DBSJ Journal, Vol.11, No.3
February 2013

複数素性の同時テストによるオンライングラフティングの効率化

Improving Efficiency of Online Grafting by Simultaneously Testing Multiple Features

大井 健吾♥　二宮 崇♦

Kengo OOI　Takashi NINOMIYA

本稿はオンライングラフティングの効率を改良する二つの手法について述べる．オンライングラフティングは機械学習における素性選択の一手法であるが，逐次的に素性を選択すること，およびℓ_1-正則化ロジスティック回帰に基づき識別器学習と素性選択の両方が同時に行われることから，識別器学習のための素性選択手法として優れていると考えられている．しかし，オンライングラフティングは学習のためにパラメータ最適化を頻繁に行う必要があり，非常に高い計算コストを要するという問題がある．本研究は複数の素性を同時にテストすることによりオンライングラフティングの効率を改良する２つの近似手法を提案する．提案手法はオンライングラフティングの近似手法となっているが，実験により予測性能を低下することなく学習効率が大きく改善されることを示した．

This paper proposes two efficient methods to improve online grafting, which is one solution for selecting features for machine learning classifiers. Online grafting is preferable in that it incrementally selects features, and it is defined as an optimization problem based on ℓ_1-regularized logistic regression. However, its learning is computationally expensive because the algorithm for online grafting requires frequent parameter optimization. We propose two efficient methods that approximate online grafting by testing multiple features simultaneously. The experiments have shown that our methods significantly improved efficiency of online grafting.

1. はじめに

現在，機械学習の分野で開発された多くの技術は，情報検索，自然言語処理，音声認識，バイオインフォマティクスを含む様々な分野で使用されており，特に，未知の入力に対し正しいラベルを予測する識別器は多くの用途に供することができるため，より精度の高い識別器を実現することは最も重要な課題の一つとなっている．識別器の学習および予測においては，まず，入力に対して素性関数を適用することにより数万から数百万に及ぶ次元から成る素性ベクトルを生成し，得られた素性ベクトルに対し学習および予測が行われる．識別器の精度は素性関数の設計に大きく依存するため，良い素性関数を定義することは非常に重要な課題となっているが，良い素性関数を自動的に得ることは容易ではなく，多くのアプリケーションにおいては人間の専門家が直感や経験を頼りに長い時間と人的コストをかけて設計している．例えば，自然言語処理における文書分類，品詞推定，構文解析などのタスクにおいては，局所的に組み合わせた単語や品詞が素性として用いられているが，どの組合せを用いるかについては専門家によって判断され設計されている．

機械学習の分野において，素性候補集合から有用な素性を選択する技術は，素性選択[1]と呼ばれ，冗長または不要な素性の排除による計算コストの削減および汎化性能の向上を目的として研究されている．素性選択は，本来，そのような汎化性能やコスト削減のための技術であるが，組み合わせ素性を大量に生成し，生成された素性集合から有用な素性だけを選択すれば，有用な素性関数を自動的に獲得する手法としても用いることができる[2]．ℓ_1-正則化ロジスティック回帰は，正解ラベルが付与された訓練データから識別器を学習する確率的識別モデルの一種であるが，ℓ_1-正則化項の特性により，損失関数を最小化する上で不要な素性に対する重みは0となるため，それらの素性を削除することで，素性選択の手法としても用いられている．ℓ_1-正則化ロジスティック回帰は，識別器として代表的なSVMやℓ_2-正則化ロジスティック回帰と並んで高い予測性能を実現し，また，訓練データに対する損失を最小化するように多くの素性が削除されるため，識別器のための素性選択として非常に優れている[3]．しかし，ℓ_1-正則化ロジスティック回帰は，素性選択としては後ろ向き探索に属し，そのすべての素性を使用してパラメータを推定してから不要な素性を削除する必要がある．もし，素性の候補として非常に大きな素性集合に対し素性選択を行う場合，そのパラメータ推定には非常に大きな計算コストが必要となってしまう．この問題を解決するため，Perkinsらはグラフティング[4]およびオンライングラフティング[5]と呼ばれるℓ_1-正則化ロジスティック回帰のためのオンライン素性選択を提案した．グラフティングおよびオンライングラフティングは，空の素性集合から始まり，候補の素性集合から素性を一つずつ選択する前向き素性選択となっているため，ℓ_1-正則化ロジスティック回帰の学習のようにすべての素性に対するパラメータ推定を行う必要がない．しかし，現在選択されている素性のためにパラメータ最適化を頻繁に行う必要があり，累積的に非常に高いコストを要する．そのためグラフティングに関する研究は，近年では岡野原ら[2]やZhuら[6]による研究があるのみで他にはほとんど存在していない．

