評価指標をマージンに反映したオンラインランキング学習

数原 良彦† 鈴木 潤‡ 安田 宜仁† 小池 義昌† 片岡 良治†
†日本電信電話株式会社 NTT サイバーソリューション研究所
‡日本電信電話株式会社 NTT コミュニケーション科学基礎研究所
††{suhara.yoshihiko, suzuki.jun, yasuda.n, koike.y, kataoka.ryoji}@lab.ntt.co.jp

1 はじめに

情報検索においては，ユーザの入力に対して適切な検索結果を提示するために，様々なランキング手法が研究されてきた．一般的な検索システムでは，単一のランキング手法ではなく，複数の手法によるスコア (ランキング素性) をランキング関数へ入力し，ランキング関数の出力によって最終的なランキングを決定する [1]．多数のランキング素性が与えられた場合，人手によるランキング関数の設定は困難であるため，機械学習を用いてランキング関数を生成するランキング学習 (learning to rank) が盛んに研究されている [2][3][4][5]．

ランキング学習では，用意したクエリ群についてあらかじめ取得した検索結果それぞれに対して，人手によって多段階の適合性評価を付与することで訓練データを作成する．そして，この訓練データを用いて教師あり学習の枠組みで最適なランキング関数の生成を目指す．

従来のランキング学習では，バッチ学習に基づく手法 [2] が数多く提案されてきたが，利用可能なデータの大規模化，ウェブ文書の激しい日々の変化に対応するために短期間でランキング関数を生成する必要性など，最近では高速に学習可能なオンライン学習に基づくランキング学習手法が注目されている [5]．たとえば，検索システムに日々蓄積されるログを用いて逐次的にランキング学習を実現できれば，従来バッチ学習では達成が困難であった即時性を持ったランキングをユーザに提供することが可能となる．

ランキング学習における既存のアプローチとしては，あるクエリに対する検索結果集合のうち，適合性評価が異なる組み合わせを順序ペアに変換し，その順序ペアの誤りに基づく損失関数を最適化するペアワイズ手法と，あるクエリに含まれる文書全ての順序に対する誤りに基づく損失関数を最適化するリストワイズ手法と呼ばれるアプローチが存在する．

ペアワイズ手法は，同一クエリに含まれる適合性評価が異なる文書ペアに対する順序誤差を最適化するため，検索結果指標が必ずしも最適化されるわけではない．リストワイズ手法では，クエリ内の文書群全てを考慮した最適化が可能であるため，検索評価指標を近似的に最適化することが可能であるが，クエリ全体を眺める必要があり，コストが大きいという問題がある．一般的にリストワイズ手法がペアワイズ手法に比べて高い検索精度を示すとされている [4]．

Sculley [5] は，ランダムサンプリングを行ったペアに対してオンライン学習を行うことにより，高速に学習を行うペアワイズ手法の枠組みを提案している．しかしながら，この方法はランダムサンプリングに基づいたペアワイズ手法であるため，リストワイズ手法のように検索評価指標を最適化しているわけではない．

そこで我々は，大規模データに対しても高速に学習が可能であり，そして検索評価指標を考慮した損失関数の最適化を行うことにより，従来のオンライン学習手法を上回る検索精度を実現するオンラインランキング学習手法の開発を目指す．具体的には，検索評価指標をマージンに取り入れ，オンラインマージン最大化学習である Passive-Aggressive (PA) [6] を用いてランキング学習を行う手法 PARank を提案する．そして，既存研究で提案されたアイディアを導入することで提案手法の更なる検索精度向上を目指し，評価実験を通じてその有効性と効果を検証する．

2 ランキング学習

ランキング学習では，人手による評価データを用いて，教師あり学習の枠組みでランキング関数を生成する．人手による適合性評価は，検索結果に含まれる文書が入力されたクエリにどれだけ適合しているかという観点で付与される．このため，たとえ同一文書であってもクエリによって適合性評価が異なる．また，ランキング素性の中には TF-IDF や BM25 スコア [1] のようにクエリ依存のものが存在するため，ある文書の特徴表現は，検索されたクエリによって異なることがある．

