L_1 正則化特徴選択に基づく 大規模データ・特徴集合に適した半教師あり学習

鈴木 潤　　磯崎 秀樹　　永田 昌明

NTT コミュニケーション科学基礎研究所

{suzuki.jun, isozaki.hideki, nagata.masaaki}@lab.ntt.co.jp

1　はじめに

形態素解析，固有表現抽出，係り受け解析といった自然言語解析問題では，正解データを用いた教師あり学習により良好な解析精度が得られることが多くの研究で示されてきた．近年では，正解が付与されていないデータ (ここでは機械学習での用語に従って「ラベルなしデータ」と呼ぶ) を正解データと共に利用する半教師あり学習法により，教師あり学習で得られる解析精度を大幅に向上させることが可能であることも多くの研究で示された [1, 2, 3, 4].

ただし，これら自然言語処理分野で発展した半教師あり学習法は，大規模なデータと特徴集合を用いることが大幅な性能向上の必須条件となっている．つまり，大規模なデータと特徴集合をいかに効率的に扱うかが，これら半教師あり学習の重要な要素の一つであると言える．しかし，従来の研究報告では，解析精度の向上のみに焦点があてられ，この点を主たる論点としている論文は今のところ存在しない．そこで本稿では，この点に着目し，特に大規模データ・特徴集合を扱うのに適した方法を提案する．

提案法では，はじめに大規模なラベルなしデータ上で各特徴の重要度に相当する統計量の計算を L_1 正則化付き一般化 KL 距離最小化問題として定式化し，その解を求める．またこの時に，解がゼロとなった特徴は重要度が低いと仮定して特徴集合から削除する．次に，選出された特徴とその統計量を用いて，正解データを使った教師あり学習により対象タスクのモデルを学習する．これらの処理により，大規模な特徴集合に対しても，最終的には限定された特徴数で解析精度の高いモデルが学習できるようになる．また，提案法で最も計算コストの高い大規模ラベルなしデータ上での特徴の重要度判定処理に関しては，問題の定義そのものを分散並列計算モデル MapReduce に適した形で定義し，分散並列処理を行うことで，比較的容易に大規模データを扱えるように工夫する．

2　自然言語処理に適した半教師あり学習法

2.1　構造予測問題

まず，提案法を説明する前に，本稿が対象とする問題を明確にする．本稿では，固有表現抽出や，係り受け解析といった，機械学習の分野では「構造学習」と呼ばれる問題を扱う．これらの問題の特徴は，個々の出力 $\mathbf{y}$ がグラフ等で表せる離散構造となっているところにある．例えば，固有表現抽出では，出力はラベル系列として表され，係り受け解析では木構造となる．このような問題の場合には，個々の問題を解く際に離散最適化などの処理が必要となり，それに合わせたモデルとパラメタ推定法が求められる．条件付確率場 [5] は，その代表的な例である．

入力: (0) 学習データ $\mathcal{D} = \{\mathcal{D}_L, \mathcal{D}_U\}$
正解データ集合 $\mathcal{D}_L = \{(\mathbf{x}^i, \mathbf{y}^i)\}_{i=1}^I$,
ラベルなしデータ集合 $\mathcal{D}_U = \{\mathbf{x}^j\}_{j=1}^J$,
(1) 特徴ベクトル $\mathbf{f}$ の定義,
(2) **参照関数** $\tilde{r}$ の定義,
(3) 教師あり学習アルゴリズム
第 1 処理: 特徴重要度判定/選択処理，ラベルなしデータ $\mathcal{D}_U$ を使用
1.1 参照関数 $\tilde{r}$ と補助関数 q からパラメタ $\mathbf{u}^*$ を推定
1.2 $u_n^* = 0$ となった全ての f_n を削除した特徴集合 $\mathbf{f}' \subseteq \mathbf{f}$ を獲得
第 2 処理: 教師あり学習によるモデルを学習，正解データ $\mathcal{D}_L$ を使用
2.1 第 1 処理で得た $\mathbf{f}', \mathbf{u}^*$ を用いてパラメタ $(\mathbf{w}^*, \mathbf{v}^*)$ を推定
出力: $(\mathbf{w}^*, \mathbf{v}^*, \mathbf{u}^*)$

