テキスト分類
－学習理論の「見本市」－

永田 昌明
NTTサイバースペース研究所
nagata@nttnly.isl.ntt.co.jp

平　博順
NTTコミュニケーション科学基礎研究所
taira@cslab.kecl.ntt.co.jp

■情報を捨てる技術■

今流行りの「捨てる技術」という言葉は，現代の個人あるいは組織が直面している情報管理の本質が，「必要」または「不必要」という2つのカテゴリーのいずれかへ物事を分類する作業にあることをよく表している．しばしば「情報洪水」と形容されるように膨大かつ多様なコンテンツが流通する現在のインターネットでは，テキスト情報を自動的に分類する技術がさまざまなところで非常に重要な役割を果たす．

たとえば，暴力やポルノなど，好ましくないWebサイトへ子供たちがアクセスすることを防ぐWebコンテンツフィルタは，Webサイトを「有害」と「無害」に分類する．また，「楽して金儲けする方法」など，不特定多数に一方的に送りつけられる広告や勧誘等の電子メール（unsolicited business emailまたはspam）をシャットアウトするスパムフィルタは，電子メールを「普通のメール」と「スパム」に分類する．

社会的な要請から，Webおよび電子メールのコンテンツフィルタリング機能は，インターネットサービスプロバイダ（ISP）側およびクライアント（ブラウザ・メーラなど）側の両方で標準的に提供すべき機能となりつつある．多くの場合，これらのコンテンツフィルタは，好ましくない情報源のリスト（ブラックリスト）を管理する技術，および，内容に基づくフィルタリング技術（テキスト分類技術）の組合せにより実現される[9]．

そのほか，キーワードなどで指定することにより自分が興味を持つ分野の記事だけを配信してくれる電子メールニュースサービス，過去の購買履歴から自分の興味に合う本を推薦してくれるオンライン書店，顧客からの電子メールを専門分野別にオペレータへ振り分けるオンラインヘルプデスクなどに，テキスト分類技術を応用することができる．

テキスト分類技術の歴史的な変遷を振り返ると，1980年代後半までは知識工学的アプローチ，すなわち人手で分類規則を書く方法が主流であった．しかし，1990年代に入ると，大量のテキストデータが利用可能になったことや，コンピュータの性能が大幅に向上したことから，機械学習的アプローチ，すなわち人手によりカテゴリーラベルを付与したテキストデータから自動的に分類器を作成する方法が，分類精度・省力性・保守性に優れているために主流となった．さらに近年では，ブースティングやサポートベクタマシンなどの最先端の機械学習アルゴリズムが次々とテキスト分類に適用された結果，さまざまな学習理論の実用性を共通のベンチマークに基づいて比較検討することが可能となり，テキスト分類は学習理論の「見本市」と呼んでよい状況になっている．

本稿では，まずテキスト分類の基本的な問題設定および代表的な手法について説明し，次に近年その分類性能の高さが注目されているサポートベクタマシンを日本語のテキスト分類に適用した事例を紹介する．

■テキスト分類の数学的定義■

以下では，テキスト分類（text classification）というタスクを，あらかじめ設定された2つ以上のクラス☆1に文書を自動的に分類することと定義する．

一般に，テキスト分類では，文書を多次元のベクトル

$$\mathbf{x}=(x_1, x_2, \ldots, x_l)$$

☆1 クラスとカテゴリーは同義である．しかし，人手による分類体系ではカテゴリー，計算機による自動分類ではクラスという用語を使用する習慣があるので，本稿でも両者を使い分ける．

