SVMを用いた文書分類と構成的帰納学習法

高村大也[†] 松本裕治[†]

Latent Semantic Indexing（LSI）などの次元圧縮手法による構成的帰納学習法を，サポートベクターマシン（Support Vector Machine，SVM）と組み合わせて文書分類に応用した場合の振舞いを論じる．SVMの分類能力は，通常用いられる次元圧縮では向上させることが困難である．しかし，次元圧縮手法により変換された文書ベクトルを素性として元のベクトルに追加することにより，その向上が可能であることを示す．実験では，次元圧縮に用いる未知データの量が十分大きい場合に精度改善が見られた．

Constructive Induction and Text Categorization with SVMs

Hiroya Takamura[†] and Yuji Matsumoto[†]

In this paper, we discuss text categorization with Support Vector Machines (SVMs) combined with the constructive induction using dimension reduction methods, such as Latent Semantic Indexing (LSI). It is difficult to improve the categorization ability of SVMs only with usual dimension reduction methods. We show, however, that the categorization ability is improved by adding new features extracted by dimension reduction methods. Using this method, we succeeded in improving the performance of SVMs in text categorization, especially when a number of unknown examples can be used for feature extraction.

1. はじめに

従来の文書分類の方法は，その多くが十分な量の訓練データに基づいたものである．しかし，訓練データの収集には多大な労力がかかり，実際の応用を考慮すると，文書分類は訓練データが小さい場合においても良い精度を実現しなくてはならない．もちろん，少量の訓練データでの分類方法も今までいくつか提案されているが[12]，さらなる改良が必要である．そのような方法の実現のためには，未知データが与える貴重な情報を十分に活用することが重要である．本稿では，Latent Semantic Indexing（LSI）[5]などの次元圧縮手法による構成的帰納学習法を用いて特徴次元を拡張し，サポートベクターマシン（Support Vector Machine，SVM）[21]の分類性能を向上させる手法について論ずる．

SVMは，画像処理や自然言語処理など多くの分野で応用されてきた．SVMを文書分類に適用するという考えは文献 9) で初めて導入され，良い結果を残している．しかし，訓練データ量が少ない場合は，しばしば高精度の分類に失敗する．このような問題に対しては，大きく分けて2つのアプローチがある．1つは未知データを疑似的に訓練データとして扱えるようにするなど学習アルゴリズムそのものを改良するもので[8),10)]，もう1つは素性選択など，データに働きかけるものである[22]．本稿で論じる手法は後者に属する．k-Nearest Neighbor法[11]などに対してはLatent Semantic Indexing（LSI）により素性空間の次元を圧縮する方法がよく用いられる．LSIは，文献 5) で提案された手法で，単語の共起情報を用いて，同様のトピックに関連していると思われる単語をまとめあげるものである．自然言語処理では幅広く使用され，その効果が認められている．

しかし後に実験で示すように，LSIなどによる従来の次元圧縮に基づく方法は，SVMに対して必ずしも良い結果を生まない．また，相互情報量による素性選択も分類性能を低下させるという実験結果が報告されている[19]．もちろん，素性選択には次元低下にともなう学習の効率化という長所があるが，文献 19) における実験で素性選択による精度向上が見られている決定木学習と比較すると，その相違は明らかである．

そのような点から，SVMの高い分類性能を十分に利用するためのデータの前処理方法は，詳細に研究さ

† 奈良先端科学技術大学院大学自然言語処理学講座
Computational Linguistics Laboratory, Nara Institute of Science and Technology

れるべき研究題材であるといえる．本稿で我々が用いる手法は，LSI などの次元圧縮手法によって得られた成分を素性空間の次元を拡張するために用いる，一種の構成的帰納学習法である．具体的には，まず，次元圧縮手法により文書ベクトルを圧縮し，次にその圧縮されたベクトルを元のベクトルに新しい素性として追加することにより新しいベクトルを構成する．そのベクトルを新たな素性ベクトルとして，SVM を用いて文書分類を行う．これは 4 章で示すように，与えられたベクトルの情報をできる限り保持しつつ，素性空間のある部分空間を重み付けしていると考えられる．

LSI を構成的帰納学習に用いるという考え方は新しいものでなく，文献 13) で扱われている．しかし，文献 13) では決定木を用いた小規模な実験のみが行われていて，文書分類のような高次元かつ疎であるようなデータに対しての分類性能は未知である．しかも，次元圧縮や素性選択によって精度低下が見られる SVM に対して，その分類性能を向上させられるかという疑問は，より高精度の分類器が望まれるという事実を鑑みれば，解決されるべきであるといえる．

また，LSI とは性質が異なると思われる，agglomerative な hard クラスタリングを次元圧縮のために使用した場合についても調査する．

提案手法の有効性を示すために，文書分類の標準的なデータセットである Reuters-21578 と 20-newsgroup を用いて実験を行った．実験では以下の 5 種類の素性空間：(1) 元の素性空間，(2) LSI によって次元圧縮された素性空間，(3) hard クラスタリングによって次元圧縮された素性空間，(4) LSI によって得られた素性が追加された素性空間，(5) hard クラスタリングによって得られた素性が追加された素性空間，において SVM による文書分類を試した．

その結果，次元圧縮に用いる未知データの量が十分に大きい場合は，提案手法が SVM の分類性能を向上させることが示された．

2. サポートベクターマシン

2.1 サポートベクターマシンの概観

サポートベクターマシン（SVM）は，Large-Margin Classifier [18] の一種である．与えられた素性ベクトルとラベルのペアの集合，

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_l, y_l) \quad (1)$$
$$\forall i,\ \mathbf{x}_i \in \mathbf{R}^d, y_i \in \{-1, 1\}$$