本論文ではオンライングラフティングの効率を改良する二つの手法，倍数分割オンライングラフティング，定数分割オンライングラフティングを提案する．どちらの手法も元のオンライングラフティングより低い頻度でパラメータ最適化を行うことで効率化を実現しており，定数分割オンライングラフティングでは定数個の素性がテストされた後にのみパラメータが最適化され，倍数分割オンライングラフティングでは倍数個の素性がテストされた後にのみパラメータが最適化される．本研究の提案手法はオンライングラフティングの近似手法となっているため，効率と予測性能の間にトレードオフが存在するが，実験により，予測性能を低下するこ

♥ 非会員　愛媛大学　大学院理工学研究科博士前期課程
ooi@ai.cs.ehime-u.ac.jp
♦ 正会員　愛媛大学　大学院理工学研究科
ninomiya@cs.ehime-u.ac.jp

となく学習効率が大きく改善されることを経験的に示す.

2. 研究背景

2.1 ℓ_1-正則化ロジスティク回帰

ロジスティック回帰は，二値分類のための確率的識別モデルとしてよく用いられており，多値分類のためのロジスティック回帰は最大エントロピーモデルとしても知られている．ロジスティック回帰識別器は，入力$\mathbf{x}$に対し出力$y \in \{-1, +1\}$を返す．ここで，-1が負のラベルであり，$+1$が正のラベルである．ロジスティック回帰識別器は，正解付き訓練データからその確率モデルを学習する．

入力$\mathbf{x}$，出力yとすると，ロジスティック回帰の確率モデルは，次のように定義される．

$$\mathrm{P} = (y = +1|\mathbf{x}) = \frac{exp\big(f(\mathbf{x})\big)}{1 + exp\big(f(\mathbf{x})\big)} \qquad (1)$$

$$f(\mathbf{x}) = \sum_{j=1}^{d} x_j \cdot w_j + b \qquad (2)$$

ここで，dは素性数，bはバイアス項，x_jはxのj番目の素性，w_jはj番目の素性の重みである．

ロジスティック回帰識別器は，訓練データの負の対数尤度（BNLL）損失関数[7]を最小化することにより訓練されており，これは，訓練データの対数尤度を最大化することに等しい．ロジスティック回帰は，多くの場合，訓練データに対する過学習を防ぐため正則化項を用いて訓練されている．通常，ℓ_1−ノルムを使用したℓ_1−正則化項またはℓ_2−ノルムを使用したℓ_2−正則化項が正則化項として用いられている．ℓ_1−正則化項を使って訓練した場合，多くの無駄な素性に対する重みは0になるのでℓ_1−正則化項は素性選択の手法としても有用であることが知られている[8].

データセットDが与えられたとき，ℓ_1−正則化ロジスティック回帰の目的関数Cは次のように定義される．

$$C(\mathbf{w}, D) = \frac{1}{|D|} \sum_{\langle \mathbf{x}, y \rangle \in D} \log\big(1 + e^{-yf(\mathbf{x})}\big) + \lambda \sum_{j=1}^{d} |w_j| \qquad (3)$$

ここで，λは手動で調整されている調整係数である．これはBNLL損失関数とℓ_1−正則化項の和である．また，パラメータに対する目的関数は制約なし凸最適化問題となるため，得られる局所的最適解は大域的最適解となる．

2.2 グラフティング

グラフティングはℓ_1−正則化ロジスティック回帰に基づきパラメータの最適化と素性選択の両方を与える学習手法である．素性集合Fが与えられたとき，グラフティングはFから一つずつ素性を選択する．素性選択を繰り返すたびに，グラフティングは汎用の勾配降下法を用いて重みパラメータを最適化し，最も目的関数Cを減少させる素性，すなわち，最大の勾配$\frac{\partial L}{\partial w_j}$である素性を選択する．残っているすべて素性の損失関数に対する勾配の大きさ$\frac{\partial L}{\partial w_j}$がλ($\ell_1$−正則化項の調整係数)以下になったとき，グラフティングは学習を終了する．得られた重みは局所的に最適であることが保証されており，グラフティングの目的関数は凸であるため，この局所的最適解は大域的最適解となっている．これは，グラフティングはℓ_1−正則化ロジスティック回帰問題の最適解を学習できることを意味している．

