Vol.2011-MPS-82 No.18
2011/3/8

特徴の出現回数に応じた L_1 正則化を実現する教師ありオンライン学習手法

大岩秀和　松島慎　中川裕志[†1]

オンライン学習（逐次学習）とは，訓練データを 1 つ受け取るたびに逐次的に学習を行う手法であり，大規模な訓練集合からの学習が効率化される．L_1 正則化とは，学習上不要なパラメータを零化する手法で，学習の高速化やメモリ効率の改善が期待される．2009 年に提案された FOBOS[7] は上記の二手法を組み合わせた，教師あり学習のための L_1 正則化付きオンライン学習手法である．しかし FOBOS では，特徴の出現回数が不均一な訓練集合では，低頻度の特徴が排除されやすい性質を持つ．FOBOS では各特徴の出現頻度やパラメータの累積更新幅とは無関係に全特徴に共通の零化を施すためである．しかし既存の L_1 正則化付きオンライン学習アルゴリズムでは，この性質は分析されてこなかった．本稿では，特徴の出現回数の情報を用いた L_1 正則化を導入した教師あり学習のためのオンライン学習手法 (HF-FOBOS) を提案する．さらに，HF-FOBOS は既存手法と同様の計算コスト・収束速度でパラメータの累積更新幅に応じた L_1 正則化を実現する学習手法である事を確認する．また，HF-FOBOS と FOBOS に対して実問題に基づく実験を行い，出現頻度を利用した L_1 正則化が精度向上へ寄与することを示した．

L_1 regularized online supervised learning using feature frequency

Hidekazu Oiwa , Shin Matsushima
and Hiroshi Nakagawa [†1]

Online learning is a method that updates parameters whenever it receives a single data. Online learning can learn efficiently from large data set. L_1 regularization is used for inducing sparsity into parameters and exclude unnecessary parameters. FOBOS[7] combines these two methods described above and presented a supervised online learning method with an efficient L_1 regularization. FOBOS has the property the parameters of low frequency features are zeros in a heterogeneous data set. However, this property is not analyzed enough in the field of online learning. In this paper, we presented a new online supervised learning method with L_1 regularization based on the number of occurrences of feature, named Heterogeneous Frequency FOBOS (HF-FOBOS). HF-FOBOS can solve optimization problems at same computational costs and convergence rate as FOBOS. Moreover, we examined the performance of our algorithms with classification tasks, and confirmed L_1 regularization based on the frequency of features improve accuracy.

1. はじめに

機械学習における教師あり学習では，入力データと出力データのペア事例を大量に集めた訓練集合を用いて，入力から出力への関数を学習する．教師あり学習は，株価予測やメールフィルタリング・画像認識等，様々な分野に応用されている．そのため，高精度かつ高速な教師あり学習のためのアルゴリズムが長年研究されており，現在にいたるまで様々な手法が提案されてきた．

2000 年頃からネットワーク技術やデータベース技術の急激な進歩により，訓練集合として利用可能なデータが爆発的に増大した．訓練集合の大規模化は，高精度な学習器を設計するうえで重要な役割を果たす．しかし大規模データを扱うには，学習にかかる計算時間や空間計算量コストが制約となる事が多い．

オンライン学習は，訓練集合中のデータが 1 つずつ逐次的に与えられ，そのたびに学習器を更新する手法である．また，学習に不要な特徴を排除する働きを持つ L_1 正則化 (Lasso 正則化) が注目されている．上記の 2 手法は，高速かつ少ない作業領域での学習を実現できる．

オンライン学習と L_1 正則化を同時に実現する Forward Backward Splitting (FOBOS) が 7) で提案された．ただし，FOBOS の L_1 正則化は全パラメータに対して共通の零化が施されているため，低頻度の特徴に対応するパラメータは学習への有用性と無関係に 0 になりやすい．また，テキスト情報や画像情報を扱う場合，一般的に訓練集合中の特徴出現頻度が不均一である．

そこで，本稿では FOBOS を拡張した新たな L_1 正則化付きオンライン学習手法を提案する．提案手法は，特徴の出現回数の情報を用いた L_1 正則化を実現する．従って，各特徴の出現回数が不均一な場合にも頻度情報に影響を受けにくい零化を可能にした．さらに，提