に対して，SVM はマージン（分離平面とベクトルとの距離）が最大になるような分離平面を構成する（**図 1**）：

$$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b. \quad (2)$$

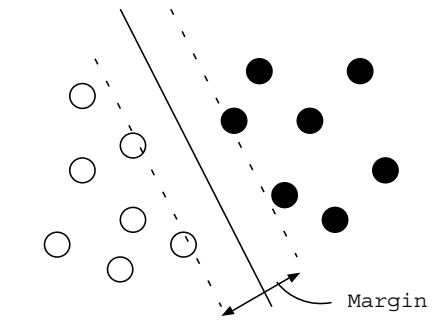

図 1 サポートベクターマシン
Fig. 1 Support Vector Machine.

最大マージンを達成することはノルム $|\mathbf{w}|$ を最小にすることと同値になる．この問題は次のように書ける．

$$\min. \quad \frac{1}{2}|\mathbf{w}|^2, \quad (3)$$
$$\text{s.t. } \forall i,\ \ y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 \geq 0.$$

この問題の解は次の双対問題を解くことによって得られる：

$$\max. \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \quad (4)$$
$$\text{s.t.} \quad \sum_i \alpha_i y_i = 0,$$
$$\forall i,\ \ \alpha_i \geq 0.$$

ここで $\alpha_i(\forall i)$ はラグランジュ乗数である．式 (4) を最大にする $\alpha_i(\forall i)$ を用いて，最適な $\mathbf{w}^*$，b^* は，

$$\mathbf{w}^* = \sum_i \alpha_i y_i \mathbf{x}_i, \quad (5)$$

$$b^* = -\frac{b_{neg} + b_{pos}}{2} \quad (6)$$

と表される．ただし，

$$b_{neg} = \max_{i:y_i=-1} (\mathbf{w}^* \cdot \mathbf{x}_i), \quad (7)$$

$$b_{pos} = \min_{i:y_i=1} (\mathbf{w}^* \cdot \mathbf{x}_i) \quad (8)$$

である．式 (5) および (6) を式 (2) に代入することにより，

$$f(\mathbf{x}) = \sum_i \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x} + b^*. \quad (9)$$

を得る．

テスト事例は式 (9) の符号に従って分類される．

2.2 カーネル法

SVM は線形分類器であるので，その分離能力には限界がある．この限界を超えるために，通常はカーネ

ル法が SVM と組み合わせて用いられる[21]．

カーネル法では，式 (4) や式 (9) における内積が，カーネル関数と呼ばれるより一般的な内積 $K(\mathbf{x}_i, \mathbf{x})$ に置き換えられる．もちろんカーネル関数は一定の条件（Mercer の条件）を満たしていなくてはならない．よく使用されるカーネル関数としては，多項式カーネル $(\mathbf{x}_i \cdot \mathbf{x}_j + 1)^d$ $(d \in \mathbf{N}_+)$，RBF（Radial Basis Function）カーネル $\exp\{-|\mathbf{x}_i - \mathbf{x}_j|^2/2\sigma^2\}$ などがある．

カーネル関数により，素性ベクトルは（高次元の）ヒルベルト空間に写像され，その空間において線形分離される．この写像によって，SVM は線形分類器であるにもかかわらず，非線形分離が可能になる．

カーネル法を用いることのもう 1 つの利点は，高次元の空間を扱うにもかかわらず，高次元ベクトルを明示的に計算する必要がないことである．カーネル関数の値だけを計算すればよい．これは計算量の大幅な節約につながる．

3. 次元圧縮手法

本研究では，文書を表現するためにベクトル空間モデルを採用する[16]．ベクトル空間モデルでは，文書 $\mathbf{d}$ はその文書内の各単語の頻度（あるいはその単語が出現したか否かの二値素性）を要素とするベクトル $(f_1, \cdots, f_d)$ で表される．ベクトル空間モデルに基づいて作られた単語文書行列 X に対し，その単語の次元を圧縮する方法をここで説明する．我々は以下の 2 手法を試した．

3.1 Latent Semantic Indexing（LSI）

LSI[5] とは，元々の素性空間の次元 n より少ない数 m の直行軸を用いて，素性空間の次元を最小自乗の意味で最適に圧縮する方法である．このような圧縮は，共起性の高い単語をまとめあげることに相当する．

具体的には，まず単語文書行列 X の特異値分解を用いて行列の分解が行われる．特異値分解によると，X ($n \times l$ 次元) は，直行行列 T ($n \times r$ 次元)，D ($l \times r$ 次元) と，対角行列 S ($r \times r$ 次元) を用いて，

$$X = TSD^t \tag{10}$$

と表される．ここで r は X のランクであり，S の対角成分は，X の特異値が降順に並んだものである．r の代わりにある整数 m ($\leq r$) を用いることを考える．S_m を大きい方から m 個の特異値を降順に持った対角行列であるとし，また T_m を T のはじめの m 個の列ベクトルから成る行列，D_m を D のはじめの m 個の列ベクトルから成る行列とする．すると X の近似行列 X_m は，

$$X_m = T_m S_m D_m^t \tag{11}$$

と表される．さらに，X を m 次元部分空間に射影した行列 $S_m D_m^t$ は，$T_m^t X$ で表されることが分かる．結局，$T_m^t X$ が単語文書行列の圧縮表現となる．

