自然言語処理 139−10
(2000. 9. 22)

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

平 博順[†] 春野 雅彦[††]

[†] 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
taira@cslab.kecl.ntt.co.jp
[††] ATR 人間情報通信研究所
〒619-0288 京都府相楽郡精華町光台 2-2
mharuno@hip.atr.co.jp

概要: 本稿では，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手法を提案する．テキスト分類法の学習で使用する大規模な訓練データの作成にはコストや時間がかかる．そのため訓練データが少ない場合でも高い分類精度が得られる学習法が求められている．トランスダクティブ法は訓練データだけでなく，分類クラスの付与されていないテストデータの分布も学習の考慮に入れることにより分類精度を上げる方法である．本稿ではこれをブースティングに対し適用し，実験を行なった．その結果，従来のブースティングによる学習に比べて高精度のテキスト分類器を学習できることが確認された．特に少数の訓練データしかない場合にも高い精度が得られた．

Text Categorization Using a Transductive Boosting Method

Hirotoshi Taira[†] Masahiko Haruno[††]
[†] NTT Communication Science Laboratories
2-4, Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0237 Japan
taira@cslab.kecl.ntt.co.jp
[††] ATR Human Information Processing Research Laboratories
2-2, Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0288 Japan
mharuno@hip.atr.co.jp

Abstract: This paper describes a new text categorization method using transductive boosting. All learning-based classification methods share on one common problem: It is not possible to have a large corpus of categorized text. This is because it is difficult to make a corpus easily and cheaply. It is therefore important that the learner is able to make efficient generalizations using a small amount of training data. We tackle here the problem of learning from small training samples by taking a transductive approach, instead of an inductive approach. We adopt a transductive method in a boosting algorithm for the task of text categorization. The categorization performance was improved significantly.

1 はじめに

インターネットの発達，コンピュータ環境の充実とともに，一般の人でも大量のオンライン情報にアクセスできるようになってきた．しかし，アクセスできる情報が大量になればなるほど，その中から必要かつ十分な情報を的確に得ることが困難になってきている．情報を的確に得るための技術の一つとしてテキスト自動分類技術が注目されている．特に機械学習手法を用いて分類器を構成する方法は分類対象テキストが大規模であったり，頻繁に更新されるような場合にも比較的容易に高水準の分類精度が得られるため，近年主流になりつつある．

これまで機械学習により分類器を構成する際には，一定数の訓練データを学習して得られた分類器で分類クラスが未知のテストデータを分類するという帰納的学習がとられてきた．この枠組みでこれまでk-最近傍法 [18]，Rocchio法 [12] [5]，決定木 [8]，Naive-Bayes [8]，SVM [6] [20] など様々な手法がテキスト分類に適用されてきた．しかしながら，帰納的学習を用いた分類では訓練データが少ない場合，訓練データとテストデータの分布の違いが大きく，十分な分類精度が得られなかった．実際にオンライン情報などのテキストを分類しようとした場合，高精度の分類器を構成するのに十分な量の訓練データが得られない場合が多い．これはデータを人手で分類し分類クラスを付与することは非常に労力がかかるためである．そこで訓練データが少ない場合でも高精度の分類器を生成する手法が期待される．

訓練データが少ない場合にテストデータの分布も考慮して分類の学習を行う方法として，NigamらがEMアルゴリズムとNaive-Bayesを組み合わせた方法を提案している [11]．また，JoachimsはSupport Vector Machine(SVM) に対してトランスダクティブ法を適用し高精度の分類結果を得ている [7]．通常の帰納的学習が全体のデータの分布に対して分類誤りを最小化するような学習であるのに対し，トランスダクティブ法は，与えられたテストデータの分布に注目しテストデータの分類誤りを最小化する学習方法である [17]．

Joachimsの用いたSVMは汎化能力が高いことで最近注目を浴びているLarge Margin Classifierと呼ばれる分類学習法の一つである．同じLarge Margin Classifierの一つにブースティング [13] [2] [4] がある．ブースティングは分類精度の低い分類器 (弱分類器と呼ぶ) を組み合わせることで高精度の分類器 (強分類器と呼ぶ) を得る方法である．弱分類器には，50% 以上の精度を持つ様々な分類器を使用できる．例えば，SVMによる学習では，明示的にルールが導出できないという問題があるのに対し，ブースティングではルールが生成できる弱学習器を用意すれば高精度のルールも学習できる．しかしながら，従来のブースティングでは，他の帰納的学習法と同様，訓練データが少ない場合には十分な分類精度が得られない．そこでブースティングにトランスダクティブ法を適用し訓練データの少ない場合にも高精度の分類器を構成することを試みた．

