Support Vector Machine を用いた Chunk 同定

工藤　拓[†]　松本　裕治[†]

本稿では，Support Vector Machine (SVM) に基づく一般的な chunk 同定手法を提案し，その評価を行う．SVM は従来からある学習モデルと比較して，入力次元数に依存しない高い汎化能力を持ち，Kernel 関数を導入することで効率良く素性の組み合わせを考慮しながら分類問題を学習することが可能である．SVM を英語の単名詞句とその他の句の同定問題に適用し，実際のタグ付けデータを用いて解析を行ったところ，従来手法に比べて高い精度を示した．さらに，chunk の表現手法が異なる複数のモデルの重み付き多数決を行うことでさらなる精度向上を示すことができた．

キーワード: *Base Phrases Chunking*, 文節まとめ上げ, 機械学習, *Support Vector Machines*, 重み付き多数決

Chunking with Support Vector Machines

Taku Kudo[†] and Yuji Matsumoto[†]

In this paper, we apply Support Vector Machines (SVMs) to identify English base phrases (chunks). It is well-known that SVMs achieve high generalization performance even using input data with a high dimensional feature space. Furthermore, by introducing the Kernel principle, SVMs can carry out training with smaller computational cost independent of the dimensionality of the feature space. In order to improve accuracy, we also apply majority voting with 8 SVMs which are trained using distinct chunk representations. Experimental results show that our approach achieves better accuracy than other conventional frameworks.

KeyWords: *Base Phrases Chunking, Machine Learning, Support Vector Machines, Majority Voting*

1　はじめに

自然言語処理において chunk 同定問題 (chunking) とは，単語列 (一般にこれを token 列とよぶ) をある視点からまとめ上げていき，まとめ上げた固まり (chunk) をそれらが果たす機能ごとに分類する一連の手続きのことを指す．この問題の範疇にある処理として，英語の単名詞句同定 (base NP chunking)，任意の句の同定 (chunking)，日本語の文節まとめ上げ，固有名詞/専門用語抽出などがある．また，各文字を token としてとらえるならば，英語の tokenization，日本語のわかち書き，品詞タグ付けなども chunk 同定問題の一種としてとらえることができる．

一般に，chunk 同定問題は，文脈から得られる情報を素性としてとらえ，それらの情報から精

† 奈良先端科学技術大学院大学情報科学研究科, Graduate School of Information Science, Nara Institute of Science and Technology

度良く chunk を同定するルールを導出する手続きとみなすことができる．そのため，各種の統計的機械学習アルゴリズムを適用可能である．実際に機械学習を用いた多くの chunk 同定手法が提案されている (Ramshaw, Marcus 1995; Tjong Kim Sang 2000a; Tjong Kim Sang, Daelemans, Déjean, Koeling, Krymolowski, Punyakanok, and Roth 2000; Tjong Kim Sang 2000b; 内元，馬青，村田，小作，内山，井佐原 2000; 颯々野，宇津呂 2000)．

しかしながら，従来の統計的手法は，いくつかの問題がある．例えば，隠れマルコフモデルや最大エントロピー (ME) モデルは素性どうしの組み合わせ (共起関係) を効率良く学習できず，有効な組み合わせの多くは人手によって設定される．また多く機械学習アルゴリズムは高い精度を得るために慎重な素性選択を要求し，これらの素性選択も人間の発見的な手続きにたよっている場合が多い．

一方，統計的機械学習の分野では，Boosting (Freund, Schapire 1996)，Support Vector Machines (SVMs) (Cortes, Vapnik 1995; Vapnik 1998) 等の学習サンプルと分類境界の間隔 (マージン) を最大化にするような戦略に基づく手法が提案されている．特に SVM は，学習データの次元数 (素性集合) に依存しない極めて高い汎化能力を持ち合わせていることが実験的にも理論的にも明らかになっている．さらに，Kernel 関数を導入するとこで，非線形のモデル空間を仮定したり，複数の素性の組み合せを考慮した学習が可能である．

このような優位性から，SVM は多くのパターン認識の分野に応用されている．自然言語処理の分野においても，文書分類や係り受け解析に応用されており，従来の手法に比べて高い性能を示している (Joachims 1999; 平，春野 2000; Kudo, Matsumoto 2000a, 2001; 工藤，松本 2002)

本稿では chunk 同定問題として，英語の単名詞句のまとめ上げ (base NP chunking) および英語の任意の句の同定 (chunking) を例にとりながら学習手法として SVM を用いた手法を述べる．さらに，chunk の表現方法が異なる複数の学習データから独立に学習し，それらの重み付き多数決を行うことでさらなる精度向上を試みる．その際，本稿では，各モデルの重みとして SVM に固有の新たな 2 種類の重み付けの手法を提案する．

本稿の構成は以下の通りである．2 章で SVM の概要を説明し，3 章で一般的な chunk 同定モデルおよび SVM の具体的な適用方法，重み付け多数決の方法について述べる．さらに 4 章で実際のタグ付きコーパスを用いた評価実験を提示し，最後に 5 章で本稿をまとめる．

2 Support Vector Machine

2.1 最適分離平面

分類問題において，正例，負例 の 2 つのクラスに属す学習データのベクトル集合を，

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

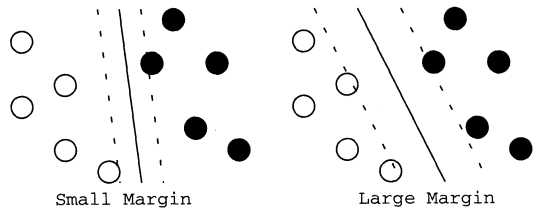

図 1　マージン最大化

とする．ここで $\mathbf{x}_i$ はデータ i の特徴ベクトルで，一般的に n 次元の素性ベクトル ($\mathbf{x}_i = [f_1, f_2, \ldots, f_n]^T \in \mathbf{R}^n$) で表現される．$y_i$ はデータ i が正例 (+1) あるいは負例 (-1) のいずれかを表わす値である．パターン認識とは，この学習データ $\mathbf{x}_i \in \mathbf{R}^n$ から，クラスラベル出力 $y \in \{\pm 1\}$ への識別関数 $f : \mathbf{R}^n \to \{\pm 1\}$ を導出することにある．