3.2 Agglomerative Hard クラスタリング

Hard クラスタリングのために我々が用いる確率モデルにおいて，単語と文書の同時確率は次のように表される：

$$P(w, \mathbf{d}) = P(C_w, \mathbf{d})P(w|C_w), \quad w \in C_w. \tag{12}$$

ここで，w はある単語，$\mathbf{d}$ はある文書を表す．また，C_w は w が属するクラスタを表す．

単語と文書の共起データ：

$$S = \{(w_1, \mathbf{d}_1), (w_2, \mathbf{d}_2), \cdots, (w_k, \mathbf{d}_k)\}, \tag{13}$$

が与えられたとき，その対数尤度は，

$$\sum_{(w,\mathbf{d})\in S} \log P(w, \mathbf{d}) = \sum_{(w,\mathbf{d})\in S} \log P(C_w, \mathbf{d})P(w|C_w) \tag{14}$$

と計算される．モデル (12) のパラメータは，最大尤度推定に基づいて，

$$P(C_w, \mathbf{d}) = \frac{N(C_w, \mathbf{d})}{|S|}, \tag{15}$$

$$P(w|C_w) = \frac{N(w)}{N(C_w)} \tag{16}$$

と計算される．ここで，$N(x)$ は x の頻度を指す．

この確率モデルをベースにして，最小の尤度減少を引き起こす単語クラスタペアのマージが繰り返される．このクラスタリング手法は，文献 2) に記述されているものと類似している．しかし，彼らの手法は単語と分類カテゴリの共起を扱った教師付きクラスタリングであるのに対し，本稿で用いるクラスタリング手法は，単語と文書の共起を扱った教師なしクラスタリングであるという点で異なっている．また，単語のクラスタリングとしては，直前に出現する単語などのローカルな情報を手がかりに単語をクラスタリングする手法なども考えられるが，用いる情報を LSI の場合と等しくするために，今回は上述のクラスタリング手法を用いた．

また，第 i クラスタに単語 w_j が含まれている場合に (i, j) 成分が 1 となり，そうでないとき 0 となるような行列 H を考えることにより，単語文書行列 X のここでの圧縮表現は HX と表される．

4. 構成的帰納学習法への適用

構成的帰納学習法とは，元の入力データに現れてい

ない新しい素性を生成するタイプの帰納学習であり，元の素性空間が十分にデータの特徴を表現できていない場合に有効である[7]．構成的帰納学習法を実現した代表的なシステムとしては，文献 4)，14) などで記述されたものがある．本稿では，前章で説明した次元圧縮手法をこの構成的帰納学習法に適用する☆．

まず，LSI もしくは hard クラスタリングを用いて文書ベクトルの次元を圧縮する．圧縮に用いた行列 T_m^t と H を統一的に M と表すことにすると，文書ベクトル $\mathbf{d}$ と圧縮されたベクトル $\mathbf{s}$ の関係は，

$$M\mathbf{d} = \mathbf{s}, \tag{17}$$

と表現できる．次に，元のベクトル $\mathbf{d}$ と圧縮されたベクトル $\mathbf{s}$ を連結（concatenate）する：

$$\hat{\mathbf{d}} = \begin{bmatrix} \mathbf{d} \\ \mathbf{s} \end{bmatrix}. \tag{18}$$

$\hat{\mathbf{d}}$ を入力として SVM により分類を行う．

このような形で文書ベクトルの次元を拡張することは，元の空間において特殊なカーネル関数を使うことと同等である．このことを線形の場合について説明する．2 つのベクトル $\mathbf{d}_1$，$\mathbf{d}_2$ が与えられたとき，拡張空間のカーネル関数 K は次のように表される：

$$\begin{aligned} K(\hat{\mathbf{d}}_1, \hat{\mathbf{d}}_2) &= \hat{\mathbf{d}}_1^t \hat{\mathbf{d}}_2 \\ &= \mathbf{d}_1^t \mathbf{d}_2 + \mathbf{s}_1^t \mathbf{s}_2 \\ &= \mathbf{d}_1^t \mathbf{d}_2 + \mathbf{d}_1^t M^t M \mathbf{d}_2. \end{aligned} \tag{19}$$

このカーネルは，一般的に使用される多項式カーネルなどと異なり，高次元空間への写像関数がデータに依存した形で決定するという特色を持つ．

式 (19) は，次元圧縮手法によって決定される *Latent Semantic Space* に重みを与えていることになる．ここで重みを与えることと圧縮することは異なることに注意されたい．一般の次元圧縮の適用方法においては，*Latent Semantic Space* だけが考慮されるが，我々の方法では元の空間が依然として計算結果に影響を与えている．提案手法におけるこの性質は，元の空間によって与えられる情報を保持しつつ *Latent Semantic Space* に注目することを可能にする．

一般に文書分類などのタスクにおいては，単語を素性として用いた素性空間は非常に疎となり，訓練データに出現しなかった単語がテストデータで出現することは稀ではない．そのような単語は分類に何ら寄与することはできない．本研究で用いる次元圧縮手法は，教師なし学習手法であるので，未知（ラベルなし）データを用いて計算することが可能である．よって次元圧縮手法により，上記のような訓練データに出現しなかった単語を（圧縮された）素性として取り込むことが可能である．一方，文献 3) などに示されているように，文書分類では特定の単語の出現が分類に大きく寄与することがある．次元圧縮を行うだけでは，そのような有用な単語の出現を適切に利用することができなくなる．提案手法では，次元圧縮により未知データを利用すると同時に，特定の単語の出現を利用することも可能である．特に，分類における各素性に対する重みを適切に推定できる SVM を用いていることから，新たに追加した素性に対する重みも適切に推定されるものと考えられる．このような考察から，提案手法により精度向上が期待できると考えられる．