ところで，ブースティングはもともと，繰り返しアルゴリズムを伴うため，単純にJoachimsがSVMに対してとったようなトランスダクティブ法は使用できない．しかし，最近，ブースティングが関数空間の中でコスト関数曲面の最急降下方向に弱分類器を選ぶアルゴリズムとして解釈できることが明らかになってきている [9]．そこでブースティングをコスト関数の最小化として解釈し，トランスダクティブ法の適用を試みた．

これまでブースティングを使ったテキスト分類の例にBoosTexter [15] がある．これは深さ1の決定木を一つの弱分類器とし，ブースティングとしてAdaBoostと呼ばれるアルゴリズムで学習を行うテキスト分類器であり，Naive-BayesやRocchio法を使った分類を上回る精度を上げている．この方法にトランスダクティブ法を適用し比較実験を行った．

本論文の構成は以下の通りである．次章でブースティングとその汎化誤差について述べる．3章でブースティングが関数空間での最急降下法に相当することを示し，ブースティングにトランスダクティブ法を適用する方法を説明する．またテキスト分類への適用についても述べる．4章で実験結果についての考察を行い，最終章で結論を述べる．

2 ブースティング

2.1 AdaBoost アルゴリズム

ブースティングはSchapireによって最初のアルゴリズム [13] が発見された後，FreundとSchapireらによってAdaBoostと呼ばれるアルゴリズム [4] が提案され，実用的にも注目されるようになった．AdaBoostアルゴリズムを以下に示す．

(手順1) m 個の訓練データ $(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_m, y_m)$ が入力として与えられる．ここで $\boldsymbol{x}_1, ..., \boldsymbol{x}_m$ は特徴ベクトル，$y_1, ..., y_m$ は各々 $\boldsymbol{x}_1, ..., \boldsymbol{x}_m$ に対する分類クラスで，正例の時 $+1$，負例の時 -1 とする．

(手順2) 各訓練データに対する重みの初期値として $D_1(i) = \frac{1}{m}$ を与える．ただし $i = 1, ..., m$ とする．

(手順3) 各ラウンド $t = 1, ..., T$ に対し，以下の(手順4)-(手順7)を繰り返す．

(手順4) 重み D_t にしたがって訓練データを学習し，$\boldsymbol{x} = \boldsymbol{x}_i$ に対して正例と判定するときは $+1$，負例と判定するときは -1 を出力する弱分類器 $h_t(\boldsymbol{x})$ を得る．

(手順5) 重み付き誤分類率 $\epsilon_t = \sum_{i:h_t(\boldsymbol{x}_i) \neq y_i} D_t(i)$ を計算する．

(手順6) パラメータ $\alpha_t = \frac{1}{2} \ln(\frac{1-\epsilon_t}{\epsilon_t})$ を計算する．

(手順7) 次式によって各訓練データの重みを更新する．

$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(\boldsymbol{x}_i))}{Z_t}$$

ここで Z_t は $\sum_{i=1}^{m} D_{t+1}$ を1とするための正規化定数である．

(手順8) 最後に以下の線形和で最終的な分類器(強分類器)を得る．

$$H(\boldsymbol{x}) = \sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x}) \quad .$$

以上のようにAdaBoostは各ラウンドで1つずつ弱分類器を学習・生成するとともに，訓練データに対する重みの更新を行う．(手順7)を見て分かるように重み $D_t(i)$ はデータ i が弱分類器によって正しく分類された場合(つまり $h_t(\boldsymbol{x}_i) = y_i$ のとき)には $\exp(-\alpha_t)$，間違って学習された場合(つまり $h_t(\boldsymbol{x}_i) \neq y_i$ のとき)には $\exp(\alpha_t)$ が乗ぜられる．分類誤り率 ϵ_t が50%未満の時，(手順6)よりパラメータ α_t は正値をとる．誤って学習されたデータの重みには1より大きな数が乗ぜられ，次ラウンドの弱分類器の学習ではこのデータに重点を置いて学習することになる．最後に(手順8)でパラメータ α_t を重みとした弱分類器の線形和をとり，最終的な分類器(強分類器) $H(\boldsymbol{x})$ を得る．