SVM では，以下のような n 次元 Euclid 空間上の平面で正例，負例を分離することを考える．

$$\mathbf{w}^T\mathbf{x} + b = 0 \qquad \mathbf{w} \in \mathbf{R}^n, b \in \mathbf{R} \tag{1}$$

この時，近接する正例と負例の間の間隔 (マージン) ができるだけ大きいほうが，汎化能力が高く，精度よく評価データを分類できる．図 1に，2 次元空間上の正例 (白丸)，負例 (黒丸) を分離する問題を例にこのマージン最大化の概略を表す．図 1中の実線は式 (1) の分離平面を示す．一般にこのような分離平面は無数に存在し，図 1 に示す 2 つの分離平面はどちらも学習データを誤りなく分離している．分離平面に平行する 2 つの破線は分離平面が傾き $\mathbf{w}$ を変化させないまま平行移動したときに，分類誤りなく移動できる境界を示す．この 2 つの破線間の距離をマージンと呼び，SVM はマージンが最大となる分離平面を求める戦略を採用している．図 1 の例では，右の分離平面が左の分離平面にくらべて大きなマージンを持っており，精度よくテスト事例を分離できることを意味している．

実際に 2 つの破線を求めてみる．破線は，正例 (+1) もしくは負例 (-1) のラベルを出力する境界面になるように正規化を行えば，

$$\mathbf{w}^T\mathbf{x} + b = \pm 1 \qquad \mathbf{w} \in \mathbf{R}^n, b \in \mathbf{R}$$

で与えられる．さらにマージン d は，分離平面上の任意の点 $\mathbf{x}'$ から各破線までの距離の和であり，$\mathbf{x}'$ は，$\mathbf{w}^T\mathbf{x}' + b = 0$ を満たすため，

$$\begin{aligned} d &= \frac{|\mathbf{w}^T\mathbf{x}' + b - 1|}{\|\mathbf{w}\|} + \frac{|\mathbf{w}^T\mathbf{x}' + b + 1|}{\|\mathbf{w}\|} = \frac{|-1|}{\|\mathbf{w}\|} + \frac{|1|}{\|\mathbf{w}\|} \\ &= \frac{2}{\|\mathbf{w}\|} \end{aligned}$$

となる．このマージンを最大化するためには，$\|\mathbf{w}\|$ を最小化すればよい．つまり，この問題は

以下の制約付き最適化問題を解くことと等価となる[1].

$$\begin{aligned} &\text{目的関数：} \quad L(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 \rightarrow \text{最小化} \\ &\text{制約条件：} \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \ (i = 1 \ldots l) \end{aligned}$$

ここで，2つの破線上の分類を決定づける事例を サポートベクター と呼び，サポートベクター以外の事例は実際の学習結果に影響を及ぼさない．

さらに，一般的な分類問題においては，学習データを線形分離することが困難な場合ある．このような場合，各素性の組み合わせを考慮し，より高次元な空間に学習データを写像すれば線形分離が容易になる．実際の証明は省略するが SVM の学習，分類アルゴリズムは事例間の内積しか使用しない．この点を生かし，各事例間の内積を任意の Kernel 関数におきかえることで，SVM は低次元中の非線形分類問題を高次元中の線形分離問題としてみなし分類を行うことが可能となっている．多くの Kernel 関数が提案されているが，我々は以下の式で与えられる d 次の 多項式 Kernel 関数を用いた．

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T\mathbf{x}_j + 1)^d$$

d 次の 多項式 関数は d 個までの素性の組み合わせ (共起) を考慮した学習モデルと見なすことができる．

2.2 SVM の汎化能力

ここで，汎化能力に関する一般的な理論について考察する．学習データおよびテストデータがすべて独立かつ同じ分布 $P(\mathbf{x}, y)$ から生成されたと仮定すると，識別関数 f のテストデータに対する汎化誤差 $E_g[f]$ ，学習データに対する誤差 $E_t[f]$ は以下のように与えられる．

$$\begin{aligned} E_g[f] &= \int \frac{1}{2}|f(\mathbf{x}) - y| dP(\mathbf{x}, y) \\ E_t[f] &= \frac{1}{l}\sum_{i=1}^{l} \frac{1}{2}|f(\mathbf{x}_i) - y_i| \end{aligned}$$

さらに，$E_g[f], E_t[f]$ には以下のような関係が成立することが知られている (Vapnik 1998).

定理 1 (Vapnik) 学習データの事例数を l，モデルの VC次元を h とする時，汎化誤差 $E_g[f]$ は，$1-\eta$ の確率で以下の上限値を持つ．

$$E_g[f] \leq E_t[f] + \sqrt{\frac{h(\ln \frac{2l}{h} + 1) - \ln \frac{\eta}{4}}{l}} \tag{2}$$

1 実際の我々の実験では多少の解析誤りを認める Soft Margin の項を追加した最適化問題を解いている

ここで VC 次元 h とは，モデルの記述能力，複雑さを表すパラメータである．式 (2) の右辺を VC bound と呼び，汎化誤差を小さくするには，VC bound をできるだけ小さくすればよい．

従来からある多くの学習アルゴリズムは，モデルの複雑さである VC 次元 h を固定し，学習データに対するエラー率を最小にするような戦略をとる．そのため，適切に h を選ばないとテストデータを精度良く分類できない．また適切な h の選択は一般的に困難である．

一方 SVM は，学習データに対するエラー率を Soft Margin や Kernel 関数を使って固定し，そのうえで右辺の第二項を最小化する戦略をとる．実際に式 (2) の右辺第二項に注目すると，h に対して増加関数となっている．つまり，汎化誤差 $E_g(h)$ を小さくするには，h をできるだけ小さくすればよい．SVM では VC 次元 h とマージン M には以下の関係が成立することが知られている (Vapnik 1998).