次元圧縮を適用する段階で問題になるのが，その圧縮率の決定である．情報理論に基づいた，Minimum Description Length [15] や Akaike InformationCriterion [1] などといったモデル選択基準などを用いて圧縮率を決定する方法もあるが，我々は非常にシンプルな方法でこの問題を回避する．我々のとった方法は，いくつかの異なる圧縮率において圧縮されたベクトルをすべて新しい素性として追加するというものである．このような方法は，SVM が高い汎化性能を持つことから可能になる．

5. 実　　験

提案手法の有効性を示すために，いくつかの実験を行った．使用したデータは Reuters-21578 データセット☆☆と 20-newsgroup☆☆☆である．

まず Reuters-21578 を用いた実験について説明する．データセットを訓練データとテストデータに分割するため，最もよく使用されている ModApte-split と呼ばれるデータ分割を施した．さらにテキスト部分がほとんど含まれていないような文書を削除することにより，8,815 の文書から成る訓練データと 3,023 の文書から成るテストデータを得た．最も頻度の高い 10 個のカテゴリを選び（**表 1**），各カテゴリについてそのテスト事例がカテゴリに属するか属さないかの二値分類を行った．

また，前処理として，TreeTagger [17] を用いて stem-

☆ 文献 20) では，構成的帰納学習法とそうでないものを厳密に分類しており，彼らは素性どうしの関係を考慮した新しい素性を追加するものが構成的帰納学習を呼ぶべきだと主張している．しかし，本稿では元の素性から新たな素性を作り出すものという一般的な意味で，構成的帰納学習という言葉を用いている．

☆☆ http://www.daviddlewis.com/resources/より入手可能.

☆☆☆ http://kdd.ics.uci.edu/より入手可能.

表 1 実験で使用されたカテゴリ
Table 1 The categories used in experiments.

カテゴリ	訓練文書数	テスト文書数
earn	2725	1051
acq	1490	644
money-fx	464	141
grain	399	135
crude	353	164
trade	339	133
interest	291	100
ship	197	87
wheat	199	66
corn	161	48

ming を行った．文書ベクトルを作成する際，stop-word 削除は行っていないが，8,815 訓練文書中で出現回数が 4 回以下の低頻度単語は削除した．また，素性の値は文書中の出現頻度を用いた．SVM の学習と分類には TinySVM☆を使用した．LSI や hard クラスタリングの計算には，訓練事例のみ 8,815 事例もしくはその一部を用い，テスト事例は使用していない．

Reuters-21578 を用いた実験は大きく 4 種類に分かれる．1 つは固定したラベル付事例数に対し，各カテゴリにおける分類性能の振舞いを調べる目的で行われた（5.1 節）．2 つ目はラベル付事例数が変化したときの分類性能の振舞いを調べるためである（5.2 節）．3 つ目に，訓練事例数が大きい場合にどのような振舞いが見られるかを調べる実験をした（5.3 節）．最後に，訓練事例数を固定し，次元圧縮に用いるデータ量だけを変化させて実験を行った（5.4 節）．

結果は F 値を用いて評価する．F 値とは，適合率 P（正例と予測された事例における正解の割合）と再現率 R（全正例における正例と予測された事例の割合）の調和平均（$= 2PR/(P+R)$）で定義される値である．また，結果の数値は，全 8,815 訓練文書の中から該当する個数の訓練文書をランダムに 10 回選び，それぞれについて実験を行い，そのマクロ平均[23] を記した．ただし，10 回の各実験において，すべてカテゴリに関して少なくとも 1 つの正例が存在するようにランダムサンプリングを行った．

ここでは，LSI における抽出成分数は 100，200，300 の 3 種類を用い，hard クラスタリングにおけるクラスタ数は 100，500，1,000 の 3 種類を用いた．LSI と hard クラスタリングで異なる次元を用いた理由は，両圧縮手法が持つパラメータ数が大きく異なり，同じ次元で両者を比較することは公平さを欠くと考えられるからである．

また，SVM の学習で用いたカーネル関数は線形カーネルである．

20-newsgroup に関しては，Reuters-21578 について効果が見られた実験設定において，別のデータセットでも効果が見られるかを調査するための補足的な実験を行った．20-newsgroup は，20 のカテゴリを含む．各カテゴリの文書数は約 600 から 1,000 であり，Reuters-21578 と比較するとカテゴリ間のサイズの差が小さい．まずデータを訓練データ（9,420 文書）とテストデータ（9,408 文書）に分割した．訓練データとテストデータに含まれる各カテゴリの文書数は等しくなるようにしてある．stemming や低頻度語の削除などの前処理は，Reuters-21578 に対するものと同じである．5.2 節で Reuters-21578 に対して行ったものと同様の実験を行った（5.5 節）．

5.1 訓練事例数固定での各カテゴリにおける分類性能

ここでは，訓練事例が 1,000 文書の場合について各カテゴリにおける分類性能を調べる．

結果を**表 2**，**表 3** に示す．“方法” の列には，素性空間の構成方法が記されている．“Original” は元の素性空間，“LSI” と “hard” は圧縮された素性空間に相当する．“Original+LSI” と “Original+hard” は 4 章で説明した次元拡張された空間に相当する．“次元” の列には，圧縮された素性空間の次元が記されている．また，表 3 を含め図や表中の “100+200+300” などは，対応する圧縮素性をすべて追加したことを意味する．

