

学位論文 博士 (工学)

多様な情報源に対するランキング学習
に関する研究

2014年度

慶應義塾大学大学院理工学研究科

数原 良彦

要旨

ユーザが望む順序で結果を提示するランキングは様々なアプリケーションにおいて重要である。特に情報検索や推薦においては、ユーザが求める情報を優先的に上位に表示することにより、より少ないコストでユーザが目的を達成できる。このように、情報処理の広い分野において、ユーザの希望に合うアイテムに優先順位をつけて提示する方法の基盤技術としてランキング技術が用いられてきた。近年では、教師あり学習の枠組みを用いて最適なランキングモデルを構築するランキング学習と呼ばれる手法が情報検索分野で盛んに研究されている。ランキング学習においては、あらかじめ評価ラベルが付与された訓練事例集合をもとに予測モデルを構築し、未知の事例集合に対して評価ラベルと整合する順位を予測する。本研究では、既存のランキング学習が抱える (1) 複数情報源に対する適用, (2) 大規模データに対する適用, (3) 学習データに含まれるノイズへの対応, (4) 少量の特徴のみ利用可能なデータへの適用という4つの課題を達成し、多様な情報源に対する実用的なランキング学習実現を目指す。

以下に本論文の構成を示す。はじめに第1章において本研究の背景と目的について述べる。第2章では、ランキング学習とその関連研究について述べる。第3章では、人手評価データとクリックログデータという適合性分布が異なる複数情報源を用いる、ブースティングに基づくランキング学習アルゴリズムである TRankBoost を提案し、実データを用いた評価実験の結果を述べる。第4章では、検索評価指標のマージンへの導入、ノイズに頑健な損失関数の採用、最大損失を与える事例の選択といった特長を持つ選択的ペアワイズ手法によるオンラインランキング学習である PARank-NDCG を提案し、従来のオンラインランキング学習に対して優れた性能を示すことについて述べる。第5章では、ランキングに用いる多数の要因を抽出できない状況において高精度なランキングを実現するため、自己組織化マップを教師あり順序学習に拡張した OrderSOM を提案し、その評価について述べる。最後に第6章で本論文のまとめと今後の課題および展望について述べる。

目次

第1章	序論	1
1.1	背景と目的	1
1.2	適合性分布が異なる情報源を用いたランキング学習 (課題1の達成)	2
1.3	評価指標を考慮した選択的ペアワイズ手法によるオンラインランキング学習 (課題2, 3の達成)	3
1.4	順序ラベルを用いた教師あり自己組織化マップ (課題4の達成)	4
1.5	論文の構成	4
第2章	関連研究	5
2.1	検索ランキング	5
2.1.1	ランキング手法	6
2.1.2	検索ランキング評価指標	7
2.2	ランキング学習	8
2.3	ランキング学習の関連研究	11
2.3.1	クリックログを用いたランキング学習	11
2.3.2	複数情報源を用いたランキング学習	11
2.3.3	オンラインランキング学習	13
2.4	自己組織化マップの関連研究	14
第3章	適合性分布が異なる情報源を用いたランキング学習	16
3.1	概要	16
3.2	複数情報源を用いたランキング学習	17
3.2.1	動機と課題	18
3.2.2	転移学習によるランキング学習	19
3.2.3	TRankBoost	21
3.3	評価	24

3.3.1	実験条件	25
3.3.2	結果	27
3.4	考察	28
3.4.1	TRankBoostI の性能が低い原因	30
3.5	まとめ	30
第4章	評価指標を考慮した選択的ペアワイズ手法によるオンラインランキン グ学習	33
4.1	概要	33
4.2	PARank-NDCG	35
4.2.1	PA に基づくランキング学習	35
4.2.2	NDCG に基づくマージン設定	36
4.2.3	最大損失更新と高速選択手法	39
4.2.4	Ramp loss を用いたロバストな学習	40
4.2.5	アルゴリズム	43
4.3	評価	43
4.3.1	データセット	45
4.3.2	MSLR-WEB を用いた評価	46
4.3.3	学習効率性の比較	47
4.4	考察	49
4.4.1	マージンサイズ導入の効果	50
4.4.2	Ramp loss 導入の効果	50
4.5	まとめ	51
第5章	順序ラベルを用いた教師あり自己組織化マップ	53
5.1	概要	53
5.2	自己組織化マップと順序学習	55
5.2.1	順序学習	55
5.2.2	自己組織化マップ	56
5.3	順序型自己組織化マップ	57
5.3.1	OrderSOM の概要	58
5.3.2	1次元 OrderSOM	59