†1 東京大学
The University of Tokyo

案手法はパラメータ更新の計算コストが FOBOS と同程度であること，データ数の増加に応じて最適解に収束することを示す．次に，実データに基づくテストデータを利用して提案手法と FOBOS の性能の比較と検証を行った．その結果，提案手法は全データセットで FOBOS よりも高い精度を達成し，予測の向上に出現頻度情報を利用することの重要性が示唆された．

本稿の構成は以下の通りである．第 2 章では，本稿で用いる記号の定義を行い，問題設定について解説する．第 3 章では，オンライン学習と L_1 正則化を組み合わせた FOBOS のアルゴリズムとその性質について述べる．第 4 章では，特徴の出現回数の情報を用いた L_1 正則化を実現する手法を提案し，そのアルゴリズムと性質について説明する．第 5 章では，実データを用いた従来手法との比較実験を行い，その性能を比較する．最後に，第 6 章で結論について述べる．

2. 問題設定

本稿では，スカラーは小文字 λ，ベクトルは太字の小文字 $\mathbf{x}$，行列は太字の大文字 $\mathbf{X}$ で表記する．ベクトルの第 i 成分は，$\mathbf{x}^{(i)}$，行列の第 i,j 成分は，$\mathbf{x}^{(i,j)}$ と表す．L_p ノルムは $\|\mathbf{v}\|_p$ と表記する．ただし，L_1 ノルム $\|\mathbf{v}\|_1$ は $|\mathbf{v}|$，L_2 ノルム $\|\mathbf{v}\|_2$ は $\|\mathbf{v}\|$ と略記することもある．$\langle\cdot,\cdot\rangle$ はベクトルの内積を表す．また $[a]_+$ は，

$$[a]_+ = \begin{cases} a & if \quad a \geq 0 \\ 0 & if \quad a < 0 \end{cases}$$

$sign(a)$ は，$sign(a) = a/|a|$（但し，$a = 0$ であれば $sign(a) = 0$）を返す関数と定義する．

本稿で対象とするのは，正則化項付きオンライン学習による教師あり学習である．教師あり学習の目的は，入力データ $\mathbf{x}$ と対応する出力データ y のペアが大量に含まれる訓練集合 $S = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots (\mathbf{x}_T, y_T)\}$ を用いて，入力から出力への関数を設計することである．この関数は，入力ベクトルから決定される特徴ベクトル $\Phi(\mathbf{x}) \in \Re^n$ と重みベクトル $\mathbf{w} \in \Re^n$ の内積で定義される．従って，出力データの予測値 $\hat{y}$ は $\mathbf{w} \cdot \Phi(\mathbf{x})$ で表す．

正則化項付きオンライン学習による教師あり学習では，入力と出力のペアが 1 組与えられるたびに重みベクトル $\mathbf{w}$ を逐次的に更新する．具体的には，次の手順で重みベクトルの更新が行われる．

(1) t 番目の入力データ $\mathbf{x}_t$ を受け取る
(2) 現在の重みベクトル $\mathbf{w}_t$ と特徴ベクトル $\Phi(\mathbf{x}_t)$ の内積を計算し，出力データの予測値 $\hat{y}_t$ を求める
(3) 入力データ $\mathbf{x}_t$ に対応する真の出力データ y_t を受け取る
(4) 予測値 $\hat{y}_t$ と真の値 y_t を用いて重みベクトルを $\mathbf{w}_{t+1}$ に更新する
(5) $t+1$ 番目のデータが存在する場合，(1) に戻る．

このように，オンライン学習ではデータを 1 つ受け取るたびに逐次的に重みベクトルを更新する．オンライン学習を用いる主な利点として，空間計算量の効率化や再学習の容易性等が挙げられる．

重みベクトルの更新基準は，損失関数 $\ell(\cdot) : \Re^n \to \Re_+$ と正則化項 $r(\cdot) : \Re^n \to \Re_+$ の和で定められる．損失関数 $\ell(\cdot)$ とは，重みベクトル $\mathbf{w}_t$ から導出される予測値 $\hat{y}_t$ が真の値 y_t から乖離しているほど，値が大きくなる関数である．損失関数には，二乗損失 $\ell_t(\mathbf{w}) = (y_t - \mathbf{w} \cdot \Phi(\mathbf{x}_t))^2$，Hinge-Loss $\ell_t(\mathbf{w}) = [1 - y_t \mathbf{w} \cdot \Phi(\mathbf{x}_t)]_+$ 等が用いられる．正則化項は，重みベクトルの複雑さを表現する関数で，平滑化の働きを持つ．正則化項には，L_1 ノルム $r(\mathbf{w}) = \lambda|\mathbf{w}|$ や L_2 ノルム $r(\mathbf{w}) = \lambda\|\mathbf{w}\|$ が良く用いられる．ここで，λ は損失関数と正則化項それぞれの学習への寄与度を調節するスカラーである．