ランキング学習に用いる訓練データ D は，人手によって適合性評価が付与されたデータ $(\mathbf{x}, y, q)$ から構成される．ここで，q は入力クエリ，$\mathbf{x} \in \mathbb{R}^m$ はクエリ q に対する検索結果文書の特徴ベクトル，$y \in \mathbb{N}$ はクエリ q に対する文書の適合性評価を表している．

ペアワイズ手法では，適合度の異なるペアを学習に利用する．同一クエリ q に含まれる適合性評価が異なる検索結果文書 a と文書 b について，$\mathbf{x} = \mathbf{x}_a - \mathbf{x}_b$, $y = \mathrm{sign}(y_a - y_b)$ とおくことで，$y \in \{-1, +1\}$ となり，通常の二値分類問題として解くことが可能となる．

検索において，検索結果下位の誤りに比べて上位の誤りがより少ない方が良いランキングとされる．多段階の適合性評価が付与されている場合には，より高い評価の文書がより上位にランキングされる方が好ましい．しかしながらペアワイズ手法では，たとえば $(y_a, y_b) = (5, 1)$ と $(y_a, y_b) = (2, 1)$ というペアに対し

て等しく損失を与える．我々は，特に情報検索に一般的に用いられる多値の適合度を考慮する評価指標においては，適合度スコアの差が大きい文書ペアに対して，差が小さい文書ペアよりも損失を大きく見積もることによって，更なる検索精度向上が可能であると考えた．そこで，ペアワイズ手法において，適合性スコアの異なる組み合わせに対して，異なる損失を設定することで検索精度向上を目指す．

2.1 PARank

我々は，検索評価指標に基づいて，それぞれの順序ペアに対して異なるマージンを設定し，PA を用いてマージン最大化学習を行うオンラインランキング学習アルゴリズム PARank (Passive-Aggressive Rank) を提案する．

本稿では評価指標としては，情報検索分野で一般的に用いられる Normalized Discounted Cumulative Gain (NDCG) [7] を用いる．NDCG は多値の適合性評価に対して用いられ，適合性評価スコアを 2 の指数とした値を，順位の値の対数で割ることによって，検索結果上位の評価結果を重視するよう設計された評価指標である．

クエリ q における i 番目の文書の適合度を $y_{q,i}$ とすると，クエリ q に対する検索結果上位 k 件に対する NDCG の値は，

$$\mathrm{DCG}_q@k \quad \sum_{i=1}^{k} \frac{2^{y_{q,i}}-1}{\log(1+i)} \tag{1}$$

$$\mathrm{NDCG}_q@k \quad \frac{\mathrm{DCG}_q@k}{\mathrm{maxDCG}_q@k} \tag{2}$$

によって計算できる [7]．ここで $\mathrm{maxDCG}_q@k$ は，クエリ q において適合度が高い順番に文書を並べた理想的なランキングに対する DCG@k の値を表す．このように正規化されているため，$\mathrm{NDCG} \in (0, 1]$ となる．

訓練データに含まれる適合性評価はクエリによって分布が異なるといわれている [8]．そこで我々は，たとえ同じ適合性評価の組み合わせでも，クエリによって異なるマージンサイズを設定するのが適切であると考え，提案手法では各クエリ，各組み合わせに対して異なるマージンサイズを設定する．

クエリ q における適合性スコア r_1 と r_2 (ただし，$r_1 > r_2$ とする) のマージンサイズ $E_q(r_1, r_2)$ は，クエリ q における理想的なランキングリストから，適合度スコア r_1 と適合度スコア r_2 の文書を交換した際の NDCG の減少値 ΔNDCG に基づいて計算する．

たとえば，クエリ q に対して付与された適合度スコア (4, 3, 2, 1) の文書が (3 件, 3 件, 2 件, 3 件) 存在する例を考える．この例において，適合度スコア 4 とスコア 3 の文書を交換する場合には，3　3 = 9 通りの組み合わせが存在する．我々の方法では，最も NDCG 値の減少値が大きい組み合わせ，すなわち適合度スコア 4 の最上位の文書と，適合度スコア 3 の最下位の文書を交換した際の NDCG の減少値を用いる．この値を $\Delta\mathrm{NDCG}_q(4, 3)$ とする．この場合，文書を交換した後の NDCG の値は 0.880 となるため，$\Delta\mathrm{NDCG}_q(4, 3) = 1.0 - 0.880 = 0.120$ と算出する．こ

