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[††]

本論文では，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．テキスト分類器の学習に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に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学習の際に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．本論文ではこれをブースティングに対し適用し，実験を行った．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[†] and Masahiko Haruno[††]

This paper describes a new text categorization method using transductive boosting. It is time-consuming and expensive to assemble a large corpus of categorized text for use with learning-based classification methods. Therefore, we require learning methods that are able to learn classifiers extremely accurately from a small quantity of training data. The transductive method takes account of both training data and test data distribution and provides a highly accurate classifier. We adopt a transductive method in a boosting algorithm for text categorization. The categorization performance was better than that of the original boosting. Specifically the performance was improved significantly for small quantities of training data.

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 1 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，頻繁に更新されたりす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[20]，Rocchio 法[7),13)]，決定木[10]，Naive-Bayes[10]，SVM[8),18),23)]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[12]．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[9]．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[19]．

Joachims の用いた SVM は汎化能力が高いことで

† NTT コミュニケーション科学基礎研究所
NTT Communication Science Laboratories
†† ATR 人間情報科学研究所
ATR Human Information Science Laboratories

最近注目を浴びている Large Margin Classifier と呼ばれる分類学習法の 1 つである．同じ Large Margin Classifier の 1 つにブースティング[3),5),6),14)]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．分類対象に応じて，弱学習器を選択すればブースティングにより，その分類対象に対して高精度の分類を行う学習ができる．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[16)]がある．これは深さ 1 の決定木を 1 つの弱分類器とし，ブースティングの手法として 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．しかしながら，他の帰納的学習法と同様，訓練データが少ない場合に十分な分類精度が得られない．そこで SVM と同様にブースティングにトランスダクティブ法を適用することが考えられるが，ブースティングはもともと，繰返しアルゴリズムをともなうため，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ず，これまでブースティングに対しトランスダクティブ法を適用した例はなかった．

ところで，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最近明らかになってきている[1),11)]．我々はブースティングをコスト関数の最小化として解釈し，ブースティングに適したトランスダクティブ法を提案する．この方法を用いてテキスト分類実験を行い，従来方法との比較を行った．

本論文の構成は以下のとおりである．次章で従来法について述べ，3 章で，提案手法であるトランスダクティブ・ブースティング法について述べる．またテキスト分類への適用についても述べる．4 章では実験結果を示し考察を行う．最終章で結論を述べる．

2. 従 来 法

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[14)]が提案された後，Freund と Schapire らによって AdaBoost と呼ばれるアルゴリズム[5)]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 $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)$ が入力として与えられる．ここで $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$ は特徴ベクトル，$y_1, \ldots, y_m$ は各々 $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$ に対する分類クラスで，正例のとき $+1$，負例のとき -1 とする．また，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 $i = 1, \ldots, m$ とする．

（手順 2） 各ラウンド $t = 1, \ldots, T$ に対し，以下の（手順 3）〜（手順 5）を繰り返す．

（手順 3） 重み D_t に従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 4） パラメータ α_t を $\alpha_t = \frac{1}{2}\ln(\frac{1-\epsilon_t}{\epsilon_t})$ によって計算する．ここで，ϵ_t は重み付き誤分類率で $\epsilon_t = \sum_{i:h_t(\boldsymbol{x}_i)\neq y_i} D_t(i)$ で計算される．

（手順 5） 各訓練データの重みを次式によって更新する．

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

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

（手順 6） 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 5）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%未満のとき，（手順 4）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 6）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．

Schapire らはマージンの概念を導入し AdaBoost の汎化誤差（ラベルのな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[15)]．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．大きいマージンをとるデータの数を多くすることができれば，汎化誤差が小さくなることが証明されている[15)]．

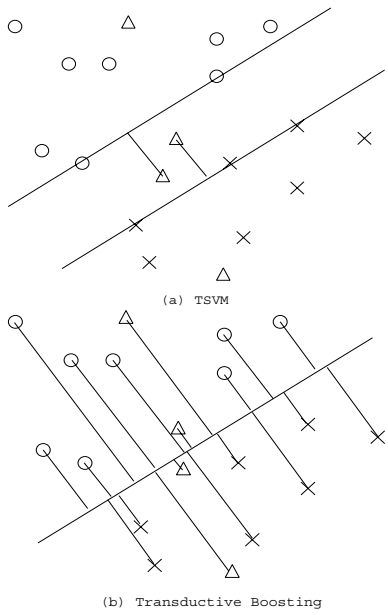

図 1 (a) TSVM と (b) トランスダクティブ・ブースティング法
Fig. 1 TSVM (a) and Transductive Boosting (b).