2.2 汎化誤差

Schapireらはマージンの概念を導入しAdaBoostの汎化誤差(ラベルの無い未知のテストデータに対する分類誤差)の解析を行っている [14]．ブースティングにおける訓練データに対するマージンを $y \sum_t \alpha_t h_t(\boldsymbol{x}) / \sum_t \alpha_t$ とする．$\sum_t \alpha_t = 1$ となるように正規化するとマージンは $yH(\boldsymbol{x})$ となる．マージンについて以下の定理が成り立つことが証明されている．

定理 1 (Schapire) D を $X \times \{-1, +1\}$ 上の分布，S を D とは独立に無作為に選ばれた m 個の訓練事例とする．また，仮説空間 H が VC 次元 d を持ち $\delta > 0$ であるとする．$m \geq d \geq 1$ を仮定すると訓練事例集合 S 上で無作為に事例を選択したときすべての $\theta > 0$ に対して少なくとも $1 - \delta$ の確率で次式が成立する．

$$\begin{aligned} & P_D[yH(\boldsymbol{x}) \leq 0] \\ \leq \ & P_S[yH(\boldsymbol{x}) \leq \theta] \\ + \ & O(\frac{1}{\sqrt{m}}(\frac{d \log^2(m/d)}{\theta^2} + \log(\frac{1}{\delta}))^{1/2}) \end{aligned}$$

ここで $P_D[A]$ は事例 $(\boldsymbol{x}, y)$ が分布 D にしたがって選択されたときの A の確率を表す．上式の上限はブースティングのラウンド数 T とは独立であり，θ を大きくとれれば(すなわちマージンを大きくできれば)汎化誤差が小さくなることが分かる．また $O(\frac{1}{\sqrt{m}})$ のオーダで訓練事例数 m が大きいほど汎化誤差が小さくなることが分かる．

3 トランスダクティブ法とテキスト分類

3.1 TSVM でのトランスダクティブ法

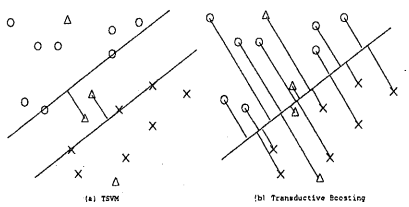

図 1: (a)TSVM と (b) トランスダクティブ・ブースティング法.

図 1にトランスダクティブSVM(TSVM) とトランスダクティブ・ブースティング法の概念図を示す. ○印は正例の訓練データ, ×印は負例の訓練データ, △印は分類クラスが付与されていないテストデータを示す. TSVM ではテストデータも含めて最も負例側よりの正例と最も正例側の負例のデータに注目し, 分離超平面が構成される. TSVM では二つの分離超平面の間の距離をマージンと呼び, 一定割合の分類誤りを許しつつ, マージンを最大化するように分離超平面が選ばれる. Joachims が SVM に対してトランスダクティブ法を適用した際には, 訓練データだけで SVM により分類器を構成し, その分類器によりすべてのテストデータを判別し, 仮の分類クラスを与えている. そして仮のクラスの付与されたテストデータも含めて SVM による学習を行う. その後, 図にあるように仮のクラスとして負例が与えられたテストデータと正例が与えられたテストデータについて, そのクラスを入れ替えた方が分類誤りを減らせる組を見つけ入れ替え, 再度 SVM による学習を行う. 入れ替えるテストデータの組が無くなるまで, クラスの入れ替えと学習を繰り返すことでテストデータの分布に合った分類超平面を得る.

一方ブースティングにおけるマージンは SVM とは異なり, 図の右のように, すべての各訓練データの分類境界面までの距離である. ブースティングではこれらすべてのマージンの平均を大きくすることを目的とする. ブースティングでは弱分類器を線形結合して最終的な強分類器を得るため, 初期のラウンドで生成される分類器を線形結合して得られた強分類器では十分な分類精度が得られず, TSVM のような分類クラスを入れ替える方法はとりにくい.