Algorithm 1 PARank

Input: D, T, C
Output: $\mathbf{w}^*$

create E_q for all queries q
$\mathbf{w}_0 \leftarrow \mathbf{0}$
for $i = 0$ to T **do**
　for all queries q in Q **do**
　　$(\mathbf{x}_a, \mathbf{x}_b, y_a, y_b) = \underset{(\mathbf{x}_a, \mathbf{x}_b, y_a, y_b) \in \mathbf{Z}}{\mathrm{argmax}} \ell_t(\mathbf{w}_t; \mathbf{x}_a, \mathbf{x}_b, y_a, y_b, q)$
　　$\mathbf{x}_t = \mathbf{x}_a - \mathbf{x}_b$
　　$\tau_t = \min\left\{C, \frac{\ell_t}{\|\mathbf{x}_t\|^2}\right\}$
　　$\mathbf{w}_{t+1} = \mathbf{w}_t + \tau_t \mathbf{x}_t$
　end for
end for
$\mathbf{w}^* = \frac{1}{T|Q|} \sum_{t=1}^{T|Q|} \mathbf{w}_t$
return $\mathbf{w}^*$

のように ΔNDCG は，

$$\Delta\mathrm{NDCG}_q(r_1, r_2) = 1.0 - \mathrm{NDCG}_q(r_1, r_2) \tag{3}$$

に基づいて計算する．ここでスケールパラメータ E_{scale} を，最小のマージンが 1.0 になるように設定し，

$$E_q(r_1, r_2) = E_{\mathrm{scale}} \Delta\mathrm{NDCG}_q(r_1, r_2) \tag{4}$$

とスケール調整を行う．

次に PA について説明を行う．PA では，重みベクトル $\mathbf{w}_t$ の更新を以下の最適化問題として定式化する:

$$\mathbf{w}_{t+1} = \underset{\mathbf{w} \in \mathbb{R}^n}{\mathrm{argmin}} \frac{1}{2}\|\mathbf{w} - \mathbf{w}_t\|^2 \quad \text{s.t.} \quad \ell_t(\mathbf{w}_t; \mathbf{x}_t, y_t) = 0. \tag{5}$$

$\ell_t = 0$ ならば何もせず，$\ell_t > 0$ ならば，更新の大きさが最小になるように，$\mathbf{w}$ を更新する．式 (5) の最適化問題は，ラグランジュの未定乗数法を用いて解くことにより，閉じた解で $\mathbf{w}_{t+1}$ を求めることができる．誤りを許容するためにスラック変数　を導入した場合も，同様に閉じた解で重みベクトルの更新が可能である．正則化項を C　と設定した手法は PA-I と呼ばれる [6]．本稿では，PA 手法として PA-I を用いる．

提案手法の処理の流れを Algorithm 1 に示す．訓練データとイテレーション回数，パラメータ C を入力とする．まず，訓練データに含まれる各クエリについて E_q を計算する．次に各クエリについて重み更新を行う．クエリ q における重み更新には，PA の max-loss update [6] を用いる．これは，クエリ毎に最大の損失を与えるペアを選択し，重みベクトルの更新を行う戦略である．ここで $\mathbf{Z} = \{\mathbf{x}_a, \mathbf{x}_b, y_a, y_b | \mathbf{x}_a, \mathbf{x}_b \in \mathbf{X}, y_a, y_b \in \mathbf{y}, y_a > y_b\}$ である．以上の処理を全てのクエリに対して行い，全クエリに対する処理を入力されたイテレーション回数 T だけ処理を繰り返す．最後に，更新した重みベクトル $\mathbf{w}_t$ 全ての平均を計算する．ここで $|Q|$ は訓練データに含まれるクエリ数を表している．

