SVMを用いたチャンキングタスクにおける素性の自動選択

工藤 拓　　山田 寛康　　中川 哲治　　松本 裕治
奈良先端科学技術大学院大学情報科学研究科
{taku-ku,hiroya-y,tetsu-na,matsu}@is.aist-nara.ac.jp

1　はじめに

自然言語処理において チャンク同定問題 (チャンキング) とは, 任意のトークンをある視点からまとめ上げていき, まとめ上げた固まり (チャンク) 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示す. 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), 品詞タグ付け, 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.

我々は以前, Support Vector Machine (SVM) による チャンク同定手法を提案し, 過去の MBL や ME に基づくモデルよりも高い精度を示すことに成功した [10, 9, 11]. また, [10] では, チャンクの表現手法 (IOB1/IOB2/IOE1/IOE2) や解析方向 (右向き/左向き) 変更し, SVM の理論的な背景となっている VC-Bound や *Leave-One-Out* Bound が最良の表現方法を選択する能力を持ちあわせている事を示した. しかし, 上記の実験に用いた素性は, 着目している単語の前後 2 つの単語や品詞のみに限定しており, 我々の直観で発見的に選択していると言ってよい.

SVM は素性の次元数に依存しない汎化能力を持ち, 従来手法に比べ過学習しにくいとされているが, チャンク同定に使用する素性, つまり文脈長を軽率に長く選ぶと, ノイズの混入から精度が低下したり, 過学習に陥る可能性がある. また, 計算量や解析時間の観点からみても無意味に長い文脈長を考慮することは賢明でない.

本稿では, まず, SVM に基づくチャンク同定問題において, 精度に対する素性 (文脈長) の影響力の調査を行なう. さらに, 汎化誤差を最小にするという観点から最適な素性 (文脈長) を選択する手法を提案する.

2　SVM と モデル選択

2.1　Support Vector Machines

正例, 負例 の二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. 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 $y = f(\mathbf{x}, \theta)$ を導出することにある. SVM の識別関数は, l 個の学習データ $\mathbf{x}_i$ と, テストデータ $\mathbf{x}$ との類似度 (内積) を計算し, それらを線形結合した形となる.

$$y = f(\mathbf{x}, \theta) = \mathrm{sgn}\Big(\sum_{i=1}^{l} y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b\Big) \qquad (1)$$
$$\theta = (\alpha_1, \alpha_2, \ldots, \alpha_l, b) \quad \alpha_i \geq 0$$

式 (1) から, SVM は事例ベースの学習アルゴリズムの一種と解釈することが可能である. この時, 類似度を計算するための関数 K は, Kernel 関数と呼ばれ, 事例間の一般化された内積に値する. この Kernel 関数を変更することで SVM は非線形問題を解く事を可能にしている. また, 線形結合の重み α_i が求めるパラメータとなり, 実際には以下のような最適化問題によって導かれる.

$$\text{min.} \quad \sum_{i=1}^{l}\sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \qquad (2)$$
$$\text{s.t.} \quad y_i\Big(\sum_{j=1}^{l} y_j \alpha_j K(\mathbf{x}_j, \mathbf{x}_i) + b\Big) \geq 1 \quad (i = 1, \ldots, l)$$

上式の直感的な解釈は, 学習データを誤りなく分類しつつ, 可能な限り最小限のデータで識別関数を表現すること ($\alpha_i = 0$ となる事例をできるだけ多くすること) を意味する. この時, $\alpha_i > 0$ となる事例 $\mathbf{x}_i$ を **Support Vector** と呼び, Support Vector の集合は, 学習データの振舞い, 及び最終的な識別関数を決定付ける最小限の事例集合となる.

一般に学習における素性選択とは, 無限の素性集合から有限個の要素をある基準で選択することを意味する. SVM の学習, 分類における事例は Kernel 関数という一般化された内積の中にのみ現われるため, SVM における素性選択とは適用する Kernel 関数を選択することと解釈できる.