図 1: 本稿で提案する半教師あり学習の概要

特徴ベクトルを $\mathbf{f}$，そのパラメタを $\mathbf{w}$ とする．次に，z を，ある (構造をもった) 出力 $\mathbf{y}$ のある一つの部分構造とし，$\mathcal{Z}(\mathbf{y})$ を $\mathbf{y}$ を構成する全ての部分構造の集合とする．ここで，入力 $\mathbf{x}$ に対する最尤出力 $\hat{\mathbf{y}}$ を，$\mathbf{x}$ が与えられた時の解候補集合 $\mathcal{Y}(\mathbf{x})$ から選択する問題は，線形判別式を用いて以下の式により表現できる．

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y} \in \mathcal{Y}(\mathbf{x})} \sum_{k=1}^{K} \sum_{z \in \mathcal{Z}(\mathbf{y})} \mathbf{w}_k \quad \mathbf{f}_k(\mathbf{x}, z) \tag{1}$$

上式は，K 種類の特徴ベクトル $\mathbf{f}_k$(ただし $k = 1, ...K$) を用いて構造予測問題をモデル化していることを仮定している．これは，一般的に，固有表現抽出や係り受け解析では，単語，品詞，単語と品詞の組み合わせ，一つ前の単語と一つ後ろの単語の組み合わせといった複数の異なる性質の特徴をまとめて利用していることを明示的に記述した形としているためである．

2.2　提案法の概要

提案法では，大きく分けて二つの処理にて対象タスクのモデルを学習する．第一段階の処理として，各特徴の重要度を後述する統計量に基づいて判定し，対象タスクのモデル学習に有効と思われる特徴集合を選出する．第二段階の処理では，選出した特徴集合と計算した統計量を用いて，対象タスクのモデルを教師あり学習アルゴリズムを用いて学習する．ここでのポイントは，第一処理では，ラベルなしデータのみを用いて特徴の重要度判定/選択を行い，第二段階では正解データのみを用いて教師あり学習する点である．図 1 に提案法の学習アルゴリズムの概要を示す．ここで，図中の入力の (1) と (3) に関しては，対象タスクに応じて設計すればよい．(1) の設計の制約は，後述する特徴重要度判定/選択法の効率的に計算するために，各特徴が取る値を 0 または任意の実数 e_n の二値しかとらないと

仮定する点である．ここで，全ての n で $e_n = 1$ とおけば，自然言語処理でよく用いられる二値特徴集合となる．よって，この仮定は，自然言語処理タスクでは，容易に許容できる仮定である．次に，(3) の設計の制約は，以下の構造予測式となるモデルとその教師あり学習アルゴリズムであることである．

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y} \in \mathcal{Y}(\mathbf{x})} \sum_{k=1}^{K} \sum_{z \in \mathcal{Z}(\mathbf{y})} (\mathbf{w}_k + v_k \mathbf{u}_k) \quad \mathbf{f}_k(\mathbf{x}, z) \qquad (2)$$

式 (1) で示した従来の構造予測式と比較してわかるように，提案法では，第一段階で計算した統計量 $\mathbf{u}$ とその重み v の積を，従来の追加の式に追加した形になっている．ただし，$\mathbf{u}$ は第二段階では変化しないので，それを $\mathbf{u}$ $\mathbf{f}$ を新たな特徴，$\mathbf{v} = (v_1, ..., v_K)$ を新たなパラメタとみなせば，従来の一般的な教師あり学習アルゴリズムをそのまま用いることができる．