次元拡張を施したものの平均値については，ウィルコクソンの符号順位検定[6] を行った．ウィルコクソンの符号順位検定とは，対応のある 2 変数の組に対し，その代表値に差があるか否かを検定する方法である．ここでは “Original” の平均 F 値と各提案手法の平均 F 値の差が有意であるか否かを検定する．各カテゴリについて 10 回，合計 100 回の実験が行われているので，2 変数の組が 100 個あるものとして検定を行った．検定結果は**表 4** に示す．表 4 には，手法とそれに対応する p 値☆☆の上限値を記した（0.001 以下の上限値が算出されたものに関しては，統一して 0.001 と記した）．表中では括弧内に，圧縮された空間の次元数を記した．

☆ http://cl.aist-nara.ac.jp/~taku-ku/software/TinySVM/ より入手可能．

☆☆ 与えられたデータに対し，帰無仮説のもとで算出された統計量が実現する確率のことである．いい換えると，帰無仮説を棄却できる最小の有意水準であるので，p 値が有意水準（通常 0.001 や 0.05 が使用される）より小さい場合は仮説を棄却でき，その場合ここでは F 値の平均の差が有意であるといえる．

表 2 F-値（1,000 訓練事例）
Table 2 F-measures (1,000 labeled examples).

方法	Original	LSI			Original+LSI			hard			Original+hard		
次元	–	100	200	300	100	200	300	100	500	1000	100	500	1000
earn	**96.8**	95.3	94.9	95.2	96.6	96.6	96.5	92.6	95.4	96.3	96.3	96.4	96.7
acq	88.8	86.8	86.3	86.8	**89.5**	89.4	89.1	79.0	84.5	87.2	87.3	87.9	88.7
money-fx	61.3	63.1	61.1	62.3	63.1	62.9	62.3	56.7	60.5	60.4	63.4	62.0	60.9
grain	70.1	68.1	68.8	70.8	72.0	71.0	71.2	57.3	70.9	68.6	71.7	73.3	70.5
crude	63.5	68.9	69.3	69.3	67.0	67.2	66.5	66.1	65.6	66.1	**70.4**	66.2	65.9
trade	63.3	65.2	62.8	63.0	65.1	63.8	64.1	58.0	63.3	63.3	**67.9**	64.1	63.3
interest	58.2	56.4	55.9	56.3	**58.9**	**58.9**	58.0	40.9	55.7	57.1	55.7	57.7	58.1
ship	43.6	45.1	58.5	58.8	49.4	54.2	53.6	40.1	56.4	57.3	52.9	57.3	54.6
wheat	65.8	66.1	**71.3**	70.8	68.1	68.6	67.9	40.7	57.5	65.1	62.8	65.1	65.2
corn	52.1	37.2	46.6	50.4	52.0	52.8	53.2	34.2	44.8	**55.9**	50.2	52.8	55.1
平均	66.4	65.2	67.5	68.4	68.2	68.5	68.2	56.5	65.5	67.7	67.9	68.3	67.9

表 3 F-値（1,000 訓練事例）
Table 3 F-measures (1,000 labeled examples).

方法	Original+LSI	Original+hard
次元	100+200+300	100+500+1000
earn	96.4	96.4
acq	**89.5**	87.7
money-fx	63.3	**63.9**
grain	72.0	**73.7**
crude	69.0	69.8
trade	64.2	66.9
interest	58.6	56.9
ship	57.9	**59.9**
wheat	70.5	65.0
corn	52.7	53.4
平均	**69.4**	**69.4**

表 4 ウィルコクソン検定の結果（p 値）
Table 4 Results of Wilcoxon tests (p-value).

方法	p 値の上限
Original+LSI(100)	0.001
Original+LSI(200)	0.001
Original+LSI(300)	0.001
Original+LSI(100+200+300)	0.001
Original+hard(100)	0.120
Original+hard(500)	0.003
Original+hard(1000)	0.014
Original+hard(100+500+1000)	0.001

訓練事例数が 1,000 のときは，“Original+LSI” と “Original+hard” は圧縮空間の次元にかかわらず，F 値の平均値に関して “Original” を上回っている．しかし，表 4 に示したウィルコクソン検定の結果を見ると，“Original+hard” は “Original+LSI” と比較して p 値の上限が高いことが分かる．特に圧縮空間の次元が 100 のときは有意とはいえない．一方，“Original+LSI” はいずれの場合も差が有意である．さらに，“Original+LSI(100+200+300)” と “Original+hard(100+500+1000)” はどちらも有意の差を持ち，複数の圧縮空間を追加した効果が見られる．

表 5 各カテゴリの平均正例数
Table 5 The averaged numbers of positive examples.