	“愛”	“逆転”	“国会”	“ホームラン”
文書1 ($\mathbf{x}_1$)	0	1	0	1
文書2 ($\mathbf{x}_2$)	0	1	1	0

図-1 文書の特徴表現

で表現する．文書ベクトルの各要素は，ある単語がその文書に出現するか否かという2値の場合もあれば，適当な方法（TF-IDFなど）で重みを付けた実数値の場合もある．

たとえば，（“愛”，“逆転”，“国会”，“ホームラン”）という4つの単語の出現の有無を素性（feature）として，次の2つの文書

文書1:「最終回に逆転満塁ホームランが飛び出した．」

文書2:「国会で与野党の勢力が逆転した．」

をベクトル表現すると，図-1のような文書ベクトル$\mathbf{x}_1$, $\mathbf{x}_2$が得られる．

上記の例では4次元のベクトルで文書の特徴を表現したが，多様な文書を高精度で分類するためには，できるだけ多く（数万以上）の素性を使用することが望ましい．しかし，学習を行う際の過学習や計算時間の問題から，多くの分類器は数百から数千程度に素性を削減する必要がある．そこで，単語出現頻度，文書頻度，相互情報量，情報利得などさまざまな評価基準を用いた素性選択法が提案されている[20]．

一方，各文書に対しては，その文書が所属するクラスのラベルyが与えられる．クラスの数は，最も簡単な2クラス（所属するか否か）の場合もあれば，複数クラス（multi-class）の場合もある．また1つの文書が1つのラベルしか持たない場合と複数のラベルを持つ（multi-label）場合がある．一般には，複数の2クラス分類器の組合せにより，複数クラス・複数ラベルの分類問題を解くことが多い．

たとえば，テキスト分類の代表的なベンチマークの1つであるReuters-21578は12,902個のニュース記事（平均200語）が118カテゴリー（corporate acquisitions, earnings, money market, grain, など）に分類され，1つの記事は平均1.2個のカテゴリーが割り当てられている．

上記の設定のもとで，テキスト分類のための分類器を作成する問題は，訓練データ

$$S=\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2),\ldots, (\mathbf{x}_n, y_n)\}$$

が与えられたとき，予測されたラベルが本当のラベルと異なる回数

$$\sum_{i\in S} f(x)\neq y_i$$

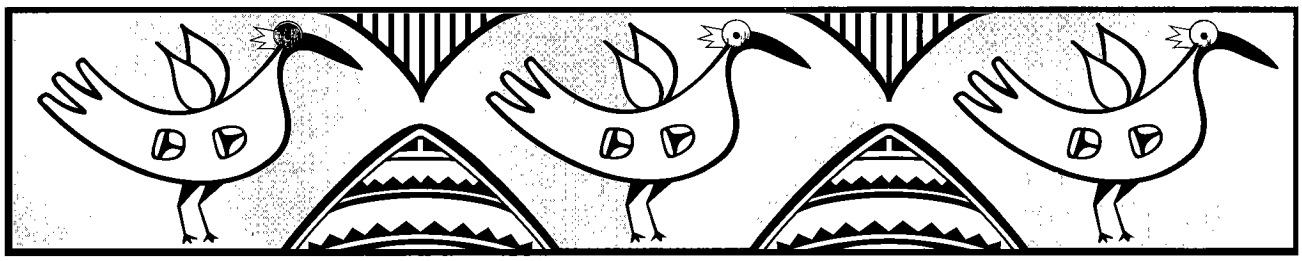
を最小化するような関数$f(\mathbf{x})$を求める問題と定式化できる．なお，2クラス分類の場合は，クラスラベルを$y\in\{-1, +1\}$とし，$sign(f(\mathbf{x}))$を予測されたラベル，$|f(\mathbf{x})|$を予測の確信度とする．

■代表的なテキスト分類アルゴリズム■

テキスト分類は，文書の表現法および分類器の構築法の違いにより，さまざまな手法が提案されている[16]．文書の表現法，すなわち，文書ベクトルの要素となる単語素性の選択（次元削減），重み付けなどは，どちらかといえば自然言語処理および情報検索に関する知見が重要な役割を果たす問題であるが，実証的な比較検討が進み，標準的な手法が固まりつつある．