定理 2 (Vapnik) 事例の次元数を n，マージンを d，全事例を囲む球面の最小直径を D とすると，*SVM* の *VC*次元 h は，以下の上限値を持つ．

$$h \leq \min(D^2/d^2, n) + 1 \tag{3}$$

式 (3) から，h を最小にするためには，マージンを最大にすればよく，これは SVM がとる戦略そのものであることが分かる．また，学習データの次元数が十分大きければ，VC 次元 h は，学習データの次元数に依存しない．さらに，D は，使用する Kernel 関数によって決まるため，式 (3) は Kernel 関数の選択の指針を与える能力も持ちあわせていることが知られている (Vapnik 1998)．また，Vapnik は式 (2) とは別に，SVM に固有のエラー率の上限を与えている．

定理 3 (Vapnik) $E_l[f]$ を *Leave-One-Out* によって評価されるエラー率とする場合

$$E_l[f] \leq \frac{\text{サポートベクター数}}{\text{学習サンプル数}} \tag{4}$$

となる．

Leave-One-Out とは，l 個の学習データのうち 1 個をとりのぞいてテストデータとし，残り $l-1$ を使って学習することをすべてのデータについて l 回繰り返すことで，未知データに対するエラー率を予測する手法である．式 (4) は容易に証明可能である．つまり，SVM の特徴として support vector 以外の事例は最終の識別関数には一切影響を及ぼさない．そのため個々の support vector すべてが誤ったときが最悪のケースとなり，式 (4) が導かれる．この bound は，単純明解で汎化誤差のおおまかな値を予測することを可能にする．しかし，support vector の数が増えても汎化能力が向上する事例もあり，式 (4) の汎化誤差の予測能力は式 (2) には劣ることが知られている．

3 SVM に基づく Chunk 同定

3.1 Chunk の表現方法

Chunk 同定の際，各 chunk の状態をどう表現するかが問題となる．一つの手法として，各 chunk 同定を分割問題とみなし，各単語の間 (ギャップ) にタグを付与する手法が考えられる．しかし，この手法は単語とは別の位置にタグを付与する必要があり，従来からある形態素解析などのタグ付けタスクとは異なる枠組が必要となる．

その一方で，各単語に chunk の状態を示すタグを付与する手法がある．この手法は，従来からあるタグ付け問題と同じ枠組でモデル化ができる利点がある．

後者の単語にタグを付与する表現法として，以下 2 種類の手法が提案されている．

(1) Inside/Outside

この手法は英語の base NP 同定でよく用いられる手法の一つである (Ramshaw, Marcus 1995)．この手法では，chunk の状態として以下の 3 種類を設定する．

- I　現在位置の単語は，chunk の一部である．
- O　現在位置の単語は，chunk に含まれない．
- B　現在位置の単語は，ある chunk の直後に位置する chunk の先頭である．

さらに Tjong Kim Sang らは，上記のモデルを IOB1 と呼び，このモデルを基に IOB2/IOE1/IOE2 の 3 種類の表現方法を提案している (Tjong Kim Sang and Veenstra 1999)．

- IOB2　IOB1 と基本的に同じだが，B タグの意味づけがことなる．IOB2 の場合，B タグはすべての chunk の先頭に付与される．
- IOE1　IOB1 と基本的に同じだが，B タグの代わりに E タグを導入する．E タグは，ある chunk の直前に位置する chunk の末尾の単語に付与される．
- IOE2　IOE1 と基本的に同じだが，E タグはすべての chunk の末尾の単語に付与される．

(2) Start/End

この手法は日本語固有名詞抽出において用いられた手法 (内元他 2000) で，各単語に付与するタグとして以下の 5 種類を設定する[2]．

2 内元らは，C/E/U/O/S の 5 種類のタグを用いているが，IOB1/IOB2/IOE1/IOE2 モデルとの整合性から，便宜的に B/E/I/O/S タグを用いる．タグの名称の変更のみで本質的なタグの意味づけに変更はない．

B　現在位置の単語は，2 つ以上の単語から構成される chunk の先頭の単語である．

E　現在位置の単語は，2 つ以上の単語から構成される chunk の末尾の単語である．

I　現在位置の単語は，3 つ以上の単語から構成される chunk の先頭，末尾以外の中間の単語である．

S　現在位置の単語は 単独で一つの chunk を構成する．

O　現在位置の単語は chunk に含まれない

これら 5 種類のタグ付け手法を 英語の単名詞句抽出 (base NP chunking) を例に以下に示す．

	IOB1	IOB2	IOE1	IOE2	IOBES
In	O	O	O	O	O
early	I	B	I	I	B
trading	I	I	I	E	E
in	O	O	O	O	O
busy	I	B	I	I	B
Hong	I	I	I	I	I
Kong	I	I	E	E	E
Monday	B	B	I	E	S
,	O	O	O	O	O
gold	I	B	I	E	S
was	O	O	O	O	O

各 chunk に対し，その chunk の役割を示すタグを付与する場合は，B/E/I/O/S といった chunk の状態を示すタグと，役割を示すタグを '-' で連結し新たなタグを導入することによって表現する．例えば，IOB2 モデルにおいて，動詞句 (VP) の先頭の単語は B-VP というタグを付与すればよい．

3.2 SVM による Chunk 同定

基本的に SVM は 2 値分類器である．そのため，chunk のタグ表現のように多値の分類問題を扱うためには SVM に対し何らかの拡張を行う必要がある．一般に，2 値分類器を多値分類器に拡張する手法として，以下に述べる 2 種類の手法がある．一つは，*one class vs. all others* と呼ばれる手法で，K クラスの分類問題に対し，あるクラスかそれ以外かを分類する計 K 種類の分類器を作成する手法である．もう一つは，*pairwise* 法であり，各クラス 2 つの組み合わせを分類する $K \times (K-2)/2$ 種類の分類器を作成し，最終的にそれらの多数決でクラスを決定する手法である．また，Dietterich や Allwein らは，上記の二つを含む形で，二値分類を多値分類

器に拡張するための統一的な手法を提案している (Dietterich, Bakiri 1995; Allwein, Schapire, Singer 2000).

