称）	語彙数	カテゴリ数	異なり	ラベルサイズ毎のデータ数（%）				
			ラベル数	1	2	3	4	≥5
Arts & Humanities (Ar)	23,146	26	599	55.6	30.5	9.7	2.8	1.4
Business & Economy (Bu)	21,924	30	233	57.6	28.8	11.0	1.7	0.8
Computers & Internet (Co)	34,096	33	428	69.8	18.2	7.8	3.0	1.1
Education (Ed)	27,534	33	511	66.9	23.4	7.3	1.9	0.6
Entertainment (En)	32,001	21	337	72.3	21.1	4.5	1.0	1.1
Health (He)	30,605	32	335	53.2	34.0	9.5	2.4	0.9
Recreation (Rc)	30,324	22	530	69.2	23.1	5.6	1.4	0.6
Reference (Rf)	39,679	33	275	85.5	12.6	1.5	0.3	0.1
Science (Si)	37,187	40	457	68.0	22.3	7.3	1.9	0.5
Social Science (SS)	52,350	39	361	78.4	17.0	3.7	0.7	0.3
Society & Culture (SC)	31,802	27	1054	59.6	26.1	9.2	2.9	2.2

表 2: 実験データの概要

	ラベルF値					検索F値				
データ	MML	PMM	Boost	SVM	kNN	MML	PMM	Boost	SVM	kNN
Ar	**0.552**	0.495	0.376	0.459	0.419	**0.302**	0.244	0.220	0.288	0.262
Bu	**0.801**	0.753	0.748	0.758	0.742	0.248	0.203	0.199	0.292	**0.326**
Co	**0.619**	0.606	0.474	0.549	0.531	0.274	0.191	0.172	0.300	**0.312**
Ed	**0.559**	0.508	0.368	0.480	0.455	0.248	0.209	0.160	**0.254**	0.252
En	**0.642**	0.606	0.488	0.541	0.522	**0.369**	0.304	0.285	0.353	0.348
He	**0.735**	0.662	0.603	0.668	0.596	0.347	0.228	0.263	**0.351**	0.318
Rc	**0.627**	0.548	0.443	0.486	0.505	**0.470**	0.361	0.328	0.395	0.403
Rf	**0.665**	0.625	0.502	0.561	0.564	0.285	0.235	0.163	0.245	**0.294**
Si	**0.609**	0.519	0.388	0.469	0.489	**0.365**	0.276	0.189	0.313	0.321
SS	**0.725**	0.657	0.587	0.643	0.595	**0.361**	0.178	0.148	0.306	0.324
SC	**0.600**	0.535	0.439	0.494	0.470	**0.286**	0.249	0.202	0.257	0.268
平均	**0.649**	0.592	0.492	0.555	0.535	**0.323**	0.243	0.212	0.305	0.312

表 4: 平均ラベルF値および平均検索F値

表 3 に，選択のために試した組合せと，そこから選ばれてテストに使用された組合せを示す．

5.1.5 その他の事項

SVM においては，正例と負例のどちらかが極端に少ない場合，正負例で異なるペナルティを使用することで分類性能が向上することが知られている [14, 15]．実験に用いる Web ページデータでも出現頻度が極端に低いトピックが存在するため，本実験では [15] と同様に，正例と負例のペナルティの比が，負例と正例の数の比と等しくなるようにした．

BoosTexter には弱学習器の違いにより 4 つの種類が存在する [2]．本実験では，そのうち real abstaining AdaBoost.MH と呼ばれるものを用いた．これは，（1）[2] の実験において，4 種のうち最も高精度だった real AdaBoost.MH とほぼ同等の精度を達成していること，（2）計算時間が短くて済み，繰り返し実験をおこなうことが可能であったこと，が理由である．

5.2 実験結果

最初の実験として，2000 件の Web ページで訓練した学習器にたいして，テストデータにおける平均ラベルF値と平均検索F値を計算した結果を，表 4 に示す．なお，訓練はランダムに作成したトレーニングデータで 5 回おこない，表 4 中の値はその平均である．また，スペースの関係上示すことができなかったが，各F値の標準偏差は 0.01 未満であった．

サイズ	MML	PMM	Boost	SVM	kNN
1	**0.718**	0.654	0.538	0.604	0.573
2	**0.543**	0.486	0.422	0.485	0.479
3	**0.457**	0.447	0.372	0.417	0.441
4	0.351	**0.375**	0.286	0.319	0.352
≥5	0.309	0.322	0.277	0.291	**0.332**