2.2 TSVM でのトランスダクティブ法

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 2 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なくなるまで，クラスの入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，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大きいマージンをとるデータの数を多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．

2.3 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[1),11)]．

弱学習器 $h : X \to \{+1, -1\}$（ここで X は特徴ベクトル空間を表す）のある族 $\mathcal{H}$ を考える．族 $\mathcal{H}$ に含まれる関数の線形結合すべての集合を $lin(\mathcal{H})$ と書く．$F, G \in lin(\mathcal{H})$ なるすべての F，G に対し内積を $< F, G > \stackrel{\text{def}}{=} \frac{1}{m}\sum_{i=1}^{m} F(\boldsymbol{x}_i)G(\boldsymbol{x}_i)$ と定義する．ここで，$\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$ は m 個の訓練データの各特徴ベクトルを示す．この内積により内積空間 $(\mathcal{X}, <, >)$ を定義する．ここで，$\mathcal{X}$ は $lin(\mathcal{H})$ を含む線形空間，$<, >$ は内積を示す．今，$H \in lin(\mathcal{H})$ なる関数を仮定しこの H に新たに関数 $h \in lin(\mathcal{H})$ を加えコスト $Cost(H + \epsilon h)$ を減らすことを考える．ここで ϵ はある小さい値とする．H におけるコスト関数の微分を

$$\nabla Cost(H)(\boldsymbol{x}) \stackrel{\text{def}}{=} \left.\frac{\partial Cost(H + \alpha 1_x)}{\partial \alpha}\right|_{\alpha=0}$$

と定義する．ここで $1_{\boldsymbol{x}}$ は $\boldsymbol{x}$ のインディケータ関数，すなわち $\boldsymbol{x}$ が $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$ のいずれかに一致するとき 1，そうでないとき 0 を出力する関数である．コスト関数を ϵ の一次のオーダで展開すると

$$Cost(H + \epsilon h) = Cost(H) + \epsilon < \nabla Cost(H), h >$$

と書ける．ここで関数空間の中での最急降下法を行う．すなわち，h を $- < \nabla Cost(H), h >$ を最大にするように選ぶことにより，コスト関数の値を小さくする．具体的にはあるラウンド t において次ラウンドの弱分類器 h_{t+1} を得るために，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 $\exp(-M)$ のタイプのアルゴリズムであることが Mason らによって示されている．ここで M はマージンを表す．具体的には AdaBoost アルゴリズムのコスト関数は

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

と表される．マージンを指数関数の尺度で平均をとっているため，サンプルについて分類予測の誤りをより大きなコストとして評価するアルゴリズムになっている．このコスト関数の値を小さくすることは大きいマージンをとるデータの数を多くし，汎化誤差を小さくすることに相当する．

3. トランスダクティブ・ブースティング法を用いたテキスト分類

3.1 トランスダクティブ・ブースティング法

前章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．$\boldsymbol{x}_{m+1},\ldots,\boldsymbol{x}_{m+n}$ の n 個のテストデータも含めたコスト関数は，

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

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 のアルゴリズムへテストデータに関する手順,（手順 2),（手順 7）を追加し以下のようなアルゴリズムとする．

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

（手順 2） 入力として n 個のテストデータ $(\boldsymbol{x}_{m+1},y_{m+1}^*),\ldots,(\boldsymbol{x}_{m+n},y_{m+n}^*)$ が与えられる．ここで $\boldsymbol{x}_{m+1},\ldots,\boldsymbol{x}_{m+n}$ は特徴ベクトル，$y_{m+1}^*,\ldots,y_{m+n}^*$ は各々 $\boldsymbol{x}_{m+1},\ldots,\boldsymbol{x}_{m+n}$ 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 $D_1(j)=0 \quad (j=m+1,\ldots,m+n)$ を与える．

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

（手順 4） 重み D_t に従って分類クラスが付与されている（すなわち $y_i\neq 0$ である）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） パラメータ α_t を $\alpha_t=\frac{1}{2}\ln(\frac{1-\epsilon_t}{\epsilon_t})$ によって計算する．ここで，ϵ_t は重み付き誤分類率で $\epsilon_t=\sum_{i:h_t(x_i)\neq y_i}D_t(i)$ で計算される．

