自然言語処理　138－11
（2000. 7. 19）

Support Vector Machine による日本語係り受け解析

工藤 拓　　松本 裕治

奈良先端科学技術大学院大学 情報科学研究科
〒630-0101 奈良県生駒市高山町 8916-5
Tel: 0743-72-5246
{taku-ku,matsu}@is.aist-nara.ac.jp

あらまし

本稿では，Support Vector Machine (SVM) に基づく日本語係り受け解析手法を提案し，その評価を行なう．既に，決定木や最大エントロピー法等が統計的係り受け解析の学習モデルとして提案されているが，これらの手法は慎重な素性選択が要求されたり，素性の組み合わせを効率良く学習できないといった問題がある．その一方で，SVM は従来からある学習モデルと比較して，入力次元数に依存することなく極めて高い汎化能力を持ち，さらに Kernel 関数という概念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学習することが可能である．SVM を係り受け解析に適用し，京大コーパスを用いて実験評価を行なった結果，5540 文という非常に少ない学習データにもかかわらず 88.66% の高い精度を示した．

キーワード　構文解析，係り受け解析，機械学習，Support Vector Machine

Japanese Dependency Structure Analysis based on Support Vector Machine

Taku Kudoh　　Yuji Matsumoto

Graduate School of Infomation Science, Nara Institute Science and Technology
8916-5 Takayama-cho, Ikoma-shi Nara 630-0101 Japan
Tel: +81-743-72-5246
{taku-ku,matsu}@is.aist-nara.ac.jp

Abstract

This paper describes an analysis method of Japanese dependency structure based on Support Vector Machine (SVM). Conventional parsing techniques based on Machine Learning framework, such as Decision Tree and Maximum Entropy Model, cannot analyze sentence precisely, since features in these models must be selected carefully and it is difficult to train combination of features. On the other hand, it is well-known that SVM achieves high generalization performance even with high dimensional input data. Furthermore, by introducing the kernel principle, SVM can carry out the training in high-dimensional spaces with smaller computational cost independent of their dimensionality. We apply SVM to Japanese dependency structure identification problem. Experimental results on Kyoto University corpus show that our system performs 88.66% even with small training data (5540 sentences).

key words　Parsing, Dependency Structure Analysis, Machine Learning, Support Vector Machine

1 はじめに

日本語の構文解析において係り受け解析は，自然言語処理の基本技術の一つとして認識されており，従来から多くの研究が行なわれている．係り受け解析は，文中の任意の二文節間の係りやすさを数値化した行列を作成し，その中から動的計画法を用いて文全体を最適にする係り受け関係を求めることである．従来，係り受け解析の研究は，二文節間の係りやすさを決定するルールを人手で作成していた．しかし，係り受け解析で用いられる素性集合は膨大で，それらは競合することが多いため，網羅性，一貫性という面で問題が多い．

近年，係り受け情報が付与された大規模コーパスが利用可能になりコーパスから二文節の係りやすさを統計的に求める手法が提案されている．既に提案されている統計的学習モデルとして，決定木 [10] や最大エントロピー (ME) 法 [11] などがあるが，これらの手法は学習に用いる素性を適切に選択しないと過学習を起こしてしまったり，複数の組み合わせで効いてくるような素性を効率良く学習できないといった問題が指摘されている．

一方，統計的機械学習の分野では，Support Vector Machine (SVM) [1] [7] や Boosting [2] 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．これらの手法は，例外的な事例も含め，全学習データを用い大域的な最適解を求めるため，過学習しにくい．特に SVM は従来からある学習モデルと比較して極めて汎化能力が高く，高次元の素性集合を用いても過学習しにくいとされている．さらに，Kernel 関数を変更することで，非線型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 は手書き文字の認識や3次元画像の認識等数多くの分野に応用されている．自然言語処理の分野においては，文書分類に応用されており，素性となる単語を増やしても過学習することなく非常に高い精度が得られたと報告されている [3] [5] [12] [13].