表 5: 正解ラベルサイズごとの平均ラベルF値．

	MML	PMM	Boost	SVM	kNN
既知（95%）	**0.667**	0.606	0.506	0.570	0.548
未知（5%）	0.316	**0.340**	0.242	0.275	0.308

表 6: 既知および未知ラベルデータにたいする平均ラベルF値（訓練データ数 2000）

第二の実験として，訓練データ数が 250，500，1000，1500，2000 の各場合について，全データセットで平均した平均ラベルF値と平均検索F値を計算した[*9]（図 2）．なお，計算時間の都合上，250 件と 2000 件の場合のみ，異なる訓練データで 5 回学習した結果を平均して示している．

次に，ラベルサイズと訓練データ中に出現しない未知ラベルがラベリングにどのように影響するかを調べるため

[*9] 以下，断らない限り，全データセットで平均した値を単に平均ラベルF値，平均検索F値と呼ぶ．

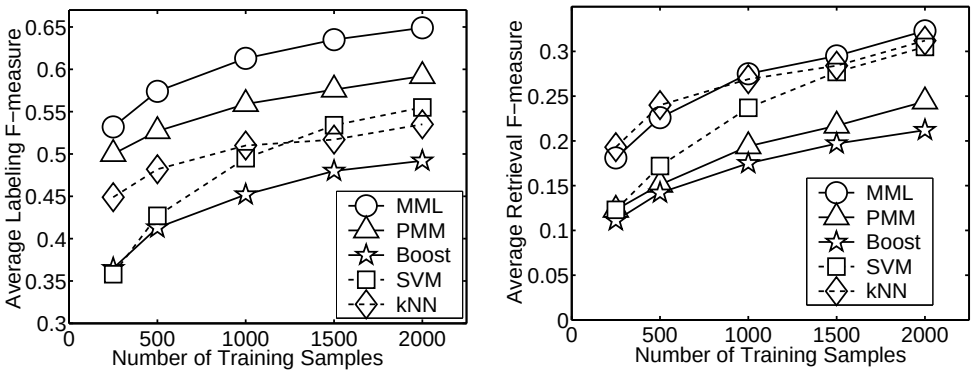

図 2: 平均ラベルＦ値（左）および平均検索Ｆ値（右）の学習曲線

	MML	PMM	Boost	SVM	kNN
既知（86%）	**0.577**	0.536	0.396	0.394	0.481
未知（14%）	0.265	**0.283**	0.179	0.144	0.253

表 7: 既知および未知ラベルデータにたいする平均ラベルＦ値（訓練データ数 250）

データ	MML	PMM	Boost	SVM	kNN
Ar	**0.441**	0.209	0.218	0.290	0.268
Bu	**0.630**	0.477	0.530	0.569	0.517
Co	**0.506**	0.349	0.337	0.408	0.392
Ed	**0.453**	0.194	0.228	0.303	0.315
En	**0.554**	0.313	0.359	0.419	0.402
He	**0.577**	0.344	0.387	0.469	0.403
Rc	**0.535**	0.246	0.325	0.366	0.388
Rf	**0.597**	0.390	0.412	0.489	0.463
Si	**0.522**	0.223	0.277	0.363	0.380
SS	**0.647**	0.445	0.487	0.553	0.484
SC	**0.443**	0.214	0.266	0.315	0.269
平均	**0.537**	0.309	0.348	0.413	0.389

表 8: 完全一致率

に，訓練データ数 2000 の場合について，平均ラベルＦ値をラベルサイズごと（表 5），および，既知[*10]／未知ラベルごと（表 6）に計算した．また，訓練データ数が 2000 の場合はテストデータ中に未知ラベルが５％しか存在しないため，訓練データ数 250 の場合についても同様の計算をおこなった．その結果を表 7 に示す．

最後に，完全一致率をデータセットごと（表 8），正解ラベルサイズごと（表 9），既知／未知ラベルごと（表 10，表 11）に計算した．ここで，表 8，表 9，表 10 は訓練データ数２０００の場合，表 11 は訓練データ数２５０の場合である．