正則化項に L_1 ノルムを導入する事は L_1 正則化と呼ばれる．L_1 正則化は学習上不要なパラメータを零化し，重みベクトルを疎な形に変化させる性質を持つ．重みベクトルを疎化させる働きはスパース化と呼ばれ，パラメータ次元数を圧縮できる．従って，膨大な次元数のデータを用いて最適化問題を解く際に計算コストを削減する効果を持つ．

本稿で扱う損失関数は

$$\ell_t(\mathbf{w} \cdot \Phi(\mathbf{x}_t); y_t) = \ell_t(\hat{y}_t; y_t) \tag{1}$$

で表現可能な関数に限定する．(1) 式が成立するとき損失関数の重みベクトルに関する勾配は，$\Phi(\mathbf{x}_t)$ のスカラー倍で表すことが出来る．さらに，損失関数は凸性を持つ関数に限定する．ここで，(2) 式を満たす関数 f を凸性を持つ関数と定義する．

$$\forall \mathbf{a}_1, \mathbf{a}_2 \in \Re^n \quad \forall \lambda \in [0,1] \quad \lambda f(\mathbf{a}_1) + (1-\lambda) f(\mathbf{a}_2) \geq f(\lambda \mathbf{a}_1 + (1-\lambda)\mathbf{a}_2) \tag{2}$$

上で挙げた二乗損失や Hige-Loss 等の損失関数はこれらの制約を全て満たす．

正則化項付きオンライン学習における教師あり学習の目的は，学習過程で生じた損失関数と正則化項の総和を最小化することである．これを式で表すと，

$$\min_{\mathbf{w}_1, \mathbf{w}_2, \ldots} \sum_t \{\ell_t(\mathbf{w}_t) + r(\mathbf{w}_t)\} \tag{3}$$

となり，この値を最小化する最適な重みベクトル $\mathbf{w}_t^*$ の導出が目的となる．(3) 式は，各時点 t の重みベクトル $\mathbf{w}_t$ に関して独立に，$\ell_t(\mathbf{w}_t) + r(\mathbf{w}_t)$ を最小化することと同値である．

つまり，各 t に関して $\ell_t(\cdot)+r(\cdot)$ の値が最小化される最適な重みベクトル $\mathbf{w}_t^*$ に更新する戦略を構築することがオンライン学習の目標である．

しかし，(3) 式から最適な戦略を設計することは不可能である．いかなる戦略を用いても (3) 式の損失関数 $\ell_t(\cdot)$ を更新後の重みベクトルに不適合な形に設計してやれば (3) 式の上限を無限大へ発散させることが可能なためである．従って，(3) 式の最小化問題からオンライン学習の戦略を評価することは非常に困難である．そこで，重みベクトル $\mathbf{w}_t$ の更新戦略の性能は Regret と呼ばれる概念で評価する．オンライン学習における T 個のデータを受け取った後の Regret は，(4) 式で定義される．

$$\mathrm{Regret}(T)=\sum_{t=1}^{T}\{\ell_t(\mathbf{w}_t)+r(\mathbf{w}_t)-\ell_t(\mathbf{w}^*)-r(\mathbf{w}^*)\}$$
$$s.t.\quad \mathbf{w}^*=\arg\min_{\mathbf{w}}\sum_{t=1}^{T}\{\ell_t(\mathbf{w})+r(\mathbf{w})\} \tag{4}$$

Regret 上限が $o(T)$ となるオンライン学習手法では，T の増加に応じて 1 データあたりの Regret 上限は減少し，0 に収束する．従って，その手法で導出される重みベクトル $\mathbf{w}_t$ は，訓練集合全体に対する最適なパラメータ $\mathbf{w}^*$ への収束が保証される．さらに，Regret 上限が小さなオーダーで抑えられるならば，重みベクトルはより高速に最適なパラメータへ収束する事が示される．この事から，Regret 上限の最小化をオンライン学習の目的と置くことが出来る．