5.3.3	多次元 OrderSOM	61
5.4	評価	63
5.4.1	実験 1: 人工データを用いた評価	63
5.4.2	実験 2: 検索ランキングデータを用いた評価	66
5.5	考察	69
5.6	まとめ	70
第 6 章	まとめ	72
6.1	結論	72
6.2	今後の課題	73
	参考文献	74

目 次

2.1	検索結果のランキングを行う検索エンジンの概要	6
2.2	1 クエリの検索結果に対する適合性評価の例	7
2.3	クエリ依存, クエリ非依存の要因を用いて検索スコアを計算するラ ンキング関数の例	9
2.4	1 クエリの検索結果に対するクリック数を適合性評価として用いる 例	12
3.1	同一クエリに対する適合性分布: (a) 人手による適合性評価 (b) ク リックログ	19
3.2	TRankBoost アルゴリズム	19
3.3	RankBoost, TRankBoostI, TRankBoostII の比較	24
3.4	NDCG@ k 値の比較結果: (a) 提案手法と human, (b) 提案手法と click, (c) 提案手法と both	32
4.1	異なる適合度間のマージンサイズ	37
4.2	マージンの計算方法	38
4.3	損失関数: (i) hinge loss, (ii) ramp loss.	41
4.4	PARank-NDCG アルゴリズム	45
4.5	MSLR-WEB10K データセットの Fold1 における学習曲線	48
4.6	損失関数の比較	50
4.7	マージンサイズ設定の比較 (i) MSLR-WEB10K データセット, (ii) MSLR-WEB30K データセット	51
5.1	1次元 OrderSOM の更新アルゴリズムの概要 . 順序ラベルと現在の OrderSOM による順位が整合していない場合	59
5.2	1d-OrderSOM の更新アルゴリズム	61
5.3	1d-OrderSOM の予測アルゴリズム	62

5.4	2d-OrderSOM の更新アルゴリズムの概要	63
5.5	実験 1a における 1d-OrderSOM と RSVM の学習結果	65
5.6	実験 1b における OrderSOM (a) $N_x = 5, N_y = 1$ (b) $N_x = 5, N_y = 6$ の学習結果	67
5.7	実験 2: 特徴次元数毎の τ の値	69

表 目 次

3.1	データセットの概要	26
3.2	実験に用いた素性集合	26
3.3	実験結果	28
4.1	MSLR-WEB10K と MSLR-WEB30K における実験結果	44
4.2	データセットの概要	45
4.3	MSLR-WEB10K データセット (Fold1-5) における平均訓練時間	49
5.1	実験 1a の結果	64
5.2	実験 1b の結果	66
5.3	実験 2: LETOR データに対する評価結果 . Small feature set 括弧内の数字は特徴次元数を表す	68

第1章 序論

1.1 背景と目的

ユーザが望む順序で結果を提示するランキングは様々なアプリケーションにおいて重要である。情報検索や推薦においては、ユーザが求める情報を優先的に上位に表示することにより、より少ないコストでユーザが目的を達成する。また、ユーザの活動をサポートする意思決定支援システムなどにおいても、ユーザの活動候補を適切にランキングして提示することで、意思決定を支援することが可能である。このように、情報処理の広い分野においてユーザに同時に提示不可能な大量の候補群から、優先順位をつけてユーザの希望に合うアイテムを提示する方法のひとつとしてランキング技術が用いられてきた。