Algorithm 1 オンライングラフティング

Input: F, D and λ
$\mathbf{f} = ()$, $\mathbf{w} = ()$
for all $f \in F$ **do**
  let $\mathbf{f} = (f_1, \ldots, f_j)$ and $\mathbf{w} = (w_1, \ldots, w_j)$
  $\mathbf{f}^{(test)} := (f_1, \ldots, f_j, f)$, $\mathbf{w}^{(test)} := (w_1, \ldots, w_j, 0)$
  if $\left|\frac{\partial L_D\left(\mathbf{w}^{(test)}\right)}{\partial w_{j+1}}\right| > \lambda$ **then**
    $\mathbf{f} := \mathbf{f}^{(test)}$, $\mathbf{w} := \mathbf{w}^{(test)}$
    Optimize $\mathbf{w}$ for D ……(*)
    ℓ_1-reduction$(\mathbf{f}, \mathbf{w})$
  end if
end for
return $\mathbf{f}$ and $\mathbf{w}$

Algorithm 2 ℓ_1-削減(ℓ_1-reduction)

Input: $\mathbf{f}$ and $\mathbf{w}$
for all $w_j \in \mathbf{w}$ **do**
  if $w_j = 0$ **then**
    remove f_j from $\mathbf{f}$
    remove w_j from $\mathbf{w}$
  end if
end for

2.3 オンライングラフティング

オンライングラフティングは，グラフティングに基づきパラメータの最適化とオンライン素性選択の両方を与えるオンライン素性選択アルゴリズムである．オンラインでないグラフティングでは素性集合から素性を選択する際に，素性集合において最良の素性を見つけるためそれぞれの素性が何度もテストされている．オンライングラフティングでは，素性列が与えられ，各素性が順番に一度だけテストされる．

オンライングラフティングのアルゴリズムを以下に説明する．Fは素性列，Dは訓練データ，λはオンライングラフティングの調整係数である．素性ベクトル$\mathbf{f}$と重みベクトル$\mathbf{w}$を持っていると仮定する．まず，Fの最初の要素である素性fを取得し，fは素性列から削除される．次に，fは素性ベクトル$\mathbf{f}$に追加され，0が重みベクトル$\mathbf{w}$に追加される．追加された素性ベクトルを$\mathbf{f}^{(test)}$，重みベクトルを$\mathbf{w}^{(test)}$とする．オンライングラフティングのアルゴリズムにおいて追加されたfに対するテストを以下に示す．

$$\left|\frac{\partial \bar{L}}{\partial w_{j+1}}\right| > \lambda \qquad (4)$$

ここで$\bar{L}$は，Dの平均損失である．fは上記の条件を満たしていれば，新たな素性としてモデルに選択される．fが選択された場合，$\mathbf{f}$と$\mathbf{w}$が$\mathbf{f}^{(test)}$と$\mathbf{w}^{(test)}$に更新され，その後，新しい$\mathbf{f}$とDについて準ニュートン法などの汎用に用いられている数値最適化手法を使用してℓ_1-正則化ロジスティック回帰問題を解き，新たな$\mathbf{w}$が再推定される．fが選択されない場

合$\mathbf{w}$，$\mathbf{f}$に変更はない．オンライングラフティングはFが空になるまでこの手順を繰り返す．ここで，Fを素性列，Dを訓練データとする．アルゴリズム1はオンライングラフティングのアルゴリズムを示している．オンライングラフティングにおいて，最適化の結果として0になる重みは削除される（アルゴリズム2）．この削除のことをℓ_1-削減と呼ぶことにする．オンライングラフティングでは，ℓ_1-削減が頻繁に実行される．

3. 複数素性でのテスト

オンライングラフティングは，百万からなる素性や数万におよぶ学習例からなる実際のデータセットに対しては非常に高い計算コストを要するため実用的ではない．

これは大きく二つの原因が考えられ，一つ目に，パラメータの最適化に計算コストの高い汎用数値最適化法を用いていることが挙げられ，二つ目に，オンライングラフティングは素性がテストに合格するたびに高いコストの最適化を実行していることが挙げられる．例えば，1,000,000 個の要素からなる素性列が与えられ，素性の 10 分の 1 が合格した場合，パラメータの最適化を 1,000 回実行する必要がある．