主なオンライン学習手法として，Perceptron[11], Online Passive-Aggressive[3], Confidence-Weighted Algorithms[4)–6)] などが提案されている．

3. Forward Backward Splitting (FOBOS)

オンライン学習と L_1 正則化を組み合わせるには，(3) 式の正則化項に L_1 正則化を導入し，劣勾配法を用いることが考えられる．ただし，この方法は，重みベクトルのスパース化の働きが損なわれる．上記の更新手法は，L_1 正則化による更新と損失関数による更新が同時に行われるため，L_1 ノルムの零化の働きが損失関数の劣勾配に妨害されるためである．

Forward Backward Splitting (以下，FOBOS) は上記の問題を解決し，オンライン学習と L_1 正則化をそれぞれの特性を損なう事無く，組み合わせるアプローチを考案した．FOBOS では，L_1 ノルムによるスパース化を実現するため，重みベクトルの更新アルゴリズムを次の 2 ステップに分解している．

ステップ 1 は，損失最小化ステップである．ステップ 1 では，正則化項を除いた最適化問題を劣勾配法[1] によって解く．

$$\mathbf{w}_{t+1/2}=\mathbf{w}_t-\eta_t\mathbf{g}_t^{\ell} \tag{5}$$

ここで，$\eta_t\in\Re_+$ は t 番目のデータにおける損失最小化ステップのステップ幅である．ステップ幅 η_t は，t 番目のデータでの更新幅を調節する．さらに，$\mathbf{g}_t^{\ell}$ は劣勾配である．劣勾配とは，(6) 式を満足するベクトル $\mathbf{g}\in\Re^n$ と定義する．

$$\forall\mathbf{y}\quad f(\mathbf{y})\geq f(\mathbf{x})+\langle\mathbf{g},\mathbf{y}-\mathbf{x}\rangle \tag{6}$$

損失関数や正則化項に微分不可能な点が存在する場合でも関数が凸であれば劣勾配は必ず存在する．劣勾配の集合を $\partial f(\mathbf{x})$ で表しており，$\mathbf{g}_t^{\ell}\in\partial\ell_t(\mathbf{w}_t)$ は，$\ell_t(\mathbf{w}_t)$ の任意の劣勾配である．

劣勾配法では，損失関数 $\ell_t(\cdot)$ の劣勾配と逆の方向に，η_t で重み付けされた幅だけパラメータを更新している．従って，劣勾配法は損失関数が最小化される方向へ逐次的にパラメータを更新するアルゴリズムである．

ステップ 2 は，L_1 正則化ステップである．ステップ 2 では，ステップ 1 で求めた重みベクトル $\mathbf{w}_{t+1/2}$ を変化させることに罰則を課すと同時に，L_1 ノルムで正則化を施す．

$$\mathbf{w}_{t+1}=\arg\min_{\mathbf{w}}\left\{\|\mathbf{w}-\mathbf{w}_{t+1/2}\|^2/2-\eta_{t+1/2}\lambda\|\mathbf{w}\|_1\right\} \tag{7}$$

$\eta_{t+1/2}\in\Re_+$ は L_1 正則化項のステップ幅である．$\eta_{t+1/2}$ は，L_1 正則化の寄与度を調整する．

ステップ 1 とステップ 2 から閉じた形のパラメータ更新式が導出できる．

$$\begin{aligned}\mathbf{w}_{t+1}^{(j)}&=sign\left(\mathbf{w}_{t+1/2}^{(j)}\right)\left[\left|\mathbf{w}_{t+1/2}^{(j)}\right|-\eta_{t+1/2}\lambda\right]_+\\&=sign\left(\mathbf{w}_t^{(j)}-\eta_t\mathbf{g}_t^{\ell,(j)}\right)\left[\left|\mathbf{w}_t^{(j)}-\eta_t\mathbf{g}_t^{\ell,(j)}\right|-\eta_{t+1/2}\lambda\right]_+\end{aligned} \tag{8}$$

(8) 式より，$w_t^{(j)}$ から直接 $w_{t+1}^{(j)}$ が導出可能であることが確認できる．また，

$$\left[|\mathbf{w}_{t+1/2}^{(j)}|-\eta_{t+1/2}\lambda\right]_+=\begin{cases}0 & if\quad \left|\mathbf{w}_{t+1/2}^{(j)}\right|<\eta_{t+1/2}\lambda\\ \mathbf{w}_{t+1/2}^{(j)}-\eta_{t+1/2}\lambda & if\quad \mathbf{w}_{t+1/2}^{(j)}>\eta_{t+1/2}\lambda\\ \mathbf{w}_{t+1/2}^{(j)}+\eta_{t+1/2}\lambda & if\quad \mathbf{w}_{t+1/2}^{(j)}<-\eta_{t+1/2}\lambda\end{cases} \tag{9}$$