本稿では，多値分類器への拡張手法として *pairwise* 法を採用した．採用の理由として以下が挙げられる．

- 一般に，SVM は $O(n^2) \sim O(n^3)$ (n は学習データのサイズ) の学習コストを要求する．そのために，個々の二値分類器に用いられる学習データのサイズが小さければ，学習コストを大幅に削減することができる．*pairwise* 法は，*one class vs. others* に比べ多くの二値各分類器を作成するが，各二値分類器に用いられる学習データは少量であり，全体的に学習のコストを小さくすることができる．
- *pairwise* 法が実験的に良い結果が得られたという報告 (Kreßel 1999) がある．

chunk タグの学習に用いる素性としては，現在の単語およびその周辺の単語や品詞といった文脈を用いる．具体的には，位置 i の chunk タグ c_i の推定を行う素性として c_i 自身の単語と品詞，および右 2 つ，左 2 つの単語と品詞を用いた．また，左 2 つの chunk タグも素性として使用した．

さらに，解析方向を逆 (右向きから左向き) にし，右 2 つの chunk を素性として使用することも考えられる．本稿では，これら 2 つの解析手法を**前向き解析/後ろ向き解析**と呼び区別する．

			→ 解析方向 →		
単語:	w_{i-2}	w_{i-1}	w_i	w_{i+1}	w_{i+1}
品詞:	t_{i-2}	t_{i-1}	t_i	t_{i+1}	t_{i+1}
chunk:	c_{i-2}	c_{i-1}	$\boxed{c_i}$		

一般に，左 2 つ (後ろ向きの場合は右 2 つ) の chunk タグは学習データに対しては付与されているが，テストデータに対しては付与されていない．そこで実際の解析時には，これらの素性は左から右向きに (後向きの場合は右から左に) 解析しながら動的に追加していくこととした．

このような処理は，一種の動的計画法 (DP) と考えることができる．すなわち，全体として最尤な chunk タグ列は，各 chunk タグに付与されるある種のスコアの和が最大になるようなタグ列を選択することにより決定される．さらに，動的計画法を行う際に，解析の ビーム幅を指定することで曖昧性の候補の爆発を抑えることができる．

CoNLL2000 の shared task において，我々はスコアとして pairwise 時の投票数，また，ビーム幅を 5 として解析を行っている (Kudo, Matsumoto 2000b)．本稿では，このような曖昧性を考慮したビーム幅付きの解析は行わず，ビーム幅 1 の決定的な解析を行った．その理由としては以下が挙げられる．

- 我々の詳細な調査の結果，ビーム幅を大きく設定しても，顕著な精度向上に繋がらず，決定的な解析でも十分な解析精度が得られることが分かった．
- 本稿の目的は，後述する重み付き多数決の手法を比較することであり，単純な設定にす

ることで，個々の重み付け手法の相違点を明確にすることができる．

3.3 重み付き多数決

重み付き多数決とは，1つの学習器で出力を得るのではなく，学習データ，学習データの表現方法，素性の選択手法，学習アルゴリズム，あるいは学習アルゴリズムのパラメータ等の異なる複数の学習器を線形結合して出力を得るアルゴリズムのことを指す．このような重み付き多数決の手法は，潜在的にマージン最大化の効果があり，汎化能力の高い強学習アルゴリズムを作成できることが理論的にも実験的にも明らかになっている．

ここで，多数決がなぜ精度向上に繋がるのか，その簡単な証明を行う．重み付き多数決に用いる学習器の1つを $f_i \in \mathbf{R}$，さらに学習すべき対象 (正解) を $t \in \mathbf{R}$ とする．また，f_i の M 個を均一な重み $1/M$ で線形結合した学習器 を $f' = \frac{1}{M}\sum_{i=1}^{M} f_i$ とする．この時，f_i と t，および f' と t の二乗誤差の期待値には以下のような関係が成立する．ただし $E[x]$ は，x の期待値を表現する．

$$\begin{aligned} E[(f'-t)^2] &= E[(\tfrac{1}{M}\textstyle\sum_{i=1}^{M} f_i - t)^2] \\ &= E[\tfrac{1}{M}\textstyle\sum_{i=1}^{M}(f_i - t)^2 - \tfrac{1}{M}\sum_{i=1}^{M}(f_i - \tfrac{1}{M}\sum_{i=1}^{M} f_i)^2] \\ &\leq E[\tfrac{1}{M}\textstyle\sum_{i=1}^{M}(f_i - t)^2] \\ &= E[(f_i - t)^2] \end{aligned} \tag{5}$$

式5より，多数決を行った学習器の二乗誤差の期待値のが，単独に学習した学習器の期待値より小さくなることが分かる．ここでは，証明を簡単にするために均一な重みとしたが，不均一な重みの場合に対する一般化も可能である．詳細については文献 (Haykin 1999) を参照されたい．この重み付き多数決の概念の一つとして Boosting (Freund, Schapire 1996) があり，自然言語処理の多くのタスクに応用され高い精度を示している．

Chunk 同定問題においても，重み付き多数決の手法が適用されている．例えば，Tjong Kim Sang らは，base NP 同定の問題に対し，弱学習アルゴリズムに MBL，ME，IGTree 等の7種類のアルゴリズム，さらに IOB1/IOB2/IOE1/IOE2 の4種類の表現を用いて独立に学習した複数のモデルの重み付き多数決を行うことで，個々のモデルのどれよりも高精度の結果が得られたと報告している (Tjong Kim Sang 2000a; Tjong Kim Sang et al. 2000)．