2.2　Leave-One-Out 推定

Leave-One-Out (以下 *LOO*) 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 $l-1$ 個を使って学習することをすべてのデータについて l 回くりかえし, それらのエラーの個数でモデルの識別能力を計測する手法である. *LOO* は, 学習アルゴリズムに依存しないモデル選択手法の一つである.

SVM の場合, Support Vector のみが最終的な識別関数を決定付けるために, *LOO* の各段階で, 個々の Support Vector すべてが誤ったときが最悪のケースとなる. つまり, m を Support Vector の数, l を学習データの数とすると, *LOO* により推定されるエラー率 $E_l[f]$ は以下の上限値を持つ.

$$E_l[f] \leq \frac{m}{l} \qquad (3)$$

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るために式 (3) は, 実際には, かなり緩い上限値となっている.

一方, Vapnik は式 (4) よりタイトな上限値として以下の推定法を提案している [7].

$$E_l[f] \leq \frac{\min(w^2D^2, m)}{l} \tag{4}$$

ただし, w^2 は式 (2) で得られる最小値であり, D は全事例を囲む球面の最小直径を指す. 本稿では, 式 (4) によって推定された エラー率の上限値を便宜的に *Leave-One-Out* 推定値と呼ぶ.

2.3 $\xi - \alpha$ 推定

LOO のよりタイトな上限値を求めるいくつかの手法が提案されている. Vapnik は, Span と呼ばれる事例の幾何的な配置を考慮することで [8], また, Chih-Wei らは SVM の最適化問題にいくつかの発見的手法を適用することにより [1], *LOO* の推定値を直接計算する手法を提案している. これらの手法は, *LOO* の推定値を少ない計算量で直接計算することを可能にするが, 学習とは別に *LOO* の推定値を計測するプロセスを必要とし, 若干処理が複雑となっている.

一方で Joachims は, 式 (4) よりもタイトな $\xi - \alpha$ 推定という手法を提案している [2].

$$E_l[f] \leq \frac{Card\{i : 2\alpha_i R^2 + \xi_i \geq 1\}}{l} \tag{5}$$

$$R^2 = \max_{i,j}(K(\mathbf{x}_i, \mathbf{x}_i) - K(\mathbf{x}_i, \mathbf{x}_j))$$

$$\xi_i = \max(0, 1 - y_i(\sum_{j=1}^{l} y_j \alpha_j K(\mathbf{x}_j, \mathbf{x}_i) + b))$$

上式の証明は, 文献 [2] を参照されたい. 一般に, 例外的な事例は, *LOO* の各段階でエラーになる可能性が高くなる. 一方で, SVM は, 例外的事例に対し, それ自身を特別視し, 他より大きな重み α_i を付与することで例外事例を分類することを試みる. 式 (5) は, α_i の値が大きいと, *LOO* におけるエラーとしてカウントされやすくなることを意味し, これは $\xi - \alpha$ 推定の直感的な解釈を与える. $\xi - \alpha$ 推定は, *LOO* 推定のための別プロセスを必要とせず, 極めて単純な手続きのみで, よりタイトな上限値を推定することを可能にする. 本稿では, 式 (5) によって推定された エラー率の上限値を便宜的に $\xi - \alpha$ 推定値と呼ぶ.

3 チャンキングと文脈長の自動選択

3.1 SVM によるチャンク同定

ここで, [10] で用いた, SVM による チャンク同定手法を簡単に説明する. まず, チャンク同定の際, 各チャンクの状態をどう表現するかが問題となるが, 本稿では, 各単語にチャンクの状態を示すタグを付与する手法の一つである IOB2 モデル [6] を実際の表現方法として採用した. IOB2 は, チャンクの先頭を示す単語に B タグを, チャンクの中にある単語で先頭以外の単語に I タグを, チャンク以外の単語に O タグを付与することで, 各チャンクの状態を区別する. また, 各チャンクに対し, そのチャンクの役割を示すタグを付与する場合は, B/I といった チャンク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, 動詞句 (VP) の先頭の単語は B-VP というタグが付与される. このモデルを採用することで, チャンク同定問題を一般的なタグ付け問題として扱うことができる.