本研究で提案する二つの手法は，複数個の素性を同時にテストすることでオンライングラフティングを近似している．つまり，複数個の素性を最適化することなく連続してテストし，後にまとめて最適化する手法である．

Algorithm 3 オンライングラフティング（倍数分割法）

Input: F, D, λ and C
$\mathbf{f} := ()$, $\mathbf{w} := ()$, $i := 0$, $j := 0$
for all $f \in F$ **do**
  $i := i + 1$
  let $\mathbf{f} = (f_1, \ldots, f_j)$ and $\mathbf{w} = (w_1, \ldots, w_j)$
  $\mathbf{f}^{(test)} := (f_1, \ldots, f_j, f)$, $\mathbf{w}^{(test)} := (w_1, \ldots, w_j, 0)$
  if $\left|\frac{\partial L_D\left(\mathbf{w}^{(test)}\right)}{\partial w_{j+1}}\right| > \lambda$ **then**
    $\mathbf{f} := \mathbf{f}^{(test)}$, $\mathbf{w} := \mathbf{w}^{(test)}$
  end if
  if i $\geq 2^j$ **then**
    Optimize $\mathbf{w}$ for D ……(∗)
    ℓ_1-reduction$(\mathbf{f}, \mathbf{w})$
    $i := 0$, $j := j + 1$
  end if
end for
return $\mathbf{f}$ and $\mathbf{w}$

図 1 倍数分割法のイメージ図
Figure 1 Image of multiplicative division method.

3.1 倍数分割法

一つ目の手法は，テストする素性数を倍数的に増やす倍数分割法である．まず，1 つ素性をテストし，パラメータの最適化を実行する．次に，2 つの素性をテストし，パラメータの最適化を実行する．次に，4 つの素性をテストし，パラメータの最適化を実行する．素性列の最後に到達するまでこの手順を繰り返し実行する．この手法では，i回目のパラメータの最適化の時点で，2^{i-1}個の素性に対するテストが実行されている．図 1 に倍数分割法のイメージ図を示している．また，アルゴリズム 3 には倍数分割オンライングラフティングを示している．

この手法では，学習の初期段階において頻繁にパラメータを最適化する．これは，学習初期段階において，まだ素性ベクトルに素性数が少なく，十分な素性選択能力がないと考えられるためである．素性ベクトルが十分な素性数を持った後，それほど頻繁にパラメータを更新せずに有用な素性を選択できるようになると推測できる．この手法は，大幅に最適化の回数減らすことができるが，誤って無駄な素性を選択したり，有用な素性を削除したりする可能性がある．そこで，実験においてオンライングラフティングとの精度を比較することにより，このトレードオフを検討する．

Algorithm 4 オンライングラフティング（定数分割法）

Input: F, D, λ and C
$\mathbf{f} := ()$, $\mathbf{w} := ()$, $i := 1$
for all $f \in F$ **do**
  $i := i + 1$
  let $\mathbf{f} = (f_1, \ldots, f_j)$ and $\mathbf{w} = (w_1, \ldots, w_j)$
  $\mathbf{f}^{(test)} := (f_1, \ldots, f_j, f)$, $\mathbf{w}^{(test)} := (w_1, \ldots, w_j, 0)$
  if $\left|\frac{\partial L_D\left(\mathbf{w}^{(test)}\right)}{\partial w_{j+1}}\right| > \lambda$ **then**
    $\mathbf{f} := \mathbf{f}^{(test)}$, $\mathbf{w} := \mathbf{w}^{(test)}$
  end if
  if i $\geq |F|/C$ **then**
    Optimize $\mathbf{w}$ for D ……(∗)
    ℓ_1-reduction$(\mathbf{f}, \mathbf{w})$
    i := 1
  end if
end for
return $\mathbf{f}$ and $\mathbf{w}$

図 2 定数分割法のイメージ図
Figure 2 Image of constant division method.

