ブール変数実数多項式による嗜好関数の学習

4U－1

中村 篤祥　馬見塚 拓　鳥羽 弘康　安倍 直樹
NEC C&C 研究所

1. はじめに

ニュース記事などに対する個人の嗜好を、記事中の出現単語から予測する方法において、各単語に１つのブール変数を割り当て、その実数係数多項式で嗜好関数を表現する方法を提案し、その係数の学習アルゴリズムについて考察及び実験を行う。特に、計算論的学習理論において盛んに研究されているオンライン学習における重みの逐次更新法を、実数係数の学習に適用する。具体的には、Kivinen & Warmuth [1] が提案・解析した加法的更新法 GD と乗法的更新法 $\mathrm{EG}^{\pm}$ に加え、新しく「誤差比例修正法」と呼ぶ重み更新アルゴリズムを提案し、その乗法的更新法である DPMU について、ある被験者の実データを用いて実験的に予測性能を評価・比較する。実験結果によれば DPMU は、既存の方式と同等以上の予測性能を有する。

2. アルゴリズム GD と $\mathrm{EG}^{\pm}$

オンライン学習モデルにおいて学習者は、事例 $\mathbf{x}$ に関係する値 y を予測し、その後実際の y の値をもらうということを繰り返す。ここでは、N 次元ブールベクトル $\mathbf{x}$ から実数値 y を予測する学習問題を扱うとし、予測値 $\hat{y}$ に対する損失を平方誤差 $(y-\hat{y})^2$ で測ることにする。l 回の試行列 $S_l=((\mathbf{x}_1,y_1),\dots,(\mathbf{x}_l,y_l))$ に対する累積損失でオンライン学習アルゴリズムを評価する。

ここでは、y の値を重みベクトル $\mathbf{w}\in\Re^N$ を使った線形関数 $\hat{y}=\mathbf{w}\cdot\mathbf{x}$ で予測する方法を考える。重みの更新を行わないで、常に同じ重み $\mathbf{w}$ で S_l を予測した場合の累積損失を $\mathrm{Loss}(\mathbf{w},S_l)$ とする。事例 $\mathbf{x}=(x_1,\dots,x_N)$ を貰い、重み $\mathbf{w}$ を使って $\hat{y}=\mathbf{w}\cdot\mathbf{x}$ と予測し、実際の値が y であったとき、重みを $\mathbf{w}'$ に更新するアルゴリズムとして、加法的更新法である GD と乗法的更新法である $\mathrm{EG}^{\pm}$ は次のように定義される [1]。但し、$||\mathbf{x}||_p$ は $\mathbf{x}$ の L_p $(p\geq 1)$ ノルムである。

1. 最急降下法 $\mathrm{GD}(s,\eta)$

$\mathbf{s}\in\Re^N$: 重みの初期値,$\eta(\mathbf{x})\in\Re$: 学習レート関数

$$\mathbf{w}'=\mathbf{w}+2\eta(\mathbf{x})(y-\hat{y})\mathbf{x}$$

2. 指数的最急降下法 $\mathrm{EG}^{\pm}(s^{\pm},\eta)$

$s^{\pm}\in(0,\infty)^{2N}$: 重みの初期値,
$\eta(\mathbf{x})\in\Re$: 学習レート関数
$\mathbf{w}=(w_1^+-w_1^-,\dots,w_N^+-w_N^-)$

$$w_i'^+=||s^{\pm}||_1 v_i^+/||\mathbf{v}^{\pm}||_1\ ,\ w_i'^-=||s^{\pm}||_1 v_i^-/||\mathbf{v}^{\pm}||_1$$

但し $\mathbf{v}^{\pm}=(v_1^+,\dots,v_N^+,v_1^-,\dots,v_N^-)$ は

$$v_i^+=w_i^+e^{2\eta(\mathbf{x})(y-\hat{y})Ux_i}\ ,\ v_i^-=w_i^-e^{-2\eta(\mathbf{x})(y-\hat{y})Ux_i}$$