[10] では, 位置 i の タグ t_i の推定を行なう素性として t_i 自身の単語と品詞, および $i+1, i+2, i-1, i-2$ の (左右 2 つの) 単語と品詞を用いた. さらに左 2 つのタグも素性として使用した. 本稿では, この前後 2 つという選択を, $i+n, i+m(n>0, m>0)$ として一般化することを考える. この時, 左側にある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, これらは左から右向きに解析しながら動的に追加していくこととした.

また, タグ付与のような多値分類問題を扱うためには 2 値分類器である SVM に対し何らかの拡張を行なう必要がある. 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 classification"* 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. 本稿では, 後者の *"pairwise classification"* を採用した. その理由として, (1) 各分類器の学習に用いられる学習データが少量であり, 学習のコストが小さいこと, (2) *"one class vs. all others"* よりも *"pairwise classification"* が実験的に良い結果が得られたという報告 [3] があること, の 2 点がある.

3.2 文脈長の自動選択

図 1 に文脈長の自動選択の手続きの概要を示す.

まず, 文脈長の異なる, 複数のモデルを個別に学習する. その後, 各タグのペアを学習した個々のモデルに対し, $\xi - \alpha$, *Leave-One-Out* 等でエラー率を推定する. 最終的に, 個々のペアそれぞれに対し, 推定されたエラー率の最も小さくなるモデル (文脈長) を選び, それらの多数決を行なうことで, タグを決定する.

上記の選択手法により, 文脈に依存しない簡単なタグの推定には, 短い文脈のモデルが, 逆に, 広範囲の文脈を考慮しないと同定できない困難なタグの推定には, 長い文脈のモデルが, それぞれ自動的に選択されることが期待される.

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.

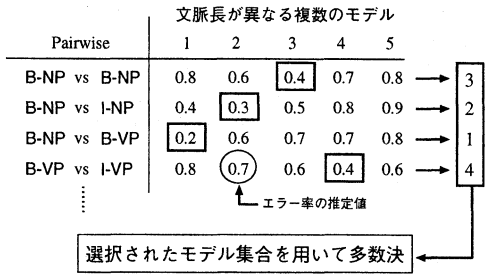

図 1: 文脈長の自動選択

- Chunking データセット (**Chunking**)
  CoNLL-2000 の Shared Task[5] に用いられたデータである[1]. 具体的には, Penn Tree-bank/WSJ の 15-18 を学習データ, 21 をテストデータとし, Brill Tagger を用いて part-of-speech (POS) を付与したデータである. 同定するチャンクタグ として, VP, PP, ADJP, ADVP, CONJP, INITJ, LST, PRT, SBAR の合計 10 種類の英語の基本句を表現するタグが IOB2 モデルにより付与されている.

- POS Tagging データセット (**POS Tagging**)
  Penn Tree-bank/WSJ の 15-18 を学習データ, 21 をテストデータとし, その part-of-speech (POS) を同定するデータセットである. この場合, 単語のみの情報では POS を推定することが困難であるため, [4, 11] らの手法を参考に, POS を推定する単語自身の部分文字列や文字種も素性として使用している.

実験には我々が開発している SVM 学習ツール TinySVM を用いた[2]. このツールは, *Leave-One-Out*, $\xi-\alpha$ 推定値を自動的に算出する機能を備えている. また, すべての実験において, Kernel 関数は 2 次の Polynomial 関数を用いた.

評価方法として, Chnking データセットでは, 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. また, POS Tagging のデータセットにおいては, 正しくタグが付与された割合 (精度) で評価を行なった.