本稿では，解析済みコーパスから統計的にモデルを学習する立場をとりながら，従来，統計的係り受け解析において有効とされてきた素性を使用し，学習モデルとして，Support Vector Machine (SVM) を用いた係り受け解析について述べる．

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章で一般的な係り受けモデル，および SVM の具体的な適用方法について述べる．さらに 4 章で京大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 Optimal Hyperplane

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，

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

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 ($\mathbf{x}_i = (f_1, f_2, \ldots, f_n) \in \mathbf{R}^n$) で表現される．$y_i$ はデータ i が，正例 (1)，負例 (-1) かを表わすスカラーである．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．

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

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．そこでまず，正例，負例のそれぞれを以下の 2 つの領域に分割する．

$$(\mathbf{w} \cdot \mathbf{x}_i) + b \geq 1 \qquad \text{if } (y_i \in 1) \tag{2}$$

$$(\mathbf{w} \cdot \mathbf{x}_i) + b \leq -1 \qquad \text{if } (y_i \in -1) \tag{3}$$

式 (2),(3) をまとめると，

$$y_i[(\mathbf{w} \cdot \mathbf{x}_i) + b] \geq 1 \qquad (i = 1, \ldots, l) \tag{4}$$

となる．式 (4) に属さない領域がマージンの領域となる．さらに，式 (1) の平面から点 $\mathbf{x}_i$ までの距離 $d(\mathbf{w}, b; \mathbf{x}_i)$ は，

$$d(\mathbf{w}, b; \mathbf{x}_i) = \frac{|\mathbf{w} \cdot \mathbf{x}_i + b|}{\|\mathbf{w}\|}$$

となり，式 (2) (3) 間のマージンは

$$\begin{aligned} & \min_{\mathbf{x}_i; y_i = 1} d(\mathbf{w}, b; \mathbf{x}_i) + \min_{\mathbf{x}_i; y_i = -1} d(\mathbf{w}, b; \mathbf{x}_i) \\ = & \min_{\mathbf{x}_i; y_i = 1} \frac{|\mathbf{w} \cdot \mathbf{x}_i + b|}{\|\mathbf{w}\|} + \min_{\mathbf{x}_i; y_i = -1} \frac{|\mathbf{w} \cdot \mathbf{x}_i + b|}{\|\mathbf{w}\|} \\ = & \frac{2}{\|\mathbf{w}\|} \end{aligned}$$

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，この問題は結局以下の制約付き最適化問題を解くことと等価となる．

$$\text{目的関数：}\quad L(\mathbf{w}) = \|\mathbf{w}\|^2 \rightarrow \text{最小化}$$
$$\text{制約条件：}\quad y_i[(\mathbf{w}\cdot\mathbf{x}_i)+b] \geq 1 \ (i = 1 \ldots l)$$

この最適化問題は，数理計画法の分野で，2 次計画法として知られており，高速ないくつかのアルゴリズムが考案されている [4]

更にこの最適化問題を，より扱いやすい形に変換すると，Lagrange 乗数 $\alpha_i \geq 0,\ (i = 1,\ldots,l)$ を使って，以下のような双対問題に帰着される．

目的関数:

$$L(\alpha) = \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j(\mathbf{x}_i\cdot\mathbf{x}_j) \qquad (5)$$
$$\rightarrow \text{最大化}$$

制約条件:

$$\alpha_i \geq 0,\ \sum_{i=1}^{l}\alpha_i y_i = 0 \qquad (i = 1,\ldots,l)$$

双対問題において $\alpha_i \geq 0$ となる $\mathbf{x}_i$ のことを Support Vector と呼び，Support Vector を使うと，$\mathbf{w}$,b は，

$$\mathbf{w} = \sum_{i;\mathbf{x}_i\in SVs}\alpha_i y_i \mathbf{x}_i$$
$$b = \mathbf{w}\cdot\mathbf{x}_i - y_i$$