カテゴリ	平均正例数
earn	31.2
acq	16.3
money-fx	5.3
grain	4.3
crude	4.7
trade	4.5
interest	3.5
ship	2.2
wheat	2.9
corn	1.7

5.2 訓練事例数を変化させたときの各手法の振舞い

訓練事例数を変化させたときに各方法がどのように振る舞うかを調べるために，この実験を行った．訓練事例数は 100 から 1,000 まで，100 ずつ増加させた．各カテゴリの平均正例数は，訓練事例数 100 の場合は **表 5** のようになっている．訓練事例のサンプリングはランダムなので，訓練事例が表 5 に記したもの以外のカテゴリに属することもある．

結果を **図 2** と **図 3** に示す．図 2 が LSI を利用して次元拡張を行った場合であり，図 3 が hard クラスタリングを利用して次元拡張を行った場合である．グラフに示した範囲で見る限り，どちらの圧縮手法を用いた場合でも，様々な訓練事例数に対してすべての圧縮素性を追加した場合が最も良い F 値を示している．また，その他の次元拡張方法も，元の素性を用いた場合と比較して高い F 値を示している．つまり，訓練事例数が小さい範囲では，事例数にかかわらず提案手法が有効であることが分かる．

5.3 訓練事例数が大きい場合

提案手法は，訓練事例数が小さい場合を対象としているが，訓練事例数が大きい場合にどのような挙動を

示すかをここで調べておく．具体的には，訓練事例数が 4,000 の場合と 8,000 の場合を調べた．LSI やクラスタリングに用いた事例数が 8,815 なので，前者は訓練事例の約 2 倍の未知データ（テストデータとは異なる）が存在する場合に相当し，後者は訓練事例以外の未知データがほとんど存在しない場合に相当する．

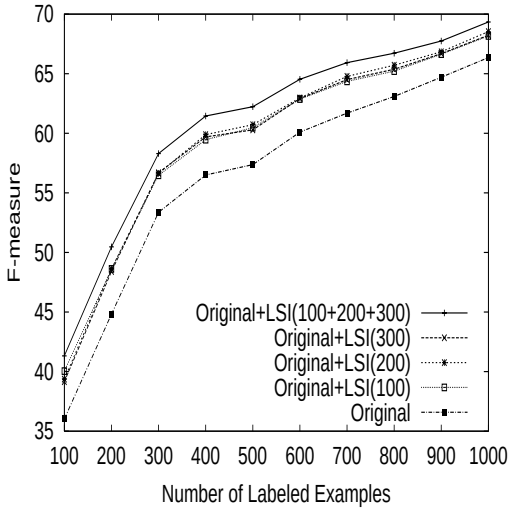

図 2 訓練事例数と分類性能の関係（LSI の場合）
Fig. 2 Training-data size and performance (LSI).

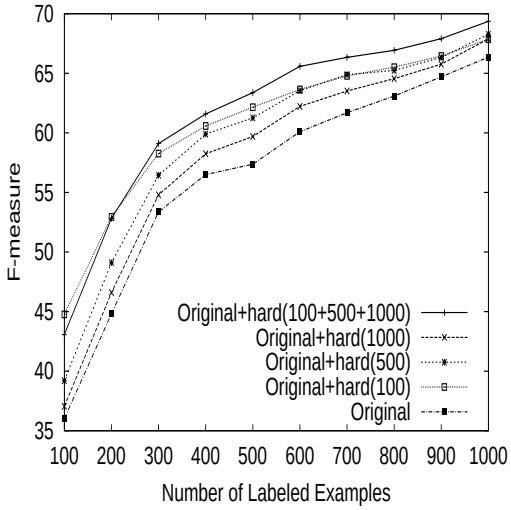

図 3 訓練事例数と分類性能の関係（hard クラスタリングの場合）
Fig. 3 Training-data size and performance (hard clustering).

実験結果を**表 6** と**表 7** に示す．結果は，F 値の平均値のみをここでは示す．これらの図から分かるように，訓練事例数 4,000 の段階で，“Original” と次元拡張したものとの差はあまり顕著に見られなくなる．そして訓練事例数 8,000 の段階では，総じて “Original” の方が良い数値を示している．

つまり，提案手法が有効なのは，LSI やクラスタリングによって情報を抽出できるような未知データが大量にある場合であると考えられる．

5.4 次元圧縮に用いるデータ量を変化させた場合

5.2 節の実験においては訓練事例数が小さい場合に提案手法が良い精度を示し，5.3 節の実験においては訓練事例数が大きくなると提案手法の優位性が薄れることが分かった．しかしこのような実験結果は訓練事例数そのものに起因するというよりも，4 章での考察から，次元圧縮に用いるデータ量に起因すると考えられる．このことをより明確に示すために，ここでは次元圧縮に用いるデータ量を変化させた場合の分類性能の振舞いを観察する．

ここでは訓練事例数は 1,000 に固定した．次元圧縮に用いるデータ量は 1,000 から 8,000 まで 1,000 単位で増加させた．結果を**図 4** と**図 5** に示す．これらの図から分かるように，訓練データのみを用いて次元圧縮を行った場合はほとんど性能の向上が見られず，特に hard クラスタリングの場合は性能の低下が見られる．しかし，未知データ量を増加させるにつれて性能が向上しているのが分かる．先の実験結果と合わせて考えると，本手法は次元圧縮に用いるデータが十分多く（今回の実験では訓練データ量の 3 倍程度以上）ある場合に，分類精度を向上させる能力があると考えられる．

5.5 別のデータセットを用いた実験

ここでは，Reuters-21578 について効果が見られた

表 7 F-値（4,000, 8,000 訓練事例）
Table 7 F-measures (4,000, 8,000 labeled examples).

方法	Original+LSI	Original+hard
次元	100+200+300	100+500+1000
4,000 訓練事例	77.9	76.1
8,000 訓練事例	79.7	77.0

表 6 F-値（4,000, 8,000 訓練事例）
Table 6 F-measures (4,000, 8,000 labeled examples).