近年、情報検索の分野で教師あり学習の枠組みを用いて最適なランキングモデルを構築する技術が盛んに研究されている [43]。教師あり学習を用いたアルゴリズムにおいては、あらかじめ評価ラベルが付与された訓練事例集合をもとに予測モデルを構築し、未知の事例集合に対して評価ラベルと整合する順位を予測する。この際、評価ラベルとしては順位そのものや、順位グループを表すラベルなどを用いる。なお、本問題においては真の順位そのものを予測する必要はなく、真の順位と整合的な順位を予測すればよい。

計算機能力の向上と計算機資源の価格低下によって、産業界においても大規模データを用いた技術運用が進んできた。大規模データ分析のアプローチとして機械学習が広く利用されている。機械学習の一分野である教師あり学習では、人手などによって与えられた教師信号を再現するようなモデルを構築し、それを用いることによって未知のデータに対して予測を行うことを可能とする。

この流れは情報検索分野にも影響を与え、ランキング学習 (learning to rank) と呼ばれる一分野が形成されるに至っている。ランキング学習では、教師あり学習の枠組みを用いることで、従来は少数の要因に対して人手で重みを設定する必要

があったのに対し，多数の要因を用意し，自動的に重みを調整することが可能となった．このように情報検索分野において，教師あり学習の枠組みを用いてランキングモデルを構築する手法が一般的に利用されるようになってきた．

しかしながら，実用的なランキング学習技術実現にあたっては，いくつかの課題が存在する．

1. 複数の異なる性質の訓練データを用いて高精度なモデルを構築すること (課題 1) .
2. 大規模データに対して学習アルゴリズムがスケールし，逐次学習が可能であること (課題 2) .
3. 学習データに含まれるノイズに強い学習アルゴリズムであること (課題 3) .
4. 低次元特徴空間においても有効な予測モデルを構築可能なこと (課題 4) .

本論文においては，上記の課題の達成に取り組む．以下，取り組む課題と本論文における解決方法を対応づけて述べる．

1.2 適合性分布が異なる情報源を用いたランキング学習 (課題 1 の達成)

ランキング学習においては，通常，人手によって構築された訓練データを用いてランキングモデルの生成を行う．しかしながら，人手による訓練データ生成はコストが高いため，作成できる量が限定されているという問題がある．一方で，検索サービスにおいてユーザが入力したクエリと表示された検索結果のクリック位置を記録したクリックログを利用し，クリック回数を擬似的に人手によって判定された評価値と見なし，これを人手評価データの代わりに用いることでランキングモデルを生成する方法が知られている．

このような異なる情報源から得られた訓練データを同時に用いて高精度なランキングモデルを生成することは容易ではない．異なる情報源から得られた訓練データにおいては，目標値の分布である適合性分布が異なるため，単純に訓練データを混合すると，かえってランキング精度が低下するおそれがある．また，実際に混合するとランキング精度が向上するような訓練データの部分集合を事前に知る

ことは困難であり，このような異なる情報源から得られた訓練データを同時に扱う枠組みが必要である．以上より，本論文では複数の情報源からの訓練データを用いる転移学習の枠組みを用いたランキング学習手法を提案し，人手評価に基づく訓練データとクリックログに基づく訓練データを用いて，単純に訓練データを混合する方法に比べて高精度なランキングを実現可能であるか評価を行う．この手法の開発を通じて課題1の達成を目指す．

1.3 評価指標を考慮した選択的ペアワイズ手法によるオンラインランキング学習（課題2，3の達成）

検索サービス運用を通じて日々蓄積されるクリックログなどを訓練データとして用いるために，大規模訓練データに対するランキング学習手法が必要となる．更に，より短いスパンでランキングモデルを更新するためには，追加された訓練データのみを用いてランキングモデルを逐次更新する必要がある．訓練データの一部をサンプリングしてモデル更新に用いるオンライン学習の枠組みを用いることで，これらの要件を満たすことが可能である．既存のオンラインランキング学習によって課題2をある程度達成しているといえるが，ランキング精度の観点からまだ向上の余地があると考えた．本稿では，更なる精度向上を目指して評価指標を損失関数に導入し，最大損失を与える文書ペアを優先的に選択し，パラメータ更新を行う手法を提案する．