となる．最終的に識別関数 $f : \mathbf{R}^n \rightarrow \{\pm 1\}$ は，

$$f(\mathbf{x}) = \text{sgn}\left(\sum_{i;\mathbf{x}_i\in SVs}\alpha_i y_i\,(\mathbf{x}_i\cdot\mathbf{x}) + b\right) \qquad (6)$$
$$= \text{sgn}\,(\mathbf{w}\cdot\mathbf{x} + b)$$

で与えられる．

2.2 Soft Margin

学習データを線型分離できない場合，多少の識別誤りは許すように制約を緩める **Soft Margin** と呼ばれる手法を適用することができる．まず，新たに非負の変数 $\xi_i \geq 0,\ (i = 1,\ldots,l)$ を導入し，式 (2),(3) の代わりに，

$$(\mathbf{w}\cdot\mathbf{x}_i) + b \geq 1 - \xi_i \qquad \text{if } (y_i \in 1)$$
$$(\mathbf{w}\cdot\mathbf{x}_i) + b \geq -1 + \xi_i \qquad \text{if } (y_i \in -1)$$

を考える．この場合，$\|\mathbf{w}\|^2$ の最小化の代わりに，

$$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l}\xi_i \qquad (7)$$

の最小化を考える．第一項はマージンの大きさに関する項であり，第二項は分離できなかった学習データがそれぞれの分離平面からどれだけ離れているかを示す項となる．C はこれら二つの項の度合を決めるパラメータである．C の値が大きいときは識別誤りを小さくするように評価され，逆に C が小さいときは多少の識別誤りを認めながら全体のマージンが大きくなるように評価される．詳細は省略するが，式 (7) は，目的関数 (5) を以下のような制約のもとで最小化することに帰着される．

$$0 \leq \alpha_i \leq C, \qquad \sum_{i=1}^{l}\alpha_i y_i = 0 \qquad (i = 1,\ldots,l)$$

通常，C の値は実験的に設定される．

2.3 Kernel 関数

一般的な分類問題においては，学習データを線型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型分離が容易になる．しかしながら，単純に学習データの組みあわせをすべて展開し，高次元空間へ写像すると莫大な計算量が必要となる．

仮に，入力パターン $\mathbf{x}$ を 写像関数 Φ によって他の高次元な空間に写像した場合を考えてみる[1]．ここで，目的関数 (5) および，識別関数 (6) に注目してみると，これらは入力パターンの内積のみに依存した形になっている．この点に着目すると，もしこの写像された空間上のベクトル $\Phi(\mathbf{x}_1)$,$\Phi(\mathbf{x}_2)$ の内積が写像関数 Φ を経由せずに $\mathbf{x}_1$,$\mathbf{x}_2$ から直接計算できるならば大幅に計算量を減らすことが可能となる．つまり、

$$\Phi(\mathbf{x}_1)\cdot\Phi(\mathbf{x}_2) = K(\mathbf{x}_1,\mathbf{x}_2) \qquad (8)$$

となる関数 K が存在すればよい．逆に，Φ は実際の学習，識別には必要ないため，適当に関数 K を選んだ時に，式 (8) を満たす Φ の存在のみが証明されればよい．実際の証明は省略するが，式 (8) を満たすには，関数 K が Mercer 条件を満足すればよいことが知られている．

[1] $\Phi(\mathbf{x})$ は一般的に Hilbert 空間への写像となる．

このようにして, 最適化問題や識別問題における内積を関数 K に置きかえることにより, 実際には, Φ により高次元な空間に写像しているにもかかわらず, その計算量を大幅に減少させることが可能となる. この時の関数 K は **Kernel 関数**と呼ばれ, 簡単なものとして,

$$K(\mathbf{x},\mathbf{y}) = \tanh(a\,\mathbf{x}\cdot\mathbf{y}-b) \quad (9)$$

$$K(\mathbf{x},\mathbf{y}) = \exp\left(\frac{-\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}\right) \quad (10)$$

$$K(\mathbf{x},\mathbf{y}) = (\mathbf{x}\cdot\mathbf{y}+1)^d \quad (11)$$

などが知られている[2].