より，$|\mathbf{w}_{t+1/2}^{(j)}|<\eta_{t+1/2}\lambda$ となるパラメータ $\mathbf{w}_{t+1/2}^{(j)}$ は，ステップ 2 で全て 0 になる．

損失関数や正則化項が一定の条件をみたすとき，適切なステップ幅を設定すれば，FOBOS における Regret 上限 (4) は $O(\sqrt{T})$ となる事が証明されている．

Vol.2011-MPS-82 No.18
2011/3/8

4. Heterogeneous Frequency FOBOS(HF-FOBOS)

3 章では，L_1 正則化を導入したオンライン学習手法である FOBOS について説明した．FOBOS では (9) 式で示したとおり，全パラメータ共通の零化を施している．従って，低頻度の特徴に対応するパラメータは，パラメータ更新の累積値そのものが小さいために高頻度の特徴に比べ 0 になりやすい．

例として，訓練集合中に出現頻度が 1/2 の特徴 a と 1/100 の特徴 b が共存する場合を考える．この時，特徴 b は，毎回のパラメータの更新幅が

$$\eta_t|\mathbf{g}_t^{\ell,(b)}| \geq \lambda \sum_{s=t}^{t+100} \eta_{s+1/2}$$

を満たさなければ，零化が必ず発生する．一方，特徴 a は毎回のパラメータの更新幅が

$$\eta_t|\mathbf{g}_t^{\ell,(a)}| \geq \lambda \sum_{s=t}^{t+1} \eta_{s+1/2}$$

ならば，零化は必ずしも発生しない．つまり FOBOS では，特徴間で劣勾配の値が同一であったとしても出現頻度に違いがあると，零化の結果が変化する危険性がある．

本稿の提案手法 (HF-FOBOS) では，L_1 正則化に各特徴の出現頻度を利用することで FOBOS の改良を行った．HF-FOBOS では，高頻度な特徴に対応するパラメータが L_1 正則化時に大きな値で重み付けされる．そのため，低頻度の特徴が零化されやすい性質が改善されるように設計されている．

初めに，FOBOS のステップ 1 で損失関数から計算される特定の特徴の劣勾配を並べたベクトルを新しく定義する．(10) 式で定義されるベクトルを劣勾配ベクトルと呼ぶ．

$$\mathbf{h}_t^{(j)} = (\mathbf{g}_1^{\ell,(j)}, \mathbf{g}_2^{\ell,(j)}, \ldots, \mathbf{g}_t^{\ell,(j)}) \tag{10}$$

HF-FOBOS では，劣勾配ベクトル $\mathbf{h}_t^{(j)}$ を用いて，FOBOS のステップ 2 を (11) 式に変更する．

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}} \left\{ \|\mathbf{w} - \mathbf{w}_{t+1/2}\|^2/2 - \lambda\eta_{t+1/2}|\mathbf{H}_t\mathbf{w}| \right\} \tag{11}$$

ここで，

$$\mathbf{H}_t = \begin{pmatrix} \mathbf{h}_{t,norm}^{(1)} & 0 & \ldots & 0 \\ 0 & \mathbf{h}_{t,norm}^{(2)} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \mathbf{h}_{t,norm}^{(n)} \end{pmatrix}$$

$\mathbf{h}_{t,norm}^{(j)} = \|\mathbf{h}_t^{(j)}\|_p$ とする．つまり $\mathbf{H}_t$ は，劣勾配ベクトルの L_p ノルムを対角項に並べた行列である．行列 $\mathbf{H}_t$ と重みベクトル $\mathbf{w}_t$ の積をとると，重みベクトルの各成分が，対応する $\mathbf{h}_{t,norm}^{(j)}$ によって重み付けされる．高頻度の特徴 j は，劣勾配ベクトルの非零の要素が増えるため $\mathbf{h}_{t,norm}^{(j)}$ の値も大きくなる．従って，高頻度の特徴 j には強い零化が作用する．