文献20）ではさまざまな素性選択の尺度を比較し，文書頻度・情報利得・カイ2乗検定の精度はほぼ同じだが，相互情報量は次元削減が進むとやや精度が落ちるので，少ない計算量でそれなりの精度が得らる文書頻度が優れていると報告している．そのほか，多変量解析の手法（特異値分解，singular value decomposition）を使って，文書ベクトルをまったく別の縮退した空間に写像してしまう潜在意味インデキシング（latent semantic indexing）という方法もある[3]．

これに対して分類器の構築は，非常に古典的な教師付き帰納学習の問題なので，naive Bayes[10]，決定木，決定リスト[11]，k-最近隣法（example-based, memory-basedを含む），オンラインアルゴリズム（Winnow, perceptron, Rocchio），最大エントロピー法，サポートベクタマシン[5), 18)]，ブースティング[15]など，主な機械学習アルゴリズムはすべて試されているといってよい．



この中で，現在，最も優れた性能を発揮しているのがAdaBoost[6]やサポートベクタマシン[19]などのlarge margin classifierである．

ブースティング

ブースティング(boosting)は，ランダム予測より少し良い予測ができる弱学習器(weak learner)を組み合わせて高精度な分類器を作成する手法の1つである．図-2にブースティングの概念図を示す．ブースティングでは訓練データに対する重みD_tを変えながら，同じ弱学習器をT回繰り返して呼び出し(t=1...T)，T個の仮説h_tを生成する．この1回の操作をラウンドと呼ぶ．最後に各仮説の分類誤り率ϵ_tを基にして計算される仮説h_tに対する重みα_tを使って線形和を求め，その符号($sign$)を分類結果とする最終仮説Hを生成する．

ブースティングアルゴリズムの1つであるAdaBoostを図-3に示す．AdaBoostでは，最初はすべての訓練データに等しい重みを与えるが，各ラウンドにおいて，分類を誤った事例の重みを指数的に増やし，より難しい事例を集中して学習する．分類誤り率ϵ_tから適応的(adaptive)に仮説h_tに対する重みα_tと次のラウンドにおける訓練データに対する重みD_tを決めることからAdaBoost(adaptive boosting)と呼ばれる．AdaBoostは，きわめて単純で容易に実装でき，かつ，計算効率がよい点が優れている．文献15)では，単語の存在をテストする弱学習器とAdaBoostの組合せにより，現在最高水準のテキスト分類精度を実現できると報告している．

サポートベクタマシン

サポートベクタマシン(support vector machine, SVM)は，訓練データを正例と負例に分け，かつ，正負例間のマージンが最大になるような超平面を求める学習器である．この超平面は構造的リスク最小化という概念のもとでの最適解となる．

図-4にSVMの概念図を示す．SVMは超空間内で正例と負例を分離する超平面を求める．最も負例よりの正例側の境界面と最も正例よりの負例側の境界面の間の距離をマージン(margin)と呼ぶ．このマージンが最大となるような超平面を求め，$\mathbf{w}\cdot\mathbf{x}+b=0$を最終的な分類境界面とする．また$\mathbf{w}\cdot\mathbf{x}+b=1$および$\mathbf{w}\cdot\mathbf{x}+b=-1$上の訓練データをサポート・ベクタ(support vector)と呼ぶ．ただし，完全に線形分離できない場合には誤差も考慮に入れて分離境界面を決定する．