5.3 考察

表 4 より，MML は平均ラベルＦ値と平均検索Ｆ値の両者で，従来手法よりも高い精度を達成していることがわかる．これは，従来手法はどちらか一方のＦ値のみ高くなる傾向があることを考慮すると，特筆すべき結果である．また，図 2 より，この傾向は訓練データ数が少ないときでも

[*10] 訓練データ中に出現するラベルのこと．

サイズ	MML	PMM	Boost	SVM	kNN
1	**0.711**	0.427	0.465	0.537	0.495
2	**0.239**	0.097	0.146	0.208	0.217
3	0.067	0.021	0.039	0.070	**0.098**
4	0.001	0.000	0.000	0.002	**0.009**
≥5	0.058	0.000	0.025	0.045	**0.071**

表 9: 正解ラベルサイズごとの完全一致率

	MML	PMM	Boost	SVM	kNN
既知（95%）	**0.566**	0.326	0.367	0.435	0.410
未知（5%）	0.000	**0.005**	0.001	0.002	0.000

表 10: 既知および未知ラベルデータにたいする完全一致率（訓練データ数 2000）

成り立っていることがわかる．特に平均ラベルＦ値に関しては，2000 件のデータで学習した従来手法と同等以上の精度を，4 分の 1 のデータ数である 500 件の学習で達成している．以上から，MML は従来手法と比べて高精度のラベリングをより少ないデータ数で実現できると言える．

表 5，表 6，表 7 より，MML は主にサイズの小さい既知のラベルを適切に予測することで，全般的な高精度を達成していることがわかる．サイズの小さい既知ラベルの方が正しく予測できるという傾向は，各学習手法とも共通であるが，中でも MML はその正しさの度合いが高い．一方，PMM はサイズの大きい未知ラベルを比較的正しく予測しているが，表 2 からわかるように，そのようなラベルを持つデータの割合は小さいため，全体の精度向上には大きく影響していない．まとめると，MML はデータ中での割合が大きい「サイズの小さい既知ラベル」を正確に予測しているため，全体の精度が高くなっていると言える．

表 8 より，MML はラベルを完全に正しく予測する率が，従来手法と比べて高いことがわかる．これは複数のトピックが組合わさったラベルを一つのクラスとみなす MML のアプローチが効果的であったためと考えられる．特に，表 4 中のラベルＦ値と比較すると，MML は完全一致が寄与する割合が高い．実際，完全一致率と平均ラベルＦ値の比を計算すると，MML 83%，PMM 52%，Boost 71%，SVM 75%, kNN 73% であり，MML においては完全に一致した予測が多いことがわかる．一方，サイズが大きく訓練データ中での出現頻度が少ないラベル，および未知ラベ

	MML	PMM	Boost	SVM	kNN
既知（86%）	**0.491**	0.310	0.286	0.324	0.368
未知（14%）	0.001	**0.010**	0.002	0.000	0.000

表 11: 既知および未知ラベルデータにたいする完全一致率（訓練データ数 250）

ルにたいしては，どの手法もほとんど完全一致する予測は出来ていない．（表 9，表 10，表 11）以上より，MML はラベルをクラスとみなしているため，データ中の出現頻度が多いラベルについて完全に一致する予測ができ，それが高いラベリング精度をもたらしていると考えられる．

6 まとめ

本論文では，文書多重ラベリングに関して，トピックの組合せであるラベルをクラスとみなして学習をおこなう最大マージンラベリング法（MML）を提案した．MML は，ラベルを高次元空間に埋め込んだ後にマージン最大化をおこなうことで，ラベルをクラスとみなすことに起因する過学習を避けている．Web 文書を対象とした実験により，MML は様々な従来手法と比べて，より高精度なラベリングをより少ないデータで実現できることを明らかにした．

今後の課題としては，ラベル埋め込みに用いた写像に関する理論的な正当化，より大規模なデータを扱うための高速なアルゴリズムの開発などが挙げられる．

謝辞

PMM のソースコードおよび実験データを提供して頂いた，ＮＴＴコミュニケーション科学基礎研究所の上田修功氏，斉藤和巳氏，金田有二氏に感謝致します．

参考文献

[1] N. Ueda and K. Saito: “Single-shot detection of multiple categories of text using parametric mixture models”, Proc. of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 626–631 (2002).

[2] R. E. Schapire and Y. Singer: “BoosTexter: A boosting-based system for text categorization”, Machine Learning, **39**, 2/3, pp. 135–168 (2000).