3.2 定数分割法

二つ目の手法は，固定の素性数でテストする定数分割法である．分割数Cを定めると，$|F|/C$個の素性を連続してテストし，後に$|F|/C$個の素性をまとめて最適化する．素性列の最後に到達すれば学習を終了する．図 2 に定数分割法のイメージ図を示している．また，アルゴリズム 4 に定数分割オンライングラフティングを示す．

この手法の利点は，パラメータの最適化の頻度を制御できるという点である．仮に，分割数Cを$|F|$とすれば，オンラインングと等しく動作する．

4. 実験

表 1　各データセットのデータサイズおよび素性数
Table 1　Specifications of data sets.

	a9a	w8a	IJCNN1	news20.binary
# 素性	123	300	22	1,355,191
# 2 次の組み合わせ素性集合	15,252	90,300	506	-
# 訓練データ	26,048	36,624	39,992	10,000
# 開発用データ	6,513	8,922	9,998	4,998
# テストデータ	16,281	13,699	91,701	4,998

4.1 データセット

a9a[9]，w8a[10]，IJCNN1[11]，news20.binary[12]の 4 つのデータセットを用いて実験を行った．a9a のタスクは，年収 50,000 ドル以上になる人を識別することであり，w8a と news20.binary は文書分類，IJCNN1 は IJCNN2001 競技会で用いられたデータセットである．各データセットは，訓練データ，開発用データ，テストデータから構成されている．各データセットの学習例，素性数を表 1 に示す．a9a，w8a，IJCNN1 は素性集合が小さいので 2 次の組み合わせの素性集合[1]を用いた．表 1 に，組み合わせ素性集合の素性数を示す．これらのデータセットはすべて二値分類のため，出力ラベルは，すべてのデータセットで$\{+1,-1\}$である．

4.2 評価

実験では，提案手法（倍数分割法，定数分割法），既存手法（オンライングラフティング），ℓ_1-正則化ロジスティック回帰（LR+L1）を比較した．定数分割法の分割数Cと正則化項の調整係数λは開発データを用いて決定した．オンライングラフティングは Python により実装した．パラメータの最適化（アルゴリズム 1, 3, 4 の (*)）のために LIBLINEAR[13] の Python パッケージを用いた．精度，学習時間，LIBLINEAR によって最適化された重みの累計数（累積最適化重み数），最適化された最大の重み数（最大最適化重み数），ℓ_1-削減の後に残っている素性数（有効素性数）を評価した．オンライン素性選択において学習例は固定されているので，学習のための計算コストは，最適化される重みの数に依存している．オンライングラフティングは基本的にはℓ_1-正則化ロジスティック回帰を近似する手法であるため，ℓ_1-正則化ロジスティック回帰の精度は，オンライングラフティングの精度よりも高くなることが期待される．しかし，ℓ_1-正則ロジスティック回帰は，入力としてすべての素性と重みを用いるため，パラメータの最適化に大幅な時間とメモリを必要とする．したがって，精度と累積最適化重み数の間にはトレードオフが存在する．実験では，このトレードオフを検証する．ここで，精度評価にはテストデータを用いた．

実験結果を表 2，3，4，5 に示す．表の結果より，2 つの提案手法を比較して，倍数分割法は累積最適化重み数が少なく，学習時間も短いことが示せた．このことより，計算コストを減少させたと言える．オンライングラフティングと 2 つの提案手法を比較した場合，提案手法は学習時間と累積最適化重み数を大きく減少させた．LR+L1 と倍数分割法の学習時間の差は比較的小さいが，倍数分割法の学習時間は LR+L1 よりも僅かに短く，最適化された重みの累積数も，LR+L1 より少なくなった．提案手法と，オンライングラフティングおよびLR+L1を比較した場合，データセット a9a，w8a，IJCNN1 において予測性能の差は非常に小さかった．これらの結果は，提案手法は予測性能の点で LR+L1 やオンライングラフティングの良い近似手法となっていることを示している．表 6 に各手法の定性的性能比較をまとめた．

4.3 組み合わせ素性の評価

組み合わせ素性の精度比較実験を行った．データセット a9a，w8a，IJCNN1 に対して 2 次の組み合わせ素性集合を生成した．組み合わせ素性集合の素性数を表 1 に示す．組み合わせ素性集合の評価実験には，組み合わせ無しの素性集合に対しては，ℓ_1-正則化付ロジスティック回帰を用い，2 次の組み合わせ素性集合に対しては，倍数分割オンライングラフティングを用いた．