$\mathbf{x}_t = \mathbf{x}_a - \mathbf{x}_b$ とすると，重みベクトル $\mathbf{w}_t$ の，クエリ q に含まれる適合度 y_a の文書 a と，適合度 y_b の文書 b に対する損失関数を，以下の hinge loss 関数で表現する:

$$\ell_t(\mathbf{w}_t; \mathbf{x}_a, \mathbf{x}_b, y_a, y_b, q) = \begin{cases} 0 & \mathbf{w}_t \quad \mathbf{x}_t \geq E_q(y_a, y_b) \\ E_q(y_a, y_b) - \mathbf{w}_t \quad \mathbf{x}_t & \text{otherwise.} \end{cases}$$

2.2 Ramp loss と loss penalty の導入

従来提案されたアイディアを導入することによって，PARank の更なる精度向上を目指す．ランキング学習においては，訓練データにノイズが含まれるという問題が知られているため [9]，ノイズに頑健な損失関数である ramp loss を PARank に導入する．また，我々が提案した検索評価指標に基づくマージンの設定とは異なるアプローチで順序ペアに対する重みづけを行う方法である損失に対する重みづけ方法 (loss penalty) を導入する．

基本的な PA で誤差関数に利用されている hinge loss の代わりに Collobert ら [10] が提案した ramp loss の適用を検討する．Hinge loss はマージンの外側に存在するデータに対しては損失を与えず，マージンの内側に存在するデータに対してはマージンからの距離に比例した大きさの損失を与える．Ramp loss は，マージンの内側に存在し，一定以上離れたデータに対しては，等しく損失を与えるような損失関数である．Ramp loss は元々サポートベクタの数を減らし，SVM を高速に学習するために考案されたものであるが，ノイズを含むようなデータに対してロバストな学習が可能になることが知られている [10][11]．PA へ導入した研究もあり，PA-I に対して ramp loss を導入した際の更新式は以下になる [11]:

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \tau_t \mathbf{x}_t \ \text{, where}$$

$$\tau_t = \begin{cases} \min\left\{C, \frac{\ell_t}{\|\mathbf{x}_t\|^2}\right\} & \text{if } |\mathbf{w}_t \quad \mathbf{x}_t| \quad 1 \\ 0 & \text{otherwise.} \end{cases}$$

ΔNDCG を損失に対する重みづけの既存手法として，Cao ら [3] が提案した異なる順位ペアの組み合わせに対する影響の度合いを hinge loss の傾きに導入する方法がある．提案手法はマージンの大きさを変更しているため，Cao らの手法を容易に組み合わせることが可能である．この方法を組み合わせた場合も効果についても検証を行う．各試行で算出された損失に対して ΔNDCG に基づく重みである $E_q(y_a, y_b)$ を hinge loss や ramp loss の損失の重み (loss penalty) として利用する．このとき更新式は以下になる:

$$\mathbf{w}_{t+1} = \mathbf{w}_t + E_q(y_a, y_b)\tau_t \mathbf{x}_t. \qquad (6)$$

3 評価実験

本稿で提案した評価指標に基づくマージンサイズの設定と ramp loss, loss penalty 導入の有効性を確認するため，既存手法との比較実験を行った．

評価実験のデータセットにはランキング学習の研究で広く利用されている LETOR4.0[12] の MQ2007 データセットを用いた．

PARank については，各設定が検索精度にどのような効果を与えるのかを検証するため，ΔNDCG に基づくマージンサイズの設定方法，ramp loss, loss penalty の全ての組み合わせについて評価を行った．損失関数には，hinge loss (hinge) と ramp loss (ramp) の 2 通り，マージンの設定方法は，1.0 に固定する方法 (const.) と NDCG の減少値に基づくマージン設定方法 (ΔNDCG) の 2 通り，loss penalty を利用しない方法 (none) と NDCG の減少値に基づく損失の重みづけ (ΔNDCG) の 2 通り，以上の組み合わせ合計 8 通りの設定を用いた．

