1R-03

モーメンタム項を付加した Online Manifold Regularization
Online Manifold Regularization with a Momentum Term

大堀 優 *　　徳山 豪 †
Yu Ohori　　Takeshi Tokuyama

1 序論

ラベル付きデータに加えてラベルなしデータを用いる半教師あり学習（SSL）は，少量のラベル付きデータしか得られない状況において，きわめて有用な学習方法である．また，データを観測する度に予測器を更新するオンライン学習は，昨今の大規模データ解析の需要増加に伴い，再び重要視されている．今回は，これらを組み合わせたオンライン半教師あり学習（OSSL）に取り組む．

SSL の一手法である Manifold Regularization (MR) [1] を OSSL の枠組みに拡張した Online Manifold Regularization (Online MR) [2] では，予測器の更新に確率的勾配降下法（SGD）[3] を用いている．この SGD は，汎用的手法であるものの，収束が遅いことが知られている．そこで，我々は，モーメンタム法を用いた Online MR を提案し，その収束速度について評価を行う．

2 問題設定

本研究では，半教師あり 2 クラス分類問題について議論する．今，入力列 $\{\boldsymbol{x}_t\}_{t=1}^T$ を考える．$\boldsymbol{x}_t \in \mathbb{R}^m$ は時刻 t に観測される入力である．このうち，l 個のデータにはラベル $y_t \in \{\pm 1\}$ が付いているとする．このとき，汎化性能に優れた判別器 $h^\star$ を学習したい．判別器 h は，判別関数 f を用いて $h = \mathrm{sgn}\,(f)$ と表せる．

3 関連研究

ラベルなしデータを学習に利用するため，「データはある低次元多様体上に分布し，ラベルはその多様体上で滑らかに変化する」と仮定する．MR では，この仮定を満たす判別関数 $f^\star$ を求めるため，従来の正則化損失に第 2 の正則化項を加えた損失 $J(f)$ を最小化する．

$$J(f) = \frac{1}{l}\sum_{t=1}^{T}\delta(y_t)c(f(\boldsymbol{x}_t), y_t) + \frac{\lambda_A}{2}\|f\|_{\mathcal{H}}^2 + \frac{\lambda_I}{2T}\sum_{s,t=1}^{T}(f(\boldsymbol{x}_s) - f(\boldsymbol{x}_t))^2 w_{st} \tag{1}$$

ここで，$\delta(y_t)$ は入力 $\boldsymbol{x}_t$ にラベルが付いていれば 1，そうでなければ 0 を返す指示関数，c は任意の凸損失関数，$\|\cdot\|_{\mathcal{H}}$ は再生核ヒルベルト空間（RKHS）$\mathcal{H}$ におけるノルム，$\lambda_A, \lambda_I \geq 0$ は正則化の強さを調節する正則化パラメータ，$w_{st} \geq 0$ は入力 $\boldsymbol{x}_s, \boldsymbol{x}_t$ 間の類似度である．

続いて，時刻 t における損失 $J_t(f)$ を $J(f) = \frac{1}{T}\sum_{t=1}^{T} J_t(f)$ を満たすように定義する．

$$J_t(f) = \frac{T}{l}\delta(y_t)c(f(\boldsymbol{x}_t), y_t) + \frac{\lambda_A}{2}\|f\|_{\mathcal{H}}^2 + \lambda_I\sum_{i=1}^{t-1}(f(\boldsymbol{x}_i) - f(\boldsymbol{x}_t))^2 w_{it} \tag{2}$$

Online MR では，時刻 t にデータを観測する度に，SGD を用いて損失 $J_t(f)$ が減少する方向へ判別関数 f を更新する．

$$f_{t+1} = f_t - \eta_t \left.\frac{\partial J_t(f)}{\partial f}\right|_{f=f_t} \tag{3}$$