詳細は省略するが，訓練データと境界面の最小距離を最大化する問題は，Lagrange乗数α_iを導入すること

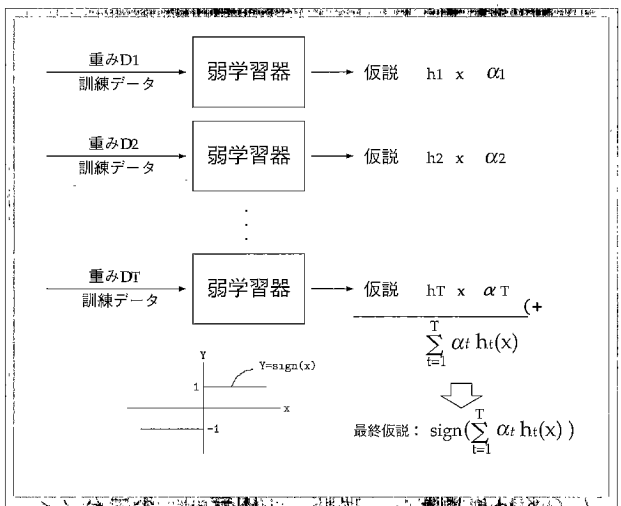


図-2 ブースティングの概念図

訓練データ: $(x_1, y_1), \ldots, (x_n, y_n)$ $y_i \in \{-1, 1\}$
ラウンド t における重みの分布D_t

初期値 $D_1(i) = 1/n$
for $t = 1, \ldots, T$

1. 学習器 h_t と分布 D_t により誤り率 ϵ_t を求める．

$$\epsilon_t = \sum_{i:h_t(x_i)\neq y_i} D_t(i)$$

2. 重み更新係数 α_t を求める．

$$\alpha_t = \frac{1}{2}\log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$$

3. 重みの分布 D_t を更新する (Z_t は正規化定数)

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } y_i = h_t(x_i) \\ e^{\alpha_t} & \text{if } y_i \neq h_t(x_i) \end{cases} = \frac{D_t(i)}{Z_t} \times e^{-\alpha_t y_i h_t(x_t)}$$

最終仮説 H

$$H(x) = sign\left\{\sum_t \alpha_t h_t(x)\right\}$$

図-3 AdaBoostアルゴリズム

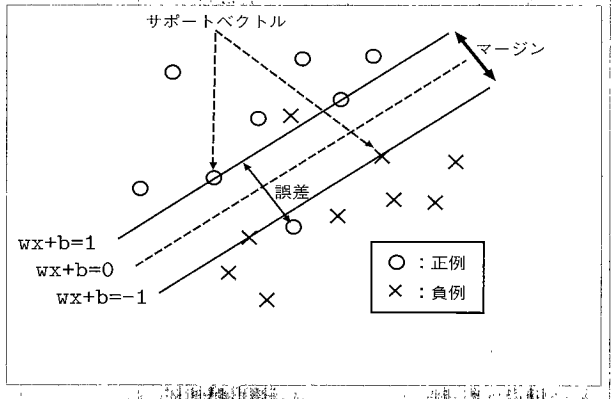


図-4 SVMの概念図

により，式(2)の制約条件のもとで，式(1)の目的関数を最小化する2次計画法(Quadratic Programming)の問題に変換できる．

$$-\sum_{i=1}^{l}\alpha_i+\frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j \mathbf{x}_i\cdot\mathbf{x}_j \quad (1)$$

$$\sum_{i=1}^{l}\alpha_i y_i=0,\ \forall_i:\alpha_i\geq 0 \quad (2)$$

ここで，y_iは$\mathbf{x}_i$に対するクラスラベルであり，$\mathbf{x}_i$が正例のとき+1，負例のとき−1の値をとる．またlは$\mathbf{x}$の次元数である．この2次計画問題を数値計算で解いて求めたα_i，正例側のサポートベクタ$\mathbf{x}_a$，負例側のサポートベクタ$\mathbf{x}_b$から，以下のようにして$\mathbf{w}$とbを決定できる．

$$\mathbf{w}=\sum_{i=1}^{l}\alpha_i y_i \mathbf{x}_i \quad (3)$$

$$b=-\frac{1}{2}(\mathbf{w}\cdot\mathbf{x}_a+\mathbf{w}\cdot\mathbf{x}_b) \quad (4)$$

また，SVMでは式(1)中の内積$\mathbf{x}_i\cdot\mathbf{x}_j$をカーネル関数と呼ばれる非線形関数$K(\mathbf{x}_i, \mathbf{x}_j)$で置き換えることによって非線形の仮説を扱うこともできる．