とする。

3. ブール変数実数多項式

$\{0,1\}^N$ から $\Re$ への任意の関数は、次のように定義されるブール変数実数多項式 f で一意に表現される。

$$f(x_1,\dots,x_N)=\sum_{S\subseteq\{1,\dots,N\}}a_S\prod_{i\in S}x_i$$

但し全ての係数 a_S は実数とする。また、高々 k 個の変数の積からなる項のみからなるブール変数実数多項式を k- ブール変数実数多項式という。k- ブール変数実数多項式は、$\sum_{i=0}^{k}\binom{N}{i}$ 次元 $\{0,1\}$ 値ベクトルに関係する値を、$\sum_{i=0}^{k}\binom{N}{i}$ 次元実数値ベクトルの重みを使って予測する線形関数とみなせる。したがって、この関数のオンライン学習に前節の GD、$\mathrm{EG}^{\pm}$ アルゴリズムが使える。

4. 誤差比例修正法

予測と実際の値の差の c 倍 $(0<c<1)$ だけ近づくように重みベクトルを更新する方法を、誤差比例修正法と名付ける。つまり、実際の値を y、予測値を $\hat{y}=\mathbf{w}\cdot\mathbf{x}$、修正後の予測値を $\hat{y}'=\mathbf{w}'\cdot\mathbf{x}$ とすれば、

$$\hat{y}'-\hat{y}=c(y-\hat{y}) \qquad (1)$$

を満たすように $\mathbf{w}$ を $\mathbf{w}'$ に更新する方法である。この節では誤差比例修正法の加法的更新法 DPAU と乗法的更新法 DPMU について考える。

1. $\mathrm{DPAU}(\mathrm{s},c)$

$\mathbf{s}\in\Re^N$: 重みの初期値,$c\in\Re$: 学習レート係数

$\mathbf{w}'=\mathbf{w}+a\mathbf{x}$ というように重みを更新するとする。但し a は定数である。式 (1) を満たすように、a の値を決める。両辺に $\mathbf{x}$ を掛けると $\hat{y}'=\hat{y}+a||\mathbf{x}||_2^2$ となる。これを式 (1) に代入して整理すると、
$a=c(y-\hat{y})/||\mathbf{x}||_2^2$ となる。従って、下式が導かれる。

$$\mathbf{w}'=\mathbf{w}+\frac{c}{||\mathbf{x}||_2^2}(y-\hat{y})\mathbf{x}$$

この式は GD の $\eta(\mathbf{x})=c/(2||\mathbf{x}||_2^2)$ の場合に等しい。

2. $\mathrm{DPMU}(\mathrm{s}^{\pm},c)$

$\mathbf{s}^{\pm}\in\Re^{2N}$: 重みの初期値,$c\in\Re$: 学習レート係数

重みを $\mathbf{w}^+=(w_1^+,\dots,w_N^+),\mathbf{w}^-=(w_1^-,\dots,w_N^-)$ とし、予測値を $\hat{y}=\hat{y}^+-\hat{y}^-$ とする。但し $\hat{y}^+=\mathbf{w}^+\cdot\mathbf{x}$, $\hat{y}^-=\mathbf{w}^-\cdot\mathbf{x}$ である。$\mathbf{w}^+,\mathbf{w}^-,\hat{y},\hat{y}^+,\hat{y}^-$ に対応する修正後の値を $\mathbf{w}'^+,\mathbf{w}'^-,\hat{y}',\hat{y}'^+,\hat{y}'^-$ とする。また $\mathbf{x}=(x_1,\dots,x_N)\in\{0,1\}^N$ とする。