Kernel 関数を用いた場合の識別関数は,

$$y = \mathrm{sgn}\left(\sum_{i;\mathbf{x}_i\in SVs}\alpha_i y_i K(\mathbf{x}_i,\mathbf{x})+b\right)$$

となる. これに各 Kernel 関数を代入すると, 式 (9) は 3 層のニューラルネットワークの形になっており, 式 (10) は, Radial Basis Function (RBF) ネットワークと呼ばれるモデルになる. また式 (11) は多項定理を用いて展開すると, 各素性の d 個の組みあわせを考慮した学習モデルに帰着することができる.

3 SVM に基づく係り受け解析

3.1 統計的係り受けモデル

本節では日本語における一般的な統計的係り受けモデル, 及び解析手法について説明する. まず, あらかじめ文節にまとめられ属性付けされた文節列 $\{b_1, b_2, \ldots, b_m\}$ を B, 係り受けパターン列 $\{Dep(1), Dep(2), \ldots, Dep(m-1)\}$ を D と定義する. ただし,$Dep(i)$ は, 文節 b_i の係り先文節番号を示す.

これ以降, D は以下の制約を満たすものと仮定する.

1. 文末を除き, 各文節はその文節の後方側に必ず 1 つの係り先を持つ.
2. 係り受け関係は交差しない.

統計的係り受け解析とは, 上記の二つの制約のもとで, 入力文節列 B に対する条件付き確率 $P(D|B)$ が最大にする係り受けパターン列 D を求めることと定義できる.

$$D_{best} = \underset{D}{\mathrm{argmax}}\, P(D|B)$$

さらにここで, それぞれの係り関係は独立であると仮定すると,$P(D|B)$ は,

$$P(D|B) = \prod_{i=1}^{m-1} P(Dep(i)=j\,|\,\mathbf{f}_{ij})$$
$$\mathbf{f}_{ij} = \{f_1,\ldots,f_n\} \in \mathrm{R}^n$$

のように変形できる. ここで, 確率 $P(Dep(i)=j\,|\,\mathbf{f}_{ij})$ は文節 b_i と文節 b_j が言語的素性集合 $\mathbf{f}_{ij}$ を持つ時に, 文節 b_i が文節 b_j に係る確率を示す. $\mathbf{f}_{ij}$ は文節 b_i と文節 b_j に関する種々の言語的特徴を表わす n 次元の特徴ベクトルである.

最終的に, これらの確率値をもとに D_{best} を決定する. これは, CYK などのボトムアップパーザを用いて解析するのが一般的であるが, 関根らは, 日本語の係り受け関係の制約をうまく利用し, 文末からビームサーチをしながら効率良く解析する手法を提案している [8]. 我々も関根の手法を実際の解析に用いることとした. 詳細なアルゴリズムに関しては, 文献 [8] を参照されたい.

3.2 SVM に基づく係り受けモデルの学習

SVM は正例, 負例の 2 値分類を行なう学習モデルである. そのためにはまず, 何を正例, 負例として学習するのかを決定しなければならない. 我々は, 非常に単純でかつ有効な方法, つまり, 係り受け候補となりうるすべての組みあわせの中で, 実際に学習データにおいて係った 2 文節を正例, 係らなかった 2 文節を負例として与えた.

$$\bigcup_{\substack{1\le i\le m-1\\ i+1\le j\le m}} (\mathbf{f}_{ij}, y_{ij}) = \{(\mathbf{f}_{12}, y_{12}), (\mathbf{f}_{23}, y_{23}), \ldots, (\mathbf{f}_{m-1\,m}, y_{m-1\,m})\}$$

$$\mathbf{f}_{ij} = \{f_1,\ldots,f_n\} \in \mathrm{R}^n \quad y_{ij} \in \{\text{係る}, \text{係らない}\}$$

また, 係り受け確率 $P(Dep(i)=j\,|\,\mathbf{f'}_{ij})$ には以下の値を与えた.