（手順 6） 各データの重みを次式によって更新する．

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

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

（手順 7） m^+ を訓練データ中の正例の数，$n_{labeled}$ をすで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) $n_{labeled}=0$ または $m^+/m\geq n_{labeled}^+/n_{labeled}$ のとき，

$y_j=0$ であるテストデータの中で

$$H(\boldsymbol{x}_j)=\sum_{k=1}^{t}\alpha_k h_k(\boldsymbol{x}_j)$$

が最大値をとるテストデータ j に対して $y_j=+1$ および $D_{t+1}(j)=\epsilon$（ϵ は小さい値でたとえば $\epsilon=0.01$）を与える．また，このとき分類クラスを付与するデータ以外ですでに分類クラスが付与されていたデータの重みを次式で更新する．

$$D_{t+1}(i)=\frac{D_t(i)}{Z_t'}$$

ここで Z_t' は j 以外の重みの和を $1-\epsilon$ にするための正規化定数である．

(ii) $n_{labeled}\neq 0$ かつ $m^+/m<n_{labeled}^+/n_{labeled}$

のとき，

$y_j = 0$ であるテストデータの中で

$$H(\boldsymbol{x}_j) = \sum_{k=1}^{t} \alpha_k h_k(\boldsymbol{x}_j)$$

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．また，このとき分類クラスを付与するデータ以外ですでに分類クラスが付与されていたデータの重みを

$$D_{t+1}(i) = \frac{D_t(i)}{Z'_t}$$

の式で更新する．ここで Z'_t は j 以外の重みの和を $1-\epsilon$ に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．

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

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山登り的に選択でき，コスト関数のとりうる値を小さくする分類器を生成することができる．

3.2 テキスト分類への適用

正例と負例の 2 つのクラスに属する l 個の訓練データのベクトルの集合を，

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

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．本論文では，テキストの特徴をテキスト中に出現する n 個の異なり単語で代表させ，単語 d がテキスト中に出現する場合，$w_d = 1$，出現しない場合を，$w_d = 0$ として 1 つのテキストをベクトル $\boldsymbol{x}_i = (w_1, w_2, \ldots, w_n)$ で表した．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[24]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[22]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．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「茶筌」[21] により形態素解析を行った後，各カテゴリに対して相互情報量の高い単語を抜き出して特徴ベクトルとした．各カテゴリごとに，相互情報量の高い順に単語を並べたとき，1,000 番目以降の単語ではカテゴリ特有のキーワードがほとんど見られないため，上位 1,000 単語を使用した．

表 1 対象データのカテゴリ別内訳
Table 1 RWCP corpus for training and test.

カテゴリ名	訓練データ数	テストデータ数
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

4.2 評価方法

分類精度の評価には，F 値[17]を用いた．各分類ごとに，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 は重み付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）アルゴリズムに対し，我々の提案したトランスダクティブ法を適用し，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 1,000 個まで増やし，1,000 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 1,000 個のテストデー

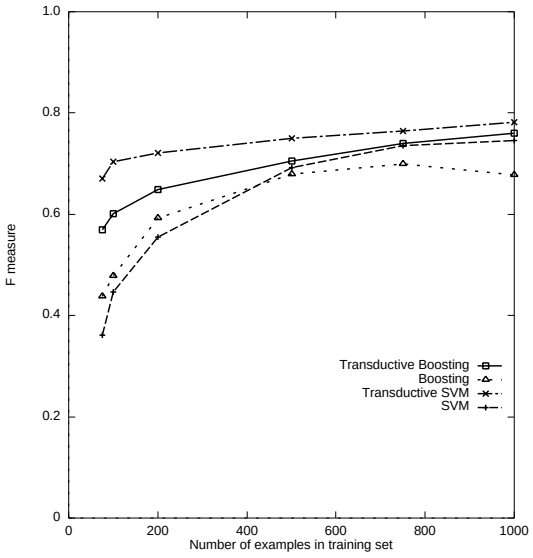

図 2 訓練データ数と学習精度（F 値）
Fig. 2 F-measure and the number of training data.