通常，分類モデルのパラメータを機械学習する場合，モデルが複雑で表現力が大きいほど，訓練データに対する分類エラーの小さい分類器が得られる反面，モデルが過度に訓練データに適合し過ぎ，未知のデータに対しての精度が下がってしまう過学習という現象が起きる．これに対してSVMは，モデルの複雑さをVC次元(Vapnik-Chervonenkis dimension)と呼ばれる尺度で数学的に定義し，モデルの複雑さと訓練データにおける分類エラーの和を最小にする構造的リスク最小化(structural risk minimization)と呼ばれる概念に基づくことで，過学習を起こさずに最適解を求めることができる．

また，SVMの登場以前の機械学習では，扱うデータの次元数が増加すると学習に必要な計算量や記憶容量が急激に増加するという，いわゆる「次元の呪い」(curse of dimensionality)のため高次元の入力空間を扱うことができないという問題があった．SVMはサポートベクタの部分のみを計算に使用するため，非常に高次元のベクトルを入力として扱えるという点でも従来の学習器より優れている．文献5)では，SVMはRocchio, 決定木, naive Bayes, Bayes netより優れていると報告している．

なお，SVMはブースティングと同じlarge margin classifierの1つであるが，ブースティングではマージンは，全サンプルの分類境界面までの距離コストの和で表現されているのに対し，SVMではサポート・ベクタの通る超平面間の距離となっているところが異なっている．

■研究例: SVMによる日本語テキスト分類■

本章では，サポートベクタマシンを用いた日本語テキスト分類の研究例[18]を紹介する．分類対象には，「RWCPテキストコーパス」を用いた．このコーパスは，1994年版の毎日新聞の約3万件の記事に，国際十進分類法に基づく分類コードを付与したものである．これらの記事の中から10種類の分類カテゴリー(スポーツ，刑法，政府，教育，交通，軍事，国際関連，言語活動，演劇，作物)を持つ訓練データ1,000記事，テストデータ1,000記事を選んだ．

英語テキストと異なり，日本語テキストでは単語が空白で区切られていないため，各記事は日本語形態素解析システム「茶筌」を用いて単語を特定した．文書ベクトルの各要素は各単語の純粋な影響を見るため，単語の出現有無を表す2値とした．クラスラベルは，記事が分類対象カテゴリに属せば+1，属さない場合には−1とする2クラス分類とした．

また過学習の傾向を，従来の代表的な帰納的学習手法である決定木アルゴリズムC4.5[14]と比較するため，素性選択を行って単語数を300から15,000としたデータを作成した．素性選択はテキスト分類の学習において頻繁に使われている，カテゴリーとの相互情報量が高い上位単語を用いる選択を行った．相互情報量は，ある単語の出現頻度が1つのカテゴリーとその他のカテゴリーの間で偏りがあるときに大きな値をとる量であるため，「する」「こと」など出現頻度は高いがどのカテゴリーにも万遍なく出現する単語に関しては小さな値をとり，分類には有効と思われるカテゴリーに特徴的な単語では大きな値をとる．

図-5に素性数を増やしていったときのSVMとC4.5の精度(カテゴリー平均でのF値)を示す．C4.5では300単語のとき精度が最大で70ポイントであるが，単語数が多くなると，急に精度が下がり，過学習が起こっていることが分かる．このようにC4.5のような従来の機械学習では，素性数を増やすと過学習が起きるが，精度を向上させるためには，より多くの素性が必要であるというジレンマがあった．一方，SVMでは300単語の時点で72ポイントですでにC4.5より精度が高く，さらに素性数を増やしていくと過学習が起こることなく精度が単調に高くなり単語数が15,000のとき精度が最高になる．素性選択で落とされたデータの中に分類に寄与する素性が確実に存在しており，SVMではそれを精度を落とすことなく学習できていることが分かる．このよ

うに高次元でスパースなデータを扱わなければならないテキスト分類の学習にSVMを用いることにより，大量の素性から情報を落とさず，高精度の分類を実現することができた．