$$P(Dep(i)=j\,|\,\mathbf{f'}_{ij}) = \tanh\left(\sum_{k,l;\mathbf{f}_{kl}\in SVs}\alpha_{kl}y_{kl}K(\mathbf{f}_{kl},\mathbf{f'}_{ij})+b\right) \quad (12)$$

式 (12) は, 評価データ $\mathbf{f'}_{ij}$ と分離平面間の距離をシグモイド関数に入力した形となっており, 厳密には確率値ではない. 我々は, SVM の距離関数を確率値の値域

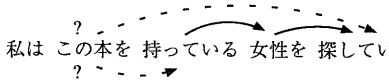
[2] $\tanh(x) = \frac{1}{1+\exp(-x)}$ 一般にシグモイド関数と呼ばれる.

に正規化し，従来からある確率モデルの枠組で解析するために，便宜的に式 (12) を採用した[3].

3.3 静的素性と動的素性

統計的日本語係り受け解析に有効とされてきた素性には，着目している 2 文節の主辞の語彙や品詞，語形の活用形，2 文節間の距離，句読点，引用符の有無などがある．これらの素性は文節のまとめあげの段階で決定される素性であり，このような素性集合のことをまとめて**静的素性**と呼ぶこととする．

日本語の係り受け関係は，語の活用形が大きな制約となり，静的素性だけで係り先の大部分を限定することができる．しかし，複数の係り先の候補がある場合，静的素性だけでは決定しにくいことがある．以下にその例を示す．



この例の場合，静的素性だけでは，「この本を」が「持っている」に係るか「探している」に係るか一意に決定するのは困難である．そこで，注目している 2 文節以外の係り関係を素性として追加することを考えてみる．具体的には，「探している」は，「女性を」という文節から修飾されており，この係り受け関係を文節「探している」の素性として追加する．一般的に同種の格が同一の文節に係ることは稀なので，「探している」はすでに「を」格で修飾されていることとなり，結果として「この本を」は「探している」に係らないことが分かる．このように，着目している 2 文節以外の係り関係を素性として追加することは，この例以外に「～ を ～ に」といった，2 重の格の組を必要とする動詞の係り関係の同定や，並列構造の同定の解析に有効であると考える．

一方で，このように係り関係そのものを素性として使うことを考えるとき，解析途中の文には素性の決定に必要な係り関係が付与されていないために素性を選択できないことが問題となる．しかしこれは一般的なボトムアップパーザを用いた場合は容易に拡張可能である．つまり，ボトムアップに解析しながら，係り関係の素性を動的に追加してゆけばよい．さらに，解析途中の候補のうち，確率値の上位 k 個のみを選択しながら，ビームサーチを行なえば，解析精度の悪化を抑えながら，解析候補数の爆発を防くことができる．

このように解析時に動的に追加されていく素性のことをまとめて**動的素性**と呼ぶこととする．

[3] シグモイド関数は，SVM の距離関数から確率値へのよい近似を与えることが実験的に示されている [6].

4 実験と考察

我々の提案する手法に基づき，実際のタグ付きコーパスを用いて実験行なった．本章ではその実験結果の報告と考察を行なう．

4.1 実験環境，設定

実験に用いたコーパスは京大コーパス (Version 2.0) [9] の一部で，学習には 1 月 1 日から 6 日分 (5540 文)，テストには 1 月 9 日分の (1246 文) を用いた．Kernel 関数には，式 (11) の多項式関数を用い，ソフトマージンのパラメータ C はすべての実験を通して 1 に固定した．

次に，学習に用いた素性を，表 1 に示す．静的素性は，内元 [11] が用いたもの基に，係り受け解析に有効だと思われる素性を追加したものである．ただし，**主辞**とは文節内で品詞が特殊，助詞，接尾辞となるものを除き，文末に一番近い形態素，**語形**とは文節内で品詞が特殊となるものを除き，文末に一番近い形態素のことを指す．