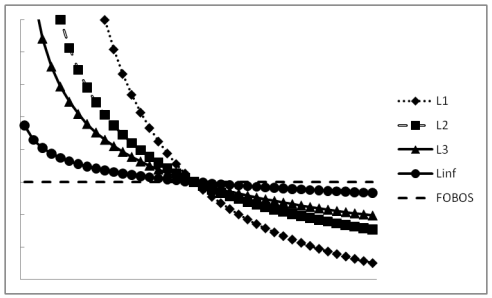

図 1 各ノルムにおける $\mathbf{h}_{t,norm}^{(j)}$ の比較
Fig. 1 comparison of the parameter of $\mathbf{h}_{t,norm}^{(j)}$ when norms change

ベクトル $\mathbf{h}_{t,norm}^{(j)}$ は L_1 ノルム，L_2 ノルム，L_3 ノルム等，様々なノルムに置き換えることが出来る．ノルムを変化させると，行列 $\mathbf{H}_t$ の対角項の値も変化する．各ノルムにおける $\mathbf{h}_{t,norm}^{(j)}$ の変化を図 1 に示す． 図 1 は，左から右へ $\mathbf{h}_{t,norm}^{(j)}$ の値が大きい順に並べている．縦軸は，各ノルムにおける特徴 j の $\mathbf{h}_{t,norm}^{(j)}$ の値である．図 1 から L_1 ノルムは出現頻度の変化に最も影響を受ける．一方，L_∞ ノルムは出現頻度には影響を受けにくく，最も FOBOS に近い．

HF-FOBOS の更新式は，FOBOS と同様の手順で導出可能である．

$$\begin{aligned} \mathbf{w}_{t+1}^{(j)} &= sign\left(\mathbf{w}_{t+1/2}^{(j)}\right)\left[\left|\mathbf{w}_{t+1/2}^{(j)}\right| - \eta_{t+1/2}\mathbf{h}_{t,norm}^{(j)}\lambda\right]_+ \\ &= sign\left(\mathbf{w}_t^{(j)} - \eta_t\mathbf{g}_t^{\ell,(j)}\right)\left[\left|\mathbf{w}_t^{(j)} - \eta_t\mathbf{g}_t^{\ell,(j)}\right| - \eta_{t+1/2}\mathbf{h}_{t,norm}^{(j)}\lambda\right]_+ \end{aligned} \tag{12}$$

この更新式から，HF-FOBOS のパラメータ更新は FOBOS のパラメータ更新と同じ計算量オーダーで実現できることが分かる．

さらに，HF-FOBOS は，FOBOS と同等の Regret 上限を持つことが示せる．

定理 4.1 $\mathbf{H}_t$ の対角成分 $\mathbf{h}_{t,norm}^{(k)}$ を L_p ノルムに設定した時，

$$\mathbf{H}_t^{(k,k)} = \begin{cases} min(\mathbf{h}_{t,norm}^{(k)}, V) & p \leq 2 \\ \mathbf{h}_{t,norm}^{(k)} & p > 2 \end{cases}$$

と定義する．損失関数と正則化項が凸性を持ち，かつ $\forall \mathbf{w}_t \quad \|\mathbf{w}_t - \mathbf{w}^*\| \leq D$, $\|\partial\ell_t\| \leq U$,

$\|\partial r\| \leq U$ が成立し，η_t を任意の定数 $c > 0$ を用いて $\eta_t = \eta_{t+1/2} = c/\sqrt{t}$ と設定する．この時，(13) 式が導出される．

$$R_{\ell+r}(T) \leq 2UD + \left(D^2/2c + 8U^2c\right)\sqrt{T} = O(\sqrt{T}) \quad (13)$$

ここで，$\|\partial f(\mathbf{w})\| = sup_{\mathbf{g}\in\partial f(\mathbf{w})}\|\mathbf{g}\|$ と定義している．定理 4.1 の条件は，一般的な損失関数，正則化項に対して成立することが知られている[9]．従って，HF-FOBOS の Regret 上限は $O(\sqrt{T})$ になる[⋆1]．