また，特にオンライン学習はデータに含まれるノイズの影響を受けやすいことが知られており，学習データに含まれるノイズに強いランキング学習手法であることももう1つの目標である．この目標を達成するために，本稿では一定以上の損失を与える事例の損失を等しいとみなす ramp loss を損失関数に導入する．実験による評価を通じて，提案手法によって既存のオンラインランキング学習手法に比べて高い精度でランキングを実現可能なことを示し，2つの課題を達成していることを検証する．

1.4 順序ラベルを用いた教師あり自己組織化マップ (課題4の達成)

低次元特徴空間においても有効なランキングモデル生成は重要な目標である。ウェブ文書の場合には、リンク情報、ウェブサーバを提供しているドメイン情報、HTMLで記述された文書の構造情報など、それらをもとに多数の要因を用いたランキングモデルを用意することが可能である。しかしながら、たとえばデスクトップ検索の対象となるテキスト文書の場合には、文書の本文と最終更新日、ファイルパス程度の情報しか利用できないため、高々数次元の要因を用いてランキングモデルを生成する必要がある。そのため、低次元特徴空間において高精度に学習可能なランキング学習アルゴリズムが必要となる。以上より、本稿では低次元特徴空間で複雑な評価分布をしている場合でも対応可能な、自己組織化マップを教師あり順序学習に拡張した OrderSOM を提案し、その評価について述べる。この手法の開発を通じて、課題4の達成を目指す。

1.5 論文の構成

本論文の構成は以下のとおりである。第2章では、本研究の関連技術として情報検索、機械学習およびランキング学習の概要と関連研究について述べる。3章では、人手評価データとクリックログデータという適合性分布が異なる複数情報源を用いる、ブースティングに基づくランキング学習アルゴリズムである TRankBoost を提案し、実データを用いた評価実験の結果を述べる。4章では、検索評価指標のマージンへの導入、ノイズに頑健な損失関数の採用、最大損失を与える事例の選択といった特徴を持つ、選択的ペアワイズ手法によるオンラインランキング学習である PARank-NDCG を提案し、従来のオンラインランキング学習に対して優れた性能を示すことについて述べる。5章では、ランキングに用いる多数の要因を抽出できない状況において高精度なランキングを実現するため、自己組織化マップを教師あり順序学習に拡張した OrderSOM を提案し、その評価について述べる。最後に第6章では、本論文のまとめと今後の課題および展望について述べる。

第2章 関連研究

本章ではまず検索ランキングにおける基本事項について述べたのちに、ランキング学習の関連研究について述べ、最後に自己組織化マップの関連研究を述べる。

2.1 検索ランキング

情報検索においては、ユーザが自分自身が望むコンテンツ (文書) を表す言語表現であるクエリを入力として受け取り、その結果をユーザに提示する。この際、検索システムは入力されたクエリに対して、ユーザの意図に合った検索結果を提示するため、検索結果を順番に並べて表示するランキング方式が広く用いられている。このように検索結果が順序づけられて提示された結果を本稿では検索ランキングと呼ぶ。特にウェブ検索のような検索システムにおいては、ユーザは検索ランキングの上位僅か数件から 10 件程度しか確認しないことが多い。そのため、ユーザ満足度が高い検索システム実現のためには、検索結果上位数件に適切な文書が現れるような検索ランキングを提供することが必須の課題となる。

ウェブ検索をはじめとする検索ランキングを用いる検索エンジンの仕組みについて述べる。図 2.1 に検索エンジンの概要を示す。このような検索エンジンは、ユーザがキーワードという形でクエリを入力し、検索エンジンは入力されたクエリに適合する文書を、入力されたクエリの意図を適切に反映した順番に並べて表示を行う。このとき、検索エンジンは、(1) 入力クエリに適合する文書群の取得、(2) 取得した文書群のランキングを行う、という二段階の処理によって検索ランキングを実現する。1 つ目の入力クエリに適合する文書群の取得には、検索対象であるすべての文書群に対してあらかじめ索引を構築しておくことで高速に条件を満たす文書群を取得可能な全文検索技術を用いる。2 つ目の処理においては、各文書に対してクエリとの関連度やそれ以外の情報を用いてスコアを計算し、取得した文書群をスコアの降順で並べる方法が一般的に用いられる。本稿においては、1 つ目