ここで，$\eta_t \in (0, \frac{1}{\lambda_A})$ は学習率といい，勾配降下の歩幅を表している．表現定理 [1] より，判別関数 f_t は特徴ベクトル $K(\boldsymbol{x}, \cdot) \in \mathcal{H}$ の線形和で表現できる．

$$f_t = \sum_{i=1}^{t-1}\alpha_i^{(t)}K(\boldsymbol{x}_i, \cdot) \tag{4}$$

ここで，K は任意の正定値カーネル関数である．したがって，判別関数 f に関する更新式 (3) は，係数ベクトル $\boldsymbol{\alpha}$ に関する更新式 (5) に変形することができる．

$$\alpha_i^{(t+1)} = \begin{cases} (1-\eta_t\lambda_A)\alpha_i^{(t)} \\ \quad -2\eta_t\lambda_I(f_t(\boldsymbol{x}_i) - f_t(\boldsymbol{x}_t))w_{it} & i < t \\ 2\eta_t\lambda_I\sum_{i=1}^{t-1}(f_t(\boldsymbol{x}_i) - f_t(\boldsymbol{x}_t))w_{it} \\ \quad -\eta_t\frac{T}{l}\delta(y_t)c'(f_t(\boldsymbol{x}_t), y_t) & i = t \end{cases} \tag{5}$$

ここで，c' は凸損失関数 c の勾配である．

Online MR は，学習が進むにつれて基底の数が線形に増加してゆき，学習，予測ともに計算が困難になっていく．この現象を回避するため，バッファサイズ τ を定め，判別関数 f_t をなす基底の数を制限する．これに伴い，損失 $J_t(f)$ と判別関数 f_t を次のように近似する．

$$J_t(f) = \frac{T}{l}\delta(y_t)c(f(\boldsymbol{x}_t), y_t) + \frac{\lambda_A}{2}\|f\|_{\mathcal{H}}^2 + \lambda_I\frac{t}{\tau}\sum_{i=t-\tau}^{t-1}(f(\boldsymbol{x}_i) - f(\boldsymbol{x}_t))^2 w_{it} \tag{6}$$

$$f_t = \sum_{i=t-\tau}^{t-1}\alpha_i^{(t)}K(\boldsymbol{x}_i, \cdot) \tag{7}$$

* 東北大学大学院情報科学研究科, Graduate School of Information Sciences, Tohoku University
† 東北大学大学院情報科学研究科, Graduate School of Information Sciences, Tohoku University

この方法では，まず，SGD の更新式 (3) を用いて判別関数 f_t を $\tau+1$ 個の基底からなる中間関数 f' へ更新する．

$$f' = \sum_{i=t-\tau}^{t} \alpha'_i K(\boldsymbol{x}_i, \cdot) \tag{8}$$

次に，バッファにある最も古いデータ $\boldsymbol{x}_{t-\tau}$（あるいは最も古いラベルなしデータ）を破棄，新しいデータ $\boldsymbol{x}_t$ をバッファに追加した後，中間関数 f' を判別関数 f_{t+1} で近似する．

$$\boldsymbol{\alpha}^{(t+1)} = \mathop{\arg\min}_{\boldsymbol{\alpha}^{(t+1)} \in \mathbb{R}^\tau} \|\boldsymbol{f}' - \boldsymbol{f}_{t+1}\|^2$$

$$\text{s.t.} \quad f_{t+1} = \sum_{i=t-\tau+1}^{t} \alpha_i^{(t+1)} K(\boldsymbol{x}_i, \cdot) \tag{9}$$

ここで，$\boldsymbol{f} = (f(\boldsymbol{x}_{t-\tau+1}), \cdots, f(\boldsymbol{x}_t))^\top$ である．式 (9) を解く際は，バッチ学習アルゴリズムの 1 つであるカーネルマッチング追跡（KMP）[4] を用いる．この方法は，Online MR (Buffering) と呼ばれる．

4 提案手法

我々は，既存の Online MR の収束速度を向上させるため，モーメンタム法を用いた Online MR を提案する．はじめに，時刻 t における更新量 Δ_t を定義する．

$$\Delta_t = f_t - f_{t-1} = \sum_{i=1}^{t-1} \beta_i^{(t)} K(\boldsymbol{x}_i, \cdot) \tag{10}$$

提案手法では，SGD の更新式 (3) の代わりに以下の更新式を用いる．

$$f_{t+1} = f_t + \mu\Delta_t - (1-\mu)\eta_t \left.\frac{\partial J_t(f)}{\partial f}\right|_{f=f_t} \tag{11}$$