本稿では，弱学習アルゴリズムに SVM を用い，IOB1/IOB2/IOE1/IOE2 の4種類の表現，さらに解析方向 (前向き/後ろ向き) の合計 $4 \times 2 = 8$ 種類の重み付け多数決を行うことで精度向上を試みる．

IOB1/IOB2/IOE1/IOE2 には，それぞれ次のような特徴がある．IOB1/IOE2 は，chunk が連続したときのみ他とは異なるタグ (B/E) が付与される．つまり，chunk が連続するような事例に特化した学習が行われる．また，IOB2/IOE2 は，chunk の開始/終了位置に他とは違った

タグ (B/E) が付与される．これらは，chunk の開始/終了位置に特化した学習が行われる

さらに，chunk 中の主辞 (Head) となる単語は，chunk の成立に必要不可欠であるために，他の単語に比べ頻出し，同定が容易である．主辞が chunk の先頭にある場合は，前向きに解析を行うことで，主辞が最初に決定され，その結果が後続するタグの素性に影響を及ぼすため，全体として高い精度が期待できる．逆に，主辞が chunk の末尾にある場合は，後ろ向きに解析を行ったほうが高い精度が得られる．前向き/後ろ向きとは，すべての chunk の主辞が先頭/末尾にあると仮定し，それぞれの仮定に特化した学習手法である．

このように，chunk の表現方法，及び解析方向の異なる複数の学習器を作成することで，それぞれ視点の異なる複数の学習器が作成される．一般に，複数の学習器の性質が異なれば異なるほど，多数決の結果の精度が高くなるために，単独のタグ表現方法，及び解析方向の手法より高い精度が期待できる．

重み付き多数決を行う場合，各モデルの重みをどう決定するかが問題となる．真のテストデータに対する精度を用いることで良い結果を得ることができるが，一般に真のテストデータを評価することは不可能である．Boosting では学習データの頻度分布を変更しながら，各ラウンドにおける学習データに対する精度を重みとしている．しかしながら，SVM は，Soft Margin パラメータ，Kernel 関数の選択次第で，学習データを完全に分離することができ，単純に学習データに対する精度を重みにすることは困難である．

本稿では，重み付き多数決の重みとして以下の 4 種類の手法を提案し，それぞれの手法の精度や計算量などを考察する．

(1)　**均一重み**

これは，すべてのモデルに対し均一の重みを付与する手法である．最も単純な手法であり，他の手法に対するベースラインとなる．

(2)　**交差検定**

学習データを N 等分し，$N-1$ を学習データ，残りの 1 をテストとして評価する．この処理を N 回行い，それぞれの精度の平均を各モデルの重みとして利用する．

(3)　**VC bound**

式 (2)，式 (3) を用いて VC bound を計算し，その値から正解率の下限を推定し[3]，重みとする手法である．ただし，式 (3) における全事例を囲む最小直径 D は各学習データから原点までのノルム最大値を用いて近似を行った．

$$D^2 \sim \max_i \{K(\mathbf{x}_i, \mathbf{x}_i) - 2K(\mathbf{x}_i, O) + K(O, O)\} \quad (O : \text{原点})$$

(4)　**Leave-One-Out(L-O-O) bound**

式 (4) の Leave-One-Out bound を求め，正解率の下限を推定し，重みとする手法である．

3 エラー率の上限であるため，1 からこの値を引き，正解率の下限とみなす．

実際の解析は以下のように行った．

(1)　学習データを IOB1/IOB2/IOE1/IOE2 の各表現に変換する．

(2)　4 つの表現に対し，前向き解析，後ろ向き解析の計 $4 \times 2 = 8$ 種類のモデルを作成し，SVM で独立に学習する．

(3)　8 種類のモデルに対し，VC bound，Leave-One-Out bound を計算し重みを求める．交差検定に関しては，(1)，(2) の処理を各分割したデータに対して行い，各ラウンドのタグ付け精度の平均を重みとする．実際の実験では，交差検定における分割数 N は，5 とした．

(4)　合計 8 種類のモデルを用いて学習データとは別のテストデータを解析する．個々の 8 種類のモデルが出力する chunk の表現は，それぞれ異なるため，そのままでは多数決を行うことができない．多数決を行うためには，個々の結果を 1 つの統一表現に変換する必要がある．この目的のために，解析後のデータを IOB1/IOB2/IOE1/IOE2 の各表現に再び変換する．

(5)　IOB1/IOB2/IOE1/IOE2 の個々に変換された結果に対し，タグレベルで合計 8 種類の重みつき多数決を行う[4]．つまり各重み付けの手法に対し，IOB1/IOB2/IOE1/IOE2 の 4 種類の表現方法で評価した結果を得ることとなる．最終的に，4 (統一表現のタイプ) × 4 (重み付けの方法) = 16 種類の結果を得ることとなる．

重み付き多数決の候補として，IOBES 前向き解析と IOBES 後向き解析の各モデルを参加させることは可能であるが，我々はそのような実験を行わなかった．その理由として，推定すべきクラスの数が IOB1，IOB2，IOE1，IOE2 モデルは 3 に対し，IOBES モデルは 5 と異なり，VC bound や，Leave-One-Out bound を 同じ条件で比較することが困難なことが挙げられる．IOBES 前向き解析と IOBES 後向き解析の各モデルの実験は，IOB1，IOB2，IOE1，IOE2 の各モデルとの精度を比較するために行った．

4　実験と考察

4.1　実験環境，設定

実験には以下の 2 種類のタグ付きデータを用いた．

- base NP 標準データセット (**baseNP**)
  Penn Tree-bank/WSJ の 15-18 を学習データ，00-14，19-24 をテストデータとし，Brill Tagger(Brill 1995) を用いて part-of-speech (POS) を付与したデータである．テストデータのサイズ以外は，base NP 抽出に用いられるデータとして一般的なものである．