[3] V. N. Vapnik: “Statistical Learning Theory”, John Wiley & Sons, Inc. (1998).

[4] 永田, 平：“テキスト分類—学習理論の「見本市」—”, 情報処理, **21**, 1, pp. 32–37 (2001).

[5] Y. Freund, R. E. Schapire, 安倍 (訳)：“ブースティング入門”, 人工知能学会誌, **14**, 5, pp. 771–780 (1999).

[6] R. O. Duda, P. E. Hart and D. G. Stork: “Pattern Classification, Second Edition”, John Wiley & Sons, Inc. (2001).

[7] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda and B. Schölkopf: “An introduction to kernel-based learning algorithms”, IEEE Transactions on Neural Networks, **12**, 2, pp. 181–201 (2001).

[8] 北, 津田, 獅々堀：“情報検索アルゴリズム”, 共立出版 (2001).

[9] T. Joachims: “Text categorization with support vector machines: learning with many relevant features”, Proc. of the 10th European Conference on Machine Learning (Eds. by C. Nédellec and C. Rouveirol), No. 1398, pp. 137–142 (1998).

[10] M. Sassano: “Virtual examples for text classification with support vector machines”, Proc. of 2003 Conference on Empirical Methods in Natural Language Processing, pp. 208–215 (2003).

[11] 田中：“凸解析と最適化理論”, 牧野書店 (1994).

[12] S. Boyd and L. Vandenberghe: “Convex Optimization”, Cambridge University Press (2004).

[13] H. Taira and M. Haruno: “Feature selection in svm text categorization”, Proc. of the 16th National Conference on Artificial Intelligence (AAAI-99), pp. 480–486 (1999).

[14] 平, 向内, 春野：“Support vector machine によるテキスト分類”, 情報処理学会研究報告, 98-NL-128-24, pp. 173–180 (1998).

[15] K. Morik, P. Brockhausen and T. Joachims: “Combining statistical learning with a knowledge-based approach - a case study in intensive care monitoring”, Proc. of the 16th International Conference on Machine Learning, pp. 268–277 (1999).

A 式（7）の導出

式（3）のラグランジアン $\mathcal{L}$ は，次のようになる．

$$\begin{aligned}\mathcal{L} = \frac{1}{2}\|W\|^2 + C\sum_{i,\lambda}\xi_i^\lambda \\ - \sum_{i,\lambda}\alpha_i^\lambda\left\{\left\langle W\mathbf{x}_i, \frac{\phi(L_i)-\phi(\lambda)}{\|\phi(L_i)-\phi(\lambda)\|}\right\rangle - 1 - \xi_i^\lambda\right\} \\ - \sum_{i,\lambda}\beta_i^\lambda\xi_i^\lambda. \qquad (14)\end{aligned}$$

ここで，$\alpha_i^\lambda, \beta_i^\lambda (\geq 0)$ は式（3）中の二つの不等式制約に対応する双対変数である．また，煩雑さを避けるため，$\sum_{i=1}^m\sum_{\lambda\in\Lambda,\lambda\neq L_i}$ を $\sum_{i,\lambda}$ であらわした．

式（3）の解は，式（14）の鞍点と一致するので，まず主変数 W, ξ_i^λ についての微分を 0 とおくと

$$\frac{\partial\mathcal{L}}{\partial W} = W - \sum_{i,\lambda}\alpha_i^\lambda\frac{\phi(L_i)-\phi(\lambda)}{\|\phi(L_i)-\phi(\lambda)\|}\mathbf{x}^T = 0,$$

$$\frac{\partial\mathcal{L}}{\partial\xi_i^\lambda} = C-\alpha_i^\lambda-\beta_i^\lambda = 0,$$

であり，結局

$$W = \sum_{i,\lambda}\alpha_i^\lambda\frac{\phi(L_i)-\phi(\lambda)}{\|\phi(L_i)-\phi(\lambda)\|}\mathbf{x}^T, \qquad (15)$$

$$\beta_i^\lambda = C-\alpha_i^\lambda, \qquad (16)$$

となる．

最後に，式（15）（16）を式（14）に代入し，式（1）を用いると，式（7）の目的関数が得られる．また，式（7）の制約 $0 \leq \alpha_i^\lambda \leq C$ は，式（16）と $\beta_i^\lambda \geq 0$ から導かれる．