4.2 実験結果

まず, 解析精度が文脈長によってどれだけ影響されるのかを調査するために, 文脈長が異なる 20 種類のモデルでの実験を行なった. それら個々の解析精度を表 1,3 に示す. ただし, 左/右 文脈長とは, 自分自身の単語は含めず, 左/右いくつの文脈を考慮するかを, タグ文脈長とは, 推定済みの左方向のタグをいくつ考慮するかを意味する.

結果から, 右方 (後方) 文脈を用いないモデルの精度が極端に低いことが分かる. これは, 本手法が, 決定的に解析を行なうためで, HMM のように文全体のコストを考慮しない事に起因する. また, 最も文脈長を長くとったモデルは, 若干ながら精度が低下している. 一般に, HMM 等のモデルでは, 文脈長を長くすると極度に過学習してしまう傾向にあるが, SVM は著しい性能低下はみられない. この結果は, SVM の持つ素性の次元数に依存しない汎化能力を実証する事実であると我々は考える.

次に, 提案手法により文脈長を自動的に選択した結果を表 2, 4 に示す. Chunking のデータセットにおいては, 二つの選択基準によらず, 20 種類のモデルのどれよりも精度が向上している. POS Tagging のデータセットにおいては, *Leave-One-Out* 推定値のみが, 20 種類のモデルのどれよりも精度が向上している. $\xi-\alpha$ 推定により選択したモデルは, 最長文脈長を選んだモデルよりも精度が低く, 期待どおりの結果は得られなかった.

4.3 データセットと推定手法の関係

なぜ, POS Tagging データセットでは, $\xi-\alpha$ 推定がうまく働かなかったのだろうか.

Chunking も POS Tagging も同一のコーパスを用いたが, 推定すべきタグの種類が, Chunkig は 22 種類, POS Tagging は, 45 種類と大きな開きがある. そのために, pairwise 推定を行なうと, Chunking に比べ, POS Tagging は個々の学習データの量が少なくなる.

一方で, 我々の適用したモデル選択基準は共に *Leave-One-Out* 推定に基づくものであり, この厳密な値を求めたところで, あくまでも真の汎化能力の近似値を与えているにすぎない. さらに, 学習データの数が少ないと, 厳密な値ですら汎化能力の推定値として信頼できなくなり, 逆に, 厳密な値は汎化能力を過大評価してしまう可能性が出てくる.

このことが, $\xi-\alpha$ 推定のうまく働かなかった要因の一つであると我々は考える. つまり, POS Tagging では個々の学習データの数が少なくなり $\xi-\alpha$ 推定値は汎化能力を過大評価してしまい精度低下に, 逆に, *Leave-One-Out* は, より緩やかな推定値であるために, 過大評価が抑えられ精度向上に繋がったものと考察される.

また, Chunking データセットにおいては, 十分な量の学習データが確保されたため, よりタイトな推定値である $\xi-\alpha$ が *Leave-One-Out* よりも高い精度を示したのではないかと考える.

4.4 今後の課題

本稿で提案した文脈長選択手法の欠点は, あらかじめ文脈長の異なる複数のモデルを作成しておかなければならないことにある. 実際, 考慮可能な文脈長の候補は無数にあるとことを考えると, このような欲張り的な手法は多大な計算量を要求するため効率が悪い. また, 本提案手法では, 個々の文脈長は推定すべきタグの種類に依存して変化するが, 本来ならば, 個々の状況 (現在の単語や品詞, 過去に推定したタグ) によって変化するの

[1] http://lcg-www.uia.ac.be/conll2000/chunking/
[2] http://cl.aist-nara.ac.jp/~ taku-ku/software/TinySVM/

表 1: 文脈長による精度比較 (Chunking)

左文脈長	右文脈長	タグ文脈長	$F_{\beta=1}$
1	0	1	89.01
1	1	1	93.05
2	0	1	89.05
2	0	2	89.46
2	1	1	93.17
2	1	2	93.22
2	2	1	93.41
2	2	2	93.44
2	0	1	89.65
3	0	2	89.63
3	0	3	89.53
3	1	1	93.28
3	1	2	93.17
3	1	3	93.08
3	2	1	**93.49**
3	2	2	93.47
3	2	3	93.35
3	3	1	93.37
3	3	2	93.39
3	3	3	93.27