4 実際には chunk レベルで行わないと chunk の整合性が取れなくなる可能性があるが，本稿では問題を簡単にするためタグレベルで多数決を取ることとした

	トークン (単語) 数	chunk 数	文数
baseNP 学習データ	211,727	53,371	8,936
baseNP テストデータ	962,039	248,656	40,272
chunking 学習データ	211,727	104,893	8,936
chunking テストデータ	962,039	483,301	40,272

表 1 実験データ

- Chunking データセット (**chunking**)
  base NP 標準データセットと基本的に同一であるが，base NP 以外に VP, PP, ADJP, ADVP, CONJP, INITJ, LST, PRT, SBAR の合計 10 種類の英語の句を表現するタグが付与されている．テストデータのサイズを除けば，CoNLL-2000 Shead Task(Tjong Kim Sang and Buchholz 2000) と同一のデータである．

それぞれのデータのサイズを表 1に示す．

実験には SVM 学習パッケージ *Tiny*SVM を用いた[5]．このツールは，本実験のようなバイナリの素性表現に特化して高速化が施されており，VC bound を自動的に推定する機能を持っている．また，すべての実験において，Kernel 関数は 2 次の 多項式 Kernel を使用した．

評価方法としては，適合率と再現率の調和平均で与えられる F 値 ($\beta = 1$) を用いた．これは chunk 同定において一般的に用いられる評価方法である．以後，特にことわらない限り F 値のことを精度と呼ぶ．

4.2 実験結果

表 2に，各 chunk の表現方法，および解析方向が異なる計 8 種のモデルで独立に学習した実験結果 (テストデータに対する精度，推定された重み) をまとめた．また，比較対象として，Start/End 法を用いた学習結果についても示している．

さらに，表 3に，これらを 均一重み， 交差検定 ($N = 5$)， VC bound， Leave on Out bound の 4 種類の重み付けで多数決を行った際の結果をまとめた． 表 4には，各の重み付け手法の中の最良の結果について，その適合率と再現率を示す．

4.3 Chunk の表現方法と解析精度

表 2から，Inside/Outside に基づく 8 つの手法を比較すると，「IOE2 + 後ろ向き」が最良の精度を，「IOE1 + 前向き」が最低の精度を示すことが分かる[6]．これは，以下に述べる我々の直観と合致する．

5 http://cl.aist-nara.ac.jp/~taku-ku/software/TinySVM/ から入手可能

6 chunking データセット の 「IOE1 + 後ろ向き」 以外は，「IOE2 + 後ろ向き」 の結果が 10% の棄却率で有意であることが確認された．

学習条件		精度	推定された重み		
学習データ	変換先	$F_{\beta=1}$	交差検定	VC bound	L-O-O bound
baseNP	IOB1-前	94.04	.9394	.4310	.9193
	IOB1-後	94.08	.9422	.4351	.9184
	IOB2-前	94.13	.9410	.4415	.9172
	IOB2-後	94.13	.9407	.4300	.9166
	IOE1-前	93.91	.9386	.4274	.9183
	IOE1-後	94.14	.9425	.4400	**.9217**
	IOE2-前	94.09	.9409	.4350	.9180
	IOE2-後	**94.23**	**.9426**	**.4510**	.9193
chunking	IOB1-前	93.56	.9342	.6585	.9605
	IOB1-後	93.58	.9346	.6614	.9596
	IOB2-前	93.54	.9341	.6809	.9586
	IOB2-後	93.52	.9355	.6722	.9594
	IOE1-前	93.46	.9335	.6533	.9589
	IOE1-後	**93.65**	.9358	.6669	.9611
	IOE2-前	93.50	.9341	.6740	**.9606**
	IOE2-後	**93.65**	**.9361**	**.6913**	.9597
baseNP	IOBES-前	93.93			
	IOBES-後	93.94			
chunking	IOBES-前	93.36			
	IOBES-後	93.41			

表 2　個々のモデルの精度比較

学習条件		各重み付けに対する精度 $F_{\beta=1}$			
学習データ	評価手法	均一重み	交差検定	VC bound	L-O-O bound
baseNP	IOB1	94.31	94.37	94.39	94.36
	IOB2	94.33	94.39	**94.41**	94.38
	IOE1	94.32	94.38	94.38	94.36
	IOE2	94.33	94.38	94.40	94.38
chunking	IOB1	93.78	93.81	93.81	93.81
	IOB2	93.74	**93.84**	**93.84**	**93.84**
	IOE1	93.79	93.81	93.81	93.81
	IOE2	93.81	93.82	93.83	93.82

表 3　重み付き多数決の結果

- 多くの場合，chunk 中の主辞は末尾の単語となる．すなわち，後ろ向きからから解析すると，主辞を最初に決定できるため優位となる．
  → (後ろ向き > 前向き)
- IOE は，主辞となりやすい chunk の末尾に特化した学習が行われるため，先頭に特化する IOB に比べ優位となる．
  → (IOE > IOB)

データセット	適合率	再現率	$F_{\beta=1}$
baseNP	94.48%	94.34%	94.41
chunking	93.85%	93.83%	93.84

表 4 各データセットに対する最良結果

- IOB は，chunk の先頭を，IOE は，chunk の末尾に特化して学習が行われる．そのため，IOB は，前向き，IOE は後ろ向きから解析すると特化して学習される単語が先に推定されるため，優位となる．
  → (IOB + 前向き > IOB + 後ろ向き，IOE 前向き < IOE 後ろ向き)
- 同一の chunk が連続することは稀である．すなわち，chunk の連続に特化する IOB1/IOE1 は，chunk の先頭/末尾に特化する IOB2/IOE2 に比べ劣る．
  → ({IOB1 IOE1} < {IOB2 IOE2})
- 同一 chunk が連続する場合は，前の chunk の末尾の単語 (主辞) よりはむしろ，後続する chunk の先頭の単語が境界の認定に役割を果たす場合が多い．そのため，chunk が連続する場合は，chunk の先頭に特化する IOB1 が IOE1 に比べ優位となる．
  → (IOB1 > IOE1)