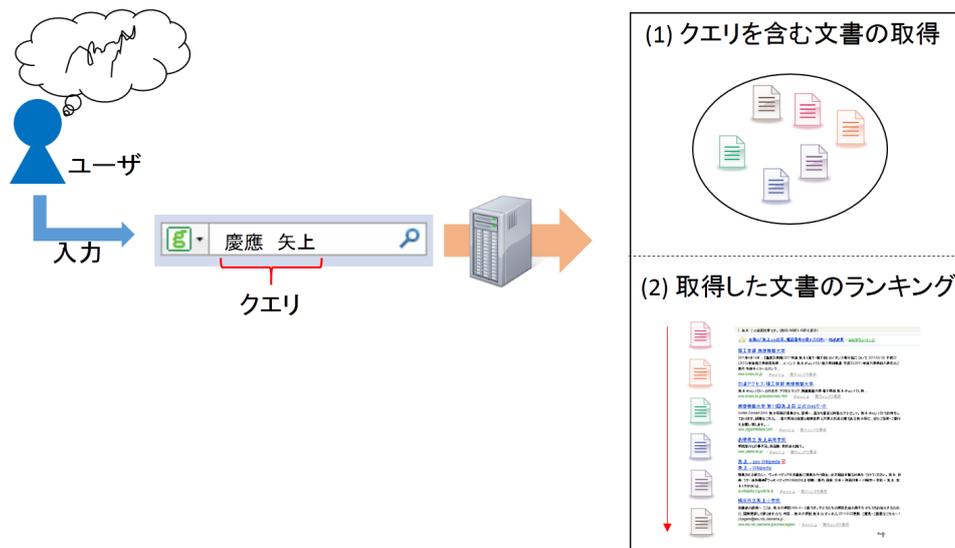


図 2.1: 検索結果のランキングを行う検索エンジンの概要 .

の入力クエリを含む文章群は所与のものとした際に、与えられた文書集合を適切に並べる検索ランキングを得る課題に取り組む。

適合度 (relevance score) とは、入力クエリに対して取得された文書がどれほど適切にユーザの検索意図を満たしているかということを表す指標である。そのため、適合度はクエリと文書に対して規定され、たとえ同じ文書であってもクエリによって適合度が異なることに注意されたい。真の適合度は測定不能であるが、複数のアノテータによってクエリと検索結果集合に対して人手で適合度を付与し、その結果を利用する方法がよく用いられる。

2.1.1 ランキング手法

情報検索においては、大きく (1) クエリ依存のランキング手法、(2) クエリ非依存のランキング手法という 2 つのアプローチに分けて研究が行われてきた。

クエリ依存のランキング手法は、入力されたキーワードと文書の関連度に基づいてスコアを計算する方法である。代表的な手法として、キーワード頻度に対してキーワードの検索対象における珍しさで重み付けを行う TF-IDF [44] や、Robertson らによる Okapi システムに実装された Best Match (BM) ランキングアルゴリズムがある。そのうち BM25 [53] がその後も情報検索ランキングのベースライン手法として長年採用されてきた。BM25 は、ウェブ文書のように文書に含まれる単語数



図 2.2: 1 クエリの検索結果に対する適合性評価の例 .

が文書ごとによって異なる状況に対応した TF-IDF と解釈することもできる . その他の方法としては , ベクトル空間モデルや言語モデルに基づく手法などがクエリ依存のランキング手法として挙げられる [44] .

クエリ非依存のランキング手法は , ウェブ文書におけるリンク解析など文書自体の重要度を計算し , ランキングに利用する方法である . 特に Google と共に有名になった PageRank アルゴリズム [4] が代表的なリンクベースのランキング手法である . その他の方法としては , HITS アルゴリズム [39] , HITS と PageRank の両方の性質を引き継いだ SALSA アルゴリズム [41] などが挙げられる .