一方，動的素性には，2 文節間にある文節で，後方の文節に係っている文節の語形の見出し語を与えた．ただし，データスパースネスを考慮し，品詞による簡単なフィルタリングを行なっている．具体的には，助詞，副詞，連体詞，接続詞 については見出し語そのものを，活用形のあるものはその活用形を，その他の品詞については，品詞と品詞細分類を与えた．

4.2 実験結果

実験条件として，解析時のビーム幅 $k = 5$, Kernel 関数の次元数 $d = 3$ のもとで行なった場合の解析結果を表 2 に示す．ただし，係り受け正解率 (A) とは，文末から 2 番目の文節の評価を含めた場合の正解率であり，係り受け正解率 (B) は，文末から 2 番目の文節の評価を除いた場合の正解率と定義する．ここからは特に指定が無い限り，正解率は (A) の定義とする．

静的素性	前/後文節	主辞見出し, 主辞品詞, 主辞品詞細分類, 主辞活用, 主辞活用形, 語形見出し, 語形品詞, 語形品詞細分類, 語形活用, 語形活用形, 括弧の有無, 句読点の有無, 文節の位置 (文頭, 文末)
	文節間	距離 (1,2-5,6 以上), 各助詞, 括弧, 句読点の有無
動的素性	2 文節間にある文節で後ろの文節に係る文節の語形の見出し	

表 1: 使用素性

学習文数	文節正解率 A/B(候補数)		文正解率
	A (11263)	B (10024)	
1172	86.52%	84.86%	39.31%
1917	87.21%	85.62%	40.06%
3032	87.67%	86.14%	42.94%
4318	88.34%	86.90%	44.07%
5540	88.66%	87.26%	45.20%

表 2: 解析結果 ($d = 3, k = 5$)

学習文数	文節正解率 A/B(候補数)		文正解率
	A (11263)	B (10024)	
1172	86.12%	84.41%	38.50%
1917	86.81%	85.18%	39.80%
3032	87.62%	86.10%	42.45%
4318	88.33%	86.89%	44.47%
5540	88.40%	86.96%	43.66%

表 3: 静的素性のみの解析結果 ($d = 3, k = 5$)

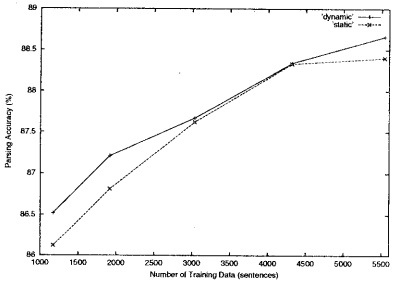


図 1: 学習データと解析精度

4.3 動的素性追加の効果

表 3 に, 静的素性のみを使用した場合の精度を示す. 全体的に, 動的素性を使用するほうが高い結果を出している. とくに学習文数が少ない状況での精度向上は著しい.

本稿における実験では, 動的素性として注目している 2 文節間にあり, 後方の文節に係る文節のみを用いた. しかし, 動的素性にはこの他に, 後方が係っている文節などが考えられる. これらの動的素性の選択について, さらなる実験, 評価が必要であろう.

4.4 学習データと解析精度

図 1 に 学習コーパスの量と精度の関係を示した. 学習データが 1172 文程度という非常に少ない量でも評価データに対して 86.52% の精度が出ており, SVM がデータスパースネスに強いことが分かる.

さらに, 学習データによる評価を行なったところ, 学習データの量によらずほぼ 100% の精度を示した. kernel 関数の次元を 3 次に設定したこともあり, 分離が困難な場合でも素性の組み合わせまで考慮して分離を行ない, 学習データはほぼ完全に分離を行なったことが分かる. 一般的に, このように学習データの細かい素性情報を考慮しながら学習を行なうと, 過学習を起こしてしまい, 評価データに対する精度が低下してしまう. しかし, 我々の提案する手法においては, 学習データ, 評価データともに高い精度が出ており, SVM の持つ極めて高い汎化能力を裏付ける結果となった.