また，これとは別にブースティングアルゴリズムの1つであるAdaBoostに基づくテキスト分類の実験も行った[7]．そこから得られた経験では，AdaBoostよりSVMの方が例外データのマージンに影響を受けにくい分，雑音耐性があるように感じる．たとえば，「国際関係」カテゴリーのように人手でも分類に迷いやすいテキストを多く含むカテゴリーでは，AdaBoostよりSVMで学習した方が分類器の性能は良いようである．しかし最近，AdaBoostの他にも雑音耐性のあるさまざまなブースティングアルゴリズムが考案されており，SVMとブースティングを単純に比較することはできない．

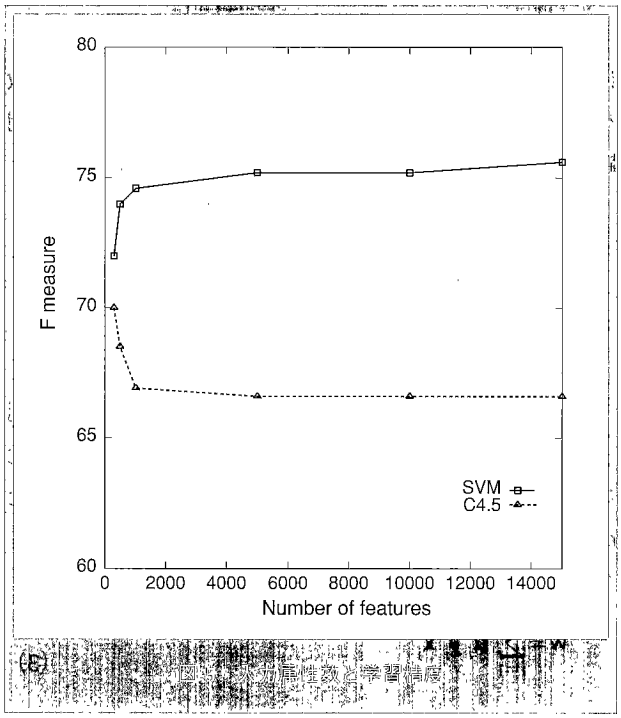


図2 高性能な学習機

■情報抽出・質問応答への応用■

近年では，テキスト分類と同様に機械学習手法を応用した新しい情報技術（IT）がいろいろと登場している．特に，質問応答（question answering）は，Ask Jeeves（http://ask.com）などの自然文入力の情報検索サービスの成功により，大きく注目されている．従来の情報検索は「文書」検索であり，ユーザは検索された文書を読んで必要な情報を取り出さなければならない．これに対して質問応答は，ユーザの質問（情報要求）に対して直接「答え」を提示することが目標である．

以下に，米国のNISTとDARPAが主催するテキスト情報検索のコンテストTREC（Text REtreival Conference）の質問応答トラックの仕様書にある質問と解答の例を示す[17]．

Q: Who was Lincoln's Secretary of State?
A: William Seward
Q: How long does it take to fly from Paris to New York in a Concorde?
A: 3 1/2 hours

非常に大雑把にいえば，「リンカーン大統領の国務長官は誰？」という最初の質問に答えるには，Lincoln, Secretary, Stateを含む文または段落を探して，そこに含まれる人名を抽出すればよい．一般に，人名・地名・組織名のような特定のカテゴリーに属する単語列をテキストから収集することを情報抽出（information extraction）という．単語列を切り出すためのパターン照合規則およびカテゴリーを付与するための分類器の作成にはさまざまな機械学習アルゴリズムを適用できる．

しかし，このような自然言語処理の問題に機械学習アルゴリズムを適用するためには非常に多くのラベル付き訓練データを必要とし，その作成コストが大きな問題になる．そこで，少量のラベル付きデータと大量のラベルなしデータを組み合わせる学習法が研究され始めたが，その中ではPAC学習風の枠組みを持つco-trainingが興味深い[1]．