3.2 最急降下法によるブースティングの解釈

最近, ブースティングが関数空間の中でコスト関数曲面の最急降下方向に弱分類器を選ぶアルゴリズムとして解釈できることが明らかになってきている. この枠組みの中では AdaBoost アルゴリズムは MarginBoost と呼ばれる抽象化されたブースティングアルゴリズムのうちコスト関数が $Cost(H) = \exp(-H)$ のタイプのアルゴリズムであることが分かっている. ここで H は強分類器を表す. あるラウンド t において次ラウンドの弱分類器 h_{t+1} を得ることは, 関数空間の中で

$$\sum_{i=1}^{m} Cost(y_i H_t(\boldsymbol{x}_i) + y_i \alpha_{t+1} h_{t+1}(\boldsymbol{x}_i))$$

を最小化するような h_{t+1}, α_{t+1} を求めることに相当する. このとき, AdaBoost アルゴリズムにおける具体的なコスト関数は

$$Cost(H(\boldsymbol{x})) = \frac{1}{m} \sum_{i=1}^{m} \exp(-y_i H(\boldsymbol{x}_i))$$

と表される. これはマージンを指数関数の尺度で平均をとったものに相当する.

3.3 トランスダクティブ・ブースティング法

前節で述べたコスト関数についてトランスダクティブ法の枠組みの中での最小化を考える. $\boldsymbol{x}_{m+1}, ..., \boldsymbol{x}_{m+n}$ の n 個のテストデータも含めたコスト関数は,

$$Cost(H(\boldsymbol{x})) = \frac{1}{m+n} \{ \sum_{i=1}^{m} \exp(-y_i H(\boldsymbol{x}_i)) + \sum_{j=m+1}^{m+n} \exp(-y_j^* H(\boldsymbol{x}_j)) \}$$

と表される. ここで y_j^* はテストデータ $\boldsymbol{x}_j$ に対する仮の分類クラスである. y_j^* は未知であり, すべてのテストデータ y_j^* の初期値を 0 としておく. 最

終的にすべてのテストデータ y_j^* に対し +1(正例) か −1(負例) のいずれかの値を正しく付与するのが目的である．TSVM と異なりブースティングでは初期のラウンドで生成される弱分類器を線形結合して構成した強分類器では学習が十分に進んでいないために分類精度が悪い．すべてのテストデータにこの強分類器による評価値を仮の分類クラスとして付与すると，高い割合で誤った分類クラスが付与されてしまう．誤った分類クラスが付与されているテストデータを多く用いて，ブースティングを行うと，誤った最急降下方向が得られ，強分類器の精度が悪くなる．そのため，仮の分類クラスの付与は高い精度で行わなければならない．そこで，各ラウンドでは分類クラスの評価が最も信頼できる 1 個のテストデータについて分類クラスの付与を行う．また，正例と負例の比は訓練データ中の正例と負例の比と同じであると仮定し，分類クラス付与を行う．2.1 節の AdaBoost のアルゴリズムの中の (手順 2)，(手順 7) の後にそれぞれ以下のような (手順 2b)，(手順 7b) を追加する．

(手順 2b) 入力として n 個のテストデータ $(\boldsymbol{x}_{m+1}, y^*_{m+1}), ..., (\boldsymbol{x}_{m+n}, y^*_{m+n})$ が与えられる．ここで $\boldsymbol{x}_{m+1}, ..., \boldsymbol{x}_{m+n}$ は特徴ベクトル，$y^*_{m+1}, ..., y^*_{m+n}$ は各々 $\boldsymbol{x}_{m+1}, ..., \boldsymbol{x}_{m+n}$ に対する仮の分類クラスで，初期値として 0 を与える．各訓練データに対する重みの初期値として $D_1(j) = 0 \quad (j = m+1, ..., m+n)$ を与える．

(手順 7b) m^+ を訓練データ中の正例の数，$n_{labeled}$ を既に分類クラスが付与されているテストデータ数，$n^+_{labeled}$ を分類クラスとして正例が付与されたテストデータ数とするとき，

(i) $m^+/m \geq n^+_{labeled}/n_{labeled}$ のとき，
$y_j^* = 0$ であるテストデータの中で

$$H(\boldsymbol{x}_j) = \sum_{t=1}^{t} \alpha_t h_t(\boldsymbol{x}_j)$$

が最大値をとるテストデータ j に対して $y_j^* = +1$ および $D_{t+1}(j) = \epsilon$ (実験では $\epsilon = 0.01$ とした) を与える．