オンライン学習のベースライン手法としては，ランダムサンプリングによってペアを選択する Stochastic Pairwise Descent (SPD)[5] を選択し，バッチ学習のベースライン手法として，代表的なペアワイズ手法である RankingSVM (RSVM) [2] と，NDCG を直接最適化するリストワイズ手法である AdaRank-NDCG[4] を選択した．SPD の実装には so a-ml[1]を利用した．また RSVM，AdaRank-NDCG については LETOR4.0 で公開されている結果を用いた．

MQ2007 に含まれるデータセットを訓練データをクエリ毎に 5 分割し，3 つを訓練データ，1 つを検証データ，1 つをテストデータとする 5-fold cross validation を用いて評価を行った．SPD の学習手法は PA-I を選択した．提案手法，SPD (PA-I) の試行回数は 10,000 とし，C パラメータは検証データにおいて最大の MeanNDCG 値を示した値を選択した．評価指標には，NDCG@1-5 を用いた．

3.1 結果と考察

それぞれの評価結果を表 1 に示す．説明のため，PARank の各設定について番号を (a) から (h) の記号で表している．また表中のボールド体の数字は，PARank の各設定方法における最大値であることを表している．表 1 から以下の結果を確認した．

- PARank におけるマージンサイズの設定に NDCG の減少値を用いた方法の比較では，hinge loss においては全ての設定，ramp loss においては (e) と (g) は NDCG@2,3,4,5 において NDCG margin を用いた設定が高い値を示した．
- PARank における hinge loss と ramp loss の比較では，(d) と (h) における NDCG@1 の値を除き，全ての設定について ramp loss を用いた設定が高い NDCG@1-5 を示した．
- PARank における loss penalty の有無の比較では，hinge loss，ramp loss を適用した場合ともに，全ての設定において NDCG@1-5 が同じ程度か低い値を示した．
- PARank と RSVM の比較では，全ての設定において，RSVM の方が高い NDCG@1-5 の値を示した．
- PARank と SPD (PA-I) の比較では，hinge loss の場合には NDCG の減少値をマージンに取り入れた場合，ramp loss の場合は全ての設定について PARank が高い NDCG@1-5 の値を示した．
- AdaRank-NDCG との比較では (e),(f) は NDCG@1 において近い値を示しているものの，その他の指標においては全ての設定において低い値を示した．

それぞれの結果について考察を述べる．PARank に対してマージンサイズ変更，損失関数変更，損失への重みづけを変更という 3 つの改善を行った．マージンサイズ変更，ramp loss を導入することによって精度

[1] http://code.google.com/p/so a-ml/

表 1: 評価実験の結果

	Method	online /batch	Loss type	Margin type	Loss penalty	NDCG @1	@2	@3	@4	@5
(a)	PARank	online	hinge	const.	none	0.370	0.368	0.372	0.377	0.386
(b)					ΔNDCG	0.370	0.368	0.372	0.377	0.386
(c)				ΔNDCG	none	0.382	0.383	0.389	0.396	**0.402**
(d)					ΔNDCG	0.383	0.378	0.384	0.390	0.397
(e)			ramp	const.	none	**0.386**	0.386	0.391	0.393	0.400
(f)					ΔNDCG	**0.386**	0.386	0.391	0.393	0.400
(g)				ΔNDCG	none	0.383	**0.389**	**0.394**	**0.397**	**0.402**
(h)					ΔNDCG	0.381	0.388	0.390	0.392	0.398
	SPD (PA-I)	online				0.378	0.378	0.386	0.392	0.397
	RSVM	batch				0.410	0.407	0.406	0.408	0.414
	AdaRank-NDCG	batch				0.388	0.397	0.404	0.407	0.410

向上が可能であることが示されたが，損失への重みづけ変更ではかえって精度が低下することを確認した．

損失への重みづけ変更によって精度が低下する原因について考察する．我々の手法では，max-loss update を用いているため，各クエリについて損失が最大のペアに対して更新を行う．損失への傾きを導入することによって，元々損失が大きいペアに対してさらに大きな損失を与えてしまう．今回用いた PA-I では，パラメータ C の値で一度の更新幅をを抑えているため，大きな損失を下げずに次のクエリを処理に移るため，うまく学習ができていないと考えられる．マージンサイズのみを変更する場合には，傾きを変更しないため，そのような問題は軽減されていると考えられる．Ramp loss を用いた場合には，一定以上の損失を持つペアは等しいものと見なされ，更新に影響を与えないため，このような問題を解決し，高い精度を示していると考えられる．