表 7 に評価結果を示す．表より a9a，w8a に対しては，2 倍から 4 倍程度素性を増やすことにより僅かではあるが精度の向上が確認できた．IJCNN1 に対しては，20 倍程度素性を増やすことにより，精度が大きく改善されることが確認できた．この実験結果より，全ての組み合わせ素性を用いずとも，可能な組み合わせ素性候補から有用な素性だけを選択することにより大きく素性数を増やすことなく精度が向上することを確認できた．

5. 結論

本論文において，オンライングラフティングの効率を改良した二つの手法を提案した．オンライングラフティングはℓ_1-正則化ロジスティック回帰に基づいた有用なオンライン素性選択手法である．しかし，頻繁にパラメータ最適化を実行する学習法のため計算コストが高く非効率であった．提案手法では，同時に複数の素性をテストした後に，パラメータを最適化することで，オンライングラフティングの学習効率を改善した．

実験では，定数/倍数個の素性を同時にテストする 2 つの手法を評価した．提案手法において，効率と予測性能との間にはトレードオフがあったが，実験結果より，予測性能を低下させることなく，大幅に学習効率を改善できることを示した．倍数分割法，定数分割法ともに最適化回数を削減したことで累積最適化重み数を大幅に削減し，学習効率を大幅に向

[1] 新たな素性は，素性集合の要素を掛け合わせることで生成できる．F個の素性の組み合わせ総数は$F^{|2|}$通り存在するので，素性の組み合わせ手法により非常に長い素性列を生成できる．$F_1 = \{f_1, f_2, \ldots, f_K\}$とすれば，新しい素性集合$F_2$は$F_1$の二つの要素を乗算することで$\{f_1f_1, f_1f_2, \ldots, f_1f_K, f_2f_1, \ldots, f_Kf_K\}$として生成される．次数 L の組み合わせは，新しい素性集合$F_1 \cup F_2 \cup \ldots \cup F_L$として生成できる．ここでは，新たな素性集合を“$L$次の組み合わせ素性集合”と呼ぶことにする．

表 2　実験結果 a9a
Table 2　Experimental results for a9a.

	精度(%)	累積最適化重み数	学習時間(s)	最大最適化重み数	有効素性数
倍数分割法	85.25	6,772	10.45	2,146	451
定数分割法 C=5	85.24	6,970	23.42	1,863	329
定数分割法 C=10	85.19	8,058	34.07	1,199	379
定数分割法 C=50	85.24	13,265	99.52	451	135
オンライン グラフティング[5]	85.19	505,822	5,952.71	116	102
LR+L1	85.19	15,252	10.78	15,252	643

表 3　実験結果 w8a
Table 3　Experimental results for w8a.

	精度(%)	累積最適化重み数	学習時間(s)	最大最適化重み数	有効素性数
倍数分割法	99.04	38,209	17.26	12,491	673
定数分割法 C=5	99.04	37,713	37.77	9,429	673
定数分割法 C=10	99.07	39,807	47.03	5,083	596
定数分割法 C=50	99.06	53,424	131.52	1,512	442
オンライン グラフティング[5]	99.05	8,833,804	56,029.97	278	269
LR+L1	99.11	90,300	24.4	90,300	958

表 4　実験結果 IJCNN1
Table 4　Experimental results for IJCNN1.

	精度(%)	累積最適化重み数	学習時間(s)	最大最適化重み数	有効素性数
倍数分割法	97.64	742	20.44	394	391
定数分割法 C=5	97.60	1,080	41.07	388	386
定数分割法 C=10	97.58	1,962	67.92	380	380
定数分割法 C=50	97.61	8,810	332.75	371	371
オンライン グラフティング[5]	97.61	68,438	2,633.61	317	314
LR+L1	97.62	506	15.67	506	414

表 5　実験結果 news20.binary
Table 5　Experimental results for news20.binary.

	精度(%)	累積最適化重み数	学習時間(s)	最大最適化重み数	有効素性数
倍数分割法	94.90	37,176	11.77	5,007	2,327
定数分割法 C=5	94.96	221,574	20.56	205,210	2,615
定数分割法 C=10	94.94	145,456	34.44	115,802	2,201
定数分割法 C=50	94.76	128,409	141.62	24,419	1,756
オンライン グラフティング[5]	95.00	16,302,937	15,994.70	1,851	1,412
LR+L1	96.22	1,355,191	12.05	1,355,191	11,224

表 6　各手法の定性的性能比較
Table 6　Qualitative performance comparison of each technique.