(ii) $m^+/m < n^+_{labeled}/n_{labeled}$ のとき，
$y_j^* = 0$ であるテストデータの中で

$$H(\boldsymbol{x}_j) = \sum_{t=1}^{t} \alpha_t h_t(\boldsymbol{x}_j)$$

が最小値をとるテストデータ j に対して $y_j^* = -1$ および $D_{t+1}(j) = \epsilon$ を与える．

このような手順をとることでコスト関数中の $\exp(-y_j^* H(\boldsymbol{x}_j))$ の項において，y_j^* の誤っている確率が小さく，かつ $y_j^* H(\boldsymbol{x}_j)$ の値がそのラウンドにおいて最大であるデータを選択でき，コスト関数のとりうる値の中で最小値をとる可能性の高い分類器を生成することができる．

表 1: カテゴリ毎のデータ数

カテゴリ名	訓練データ	テストデータ
スポーツ	146	162
刑法	138	166
政府	129	148
教育	101	133
交通	97	118
軍事	96	132
国際関連	92	101
言語活動	92	67
演劇	90	91
作物	77	73

3.4 テキスト分類への適用

正例と負例の 2 つのクラスに属す l 個の訓練データのベクトルの集合を，

$$(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_l, y_l), \ \boldsymbol{x}_i \in R^n, \ y_i \in \{-1, +1\}$$

とする．ここで，$\boldsymbol{x}_i$ はデータ i の特徴ベクトルで，y_i はデータ i の分類クラスである．テキスト分類問題では，テキストの特徴をテキスト中に出現する単語で代表させ，単語 w_i がテキスト中に出現する場合，$w_i = 1$，出現しない場合を，$w_i = 0$ として 1 つのテキストをベクトル $\boldsymbol{x}_i = (w_1, w_2, ..., w_n)$ で表す．テキストがあるカテゴリに含まれる場合を正例 ($y_i = +1$)，含まれない場合を負例 ($y_i = -1$) として，各カテゴリに対して，分類器を構成する．

4 実験結果

4.1 実験設定

実験には，RWCP テキストコーパス [21] 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 UDC コード [19] を付与したものである．これらの記事の中から頻度の高い 10 種類の分類カテゴリ (ス

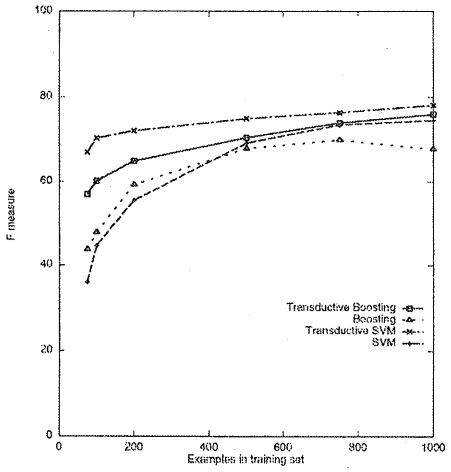

図 2: 訓練データ数と学習精度 (F 値).

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) をもつ訓練データ 1000 記事，テストデータ 1000 記事を選んだ．各カテゴリの訓練データ数，テストデータ数を表 1に示す． これらの記事に対して形態素解析システム ChaSen [10] により形態素解析を行なった後，相互情報量の高い上位 1000 単語を抜き出して特徴ベクトルとした．

4.2 評価方法

分類精度の評価には，F 値 [16] を用いた．各分類毎に，a =(正解が正例で分類器も正例と判別したデータ数)，b =(正解が負例で分類器は正例と判別したデータ数)，c =(正解が正例で分類器は負例と判別したデータ数) を考えると，適合率 (P)，再現率 (R) は，

$$P = \frac{a}{a+b}, \quad R = \frac{a}{a+c}$$

と定義される．F 値は適合率と再現率とを組み合わせた評価値であり，

$$F = \frac{1+\beta^2}{\frac{1}{P}+\beta^2\frac{1}{R}}$$

で表される．F 値は 0 から 1 の値をとり，大きな値ほど分類精度が高い．ここで，β は重みづけパラメータで今回は $\beta = 1$ とした．

4.3 訓練データ数と学習精度

深さ 1 の決定木を弱分類器とする AdaBoost (つまり BoosTexter と同じ) アルゴリズムによるトランスダクティブ法でテキスト分類実験を行った．比較のため従来のトランスダクティブ法を使わない AdaBoost，SVM，TSVM による実験も行った．まず，分類クラスがあらかじめ付与されている訓練データを 75 個から 1000 個まで増やし，1000 個のテストデータの分類を実験を行った．10 カテゴリで平均した結果 (F 値) を図 2に示す．