2.1.2 検索ランキング評価指標

検索ランキングの良し悪しを測るために用いる検索ランキングの評価指標について述べる . 定量評価を行うため , 評価対象であるクエリ集合と各クエリについて , 当該クエリの検索結果となる文書集合の一部を評価対象とし , この文書群に対して人手で評価点数をつける方法が一般である . 図 2.2 に評価点数付与の概略図を示す . この例では , goo というクエリに対して得られた検索結果集合に対して , 適合性評価の付与を行う . ここで適合性評価 (relevance judgement) とは , 検索結果集合の各文書に対して入力クエリの検索意図を適切に反映しているかを判定する評価であり , これによってあるクエリの検索結果集合に含まれる各文書に対して適合度が付与される . 適合度の判定結果として , 1 ~ 5 のような離散値で評価することが多い . 複数の被験者の適合性評価を平均する .

ウェブ検索のような典型的な検索エンジンにおいては , 検索結果上位に表示され

るものは、下位に表示されるものに比べ、重要であるべきである。したがって、適合度が高い文書をより上位の検索結果として提示する場合に、高い評価値となるような評価指標が好ましい。一部の検索評価指標は多段階の適合度に対して評価値を計算するものであり、このような要求を満たしている。本稿では、多段階評価指標の中で情報検索分野で広く利用されている Normalized Discounted Cumulative Gain (NDCG) [33] を評価指標として用いる。Mean Average Precision (MAP) [7] も情報検索でよく用いられる評価指標ではあるが、MAP は多段階の適合度に対して計算できないため本稿では対象としない。

NDCG は特にランキング上位と高い適合度を持つ文書に着目した検索評価指標である。同じ順位においては、適合度に対してスコアは指数的に加算され、順位の重要度は順位に対して対数的に減少する。NDCG@ k [33] は検索結果上位 k 件における評価指標であり、以下のとおり計算する：

$$\begin{aligned} \text{DCG}_q@k &\equiv \sum_{i=1}^k \frac{2^{y_i^{(q)}} - 1}{\log(1 + i)} \\ \text{NDCG}_q@k &\equiv \frac{\text{DCG}_q@k}{\max \text{DCG}_q@k} \end{aligned}$$

ここで $y_i^{(q)}$ はクエリ q に対するランキング結果 i 位の文書の適合度であり、 $\max \text{DCG}_q@k$ はクエリ q に対する理想的なランキングにおける $\text{DCG}@k$ の値である。すなわち、 $\max \text{DCG}_q@k$ が $\text{DCG}_q@k$ の上限であり、正規化によって $\text{NDCG}_q@k \in (0, 1]$ となる。多くのウェブ検索エンジンでは一度に表示する検索結果を 10 件に設定しているため、ランキング学習アルゴリズムの評価指標としては $\text{NDCG}_q@1 \sim \text{NDCG}_q@10$ がよく用いられる。

2.2 ランキング学習

ランキング学習とは、主に情報検索ランキングの精度向上を目標として、機械学習を用いて予測モデルを構築する方法の総称である。ランキングに用いる要因の数が膨大になるにつれて、人手による調整が困難となり、機械学習を用いてランキング関数を生成するアプローチが広く用いられるようになってきた [43]。例えば、線形モデルを仮定する場合、検索ランキングの従来手法である、クエリ依存、クエリ非依存のランキング手法のスコアを要素とするベクトルに対して、最適なランキ