	速度	精度	累積最適化重み数	最大最適化重み数
倍数分割法	高速	最適解に近い	小	中
定数分割法	高速～非常に低速	最適解に近い	大～小	大～小
オンライングラフティング（従来法）	非常に低速	ほぼ最適解	大	小
LR+L1	高速	最適解	中	大

表 7　組み合わせ素性集合による精度と素性数
Table 7　Precision with combination feature sets.

	組み合わせなし (精度/素性数)	2 次の組み合わせ素性 (精度/素性数)
a9a	84.91%/123	85.25%/451
w8a	98.94%/300	99.04%/673
IJCNN1	91.98%/22	97.64%/391

上させた．また，倍数分割法において十分に重みが学習された後は，多くの素性を一度にテストしても予測性能を下げることはなかったと考えられる．定数分割法においては，最大最適化重み数がオンライングラフティングと並んで小さかった．いくつかのデータセットにおいて，提案手法は，オンライングラフティングやℓ_1-正則化ロジスティック回帰の予測性能を上回った．

[謝辞]

14 拠点に分散して配置された 1,900 以上の CPU コアで構成される計算環境 InTrigger を提供していただいた InTrigger チームに感謝する．本研究は JSPS 基盤研究 (C) 22500121 の助成を受けたものである．

[文献]

[1] Isabelle Guyon and Andr´e Elisseeff. An introduction to variable and feature selection. Journal of MachineLearning Research, Vol. 3, pp. 1157–1182, mar 2003.
[2] Daisuke Okanohara and Jun'ichi Tsujii. Learning combination features with L1 regularization. In Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers (NAACL-short'09), pp.97–100, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.
[3] Jianfeng Gao, Galen Andrew, Mark Johnson, and Kristina Toutanova. A comparative study of parameter estimation methods for statistical natural language processing. In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL'07), pp. 824–831, Prague, Czech Republic, June 2007. The Association for Computational Linguistics.
[4] Simon Perkins, Kevin Lacker, James Theiler, Isabelle Guyon, and Andre' Elisseeff. Grafting: Fast, incremental feature selection by gradient descent in function space. Journal of Machine Learning Research, Vol. 3, pp. 1333–1356, 2003.
[5] Simon Perkins and James Theiler. Online feature selection using grafting. In International Conference on Machine Learning (ICML 2003), pp. 592–599. ACMPress, 2003.
[6] Jun Zhu, Ni Lao, and Eric P. Xing. Grafting-light: fast, incremental feature selection and structure learning of markov random fields. In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '10, pp. 303–312, New York, NY, USA, 2010. ACM.
[7] Trevor Hastie, Robert Tibshirani, and J. H. Friedman. The Elements of Statistical Learning. Springer, New York: Springer-Verlag, 2001.
[8] Robert Tibshirani. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society (Series B), Vol. 58, pp. 267–288, 1994.
[9] A. Frank and A. Asuncion. UCI machine learning repository, 2010.
[10] John C. Platt. Advances in kernel methods. chapter Fast training of support vector machines using sequential minimal optimization, pp. 185–208. MIT Press., 1999.
[11] Danil Prokhorov. IJCNN 2001 neural network competition. In IJCNN'01, Ford Research Laboratory, 2001.
[12] S. Sathiya Keerthi and Dennis DeCoste. A modified finite newton method for fast solution of large scale linear SVMs. Journal of Machine Learning Research, Vol. 6, pp. 341–361, dec 2005.
[13] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, XiangRui Wang, and Chih-Jen Lin. LIBLINEAR:a library for large linear classification. Journal of Machine Learning Research, Vol. 9, pp. 1871–1874, 2008.

大井 健吾　Kengo OOI
愛媛大学大学院理工学研究科博士前期課程在学中．2011 年愛媛大学工学部情報工学科卒業．情報処理学会学生会員．

二宮 崇　Takashi NINOMIYA
1996 年東京大学理学部卒．2001 年同大学院博士課程修了．博士(理学)．2010 年より愛媛大学准教授．機械学習，自然言語処理，計算言語学の研究に従事．