トランスダクティブ法により 1000 個のテストデータの分布も考慮することで，大きく精度が上がっている．特に訓練データの少ない場合に精度の向上が顕著で，訓練データが 75 個のときには平均で 13.1 ポイント上昇している．カテゴリ別の詳細な結果を表 2，表 3に示す．スポーツカテゴリの訓練データが 75 個，1000 個の場合のようにトランスダクティブ法を使わなくても，もともと精度が高かったカテゴリについてはかえって精度が下がることがある．しかし交通，軍事，国際関連，言語活動カテゴリのようにかなり精度が低かったカテゴリがトランスダクティブ法を使うことで大幅に精度が向上していることが分かる．

SVM と TSVM によるテキスト分類ではデータは同じものを用い，カーネル関数は線形関数を用いて実験を行った．F 値で比較するとトランスダクティブ・ブースティング法による分類は TSVM よりやや劣るが，SVM を上回る結果が得られている．

次に訓練データを 100 個に固定し，トランスダクティブ法に使用するテストデータの個数を増やしていった実験の結果 (10 カテゴリの F 値の平均) を図 3に示す．テストデータが増えるにしたがってほぼ単調に精度が上がっていくことが分かる．このようにブースティングに対してトランスダクティブ法が効果があることが分かる．

TSVM に比べてトランスダクティブ・ブースティング法による分類精度が劣るのは SVM は一定の分類誤りを許しつつマージンを最大化するのに対し，ブースティングでは分類誤りを起こしたデータに重点をおいて学習を行うために，例外的なテストデータが多い場合には精度が下がるためであることが考えられる．実際，AdaBoost アル

表 2: 訓練データ数による影響 (F 値)(Transductive Boosting)

カテゴリ名 \ 訓練データ数	75	100	200	500	750	1000
スポーツ	0.642	0.726	0.766	0.875	0.901	*0.903*
刑法	0.600	0.571	0.663	0.656	0.743	*0.750*
政府	0.723	0.560	0.622	0.689	*0.727*	0.722
教育	0.459	0.624	0.661	0.675	0.762	*0.778*
交通	0.495	0.493	0.500	0.638	0.680	*0.698*
軍事	0.507	0.561	0.688	0.748	0.754	*0.781*
国際関連	0.429	0.396	0.363	0.558	0.508	*0.560*
言語活動	0.493	0.523	0.641	0.612	*0.703*	0.692
演劇	0.583	0.749	0.756	0.795	0.857	*0.862*
作物	0.750	0.817	0.831	0.805	0.761	*0.853*
平均	0.569	0.602	0.649	0.705	0.740	*0.760*

表 3: 訓練データ数による影響 (F 値)(Boosting)

カテゴリ名 \ 訓練データ数	75	100	200	500	750	1000
スポーツ	0.675	0.681	0.826	0.867	0.891	*0.912*
刑法	0.561	0.402	0.649	0.664	0.681	*0.723*
政府	0.607	0.524	0.580	0.683	*0.692*	0.670
教育	0.287	0.525	0.563	0.646	0.667	*0.714*
交通	0.514	0.510	0.493	0.647	*0.658*	0.579
軍事	0.216	0.321	0.550	*0.728*	0.686	0.628
国際関連	0.324	0.317	0.233	0.428	0.490	*0.329*
言語活動	0.119	0.220	0.528	*0.576*	0.561	0.559
演劇	0.385	0.645	0.767	0.693	0.800	*0.813*
作物	0.690	0.643	0.734	0.855	*0.864*	0.850
平均	0.438	0.479	0.592	0.679	*0.699*	0.678

ゴリズムでは例外的なデータが多い場合には正しい分類が困難なデータに学習の重点を置いてしまい，分類精度が大きく下がることがあることが指摘されている [1]．今後，さらにトランスダクティブ・ブースティング法の分類精度を高める方法として BrownBoost [3] のような，例外的なデータの重みを減らすブースティングアルゴリズムの使用も検討したい．

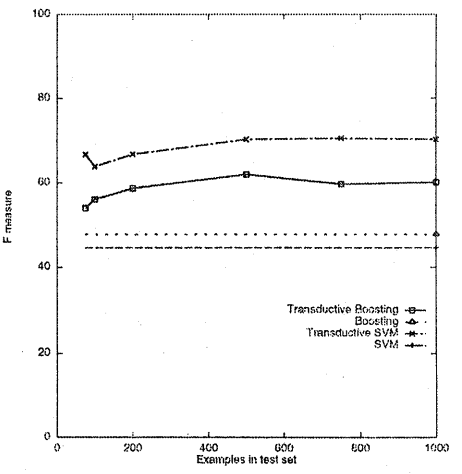

図 3: ラベルなしデータの数と学習精度 (F 値).