Co-trainingは，ラベル付きデータを正しく分類でき，かつ，ラベルなしデータの分類結果が矛盾しないような，互いに排他かつ独立な2組の素性集合が存在する場合に適用できる．まず片方の素性集合とラベル付きデータから分類器1を作成し，この分類器1によるラベルなしデータの分類結果の中で信頼性が高いものをラベル付きデータに加える．この少し増えたラベル付きデータともう一方の素性集合から分類器2を作成し，この分類器2によるラベルなしデータの分類結果の中で信頼性が高いものをさらにラベル付きデータに加える．こうして2つの分類器を交互に訓練しながらラベル付きデータ（正確には，分類結果が信頼できるラベルなしデータ）を増やすことにより，分類性能の向上を図る．

たとえば，Webページを分類する場合，あるページのテキスト，および，そのページを指すハイパーリンク上のテキストは，そのページを分類することができる互いに排他かつ独立な2組の手掛かりになる[1]．先の名詞のカテゴリーの分類の場合は，以下の例文のMr. Cooperのように，その文字列の綴り（Mr.）および文字列が出現する文脈（a vice president of）が互いに排他かつ独立な2組の手掛かりになる[2]．

.., says Mr. Cooper, a vice president of ..

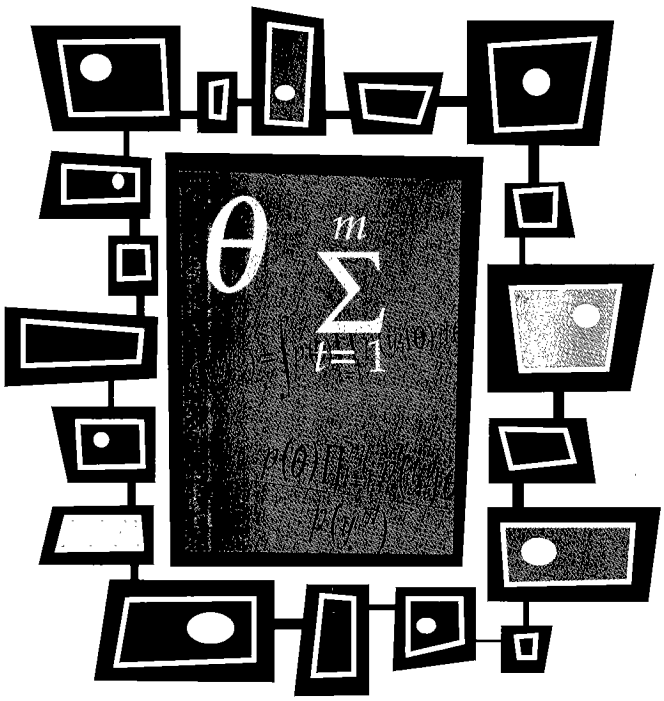
このほか，ラベル付きデータとラベルなしデータを組み合わせて分類器を構成する手法としては，Naive BayesとEMの組合せ[13]，トランスダクティブSVM法[8]，トランスダクティブブースティング法[7]などが提案されており，今後の発展が期待できる．

■究極の目標: ポータルサイトの自動構築■

おそらく多くの人が「分類」という言葉に対して思い浮かべるイメージは，図書館で使われる十進分類法や，Yahoo! などのインターネットディレクトリのような階層構造を持った分類体系だろう．本稿で紹介したテキスト分類は，すべてフラットな複数カテゴリーへの分類だったが，最近では，Web文書をYahoo! のような階層構造を持つカテゴリに分類する研究が行われている[4]．さらに，この階層的カテゴリーへのテキスト分類技術，情報抽出技術，Web文書の収集技術（crawler）を組み合わせて，インターネットのポータルサイトを自動構築するという研究もある[12]．

テキスト分類は，大量の学習データ（数万文書程度）から多数の素性（数千単語程度）を選択して分類器を構成し，2個から数十個程度のクラスに分類することが要求されるため，新しい学習理論の実用性を試す格好の実験場となっている．しかも実用性が高く，ビジネスへの展開も夢ではない．