次節では，残りの (2) に相当する「参照関数」と，第一処理の特徴重要度判定/選択方法に関して説明する．

2.3 参照関数，補助関数

提案法の第一処理にあたる特徴重要度判定処理を説明するために，まず，そこで使われる参照関数と補助関数について説明する．

入力 $\mathbf{x}$ と部分出力 z を変数とする非負の関数 $r(\mathbf{x}, z)$ があるとする．この関数 r は，z が入力 $\mathbf{x}$ の正解出力 $\mathbf{y}$ の一部である尤度を表す関数と仮定する．つまり，z が入力 $\mathbf{x}$ の正解出力 $\mathbf{y}$ の一部となる確率が高ければ，高い値となり，逆に確率が低ければ 0 に近い値を返す関数である．正解となる尤度を反映した関数と仮定することから，本稿では，関数 r を「参照関数」と呼ぶ．

ただし，正解となる尤度を正確に反映した参照関数 r を準備するのは現実にはほぼ不可能である．また準備できるのであれば，ここでのモデル学習は不要ということになる (その関数 r により 100%の解析精度が得られることを意味するので)．ここで言いたいことは，提案法のアイディアは，何かしらの観点で r の近似関数 $\tilde{r}$ を推定し，それを特徴重要度判定に利用することである．

このような近似参照関数 $\tilde{r}$ の定義には，いくつかの候補が考えられる．本稿では，実際に本稿の実験で用いた以下に二種類の参照関数の例を示す．一つ目の例としては，事前に教師あり学習したモデルを (近似) 参照関数として用いる方法である [2]．ここでは，参照関数を，$\mathbf{x}$ を入力した際に，事前に教師あり学習したモデルの推定結果 $\hat{\mathbf{y}}$ の部分構造として z が含まれている場合には $1+$ ，含まれていない場合には　を返す関数とする．ただし，　は，小さい正の値，例えば　$= 1.0$　10^{16} とする．

二番目の例としては，学習データから推定した特徴毎の尤度を利用する方法が考えられる．具体例として，各特徴 $\mathbf{f}(\mathbf{x}, z)$ が，z が $\mathbf{x}$ の正解の部分構造である場合の出現数と，正解でない場合の出現数をカウントし，単純に正解のカウントを出現数で割った値の対数を尤度として用いる．この場合は，単にラベルありデータから各特徴の出現頻度を数え上げるだけの処理でよいため，計算コストは比較的少なくて済むという利点がある．

次に，参照関数 r と同じような，以下の非負関数 q を導入する．

$$q(\mathbf{x}, z, n; \mathbf{u}) = r(\mathbf{x}) \exp[u_n f_n(\mathbf{x}, z)] \qquad (3)$$

ただし $r(\mathbf{x})$ は，$r(\mathbf{x}) = \sum_{z \in \mathcal{Z}(\mathbf{x})} \frac{\tilde{r}(\mathbf{x}, z)}{|\mathcal{Z}(\mathbf{x})|}$ であり，$\tilde{r}(\mathbf{x}, z)$ の $\mathbf{x}$ での平均である．q は，$u_n = 0$ の時に，参照関数の平均 $r(\mathbf{x})$ と一致するように定義されているところがポイントである．ここでは，q を補助関数と呼ぶ．

2.4 特徴重要度判定/選択法

提案法では，参照関数 $\tilde{r}$ と補助関数 q を用いて特徴重要度判定を行う．具体的には，提案法での特徴重要度判定処理を，各特徴の参照関数と補助関数間の L_1 正則化項付き一般化 KL 距離最小化問題として定式化する．実際には，一般化 KL 距離は，ラベルなしデータ上での経験一般化 KL 距離 $\tilde{K}_{\mathcal{D}_U}$ を用いて以下の最適化問題を解く．

$$\begin{aligned} \mathbf{u} &= \arg\min_{\mathbf{u}} \mathcal{U}(\mathbf{u}|\mathcal{D}_U) \\ \mathcal{U}(\mathbf{u}|\mathcal{D}_U) &= \tilde{K}_{\mathcal{D}_U}[\tilde{r}(\mathbf{x}, z)||q(\mathbf{x}, z, n; \mathbf{u})] + C_U||\mathbf{u}||_1 \\ \tilde{K}_{\mathcal{D}_U}[\tilde{r}||q] &= \sum \tilde{r} \log \tilde{r} - \sum \tilde{r} \log q - \sum \tilde{r} + \sum q \end{aligned} \qquad (4)$$