方法	Original	LSI			Original+LSI			hard			Original+hard		
次元	–	100	200	300	100	200	300	100	500	1000	100	500	1000
4,000 訓練事例	77.6	73.1	74.2	74.9	**78.1**	78.0	77.8	64.4	69.0	75.3	76.9	76.3	77.6
8,000 訓練事例	80.1	76.3	77.6	77.1	**80.2**	79.9	79.8	67.7	68.2	75.8	79.3	77.3	79.1

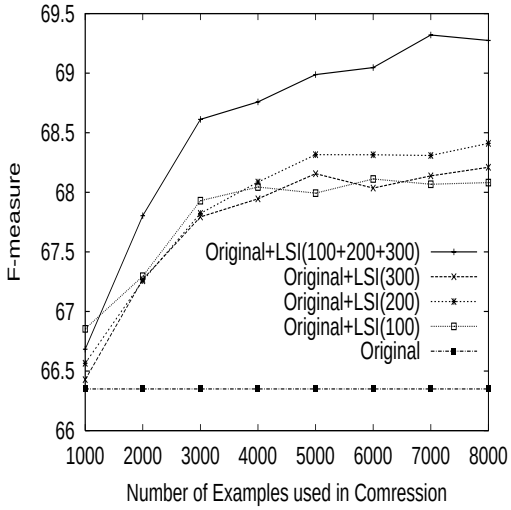

図 4 次元圧縮用データ量と分類性能の関係（訓練事例数 1,000, LSI の場合）
Fig. 4 Data size used in compression and performance (1,000 labeled examples, LSI).

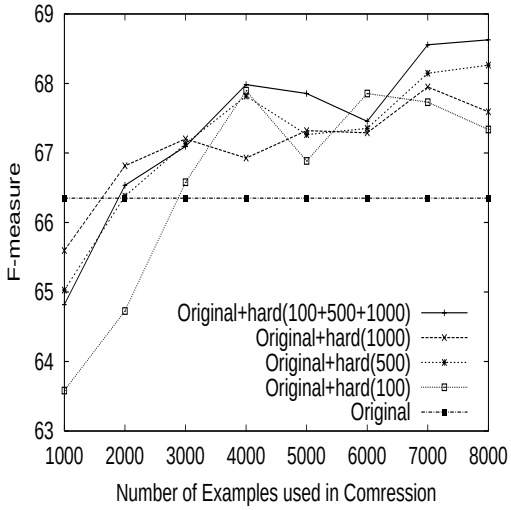

図 5 次元圧縮用データ量と分類性能の関係（訓練事例数 1,000, hard クラスタリングの場合）
Fig. 5 Data size used in compression and performance (1,000 labeled examples, hard clustering).

実験設定において，別のデータセットでも効果が見られるかを調査するために，20-newsgroup を用いて補足的な実験を行う．Reuters-21578 では，次元圧縮データが訓練データの 3 倍程度以上ある場合に提案手法の効果が見られた．よって，ここでは「次元圧縮データが訓練データの 3 倍程度以上ある」という条件が成り立つ範囲で実験を行い，精度向上が見られるかどうかを調べる．次元圧縮は全訓練データ（9,420 文書）を用いて行い，上記条件が成り立つ範囲（具体的には訓練文書数 100 から 3,000 の間）で訓練データ量を変化させる．特に精度が良かった “Original+LSI(100+200+300)” と “Original+hard(100+500+1000)” を “Original” と比較する．それ以外の実験設定は Reuters-21578 に対する実験の設定と同じである．

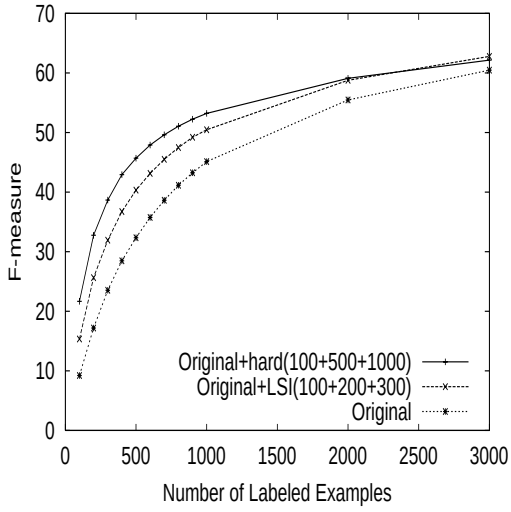

図 6 20-newsgroup に対する分類性能
Fig. 6 Performance for 20-newsgroup.

結果を**図 6** に示す．訓練データ量にかかわらず，両提案手法が “Original” を精度で上回っているのが分かる．この結果は，「提案手法は，次元圧縮に用いるデータが十分多くある場合に分類精度を向上させる」という結論を補強するものである．

6. おわりに

LSI などの特徴成分抽出手法による構成的帰納学習法を，サポートベクターマシン（Support Vector Machine，SVM）と組み合わせて文書分類に応用した場合の振舞いを論じた．

本稿で扱った手法は，次元圧縮手法により変換された文書ベクトルを元のベクトルと結合することにより素性空間を新たに構築し，得られた新しいベクトルを入力として SVM を用いて分類を行うものである．実験では，次元圧縮に用いる未知データが十分多く存在する場合に SVM の分類精度が向上することが示された．また，複数の異なる圧縮率を持つ圧縮空間を同時に用いることで，さらに分類精度が向上することが分かった．