式 $w_i'^+=w_i^+\beta^{x_i}$, $w_i'^-=w_i^-/\beta^{x_i}$ で重みを更新する。式 (1) を満たすように β を決める。$\mathbf{x}\in\{0,1\}^N$ なので、$\hat{y}'^+-\hat{y}'^-=\mathbf{w}'^+\cdot\mathbf{x}-\mathbf{w}'^-\cdot\mathbf{x}=\sum_{x_i=1}\beta w_i^+x_i-\sum_{x_i=1}\frac{1}{\beta}w_i^-x_i=\beta\hat{y}^+-\frac{1}{\beta}\hat{y}^-$ が成り立つ。従って、$\hat{y}'=\beta\hat{y}^+-\frac{1}{\beta}\hat{y}^-$ を式 (1) に代入して整理すると β が求まる。

$$\beta=\frac{\hat{y}+c(y-\hat{y})+\sqrt{(\hat{y}+c(y-\hat{y}))^2+4\hat{y}^+\hat{y}^-}}{2\hat{y}^+}$$

Learning personal preference functions using boolean-variable real-valued multivariate polynomials.
Atsuyoshi Nakamura, Hiroshi Mamitsuka,
Hiroyasu Toba, Naoki Abe.
C&C Research Laboratories NEC Corp.

5. 嗜好関数学習への応用 (実データによる実験)

1. 実験データの採集及び実験方法

ネットニュースの Newsgroup: comp.infosystems.www.announce にある期間に投稿された記事に対し、ある被験者が 5 段階 (-2[興味小],-1,0,1,2[興味大]) に点数を付けたデータを使用した。各ニュースは単語に分解された後、綴りのみから名詞の可能性があるもののみを抽出し、複数形は原形にした。また辞書[1]に頻出マークの付いている単語 (397 単語) は除いた。このように、各ニュースをキーワードの集合で表現し、それに対するユーザの嗜好度 (実数値) を与える関数を 1- ブール変数実数多項式と仮定して、実数係数を学習しながら予測した。但し、各ブール変数は各キーワードに対応させ、そのキーワードが含まれている時に 1、そうでない時には 0 が割り当てられる。学習データ S の性質は以下の通りである。

[キーワード数]

全体	平均	最大	最小
13041	45.4	310	14

[点数分布]

-2	-1	0	1	2	計
527	221	179	89	47	1063

[$\mathbf{u}_{opt} = \arg\min_{\mathbf{u}} \mathrm{Loss}(\mathbf{u}, S)$ の性質] [2]

- $\mathrm{Loss}(\mathbf{u}_{opt}, S) = 21.0333$
- $\|\mathbf{u}_{opt}\|_1 = 1043.16$
- $\|\mathbf{u}_{opt}\|_2^2 = 633.282$

2. 累積損失の上界と実験値

GD,EG$^\pm$,DPMU の 3 つのアルゴリズムを累積損失で比較する。上界は Kivinen & Warmuth の定理 [1] による。各アルゴリズムの重みの初期値は、GD が $s = (0, ..., 0)$ 、EG$^\pm$ が $s^\pm = \|\mathbf{u}_{opt}\|_1(\frac{1}{2N}, \cdots, \frac{1}{2N})$ 、DPMU が $s^\pm = (1, ..., 1)$ とした。[1] によれば、GD と EG$^\pm$ の学習レート $\eta(\mathbf{x})$ は、$\mathrm{Loss}(\mathbf{u}_{opt}, S)$、$\mathbf{u}_{opt}$ と $\mathbf{s}$ ($\mathbf{u}_{opt}$ と $\mathbf{s}^\pm$) との距離が既知であれば精密なチューニングができる。未知の場合のチューニング (下表上段) と既知の場合のチューニング (下表下段) の両方で比較する。

Alg.	$\eta(\mathbf{x})$	累積損失	
		上界	実験値
GD	$0.25/\|\mathbf{x}\|_2^2$	3.94×10^5	1388.61
	$0.495/\|\mathbf{x}\|_2^2$	2.01×10^5	1590.52
EG$^\pm$	3.06×10^{-7}	7.93×10^6	2140.12
	4.59×10^{-7}	5.31×10^6	2002.78
DPMU	0.5	?	1384.30