これより，本稿で提案した NDCG の減少値に基づくマージンサイズの設定によって，マージンサイズ固定に比べて精度向上が可能であるといえる．また，ramp loss の導入によってさらに検索精度向上が可能であることが示唆されたが，マージンサイズの設定による効果と相殺している部分もあると考えられる．

SPD (PA-I) では，ランダムサンプリングによってペアを選択し，PA-I に基づいて重み更新を行っているため，表 1 の (a) とほぼ等しい検索精度であると予想した．しかしながら，結果を見ると SPD (PA-I) の方が (a) に比べて NDCG@1-5 において高い値を示している．これより損失関数に hinge-loss を用いてマージンサイズ固定の場合においては，ランダムサンプリングを行う戦略の方がよいということが示唆される．これは先ほど述べた理由と同じく，最大の損失ペアを選択する戦略では，損失が大きいペアがノイズとなって，なかなか適切な重みに収束しないという理由が考えられる．

4 おわりに

本稿では，PA を用いたペアワイズ手法にペア毎に異なる評価指標に基づいたマージンを設定することで，検索精度を改善したオンラインランキング学習手法 PARank を提案した．評価実験を通じて，ランダムサンプリングを行うペアワイズ手法である SPD (PA-I) に比べて NDCG@1-5 において高い精度で学習可能なことを示した．また，既存研究で提案された ramp loss を導入することにより，更なる性能向上が可能であることを確認した．ランキング学習への Ramp loss 適用の効果は初めての試みであり，損失への重みづけ方法との組み合わせの設定について網羅的な検証を通じて様々な知見を得られた．

オンラインランキング学習は今後も重要な課題になると考えている．今後はリストワイズ手法の性能を目指し，オンラインランキング学習が持つ課題の分析と手法の改善を引き続き行う予定である．

参考文献

[1] S. Büttcher, C. L. A. Clarke, and G. V. Cormack. *Information Retrieval: Implementing and Evaluating Search Engines*. MIT Press, 2009.

[2] T. Joachims. Optimizing search engines using clickthrough data. In *Proc. KDD '02*, pp. 133–142, 2002.

[3] Y. Cao, J. Xu, T.-Y. Liu, H. Li, Y. Huang, and H.-W. Hon. Adapting ranking svm to document retrieval. In *Proc. SIGIR '06*, pp. 186–193, 2006.

[4] J. Xu and H. Li. Adarank: a boosting algorithm for information retrieval. In *Proc. SIGIR '07*, pp. 391–398, 2007.

[5] D. Sculley. Large scale learning to rank. In *Proc. NIPS '09 Workshop on Advances in Ranking*, 2009.

[6] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithm. *Mach. Learn.*, Vol. 7, pp. 551–585, 2006.

[7] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.*, Vol. 20, No. 4, pp. 422–446, 2002.

[8] T. Minka and S. Robertson. Selection bias in the letor datasets. In *In Proc. LR4IR '08 (SIGIR '08 Workshop)*, 2008.

[9] J. Xu, C. Chen, G. Xu, H. Li, and E. Abib. Improving quality of training data for learning to rank using click-through data. In *Proc. WSDM '10*, pp. 171–180, 2010.

[10] R. Collobert, F. Sinz, J. Weston, and L. Bottou. Trading convexity for scalability. In *Proc. ICML '06*, pp. 201–208, 2006.

[11] Z. Wang and S. Vucetic. Online passive-aggressive algorithms on a budget. In *Proc. AISTATS '10*, pp. 908–915, 2010.

[12] T.-Y. Liu, T. Qin, J. Xu, W. Xiong, and H. Li. Letor: Benchmark dataset for research on learning to rank for information retrieval. In *Proc. SIGIR '07 Workshop: LR4IR '07*, 2007.