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 個のときにはブースティングでの F 値が 0.438，トランスダクティブ・ブースティングでの F 値が 0.569 で，大きく 0.131 ポイントも上昇している．これは，75 個の訓練データしかなくても帰納的学習で 200 個の訓練データを使用して学習したときの分類精度に匹敵し，本論文で使用したトランスダクティブ法が訓練データが少数の場合に精度を向上させるのに役立っていることが分かる．また，SVM と TSVM での分類精度比較でも，トランスダクティブ法により分類精度が大きく向上しており，扱ったデータはテストデータの分布を考慮することにより分類精度の向上が期待できるデータであることが分かる．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．このように本論文で提案のトランスダクティブ法によって，ブースティングの精度が，高い分類精度で注目されている SVM を上回ることは興味深い．分類精度は，ブースティングが訓練データ数 1,000 個でやや分類精度を下げている場合を除いて，訓練データを 1,000 個まで増やすに従って単調増加しているが，テストデータの分布を考慮するトランスダクティブ法を使ったときの精度の使わなかったときの精度に対する向上は小さくなっていく．これは，訓練データ数が 1,000 個のときには訓練データとテストデータの分布がかなり似ており，帰納的学習とトランスダクティブ法の差が小さくなるためだと考えられる．

ブースティングおよびトランスダクティブ・ブースティング法に関して，カテゴリ別の詳細な結果を**表 2**，**表 3** に示す．なおカテゴリごとに最高精度の数字を太字で表している．スポーツカテゴリで訓練データが 75 個，1,000 個の場合のように，もともと精度が高かった場合には，トランスダクティブ・ブースティング法を使うと，ブースティングに比べて，かえって分類精度が下がることがある．しかし，教育，軍事，国際関連，言語活動カテゴリのように，ブースティングを使ったときに，分類精度がかなり低かったカテゴリでは，トランスダクティブ・ブースティング法を使うことにより，大幅に精度が向上していることが分かる．

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．ブースティング，SVM どちらの場合もトランスダクティブ法を使用することで精度が大きく向上している．テストデータが増えるに従って精度はほぼ単調に上がっていくが，テストデータが 500 個以上になると，ブースティング，SVM の場合もほとんど精度は変わらなくなる．これは，訓練データが少ない場合にテストデータの分布を考慮すると精度は上がるが，考慮するテストデータの数が一定数を超えると精度向上には寄与しなくなることを示している．しかし，逆にテストデータの数を増やしたことで精度の低下を招くなどの悪影響も出ていないことが分かる．

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，トランスダクティブ法を使わない元々の SVM とブースティングの分類精度の差に起因するところが大きいと思われる．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ことが考えられる．実際，AdaBoost アル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[2]．よって，全体的な精度向上のためには，ブースティングアルゴリズムとして BrownBoost[4] のような，例外的なデータの重みを減らすアルゴリズムを使用することが考えられる．なお BrownBoost は Freund の “boost-by-majority” アルゴリズム[3] を Freund 自ら適応型にしたものである．

表 2 訓練データ数による影響（F 値）（Transductive Boosting）
Table 2 F-measure for the number of training data (Transductive Boosting).

カテゴリ名 \ 訓練データ数	75	100	200	500	750	1,000
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）
Table 3 F-measure for the number of training data (Boosting).

カテゴリ名 \ 訓練データ数	75	100	200	500	750	1,000
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

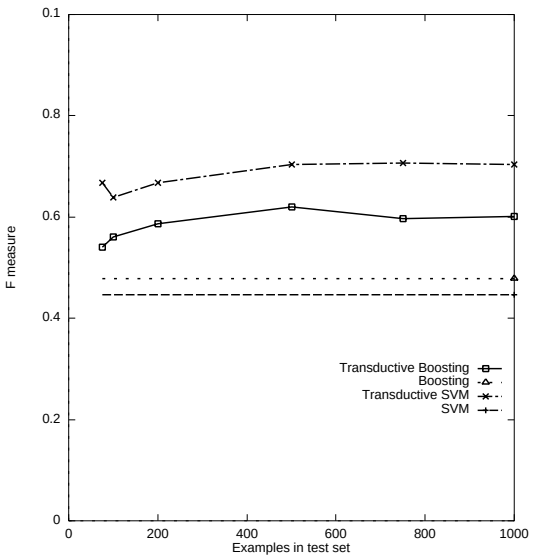

図 3 ラベルなしデータ数と学習精度（F 値）
Fig. 3 F-measure and the number of unlabeled data.