補助関数 q は，$(\mathbf{x}, z, n)$ の三種類のパラメタを持つ．よって，$\tilde{K}_{\mathcal{D}_U}$ の総和計算の部分は，$\sum_n \sum_{\mathbf{x} \in \mathcal{D}_U} \sum_{z \in \mathcal{Z}(\mathbf{x})}$ となる．ただし，$\mathcal{Z}(\mathbf{x})$ を $\mathbf{x}$ に対する可能な全ての出力 $\mathcal{Y}(\mathbf{x})$ に含まれる部分構造の集合とする．この最小化問題の意味は，参照関数の値になるべく近くなるように補助関数のパラメタ $\mathbf{u}$ を推定することである．また，同時に，L_1 正則化項の効果により，なるべく少ない非零のパラメタ数で解を求めることも意味する．

ここで，$\tilde{R}_n$ と A_n を導入する．$\tilde{R}_n$ は，n 番目の特徴が出現する $(\mathbf{x}, z)$ での $\tilde{r}(\mathbf{x}, z)$ の値の総和とする．つまり，$\tilde{R}_n = \sum_{(\mathbf{x}, z) \in \mathcal{D}_U} I(f_n(\mathbf{x}, z) = e_n) \tilde{r}(\mathbf{x}, z)$ と，$A_n = \sum_{(\mathbf{x}, z) \in \mathcal{D}_U} I(f_n(\mathbf{x}, z) = e_n) r(\mathbf{x})$ である．同様に，A_n を，$r(\mathbf{x})$ の総和とする．本稿では，紙面の都合で詳細な導出は省略するが，簡単な式変形により最終的に式 (4) の解 $\mathbf{u}$ は以下の式で解析的に求めることができる．

$$u_n = \begin{cases} \frac{1}{e_n} \Big(\log \Big[e_n \tilde{R}_n - C_U \Big] - \log[e_n A_n] \Big) \\ \qquad \text{if } C_U < e_n \tilde{R}_n - e_n A_n \\ \frac{1}{e_n} \Big(\log \Big[e_n \tilde{R}_n + C_U \Big] - \log[e_n A_n] \Big) \\ \qquad \text{if } -C_U > e_n \tilde{R}_n - e_n A_n \\ 0 \qquad \text{otherwise} \end{cases}, \qquad (5)$$

ただし，u_n は，$\mathbf{u}$ の n 番目の要素とする．ここで，解析的に解が求まるポイントは二点あり，一つは式 (4) が各特徴毎に独立に定義されている点，二つ目は各特徴が $f_n \in \{0, e_n\}$ という二値しかとらないと仮定している点である．また，この式から簡単にわかるように，n 番目の特徴が $|e_n \tilde{R}_n - e_n A_n|$　C_U の時には，u_n が 0 になる．

次に，以上の重要度判定に従って特徴選択も同時に行う．提案法では，単純に $u_n = 0$ になった特徴 f_n は，モデル学習時の重要度は低いと仮定し，元の特徴集合から削除する．

一般的に，特徴選択する際には，選択する特徴数を明示的に決定できたほうが良い．一方，提案法では C_U の値で選択する特徴数をコントロールすることになるが，C_U の値は，特徴数を明示的に表してはいない．ただし．提案法でも簡単な追加処理によって明示的に特徴数をコントロールすることができる．具体的には，はじめに $C_U = 0$ と仮定して計算を行い，$\mathbf{u}$ の要素を絶対値でソートする．欲しい特徴数が N のばあいには，ソートした $N+1$ 番目のパラメタの値を C_U に設定すればよい．この処理は，文献 [6] で紹介されている処理とほぼ同じ処理である．