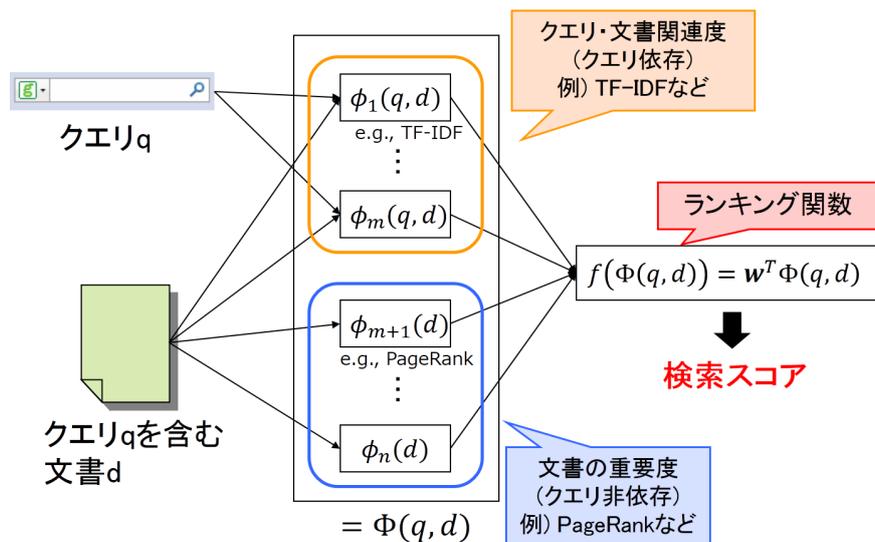


図 2.3: クエリ依存, クエリ非依存の要因を用いて検索スコアを計算するランキング関数の例 .

ングを実現する重みベクトルを生成することがランキング学習の目的である . 複数の要因を用いたランキング関数の例を図 2.3 に示す . この例では , クエリ q に対して得られた文書 d について , クエリ依存のランキング要因 $\phi_1(q, d), \dots, \phi_m(q, d)$ とクエリ非依存のランキング要因 $\phi_{m+1}(d), \dots, \phi_n(d)$ を要素とする特徴ベクトル $\Phi(q, d)$ に対して , 重みベクトル \mathbf{w} との内積によって検索スコアを算出している .

ランキング学習の訓練データはクエリ , クエリの検索結果である文書集合 , そして各文書に対して付与された適合度から構成される . 適合性評価は複数の評価者によって付与され , 評価者の平均値を用いる . この際 , ユーザの検索意図を満たしているか適切に判定するため , あらかじめガイドラインを用意し , 評価者の点数のぶれを防ぐ . ここで同じ文書であっても異なるクエリにおいては適合度評価が異なるため , 適合度評価はクエリ・文書ペアに付与されることに注意されたい .

ランキング学習はポイントワイズ (pointwise) , ペアワイズ (pairwise) , リストワイズ (listwise) の 3 つのアプローチに大別される [43] . これらのアプローチは , どのように目的関数を設定するかによっている . ポイントワイズアプローチ [17, 42] は , 個別の文書に対して設定される損失関数を最小化するようなランキング関数を生成する方法である . ペアワイズアプローチ [34, 22, 55] は , 文書ペアの順序関係に対して設定された損失関数を最小化するようなランキング関数を生成する方法である . 線形モデルを用いるペアワイズアプローチは , 2 つの事例の特徴ベク

トルの差ベクトルを 1 事例とみなすことで、二値分類の教師あり学習問題として定式化することが可能である。ペアワイズアプローチの問題は、目的関数が文書リストではなく、文書ペアの順序誤りの最小化に基づいて学習アルゴリズムが設計されている点にある。リストワイズアプローチは、クエリの検索結果である文書群のリストに対して設定された損失関数を最小化するようなランキング関数を生成する方法である。情報検索評価指標の多くはクエリ毎の文書リストによって算出されるため、リストワイズアプローチを用いることで検索評価指標を近似的、または直接的に最適化することが可能となる。NDCG や MAP を最適化するリストワイズアプローチに基づく手法 [6, 57, 60, 63, 66] は、一般的にペアワイズ手法より高精度なランキング関数の生成が可能であるといわれている。しかしながら、リストワイズアプローチはクエリ毎の文書リストに対する損失関数を計算するため、計算量が多くなるという問題点がある。