本論文で述べたトランスダクティブ法そのものについては，ラウンドごとにラベル付けするサンプル数を複数にする，ラベル付けの際の重みの与え方を現在の固定値から現ラウンドの重み中の最小値を与える，といった改良が考えられる．

5. 結　　論

本論文では，ブースティングアルゴリズムに適したトランスダクティブ法を提案し，テキスト分類問題に適用した．訓練データ数を変化させる実験を行い，トランスダクティブ法を使わない AdaBoost，SVM に加えて TSVM と比較した．その結果，訓練データが少ない場合，本論文で提案したトランスダクティブ・ブースティング法を用いることにより，従来のブースティングによる分類に比べて，SVM の精度を上回るような大幅に高い精度の分類が得られた．本論文で提案したトランスダクティブ法が，訓練データが少ない場合のブースティングを用いたテキスト分類問題に対し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いただいた毎日新聞社に感謝いたします．

参 考 文 献

1) Breiman, L.: Prediction Games and Arcing Algorithms, *Neural Computation*, Vol.11, No.7,

pp.1493–1517 (1999).
2) 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).
3) Freund, Y.: Boosting a Weak Learning Algorithm by Majority, *Information and Computation*, Vol.121, No.2, pp.256–285 (1995).
4) Freund, Y.: An Adaptive Version of the Boost by Majority Algorithm, *Proc.12th Annual Conference on Computational Learning Theory*, pp.102–113 (1999).
5) 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).
6) Haruno, M., Shirai, S. and Ooyama, Y.: Using Decision Trees to Construct a Practical Parser, *Machine Learning*, Vol.34, pp.131–149 (1999).
7) Ittner, D.J., Lewis, D.D. and Ahn, D.D.: Text Categorization of Low Quality Images, *Proc. Symposium on Document Analysis and Information Retrieval*, pp.301–315 (1995).
8) Joachims, T.: Text Categorization with Support Vector Machines: Learning with many relevant features, *Proc. 10th European Conference on Machine Learning* (*ECML-98*), pp.137–142 (1998).
9) Joachims, T.: Transductive Inference for Text Classification using Support Vector Machines, *Proc. 16th International Conference on Machine Learning* (*ICML-99*), pp.202–209 (1999).
10) Lewis, D. and Ringuette, M.: A Comparison of Two Learning Algorithms for Text Categorization, *Proc. 3rd Annual Symposium on Document Analysis and Information Retrieval*, pp.81–93 (1994).
11) Mason, L., Baxter, J., Bartlett, P. and Frean, M.: Boosting Algorithms as Gradient Descent, *Proc. 12th Advances in Neural Information Processing Systems* (*NIPS-99*), pp.512–518 (2000).
12) 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).
13) Salton, G.: *The Smart Retrieval System-experiments in Automatic Document Processing*, Prentice-Hall (1971).
14) Schapire, R.E.: The Strength of Weak Learnability, *Machine Learning*, Vol.5, No.2, pp.197–227 (1990).
15) 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).
16) Schapire, R.E. and Singer, Y.: BoosTexter: A Boosting-Based System for Text Categorization, *Machine Learning*, Vol.39, pp.135–168 (2000).
17) Sundheim, B.M.: Overview of the Fourth Message Understanding Evaluation and Conference, *Proc. 4th Message Understanding Conference*, pp.3–29 (1992).
18) Taira, H. and Haruno, M.: Feature Selection in SVM Text Categorization, *Proc. 16th National Conference on Artificial Intelligence* (*AAAI-99*), pp.480–486 (1999).
19) Vapnik, V.: *Statistical Learning Theory*, John Wiley & Sons (1998).
20) Yang, Y.: Expert Network: Effective and Efficient Learning from Human Decisions in Text Categorization and Retrieval, *Proc. 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.13–22 (1994).
21) 松本裕治, 北内 啓, 山下達雄, 平野善隆, 浅原正幸, 松田 寛：日本語形態素解析システム『茶筌』version 2.0 使用説明書第二版 (1999). NAIST Technical Report NAIST-IS-TR99012.
22) 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).
23) 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).
24) 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).

(平成 12 年 11 月 20 日受付)
(平成 14 年 3 月 14 日採録)

平 博順（正会員）

1994 年東京大学理学部卒業．1996 年同大学院修士課程修了．2002 年奈良先端科学技術大学院大学博士後期課程修了．博士（工学）．1996 年日本電信電話（株）入社．同社コミュニケーション科学基礎研究所研究員．機械学習および自然言語処理に興味を持つ．

春野 雅彦（正会員）

1991 年京都大学工学部電気工学第二学科卒業．1993 年同大学院修士課程修了．1998 年奈良先端科学技術大学院大学博士後期課程修了．博士（工学）．1993 年日本電信電話（株）入社．1997 年まで同社コミュニケーション科学研究所研究員．1997 年より ATR 人間情報科学研究所研究員．機械学習，自然言語処理およびコミュニケーションの生物的基礎に興味を持つ．