Input: unlabeled data $\mathcal{D}_U = \{\mathbf{x}^j\}_{j=1}^J$
Mappers: the p'th mapper
1, receive the p'th part of unlabeled data $\mathcal{D}_{U,p}$.
2, perform the followings for each $(\mathbf{x}, z)$ in $\mathcal{D}_{U,p}$,
2.1, calculate $\tilde{r}(\mathbf{x}, z)$ and $\bar{r}(\mathbf{x})$
2.2, make key-values $[n, (\tilde{r}(\mathbf{x}, z), \bar{r}(\mathbf{x}), e_n)]$,
where key = n, and value = $(\tilde{r}(\mathbf{x}, z), \bar{r}(\mathbf{x}), e_n)$,
for all n, when $f_n(\mathbf{x}, z)$ is non-zero, that is, $f_n(\mathbf{x}, z) = e_n$.
2.3, output the key-values made in 2.2.
Combiners: (optional) the p'th combiner
1, receive the output of p'th mapper
2, perform the following procedures for each n in the received list,
2.1, sum together $\tilde{r}(\mathbf{x}, z)$ and $\bar{r}(\mathbf{x})$ individually in values,
2.2, make key-values $[n, (\sum \tilde{r}(\mathbf{x}, z), \sum \bar{r}(\mathbf{x}), e_n)]$,
2.3, output the key-values made in 2.2.
Reducers: q'th reducer
1, receive the q'th list of shuffled mapper (or combiner) outputs,
$\{[n, \{(\tilde{r}', \bar{r}', e_n)\}]\}$ where $\tilde{r}'$ is either $\tilde{r}(\mathbf{x}, z)$ or $\sum \tilde{r}(\mathbf{x}, z)$,
and $\bar{r}'$ is either $\bar{r}(\mathbf{x})$ or $\sum \bar{r}(\mathbf{x})$.
2, perform the following procedures for each n in the received list,
2.1, sum together $\tilde{r}'$ and $\bar{r}'$ to make $\tilde{R}_n$ and A_n, respectively
2.2, calculate Equation 5 by using $(\tilde{R}_n, A_n, e_n)$.
Output: parameter set $\mathbf{u}^*$
1, either $\mathbf{u}^* = \{u_n^*\}_{n=1}^N$ or $\{(n, u_n^*)\}$ for all n if $u_n^* \neq 0$.

図 2: MapReduce 計算モデルに基づく提案法での特徴重要度判定/選択処理

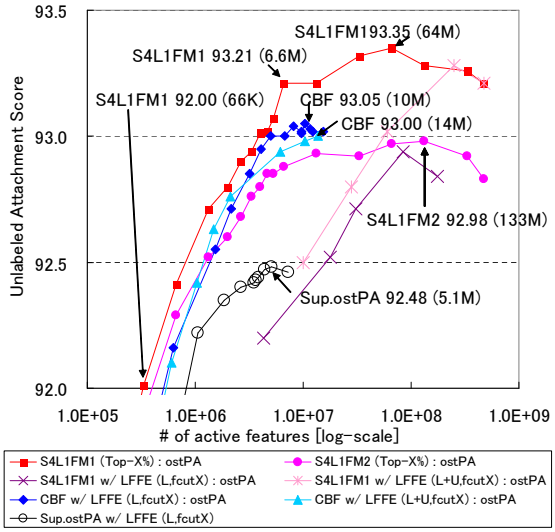

図 3: 各手法の各特徴数での解析精度 (UAS). 以下の特徴選択設定を使用 (右から左の点): S4L1FM1/S4L1FM2 (Top-{100, 50, 20, 10, 5, 2, 1, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3 0.2, 0.1, 0.05}%), S4L1FM w/ LFFE(L,fcut{0, 1, 5, 10, 50}) and (L+U,fcut{3, 10, 100}), CBF(L,fcut{0, 1, 2, 3, 4, 5, 10, 20, 50, 100, 200, 500, 1K, 5K}), CBF(L+U,fcut{10, 100, 1K, 10K, 50K, 100K, 500K}), Sup.ostPA(L,fcut{0, 1, 2, 3, 4, 5, 10, 20, 50, 100})

本稿では，提案法を 'Scalable and Sparse Semi-Supervised learning based on L_1-regularized Feature Mining **S4L1FM**' と呼ぶ.