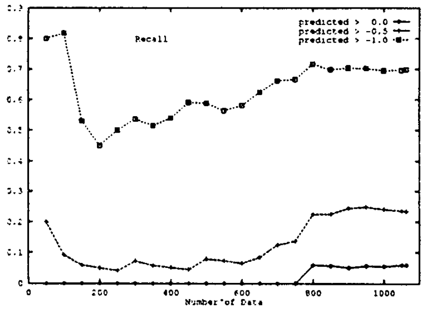
3. 実験的最適学習レートでの比較

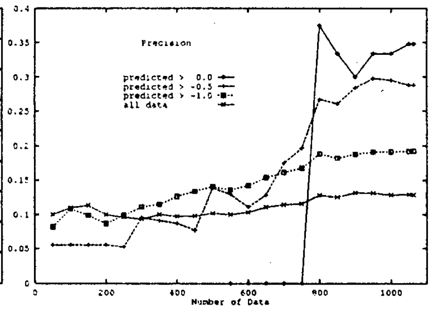
最適な学習レートを実験的に求め、そのときの累積損失で比較する。η または c の関数としての累積損失が、最小値をもつ下に凸な関数と仮定して 2 分割サーチにより最小値の近似値を実験的に求めた。具体的には、$\Delta = 1.0 \times 10^{-6}$ だけ増加させたときの累積損失の増減によりその η または c に対する関数の傾きの正負を求め、最小値を含む区間を半分に絞る操作を繰り返した。各アルゴリズムの重みの初期値は上の実験と同じである。

Alg.	$\eta(\mathbf{x})$ or c	累積損失
GD	$0.151/\|\mathbf{x}\|_2^2$	1363.43
EG$^\pm$	9.47×10^{-6}	1385.79
DPMU	0.313	1362.62

4. 実用性に関する実験

被験者の興味のある記事 (1 または 2 を付けた記事 [12.8%]) に高い予測点を付けるかどうか実験する。具体的には、予測方式によって選ばれた記事の集合を S_s 、被験者が興味のある記事の集合を S_i としたとき、再現率 $|S_s \cap S_i|/|S_i|$ と適合率 $|S_s \cap S_i|/|S_s|$ を計算する。予測値が 0.0 より大きい記事,-0.5 より大きい記事,-1.0 より大きい記事の 3 つの場合ににについて実験した結果を下に示す。アルゴリズムは DPMU$((1, ..., 1), 0.313)$ を使った。(左: 再現率, 右: 適合率)





最初の 963 の記事に対し最小二乗法 LS で累積損失を最小にする重みを求め、その重みで最後の 100 の記事を予測した場合と、DPMU の最後の 100 の記事の予測と比較すると下の表のようになる。100 の記事全部を選ぶと適合率は 0.1 である。オンライン学習方式 DPMU がバッチ型の学習方式である LS の性能を上回っている。

アルゴリズム	累積損失	誤差 $< a$ の率		
		$a = 0.5$	$a = 1.0$	$a = 1.5$
DPMU	113.173	0.35	0.7	0.86
LS	221.631	0.26	0.48	0.69

予測値 $> b$ のデータの (再現率, 適合率)		
$b = 0.0$	$b = -0.5$	$b = -1.0$
(0.1,0.500)	(0.1,0.143)	(0.7,0.200)
(0.3,0.125)	(0.8,0.195)	(0.8,0.148)

6. 結論

今回の実験データに関しては、理論値、実験値共に、乗法的更新法 EG$^\pm$ より加法的更新法 GD の方が累積平方誤差が小さい。但し、学習レートを選ぶことにより実験値の差はほとんどなくなる。また、乗法的更新法でも修正量を誤差に比例する量に押さえる DPMU では、実験で GD と累積平方誤差が変わらなかった。また、DPMU では、予測値がある点数より大きい記事集合の実際に評価値が高い記事集合に対する再現率、適合率においても学習効果がみられた。

[1] Computer-Usable Dictionary File based on the Oxford Advanced Learner's Dictionary of Current English (June 1992 version)

[2] 但し $\mathbf{u}_{opt}$ は最小二乗法により求めた近似値である。

参考文献

[1] J. Kivinen, M. Warmuth . "Exponentiated Gradient Versus Gradient Descent for Linear Predictors." *UCSC-CRL-94-16*, June 21, 1994 .