また, 学習曲線から推測するに, 学習データを増やすとさらなる精度向上が期待できる.

4.5 Kernel 関数と解析精度

学習データ 3032 文, ビーム幅 $k = 5$ という条件で, Kernel 関数の次元数 d を $d = 1, 2, 3, 4$ と変化させた時の解析精度を表 4 に示す. 結果として, $d = 4$ の時が最も精度が良く, $d = 1$ の場合は, 現実的な時間内で学習が行えなかった. この結果はある意味, 我々の直観と合致している. つまり, 2 文節間の係り受け関係であるから, 係り元あるいは係り先の素性のみで係り関係が決定されるとは考えにくい. それよりは, 係り元, 係り先, 最低でも 2 つの素性のペアで決定されると考えるのが自然であろう. また, 動的素性を使用しているため,

次元数	係り受け正解率	文正解率
1	N/A	N/A
2	86.87%	40.60%
3	87.67%	**42.94%**
4	**87.72%**	42.78%

表 4: 次元数と解析精度 (3032 文, $k = 5$)

3 つ以上の素性の組み合わせを考慮した場合にさらに精度が向上しているものと考察される.

beam 幅	係り受け正解率	文正解率
1	88.59%	45.04%
3	88.64%	**45.20%**
5	**88.66%**	45.20%
7	88.63%	45.20%
10	88.56%	45.20%
15	88.55%	45.20%
25	88.55%	45.20%

表 5: ビーム幅と解析精度 (5540 文, $d = 3$)

4.6 ビーム幅と解析精度

ビーム幅と解析精度については, 関根 [8] が非常に興味深い報告をしている. 一般に, ビーム幅を高く設定したほうが文全体の確率が最大のものが選択される可能性が高くなるために, 全体の解析精度は向上するのではないかと考えられる. しかし, 関根は, 直観に反しビーム幅が 3-10 程度の時に最高の正解率を与え, それ以上に設定するとかえって精度が低下したと報告している. これは, 日本語の係り関係が, ある局所的な係り関係の最適化から成り立っていることが予想される.

そこで, 我々も同様にビーム幅と解析精度の関連について調べてみた. 表 5 に学習データ 5540 文, kernel 関数の次元数 $d = 3$ の条件下で, ビーム幅を $k = 1 \sim 25$ と変化させた時の解析精度を示す. 我々の提案手法においても, ビーム幅が 5 の時に最高の係り受け正解率を与えている.

これらをふまえると, 最適なビーム幅の設定方法が問題となるだろう. 我々は, 最適な正解率を与えるビーム幅は, その文の長さや含まれている語彙や品詞と何らかの関係があるのでないかと考える. この点に着目し, ビーム幅についてのさらなる実験を進めて行く考えである.

4.7 関連研究との比較

内元 [11], 関根 [8] は, 京大コーパス 7958 文を用いて, ME に基づくモデルを作成し, 約 87.2%の精度を報告している. 我々は, 公平な精度比較を行うために, 内元が用いた学習データの一部を学習データとして使用し, 評価データは同じものを使用した. 我々の提案する手法では, 内元が用いたおよそ 2/3 の 5540 文で, 88.66%の精度が出ており, 単純な精度比較においては十分優位であると考える.

また, 内元も同じく, 素性の組み合わせの重要性を示しているが, ME に基づくモデルでは素性の組み合わせを展開して新たな素性として投入する必要がある. 内元は, 計算量との兼ね合いを考慮しながら, 重要だと思われる素性の組み合わせを人手により発見的に選択しているが, 必ずしも重要な素性の組み合わせを網羅しているとは限らない. 我々の提案する手法は Kernel 関数の変更という操作のみで, 計算量をほとんど変えることなく組み合わせを含めた学習が行なえるため, 網羅性, 一貫性という意味で優位であると考える.