2.5 MapReduce 上での計算方法

図 2 に，提案法の L_1 正則化経験一般化 KL 距離最小化問題を MapReduce 計算モデル上で処理するアルゴリズムを示す．ここで注目してもらいたい点は，図中に示したアルゴリズムは，MapReduce 計算モデル一回分の計算で処理が終了する点である．一般的に，MapReduce 上での EM アルゴリズムや SVM の最適化の場合は，最適パラメタを得るために反復計算が必要となる [7]．しかし，提案法では，反復計算なしに解が求まる計算で終わることから，相対的に計算コストが低く抑えられる方法となっている．特に，半教師あり学習においては，ラベルなしデータは非常に大規模なデータを扱う場合が多いため，この差が大きな差になる．

3 実験: 係り受け解析

本稿の実験では，従来の研究に従って，係り受け解析で標準的に用いられる Penn TreeeBank から得られる英語係り受け解析用のデータを用いた [4, 8]. ただし，本稿の実験内の比較にはセクション 00,01,24 のデータを用いた．先行研究との公平な比較のため，標準的な評価データであるセクション 23 は，従来法との比較にのみ用いた.

3.1 比較手法

まず，教師あり学習アルゴリズムには，構造学習用に拡張した online Passive-Aggressive algorithms (**ostPA**) [9] を用いた．提案法の実験設定として，参照関数には，第 2.3 に示した二つの例の方法を用いた．一つ目の例で示した，事前に学習した教師あり学習のモデルを利用する方法を **S4L1FM1** とする．ただし，ここで用いるモデルは前述の ostPA を用いて学習したモデルとする．二つ目の例で示したラベルありデータから推定した統計量を用いる方法を **S4L1FM2** とする.

係り受け解析において，現在単一の半教師あり学習法として最も良い成績を示しているのは文献 [4] の方法である．文献 [4] の方法とは，簡単に言うと，事前に行った (対象タスクとは基本的に無関係な) 単語クラスタリングのクラスタ番号を教師あり学習の特徴として利用する方法である．ここでは，このクラスタリングに基づく方法を **CBF** と記述する．本稿では，この方法と提案法の解析精度を，利用する特徴数との関係を考慮しながら解析精度を比較する.

ただし，文献 [4] の方法は，特に明示的に特徴数を削減することは考慮されていない．本稿では，用いる特徴数の違いによる解析精度を比較したいので，簡単な特徴選択法を導入する．具体的には，一般的によく用いられる方法である各特徴の頻度を用いて，一定の頻度以下の特徴は利用しないといった方法を用いる．ここでは，この頻度による特徴選択法を **LFFE** と記述する.

3.2 結果および考察

図 3 に各学習法で得られた最終的なモデルの非ゼロのパラメタを持つ特徴数と，そのモデルでの解析精度を示す．図中の手法は，提案法 S4L1FM:ostPA，比較手法 CBF:ostPA，ベースラインの教師あり学習 Sup.ostPA の結果である．図中の 'Top-X%' は，全特徴数に対して上位 X%の特徴を，S4L1FM:ostPA での特徴選択により抽出した場合を示している．また，'fcutX' は，X の値を閾値とした頻度による特徴選択を用いていること表す．このとき，頻度を計算するデータが正解データのみの場合に 'L'，正解データとラベルなしデータ両方を用いる場合に 'L+U' と記述した．ここで，便宜上 'fcut0' とは，頻度による特徴選択を用いていないことを意味する.