これは，勾配に対して更新量の方向に慣性をつけたものと捉えることができる．ここで，$\mu \in [0, 1)$ はモーメンタム係数といい，過去の更新量 Δ_t をどれだけ重視するかを表している．$\mu = 0$ のとき，更新式 (11) は SGD の更新式 (3) と一致する．判別関数 f に関する更新式 (11) を変形すれば，係数ベクトル $\boldsymbol{\alpha}$ に関する更新式が得られる．最後に，係数ベクトル $\boldsymbol{\beta}$ を求め，次回の更新に利用する．

$$\beta_i^{(t+1)} = \begin{cases} \alpha_i^{(t+1)} - \alpha_i^{(t)} & i < t \\ \alpha_t^{(t+1)} & i = t \end{cases} \tag{12}$$

バッファサイズ τ を定める場合も，既存の Online MR (Buffering) のときと同様に，更新式を導出できる．

5 評価実験

提案手法の収束速度を評価するため，CBCL at MIT が提供する「顔」と「非顔」の 2 クラスからなるデータセット Face を対象に，既存手法との比較実験を行った．データを予め標準化した後，以下の試行を 10 回繰り返し，その汎化誤差の平均を算出した．公平性のため，ラベル付きデータの選択やシャッフルに用いる乱数のシードは，既存手法と提案手法で同じものを使用する．

(1) 訓練データのうち 2% をラベル付きデータとみなす
(2) 訓練データを無作為にシャッフルする
(3) 学習を行いながら一定の周期で汎化誤差を測定する

既存手法，提案手法ともに類似度 $w_{st} = \exp(-\gamma\|\boldsymbol{x}_s - \boldsymbol{x}_t\|^2)$，ヒンジ損失 $c(f(\boldsymbol{x}), y) = \max(1 - yf(\boldsymbol{x}), 0)$，ガウシアンカーネル $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\gamma_K\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2)$ を用いた．また，バッファサイズを $\tau = 100$ に固定し，学習率を $\eta_t = \frac{0.1}{\sqrt{t}}$ とした．提案手法のモーメンタム係数は，$t \leq \tau$ ならば $\mu = 0.5$，そうでなければ $\mu = 0.9$ とした．最初に $\mu = 0.5$ と設定するのは，初期値の周辺は勾配が大きく変化する可能性があるからである．残りのハイパパラメータ $\gamma, \gamma_K, \lambda_A, \lambda_I$ はデータセットに応じて選択した．

実験の結果，提案手法は，既存手法を上回る性能を示した（図 1）．

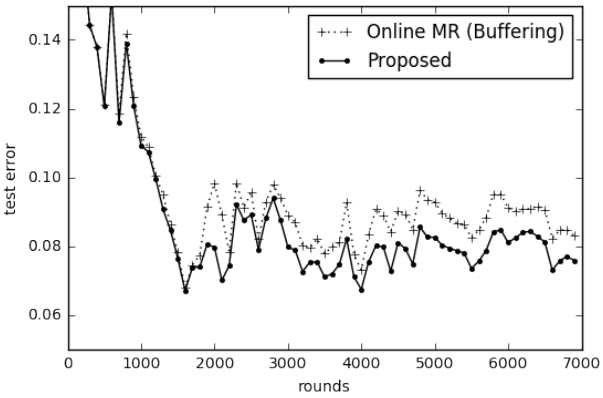

図 1: 実験結果

6 結論

本研究では，Online MR の収束速度を改善するため，モーメンタム法を用いた手法を提案した．また，既存手法との比較実験を行い，性能が向上することを確認した．今後の課題として，提案手法の理論解析が挙げられる．

謝辞

本研究は，革新的研究開発推進プログラム ImPACT から研究費を受けている．

参考文献

[1] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *JMLR 7*, 2006.
[2] A. B. Goldberg, M. Li, and X. Zhu. Online manifold regularization: A new learning setting and empirical study. In *ECML/PKDD*, 2008.
[3] J. Kivinen, A. J. Smola, and R. C. Williamson. Online learning with kernels. *IEEE Trans. Signal Processing 52(8)*, 2004.
[4] P. Vincent and Y. Bengio. Kernel matching pursuit. *Machine Learning 48(1-3)*, 2002.