5 結論

トランスダクティブ・ブースティング法をテキスト分類問題に適用し，訓練データ数を変えていく実験を行い，トランスダクティブ法を行わないAdaBoost，SVM，TSVMと比較した．その結果，トランスダクティブ法を適用することで大幅な分類精度向上が見られ，ブースティングを使ったテキスト分類問題に対してもトランスダクティブ法が有効であることが明らかになった．

謝辞

毎日新聞94年版の使用に関して，記事データの研究利用許諾を頂いた毎日新聞社に感謝致します．

参考文献

[1] Dietterich, T. G.: An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization, *Machine Learning*, Vol. 40, No. 2, pp. 139–157 (2000).

[2] Freund, Y.: Boosting a Weak Learning Algorithm by Majority, *Information and Computation*, Vol. 121, No. 2, pp. 256–285 (1995).

[3] Freund, Y.: An Adaptive Version of the Boost by Majority Algorithm, *Proc. of the Twelfth Annual Conference on Computational Learning Theory* (1999).

[4] Freund, Y. and Schapire, R.: 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).

[5] Ittner, D. J., Lewis, D. D. and Ahn, D. D.: Text Categorization of Low Quality Images, *Proc. of Symposium on Document Analysis and Information Retrieval*, pp. 301–315 (1995).

[6] Joachims, T.: Text Categorization with Support Vector Machines, *Proc. of European Conference on Machine Learning(ECML)* (1998).

[7] Joachims, T.: Transductive Inference for Text Classification using Support Vector Machines, *Proc. of the 16th International Conference on Machine Learning (ICML'99)* (1999).

[8] Lewis, D. and Ringuette, M.: A Comparison of Two Learning Algorithms for Text Categorization, *Proc. of Third Annual Symposium on Document Analysis and Information Retrieval*, pp. 81–93 (1994).

[9] Mason, L., Baxter, J., Bartlett, P. and Frean, M.: Boosting Algorithms as Gradient Descent, *Proc. of Neural Information Processing Systems 1999 (NIPS-99)* (1999).

[10] Matsumoto, Y., Kitauchi, A., Yamashita, T., Hirano, Y., Imaichi, O. and Imamura, T.: *Japanese Morphological Analysis System Chasen Manual* (1997). NAIST Technical Report NAIST-IS-TR97007.

[11] Nigam, K., McCallum, A., Thrun, S. and Mitchell, T.: Text Classification from Labeled and Unlabeled Documents using EM, *Machine Learning*, Vol. 39, pp. 103–134 (2000).

[12] Salton(Ed.), G.: *The Smart Retrieval System-experiments in Automatic Document Processing*, Prentice-Hall (1971).

[13] Schapire, R. E.: The Strength of Weak Learnability, *Machine Learning*, Vol. 5, No. 2, pp. 197–227 (1990).

[14] Schapire, R. E., Freund, Y., Bartlett, P. and Lee, W. S.: Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods, *The Annals of Statistics*, Vol. 26, No. 5, pp. 1651–1686 (1998).

[15] Schapire, R. E. and Singer, Y.: BoosTexter: A Boosting-Based System for Text Categorization, *Machine Learning*, Vol. 39, pp. 135–168 (2000).

[16] Sundheim, B. M.: Overview of the Fourth Message Understanding Evaluation and Conference, *Proc. of Fourth Message Understanding Conference*, pp. 3 – 29 (1992).

[17] Vapnik, V.: *Statistical Learning Theory*, John Wiley & Sons (1998).

[18] Yang, Y.: Expert Network: Effective and Efficient Learning from Human Decisions in Text Categorization and Retrieval, *Proc. of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 13–22 (1994).

[19] 情報科学技術協会: 国際十進分類法，日本語中間版第3版, 丸善 (1994).

[20] 平博順, 春野雅彦: Support Vector Machine によるテキスト分類における属性選択, 情報処理学会論文誌, Vol. 41, No. 4, pp. 1113–1123 (2000).

[21] 豊浦潤, 徳永健伸, 井佐原均, 岡隆一: RWC における分類コード付きテキストデータベースの開発, 電子情報通信学会研究報告 NLC96-13, pp. 27 – 32 (1996).