5. 実　　験

実問題に基づくテストデータを用いて，本稿の提案手法 (HF-FOBOS) に関する性能評価を行った．今回行った実験の手法について説明する．実験には，Amazon.com のデータセットである[2] books,dvd [⋆2], ニュース記事に基づく 20 NewsGroups[8](news20) のサブセット[⋆3]，Reuters-21578[10](reut20) [⋆4] を用いた．

books, dvd は，Amazon.com のレビュー記事からなる評価分類のデータセットである．各レビュー文章から特徴ベクトルを生成し，その文章が製品に対して positive な意見を述べているかか否かを判定するタスクである．

news20 は約 20,000 件，20 カテゴリのニュース記事からなるデータセットである．ニュース記事から各単語の tf-idf 値を並べたベクトルを生成し，そのベクトルを用いてその記事が属するカテゴリを判別するタスクである．本研究では，news20 のうち，ob-2-1, sb-2-1, ob-8-1, sb-8-1 の 4 つのサブセットを用いた．サブセット毎に，分類したいカテゴリの数や記事の精度が異なる．サブセットの名前の意味は以下の通りである．1 文字目のアルファベットは'o' は'overlapped', 's' は'separeted' を意味している．'o' の方が記事の精度が低い．2 文字目のアルファベットは，クラス間のデータ数の不均一性を表している．'b' は'balanced' を意味し，クラス間でデータ数は均等である．最後の数字は，サブセット中に登場するクラスの種類数を表している．

Reuters-21578 (reut20) も news20 と同じくニュース記事からなるデータセットである．本研究では，このコーパスから 20 クラスの分類を作成し，使用した．

表 5 に，各データセットの特徴次元数やデータ数，ニュースカテゴリの数（クラス数）を示す．

表 1　データセットの概要
Table 1　Abstract of datasets

	データ数	特徴次元数	カテゴリ数
books	4,465	332,441	2
dvd	3,586	282,901	2
ob-2-1	1,000	5,942	2
sb-2-1	1,000	6,276	2
ob-8-1	4,000	13,890	8
sb-8-1	4,000	16,282	8
reut20	7,800	34,488	20

損失関数には Hinge-Loss を使用した．ただし，3 クラス以上の分類問題の場合は FOBOS, HF-FOBOS の更新式をそのまま適用することは出来ない．そこで本実験では，3) と同様の手法で 3 クラス以上の分類問題に適用した．ステップ幅 η_t は，Regret の条件をみたすように $\eta_t = \eta_{t+1/2} = 1/\sqrt{t}$ とした．また，L_1, L_2 ノルムの HF-FOBOS では，$V = 500$ に設定した[⋆5]．さらに，FOBOS,HF-FOBOS ともに損失関数と正則化項のそれぞれの学習への寄与度を調整するパラメータ λ を設定する必要がある．本実験では，性能評価の際にはデータセットを 10 分割した交差検定法を用いて，各手法において精度が最も高くなるパラメータ λ の値を選択した．各試行では，20 回反復を行った．

実験では，HF-FOBOS の L_1 ノルム，L_2 ノルム，L_3 ノルム，L_∞ ノルムと FOBOS の 5 種類の手法で実験を行い，精度と重みベクトル中の 0 要素の割合を比較した．

実験の結果をまとめた表が**表 2** である．表 2 には，各データセットで，交差検定法で求めた最適な λ の値で実験した時の分類精度を示す．この数値が高いほど，高精度な学習器が設計されていることを表している．角括弧 [·] 内の数値は標準偏差を示す．括弧 (·) 内の数値は，各データセットの各手法で λ を固定した時の重みベクトル中の 0 要素の割合を示す．また，各データセットで最高精度を達成した値は**太字**で示す．

表 2 より，全データセットに対して，HF-FOBOS は FOBOS よりも高い精度を示している事が確認できる．特に，L_2 ノルムと L_3 ノルムの HF-FOBOS は全データセットにおいて一様に FOBOS からの精度向上が示された．この結果から，特徴の出現頻度情報を用いた L_1 正則化は精度の向上に寄与することが示唆される．

⋆1 詳細な証明手順は，http://www.r.dl.itc.u-tokyo.ac.jp/ oiwa/ref/HF-FOBOS.pdf に記載している
⋆2 http://www.cs.jhu.edu/∼mdredze/datasets/sentiment
⋆3 http://mlg.ucd.ie/datasets
⋆4 http://www.daviddlewis.com/resources/testcollections/reuters21578
⋆5 今回の実験では，$\mathbf{h}^{(j)}_{t,norm}$ の値は 500 を超えなかったため，V の値は結果に影響しない．