読者の皆さんの中に，もし分類器の学習アルゴリズムに関する優れたアイディアをお持ちの方がおられれば，ぜひ一度，この分野に挑戦してみてはいかがだろうか？



参考文献

1) Blum, A. and Mitchell, T.: Combining Labeled and Unlabeled Data with Co-Training, Proceedings of the Eleventh Annual Conference on Computational Learning Theory (COLT-98), pp.92-100 (1998).
2) Collins, M. and Singer, Y.: Unsupervised Models for Named Entity Classification, Proceedings of the 1999 Joing SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-99), pp.100-110 (2000).
3) Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K. and Harshman, R.: Indexing by Latent Semantic Analysis, Journal of the American Society for Information Science, Vol.41, No.6, pp.391-407 (1990).
4) Dumais, S. and Chen, H.: Hierachical Classification of Web Content, Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2000), pp.256-263 (2000).
5) Dumais, S., Platt, J., Heckerman, D. and Sahami, M.: Inductive Learning Algorithm and Representation for Text Categorization, Proceeding of the Seventh International Conference on Information and Knowledge Management (CIKM-98), pp.148-155 (1998).
6) Freund, Y. and Schapire, R.E.: A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting, Journal of Computer and System Sciences ,Vol.55, No.1, pp.119-139 (1997).
7) 平　博順, 春野雅彦: トランスダクティブ・ブースティング法によるテキスト分類, 情報処理学会研究報告2000-NL-139, pp.69-76 (2000).
8) Joachims, T.: Transductive Inference for Text Classification using Support Vector Machines, Proceedings of the Sixteenth International Conference on Machine Learning (ICML-99), pp.200-209 (1999).
9) 国分明男, 清水　昇: インターネットにおけるコンテンツ・レイティングとフィルタリング, 情報処理, Vol.40, No.1, pp.57-61 (Jan. 1999).
10) Lewis, D.D.: Naive (Bayes) at Forty: Independence Assumption in Information Retrieval, Proceedings of the 10th European Conference on Machine Learning (ECML-98), pp.4-15 (1998).
11) Li, H. and Yamanashi, K.: Text Classification using ESC-based Stochastic Decision Lists, Proceedings of the Eighth International Conference on Information and Knowledge Management (CIKM-99), pp.122-130 (1999).
12) McCallum, A.K., Nigam, K., Rennie, J. and Seymore, K.: Automating the Construction of Internet Portals with Machine Learning, Information Retrieval ,Vol.3, No.2, pp.127-163 (2000).
13) Nigam, K., McCallum, A.K., Thrun, S. and Mitchell, T.: Text Classification from Labeled and Unlabeled Documents using EM, Machine Learning ,Vol.39, No.2/3, pp.103-134 (2000).
14) Quinlan, J.: C4.5: Programs for Machine Learning, Morgan Kaufmann (1993).
15) Schapire, R.E. and Singer, Y.: BoosTexter: A Boosting-based System for Text Categorization, Machine Learning ,Vol.39, No.2/3, pp.135-68 (2000).
16) Sebastiani, F.: Machine Learning in Automated Text Categorisation, Submitted for publication to ACM Computing Survey (2000). http://faure.iei.pi.cnr.it/~fabrizio/Publications/ACMCS00/ACMCS00.pdf
17) Singhal, A.: 1999 TREC-8 Question Answering Track (1999). http://www.research.att.com/singhal/qatrack.html.
18) Taira, H. and Haruno, M.: Feature Selection in SVM Text Categorization, Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99), pp.480-486 (1999).
19) Vapnik, V.N.: The Nature of Statistical Learning Theory, Springer-Verlag (1995).
20) Yang, Y. and Pedersen, J.O.: A Comparative Study on Feature Selection in Text Categorization, Proceedings of the Fourteenth International Conference on Machine Learning (ICML-97), pp.412-420 (1997).

(平成12年11月30日受付)