次に Inside/Outside 法 (IOB1/IOB2/IOE1/IOE2 の各手法) と Start/End 法の精度を比較する．颯々野らは，各学習アルゴリズムの特徴を考察しながら，決定リストについては細かい組み合せを考慮する Start/End 法が，最大エントロピー方についてはより粗い情報を考慮する Inside/Outside 法が精度が良いと報告している (颯々野，宇津呂 2000)．SVM を用いた本手法では，全体的に Inside/Outside 法の法が，Start/End に比べ高い精度を示している．SVM は，決定リストのように単独の素性 (ルール) で分類するのではなく，最大エントロピーと同じく複数の素性の線型結合で分類するために，この結果は，颯々野らの分析と合致する．

さらに，別の要因として以下が考えられる．まず，Start/End は，5 種類のタグを使い表現するため，Inside/Outside と比較して，データスパースネスの問題を助長してしまう恐れがある．また，5 種類のタグを使うことで，矛盾のあるタグのシーケンスの数が増えてしまう．具体的には，S → E，I → B，O → I といったタグの連続は，タグ付けとしては不適切である．一方，IOB1 は， O → B のみ，IOB2 は O → I のみが不適切な連続である．タグ付けに関する指針，制約といった「タグ付けスキーマ」は，それらを明示的な形で与えない本手法では，システム自身がデータから学習する必要があり，それだけ余計なコストが生じてしまう．つまり，矛盾のあるタグ列が少ない表現方法が優位であると考える．

4.4 モデル選択能力

重み付き多数決を行う際の重みは，各システムの未知データに対する精度の予測値であるため，これらの大小を比較することでモデル選択が行える．

表 2から，VC bound，交差検定，それぞれが「IOE2 + 後ろ向き」に対し最高の重みを，「IOE1 + 前向き」に最低の重みを算出しており，テストデータに対する精度をうまく予想してる．これらの結果から，VC bound，交差検定がモデル選択基準として良好に機能していることが分かる．

交差検定はモデル選択に用いられる一般的な手法であるが，分割数が多くなると推定に多くの計算量を必要とする．その一方で，VC bound は学習と同時にモデル選択が行え，交差検定に比べ効率的であると考える．

Leave-One-Out bound は 他に比べ計算コストの小さいモデル選択手法であるが，その能力は VC bound や 交差検定よりも劣ることが分かった．

4.5 多数決の効果

表 3から，多数決を行うことで，重みの付与方法によらず，単独のどのモデルよりも精度が向上することが確認できる[7]．

重み付き多数決の手法間の精度差には，多くの場合，顕著な差は見られなかった．特に VC bound，交差検定，Leave-One-Out bound は，ほぼ同等の精度となった．しかし，均一重みと比較して，上記の 3 つ手法で重みを推定するほうが，若干ながら優位であることが分かる．

4.6 関連研究との比較

baseNP データセット

Tjong Kim Sang らは，弱学習アルゴリズムに MBL，ME，IGTree 等の 7 種類のアルゴリズム，さらに IOB1/IOB2/IOE1/IOE2 の 4 種類の表現を用いて独立に学習した複数のモデルの重み付き多数決を行うことで，baseNP データセットに対し 93.86 の精度が得られたと報告している (Tjong Kim Sang 2000a; Tjong Kim Sang et al. 2000)．

我々は単独の表現を用いた場合でも 93.91 - 94.23 の精度を得ている．テストデータが異なるため，厳密な比較は行えないが，SVM 単独の結果は，従来手法と同等だと考える．一方，従来手法は 7 種類の学習アルゴリズム，及び 4 つの chunk 表現の異なるシステムの多数決の結果であり，個々の学習器の学習，及びテストの計算量は，SVM 単独のシステムに比べ大きい．システムの複雑さという観点から見れば，SVM 単独のシステムは，従来手法に比べ優位であると考える．

7 棄却率 10%以下で有意差があると判定された

さらに，従来手法と同様に，各表現の重み付き多数決を行うことで 94.40 の精度を得ることができた．これは，従来法の精度 93.86 に比べ優れていると考える．多数決を実行することは，全体としてシステムが複雑になることが一つの問題点である．Tjong Kim Sang らによる手法は，MBL，ME，IGTree といった，7 種類のアルゴリズムを用いており，全体として複雑になっている．さらに，個々の学習器のパラメータは恣意的に設定されており，これらの最適なパラメータを考慮すると，設定すべきパラメータの数が多く，制御が困難であると考える．一方，本手法は，単一の SVM のみを用い，それ以外の学習アルゴリズムを用いていない．重み付き多数決を行うという観点から見れば，本手法は，従来手法に比べシステム全体の設計が，簡潔であり，設定すべきパラメータ数が少ない．この点も，本手法の優位な点と考える．

CoNLL データセット

CoNLL-2000 Shared Task において我々は SVM と IOB2 と 前向き解析の単独システム用いて 93.48 の精度を報告している (Kudo, Matsumoto 2000b) [8]．本実験結果から，多数決を行うことで,「IOB2 + 前向き」 に限らず，どの単独システムに比べても精度が向上している．また CoNLL-2000 で報告された重み付き多数決に基づく他の手法 (Tjong Kim Sang 2000b) よりも高い精度を示すことができた．

4.7　今後の課題

- 他の分野への応用
  我々の提案する手法は，日本語の文節まとめ上げや固有名詞，専門用語抽出と一般的な chunk 同定問題に応用可能である．我々の提案する手法がこれらの他の分野でも有効であるか実際に検証を行う予定である．
- 可変長モデル
  本稿では，左右 2 つの文脈のみを考慮する単純な固定長モデルを採用した．しかし実際には，個々の chunk を同定に必要な文脈長は可変であり，個々の chunk に対し最適な文脈長を選択することでさらなる精度向上が期待できる．颯々野らは日本語の固有名詞抽出において可変長モデルを提案し単純な固定長のモデルより高い精度が得られたと報告している (颯々野，宇津呂 2000)．今後 このような可変長のモデルを取りいれたいと考えている．
- より予測能力の高い bound の採用
  本稿では，重み付き多数決の重みとして，SVM に固有の概念 — VC bound，Leave-One-Out bound を提案した．その一方で Chapelle らは，これらより予測能力の高い bound を提案し，Kernel 関数の選択や Soft Margin パラメータの選択に極めて有効であるとこ