4.8 今後の課題

精度向上に直接結びつく最も簡単で効果的な方法は, 学習データを増やすことである. しかしながら, 係り受け関係となりうるすべての候補を用いる我々の提案手法は, 多くの計算量を必要する. 本稿における実験も, この学習データの量とその際の計算時間の制約から, 5540 文程度の学習データが限界であった[4]

より多くの学習データを用いるには, すべての係り受け候補から, 分類に必要な事例を何らかの方法で選択する必要があると我々は考える. これは学習の能率化はもちろん, 解析の際の探索空間を小さくし, 解析の高速化にもつながる. そこで, 我々は以下に述べる 3 つの手法に基づき事例の選択を行おうと考えている.

- 係らない制約の導入
  明らかに係らない制約ルールを人手により用意しておいて, それにマッチするものは学習を行なわな

[4] AlphaServer 8400 (617Mhz) を使用して, 4318 文を学習するのに 2 日間, 5540 文に 7 日間を要した.

いようにする．例えば，簡単なものとして，「引用符をまたいで係らない」という制約が考えられるであろう．しかし，このように人手で制約ルールを追加していくことは，網羅性，一貫性という問題が多い．

- 他のモデルとの融合
  ある程度の再現率と精度が保証されており，計算コストの小さい学習モデルを用いて，複数の冗長な解析結果を出力させておき，その複数候補からSVM を用いて学習する．統計的な枠組みは変えないために，一貫性，網羅性は維持される．問題は計算量と精度のバランスがとれたモデルとして何を選択すればよいのかということにある．
- 誤り駆動型による素性選択
  まず少量のデータで学習を行ない．他のタグ付き学習コーパスを解析させ，その時誤ったものを選択し，新たな学習データとして追加していく．我々の提案手法の枠組みをほとんど変えることはないが，例外的な現象ばかりが選択される可能性があり，それが原因で精度低下してしまうおそれがある．

5 まとめ

本稿では，Support Vector Machine (SVM) に基づく日本語係り受け解析の解析手法を提案し，実際のタグ付きコーパスを使用して実験を行なった．京大コーパスを用いた実験では，5540 文という非常に少ない学習データにもかかわらず，88.66% の精度を示し，SVM の持つ極めて高い汎化能力と，高次元の入力ベクトルに対しても過学習しにくいという特徴を実証する結果となった．また，統計的係り受け解析においては，素性の組み合わせによる学習が非常に重要であり，その観点から見ても，SVM は網羅性，一貫性，計算量の 3 つの点に関して，従来手法に比べて優位であることが分かった．

参考文献

[1] C. Cortes and V. Vapnik. Support Vector Networks. *Machine Learning*, Vol. 20, pp. 273–297, 1995.

[2] Y. Freund and Schapire. Experiments with a new Boosting algoritm. In *13th International Conference on Machine Learning*, pp. 148+, 1996.

[3] T. Joachims. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In *European Conference on Machine Learning (ECML)*, 1998.

[4] T. Joachims. Making Large-Scale SVM Learning Practical. In *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1999.

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

[6] J. Platt. Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. In *Advances in Large Margin Classifiers*. MIT Press, 1999.

[7] V. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

[8] 関根論，内元清貴，井佐原均．文末から解析する統計的係り受け解析アルゴリズム．自然言語処理，Vol. 6, No. 3, pp. 59–73, 1999.

[9] 黒橋禎夫，長尾眞．京都大学テキストコーパス・プロジェクト．言語処理学会 第 3 回年次大会，pp. 115–118, 1997.

[10] 春野雅彦，白井諭，大山芳史．決定木を用いた日本語係り受け解析．情報処理学会論文誌，Vol. 39, No. 12, p. 3117, 1998.

[11] 内元清貴，関根聡，井佐原均．ME による日本語係り受け解析．情報処理学会 自然言語処理研究会 NL128-5, pp. 31–38, 1998.

[12] 平博順，向内隆文，春野雅彦．Support Vector Machine によるテキスト分類．情報処理学会 自然言語処理研究会 NL128-24, pp. 173–180, 1998.

[13] 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, p. 1113, 2000.