表 2　各分類手法の正識別率 [%]・零特徴率 [%] 反復回数：20 回
Table 2　precision and sparseness rate of the experiments (number of iterations : 20)

	HF-FOBOS L_1	HF-FOBOS L_2	HF-FOBOS L_3	HF-FOBOS L_∞	FOBOS
books	85.23[1.52] (34.52)	**85.52**[1.24] (48.26)	85.14[1.33] (49.58)	85.05[1.41] (69.39)	84.98[1.61] (48.28)
dvd	82.49[1.68] (37.46)	84.75[1.75] (59.72)	**85.03**[2.28] (63.74)	84.02[1.66] (67.19)	83.91[1.55] (79.57)
ob-2-1	97.00[1.73] (42.78)	**97.10**[1.14] (56.73)	96.90[1.87] (59.03)	96.80[1.94] (59.78)	96.40[1.96] (49.23)
sb-2-1	**98.90**[0.83] (60.13)	98.40[0.80] (70.32)	98.40[1.11] (71.99)	98.10[1.14] (72.69)	97.20[1.78] (84.25)
ob-8-1	92.25[1.14] (62.83)	**93.10**[1.41] (62.84)	93.00[1.29] (64.64)	91.45[1.33] (77.78)	90.63[1.64] (87.90)
sb-8-1	90.90[1.72] (68.26)	92.55[1.85] (68.49)	**93.78**[2.44] (70.23)	91.25[1.44] (83.53)	90.53[1.61] (67.46)
reut20	95.23[0.65] (89.11)	**96.04**[0.56] (90.38)	95.91[0.55] (90.21)	94.80[0.67] (91.05)	95.53[0.63] (89.29)

6. ま と め

本稿では，特徴の出現頻度情報を利用した L_1 正則化付きオンライン学習のための新しい数理モデルを提案した．既存の L_1 正則化付きオンライン学習では，訓練集合中の特徴の出現頻度が不均一な場合，低頻度の特徴が優先的に零化される性質を持つ．そこで，提案手法である HF-FOBOS では，各特徴の累積更新幅を用いた L_1 正則化を導入することで，頻度の低い特徴が優先的に排除されなくなり，特徴出現頻度や各特徴が取りうる値が不均一なデータに対しても不要な特徴のみを排除する適切なスパース化を可能にした．さらに本稿では，実データを用いて，HF-FOBOS と FOBOS の性能比較を行った．その結果，HF-FOBOS は全データセットに対して FOBOS よりも高い精度を達成することを確認した．

参 考 文 献

1) Bertsekas, D.P. and Bertsekas, D.P.: *Nonlinear Programming*, Athena Scientific, 2nd edition (1999).
2) Blitzer, J., Dredze, M. and Pereira, F.: Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification, *Proceedings of ACL-07, the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, CZ, Association for Computational Linguistics, pp.440–447 (2007).
3) Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S. and Singer, Y.: Online Passive-Aggressive Algorithms, *Journal of Machine Learning Research*, Vol.7, pp. 551–585 (2006).
4) Crammer, K., Dredze, M. and Kulesza, A.: Multi-class confidence weighted algorithms, *EMNLP '09: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, Morristown, NJ, USA, Association for Computational Linguistics, pp.496–504 (2009).
5) Crammer, K., Fern, M.D. and Pereira, O.: Exact convex confidence-weighted learning, *In Advances in Neural Information Processing Systems 22* (2008).
6) Dredze, M. and Crammer, K.: Confidence-weighted linear classification, In ICML '08: Proceedings of the 25th international conference on Machine learning，ACM, pp.264–271 (2008).
7) Duchi, J. and Singer, Y.: Efficient Online and Batch Learning Using Forward Backward Splitting, *Journal of Machine Learning Research*, Vol.10, pp.2899–2934 (2009).
8) Lang, K.: Newsweeder: Learning to filter netnews, *Proceedings of the Twelfth International Conference on Machine Learning*, pp.331–339 (1995).
9) Langford, J., Li, L. and Zhang, T.: Sparse Online Learning via Truncated Gradient, *J. Mach. Learn. Res.*, Vol.10, pp.777–801 (2009).
10) Lewis, D.D.: Reuters-21578.
11) Rosenblatt, F.: The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, *Psychological Review*, Vol. 65, No. 6, pp. 386–408 (1958).
12) Zinkevich, M.: Online convex programming and generalized infinitesimal gradient ascent, *In ICML*, pp.928–936 (2003).