実際に図の X 軸にあたる特徴数を計算には，各手法間で公平な比較を行うために，それぞれに合った計算法を用いている．Sup.ostPA では，学習後の $\mathbf{w}$ の非ゼロのパラメタ数に相当する．同様に CBF では，$\mathbf{w}$ と $\mathbf{v}$ の非ゼロのパラメタ数である．一方 S4L1FM では，$\mathbf{w}$ と $\mathbf{v}$ の非ゼロのパラメタ数ではなく，$\mathbf{w}$ と $\mathbf{u}$ の非ゼロのパラメタ数となる．これは，S4L1FM は，$\mathbf{v}$ ではなくて，$\mathbf{u}$ が特徴ベクトル $\mathbf{f}$ に対応しているためである．仮に，$\mathbf{v}$ で計算すると非常に少ない特徴数となる.

ここで注意点として，S4L1FM と CBF の特徴数は，

	best			balance		
methods	dev	test	#f	dev	test	#f
L2CRF w/ LFFE	90.83	85.20	45M	85.35	79.82	1.0M
L1CRF	90.94	85.07	9M	90.54	84.43	1.0M
S4L1FM1:L2CRF	93.62	89.22	80M	90.97	87.20	1.0M
S4L1FM2:L2CRF	92.30	88.06	76M	90.53	85.98	1.0M

表 1: 固有表現抽出実験の結果．(#f: 特徴数)

Sup.ostPA より多くなる場合がある．これは，S4L1FM の場合は，ラベルなしデータにしか出現しない特徴も利用する可能性があるからであり，CBF の場合は，クラスタリングに基づく特徴を新たに追加しているためである．

まず，全体として，全ての方法でより多くの特徴を用いた方が解析精度が高くなる傾向にあることがわかる．これは，係り受け解析では，より高い解析精度を得るためには，傾向としてより多種多様な特徴を取り入れたる必要があることを示している．

次に，本実験の全ての結果の中で，S4L1FM1:ostPA が最もよい解析精度を得た．また，S4L1FM1:ostPA は同レベルの特徴数での比較でも，比較手法内で最も良い解析精度を得ている．この結果は，S4L1FM1:ostPA が特徴数を限定した状況でも，比較手法よりも良いモデルが学習できることを示している．一方，S4L1FM2:ostPA は，CBF:ostPA より若干低い解析精度となった．S4L1FM1:ostPA と S4L1FM2:ostPA の差は，用いた参照関数の違いのみである．この結果は，提案法では，いかに良い参照関数を準備するかが解析精度を決める大きな要因となることを示している．

もう一つ興味深い点として，S4L1FM1:ostPA は Sup.ostPA と同程度の解析精度を 10 分の 1 程度の特徴数で実現している点である．これは，半教師あり学習にすることで，教師あり学習と同程度の解析精度をよりコンパクトなモデルで実現できることを示している．

最後に，提案法に頻度による特徴選択 (S4L1FM1w/LFFE) を用いても効果がほとんど得られないことがわかった．この結果は，提案法による特徴の重要度判定/選択法が効果的に特徴を選択できていることを裏付ける結果となっている

3.3 上位システムとの解析精度比較

本実験で用いた PTB III の係り受け解析データにおいて，これまでに報告されている最も高い解析精度は文献 [8] の 93.79 である．それに対し，S4L1FM:ostPA の評価データ (Sec.23) の解析精度は 93,86 であった．つまり，S4L1FM:ostPA は，PTB III ベンチマークデータで，これまでで最も高い解析精度を達成した．

4 実験: 固有表現抽出

次に，簡単に固有表現抽出タスクの実験を述べる．固有表現抽出タスクの標準的なデータである CoNLL'03 [10] shared task データを用いた．実験設定は，文献 [1, 2] の方法に従う．ラベルなしデータは文献 [2] と同様に 35M 単語の Reuters corpus とした．

教師あり学習法として，L_1 および L_2 正則化条件付確率場 [11, 5] を用いた (**L1CRF** および **L2CRF** と記述する)．提案法の参照関数には，係り受け解析で用いたものと同じ二種類の参照関数を用いた (S4L1FM1, S4L1FM2).