8 テストデータが異なるため精度に若干差が出ている．

を示している (Chapelle, Vapnik 2000)．これらの予測能力の高い bound を重みとして採用することでさらなる精度向上が期待できる．

5 まとめ

本稿では，Support Vector Machine (SVM) に基づく一般的な chunk 同定問題の解析手法を提案し，実際のタグ付きコーパスを用いて実験を行った．英語の単名詞句抽出における実験では，複数のシステム混合に基づく従来のモデルと同等の精度を示し，SVM の持つ高い汎化能力を裏づける結果となった．

また，chunk の表現方法や解析方向の異なる複数のシステムの中から最適なものを選択するための「モデル選択基準」として，本稿で採用した VC bound は，従来からある交差検定と同程度の予測性能があることが確認された．VC bound は，交差検定のように学習を繰り返す必要がなく，学習と同時に計算が可能であるため，計算量の軽減に繋がる．

さらに，chunk の表現方法や解析方向の異なる複数のシステムの重み付き多数決を行うことで，個々のどのモデルよりも高い精度を示した．

参考文献

Allwein, E. L., Schapire, R. E., and Singer, Y. (2000). "Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers." In *International Conf. on Machine Learning (ICML)*, pp. 9–16.

Brill, E. (1995). "Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging." *Computational Linguistics*, **21** (4).

Chapelle, O. and Vapnik, V. (2000). "Model Selection for Support Vector Machines." In *Advances in Neural Information Processing Systems 12*. Cambridge, Mass: MIT Press.

Cortes, C. and Vapnik, V. N. (1995). "Support Vector Networks." *Machine Learning*, **20**, pp. 273–297.

Dietterich, T. G. and Bakiri, G. (1995). "Solving multiclass learning problems via error-correcting output codes." *Journal of Artificial Intelligence Research*, **2**, pp. 263–286.

Freund, Y. and Schapire, R. E. (1996). "Experiments with a new Boosting algorithm." In *International Conference on Machine Learning (ICML)*, pp. 148–146.

Haykin, S. (1999). *Neural Networks: a comprehensive foundation - 2nd ed.* Prentice-Hall.

Joachims, T. (1999). "Transductive Inference for Text Classification using Support Vector Machines." In *International Conference on Machine Learning (ICML)*.

Kreßel, U. H.-G. (1999). "Pairwise Classification and Support Vector Machines." In *Advances in Kernel Mathods.* MIT Press.

Kudo, T. and Matsumoto, Y. (2000a). "Japanese Dependency Structure Analysis Based on Support Vector Machines." In *Empirical Methods in Natural Language Processing and Very Large Corpora*, pp. 18–25.

Kudo, T. and Matsumoto, Y. (2000b). "Use of Support Vector Learning for Chunk Identification." In *Proceedings of the 4th Conference on CoNLL-2000 and LLL-2000*, pp. 142–144.

Kudo, T. and Matsumoto, Y. (2001). "Chunking with Support Vector Machines." In *Proceedings of NAACL-2001*, pp. 192–199.

工藤，松本 (2002). "チャンキングの段階適用による係り受け解析." 情報処理学会論文誌, **43** (6), 1834.

Ramshaw, L. A. and Marcus, M. P. (1995). "Text Chunking using Transformation-Based Learning." In *Proceedings of the 3rd Workshop on Very Large Corpora*, pp. 88–94.

颯々野，宇津呂 (2000). "統計的日本語固有表現抽出における固有表現まとめ上げ手法とその評価." 情報処理学会 自然言語処理研究回 NL139-2, pp. 1–8.

平，春野 (2000). "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, **41** (4), 1113.

Tjong Kim Sang, E. F. (2000a). "Noun phrase recognition by system combination." In *Proceedings of ANLP-NAACL 2000*, pp. 50–55.

Tjong Kim Sang, E. F. (2000b). "Text Chunking by System Combination." In *Proceedings of CoNLL-2000 and LLL-2000*, pp. 151–153.

Tjong Kim Sang, E. F. and Buchholz, S. (2000). "Introduction to the CoNLL-2000 Shared Task: Chunking." In *Proceedings of CoNLL-2000 and LLL-2000*, pp. 127–132.

Tjong Kim Sang, E. F., Daelemans, W., Déjean, H., Koeling, R., Krymolowski, Y., Punyakanok, V., and Roth, D. (2000). "Applying System Combination to Base Noun Phrase Identification." In *Proceedings of COLING 2000*, pp. 857–863.

Tjong Kim Sang, E. F. and Veenstra, J. (1999). "Representing text chunks." In *Proceedings of EACL'99*, pp. 173–179.

内元，馬青, 村田，小作，内山，井佐 (2000). "最大エントロピーモデルと書き換え規則に基づく固有名詞抽出." 自然言語処理, **7** (2).

Vapnik, V. N. (1998). *Statistical Learning Theory.* Wiley-Interscience.

略歴

工藤 拓: 1976 年生．1999 年京都大学工学部電気電子工学科卒，2001 年奈良先端科学技術大学院大学情報科学研究科博士前期課程修了，同年 同大学院博士

後期課程に進学．専門は統計的自然言語処理．機械学習，統計的手法に興味を持つ．

松本 裕治: 1955 年生．1977 年京都大学工学部情報工学科卒．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．1984～85 年英国インペリアルカレッジ客員研究員．1985～87 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．工学博士．専門は自然言語処理．情報処理学会，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

(2001 年 11 月 25 日 受付)
(2002 年 3 月 1 日 再受付)
(2002 年 5 月 10 日 再々受付)
(2002 年 7 月 17 日 採録)