訓練事例数を 10,000 近くにまで増やした場合については，未知データを用いることによる分類精度の向

上は，データ不足のため実験的には未確認であるが，4章での考察と 5.4 節での実験から考えて向上があるものと期待される．ここで，精度を向上させるために必要な未知データ量はどのくらいかという問題が残っているが，これはデータに依存するものと思われ，この問題に関しては今後理論的なアプローチあるいは種々のデータでの実験などを通して実証していきたい．

また，文書分類だけでなく，多義語の曖昧性解消など，他のタスクへの応用も期待される．

参 考 文 献

1) Akaike, H.: A New Look at the Statistical Model Identification, *IEEE Trans. Autom. Control*, Vol.AC-19, pp.716–723 (1974).
2) Baker, D. and McCallum, A.: Distributional Clustering of Words for Text Classification, *Proc. SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval*, pp.96–103 (1998).
3) Bekkerman, R., El-Yaniv, R., Tishby, N. and Winter, Y.: On Feature Distributional Clustering for Text Categorization, *Proc. SIGIR-01, 24th ACM International Conference on Research and Development in Information Retrieval*, pp.146–153 (2001).
4) Bloedorn, E. and Michalski, R.S.: Data-Driven Constructive Induction, *IEEE Intelligent Systems*, Vol.13, No.2, pp.30–37 (1998).
5) Deerwester, S., Dumais, T., Landauer, T., Furnas, W. and Harshman, A.: Indexing by Latent Semantic Analysis, *Journal of the Society for Information Science*, Vol.41, No.6, pp.391–497 (1990).
6) DeGroot, M.: *Probability and Statistics*, Addison-Wesley (1986).
7) Dietterich, T. and Michalski, R.: A Comparative Review of Selected Methods for Learning from Examples, *Machine Learning *An Artificial Intelligence Approach*, Morgan Kaufmann (1983).
8) Glenn, F. and Mangasarian, O.: Semi-Supervised Support Vector Machines for Unlabeled Data Classification, *Optimization Methods and Software*, pp.1–14 (2001).
9) Joachims, T.: Text Categorization with Support Vector Machines: Learning with Many Relevant Features, *Proc. European Conference on Machine Learning*, Chemnitz, Germany, pp.137–142 (1998).
10) Joachims, T.: Transductive Inference for Text Classification using Support Vector Machines, *Proc. 16th International Conference on Machine Learning* (*ICML '99*), Bled, Slovenia, pp.200–209 (1999).
11) Mitchell, T.: *Machine Learning*, McGraw Hill (1997).
12) Nigam, K., McCallum, A., 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).
13) Popelinsky, L. and Brazdil, P.: The Principal Components Method as a Pre-processing Stage for Decision Tree Learning, *Principles and Practice of Knowledge Discover in Databases, Workshop on Metalearning*, Guimaraes (2000).
14) Pagallo, G. and Haussler, D.: Boolean Feature Discovery in Empirical Learning, *Machine Learning*, Vol.5, No.1, pp.71–99 (1990).
15) Rissanen, J.: Stochastic Complexity, *Journal of Royal Statistical Society*, Series B, Vol.49, No.3, pp.223–239 (1987).
16) Salton, G. and McGill, M.J.: *Introduction to Modern Information Retrieval*, McGraw-Hill Book Company, New York (1983).
17) Schmid, H.: Probabilistic Part-of-Speech Tagging Using Decision Trees, *Proc. International Conference on New Methods in Language Processing*, Manchester, pp.44–49 (1994).
18) Smola, A., Bartlett, P., Schölkopf, B. and Schuurmans, D.: *Advances in Large Margin Classifiers*, MIT Press (2000).
19) Taira, H. and Haruno, M.: Feature Selection in SVM Text Categorization, *Proc. 16th National Conference on Artificial Intelligence/ Eleventh Conference on Innovative Applications of Artificial Intelligence* (*AAAI-99/IAAI-99*), Florida, pp.480–486 (1999).
20) Thornton, C.: What Do Constructive Learners Really Learn? *Artificial Intelligence Review*, Vol.13, No.4, pp.249–257 (1999).
21) Vapnik, V.: *The Nature of Statistical Learning Theory*, Springer (1995).
22) Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T. and Vapnik, V.: Feature Selection for SVMs, *Advances in Neural Information Processing Systems*, Vol.13, pp.668–674 (2000).
23) Yang, Y.: An Evaluation of Statistical Approaches to Text Categorization, *Information Retrieval*, Vol.1, No.1/2, pp.69–90 (1999).

(平成 14 年 10 月 1 日受付)
(平成 14 年 12 月 28 日採録)

（担当編集委員 江口 浩二）

高村　大也（学生会員）

1974 年生．1997 年東京大学工学部計数工学科卒業．2000 年同大学大学院工学系研究科計数工学専攻修了（1999 年はオーストリアウィーン工科大学にて研究）．2000 年奈良先端科学技術大学院大学博士後期課程入学，現在に至る．自然言語処理，特に学習理論等の応用に興味を持つ．

松本　裕治（正会員）

1955 年生．1977 年京都大学工学部情報工学科卒業．1979 年同大学大学院工学研究科修士課程情報工学専攻修了．同年電子技術総合研究所入所．1984 年〜1985 年英国インペリアルカレッジ客員研究員．1985 年〜1987 年（財）新世代コンピュータ技術開発機構に出向．京都大学助教授を経て，1993 年より奈良先端科学技術大学院大学教授，現在に至る．京都大学工学博士．専門は自然言語処理．人工知能学会，日本ソフトウェア科学会，言語処理学会，認知科学会，AAAI，ACL，ACM 各会員．