4.1 結果

本稿では $F_{\beta=1}$ スコアにより固有表現抽出精度を評価した [10]．表 1 に，各手法で最も良い抽出精度を示したパラメタでの結果と，特徴数が 100 万 (1M) の時の結果を示す．L2CRF w/ LFFE は特徴数を減らすと急激に抽出精度が悪くなっている．これは単純に頻度で特徴数を減らすのは抽出精度に大きな悪影響を及ぼすことを示している．一方，L1CRF は，L_1 正則化の効果で効果的に非ゼロの特徴数を削減することに成功していることがわかる．最後に，S4L1FM は，最良の結果も特徴数が 1 M の時の場合も，L1CRF より良い抽出精度を示した．この結果は，S4L1FM によるラベルなしデータからの特徴重要度判定が効果的に行われていることを示している．ここでポイントとなるのは，S4L1FM は教師あり学習を始める前の段階で特徴を選択している点である．これはつまり，L1CRF とは違い教師あり学習の段階で既に少ない特徴集合から効果的にモデルを学習ができていることを意味している．また，少ない特徴集合から教師あり学習を行っているということは，必要とする計算リソースも少なくて，かつ高速に学習することが可能となる．

5 結論

本稿では，大規模なラベルなしデータから特徴の重要度判定/選択し，限定した特徴数でモデル学習を行う半教師あり学習法を提案した．提案法での特徴選択は，参照関数と補助関数間の L_1 正則化付きの一般化 KL 距離最小化による最適化後のパラメタがゼロになる特徴を特徴集合から削除する方法である．また，この処理は，1-pass の Map-Reduce 計算モデルにより効率的に処理することができるため，比較的大規模なラベルなしデータでも容易に処理することが可能となる．本稿の実験では，係り受け解析，固有表現抽出タスクにおいて，限定した特徴数でも，非常に良好な解析精度が得られることを示した．

参考文献

[1] Rie Kubota Ando and Tong Zhang. A High-Performance Semi-Supervised Learning Method for Text Chunking. In *Proceedings of 43rd Annual Meeting of the Association for Computational Linguistics*, pages 1–9, 2005.

[2] Jun Suzuki and Hideki Isozaki. Semi-supervised Sequential Labeling and Segmentation Using Giga-Word Scale Unlabeled Data. In *Proceedings of ACL-08: HLT*, pages 665–673, 2008.

[3] Dekang Lin and Xiaoyun Wu. Phrase Clustering for Discriminative Learning. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1030–1038, 2009.

[4] Terry Koo, Xavier Carreras, and Michael Collins. Simple Semi-supervised Dependency Parsing. In *Proceedings of ACL-08: HLT*, pages 595–603, 2008.

[5] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the International Conference on Machine Learning (ICML 2001)*, pages 282–289, 2001.

[6] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient Projections onto the L1-ball for Learning in High Dimensions. In *proc. of ICML-2008*, pages 272–279, 2008.

[7] Cheng-Tao Chu, Sang Kyun Kim, Yi-An Lin, YuanYuan Yu, Gary R. Bradski, Andrew Y. Ng, and Kunle Olukotun. Map-Reduce for Machine Learning on Multicore. In *Advances in Neural Information Processing Systems 19*, pages 281–288, 2006.

[8] Jun Suzuki, Hideki Isozaki, Xavier Carreras, and Michael Collins. An empirical study of semi-supervised structured conditional models for dependency parsing. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 551–560, 2009.

[9] Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online Passive-Aggressive Algorithms. *Journal of Machine Learning Research*, 7:551–585, 2006.

[10] Erik Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2003*, pages 142–147, 2003.

[11] Galen Andrew and Jianfeng Gao. Scalable training of L1-regularized log-linear models. In Zoubin Ghahramani, editor, *Proceedings of the 24th Annual International Conference on Machine Learning (ICML 2007)*, pages 33–40. Omnipress, 2007.