表 2: 文脈長 自動選択の結果 (Chunking)

選択基準	$F_{\beta=1}$
Leave-One-Out	93.56
$\xi - \alpha$	**93.60**

表 3: 文脈長による精度比較 (POS Tagging)

左文脈長	右文脈長	タグ文脈長	精度 (%)
1	0	1	95.05
1	1	1	95.87
2	0	1	95.20
2	0	2	95.26
2	1	1	95.59
2	1	2	95.72
2	2	1	96.08
2	2	2	96.10
2	0	1	95.35
3	0	2	95.32
3	0	3	95.35
3	1	1	96.03
3	1	2	96.03
3	1	3	96.00
3	2	1	**96.15**
3	2	2	96.10
3	2	3	96.11
3	3	1	96.13
3	3	2	96.13
3	3	3	96.14

表 4: 文脈長 自動選択の結果 (POS Tagging)

選択基準	精度 (%)
Leave-One-Out	**96.16**
$\xi - \alpha$	96.12

が自然であろう. 例えば, 英語の tokenization のように, 2 種類のタグしか無いような問題は, タグのペアの個数が事実上 1 つとなるため, 文脈長は常に固定され, 動的に変化しない. 今後の課題として, 文脈長を 各状況において adaptive に選択する新たな手法の提案が挙げられる.

5 まとめ

本稿では, SVM を用いたチャンクの同定問題における素性 (文脈長) を汎化誤差を最小にするという観点から自動選択する手法を提案した. 実際のコーパスを用いた実験結果により, 学習データ数が少ない場合は, 採用した選択基準 ($\xi - \alpha$, *Leave-One-Out*) によりうまく選択が行なわれなかったものがあったが, 学習データが十分存在する場合は, 提案手法により自動的に文脈長が選択され, 精度が向上することが分かった.

参考文献

[1] Chih-Wei Hsu and Chih-Jen Lin. Automatic model selection for support vector machines. In *http://www.csie.ntu.edu.tw/~cjlin/looms/*.

[2] Thorsten Joachims. Estimating the Generalization Performance of a SVM Efficiently. In *International Conference on Machine Learning (ICML)*, 2000.

[3] Ulrich H.-G Kreßel. Pairwise Classification and Support Vector Machines. In *Advances in Kernel Methods*. MIT Press, 1999.

[4] Adwait Ratnaparkhi. A Maximum Entropy Model for Part-of-Speech Tagging. In *Proceedings of the EMNLP '96*, pp. 133–142, 1996.

[5] Erik F. Tjong Kim Sang and Sabine Buchholz. Introduction to the CoNLL-2000 Shared Task: Chunking. In *Proceedings of CoNLL-2000 and LLL-2000*, pp. 127–132, 2000.

[6] Erik F. Tjong Kim Sang and Jorn Veenstra. Representing text chunks. In *Proceedings of EACL'99*, pp. 173–179, 1999.

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

[8] Vladimir N. Vapnik and Oliver Chapelle. Bounds on Error Expectation for SVM. In *Advances in Large Margin Classifiers*. MIT Press, 1999.

[9] 山田寛康, 工藤拓, 松本裕治. 単語の部分文字列を考慮した専門用語抽出と分類. 情報処理学会 自然言語処理研究会 NL-140, pp. 77–84, 2000.

[10] 工藤拓, 松本裕治. Support Vector Machine を用いた Chunk 同定. 情報処理学会 自然言語処理研究会 NL140, pp. 9–16, 2000.

[11] 中川哲治, 工藤拓, 松本裕治. Support Vector Machine を用いた未知語の品詞推定. 情報処理学会 自然言語処理研究会 NL-141, pp. 77–82, 2000.