本稿で用いる枠組みはペアワイズ手法であるため、ペアワイズ手法について詳細を述べる。ペアワイズ手法においては、あるクエリにおいて異なる適合性評価を持つ 2 つの文書から成る文書ペアを用いる。すなわち、クエリ q の検索結果である 2 つの文書 $(\mathbf{x}_a^{(q)}, \mathbf{x}_b^{(q)})$ それぞれの適合性評価を $(y_a^{(q)}, y_b^{(q)})$ とすると、 $y_a^{(q)} \neq y_b^{(q)}$ である。重みベクトル \mathbf{w} をパラメータとする線形モデルを仮定すると、特徴ベクトルを $\mathbf{x} = \mathbf{x}_a^{(q)} - \mathbf{x}_b^{(q)}$ 、ラベルを $y = \text{sign}(y_a^{(q)} - y_b^{(q)})$ とみなすことで、順序誤りの最小化問題を二値分類問題とすることができる。

RankingSVM (RSVM) [34] はよく知られたペアワイズ手法のランキング学習アルゴリズムである。RSVM は以下の最適化問題を解くことで、パラメータを学習する:

$$\begin{aligned} \min. \quad & \|\mathbf{w}\|_2^2 + C \sum_{q \in Q} \sum_{a,b} \xi_{a,b}^{(q)} \\ \text{s.t.} \quad & \text{sign}(y_a^{(q)} - y_b^{(q)}) \mathbf{w}^T (\mathbf{x}_a^{(q)} - \mathbf{x}_b^{(q)}) \geq 1 - \xi_{a,b}^{(q)} \quad \forall q, a, b. \end{aligned}$$

この最適化問題は、たとえば SMO や Coordinate Descent など、通常の SVM と同様に解くことができる [50][23]。

2.3 ランキング学習の関連研究

2.3.1 クリックログを用いたランキング学習

ユーザが入力したクエリと当該クエリの検索結果に対するクリック履歴を保存した検索ログ (クリックログ) を用いたランキング学習の取り組みがある。クリックログの利用方法は (1) クリックログから訓練データを抽出するアプローチ [34][35][51], (2) クリックログから素性を得るアプローチ [65][1][27] の2つに分けられる。本稿では, 1つ目の訓練データとしてのクリックログ利用方法に着目する。

Joachims[34] は, あるユーザのクリックログのうち, あるクエリの検索結果として表示された文書で, クリックされた文書よりも通常は先に読まれ, しかしクリックされなかった文書よりも, 相対的に適合度が高いというアイデアに基づいてランキング関数学習手法を提案している。一方, Douら [20] は, 個別のクリックログではなく, クエリ・文書に対するクリック数を集約し, クリック数が多い方がより適合度が高いという仮定に基づいて訓練データを作成する手法を提案している。文献 [34] による訓練データ生成方法では, 複数ユーザのクリックログを利用した結果, 同じクエリと文書集合に対して矛盾する順序関係がラベルとして生成されるおそれがある。そのため, 本稿ではクリックログを用いた訓練データ生成方法として Douら [20] の方法を採用する。

図 2.4 にクリックログを人手による適合性評価の代わりに用いる例を示す。特にウェブ検索においては検索ランキング上位に強いバイアスがかかること (ランキングバイアス) が知られており, 上位に提示された検索結果はクリックされやすいという性質を持つ。そのため, クリック数が必ずしも適合度そのものを反映しているわけではないということが知られている [52]。この例の場合では, 検索結果上位にスパムが出現しているが, 一定数のユーザがスパムと気づかずにクリックしてしまい, 結果的に検索結果の3番目に出現する非スパム文書に比べてクリック回数が多くなっている。

2.3.2 複数情報源を用いたランキング学習

情報検索においてランキング学習の研究は盛んに行われている。既存のランキング学習手法は学習モデルの定式化の観点から3種類に分けることができる。ポイ



図 2.4: 1クエリの検索結果に対するクリック数を適合性評価として用いる例。

ントワイズ手法 [42] は、分類や回帰の問題としてランキング学習を実現する。ペアワイズ手法 [5, 8, 25, 31] は文書ペアの順序に対して誤差関数を設定し、これを最適化することでランキング学習を実現する。リストワイズ手法 [9, 58, 60, 64, 66] は、文書のリストに対して設定された誤差関数を最適化を行う。

人手による適合性評価とクリックログを直接比較した研究は少ない [20]。Douら [20] は、人手による適合性評価から得られた訓練データとクリックログから得られた訓練データを用いたランキング学習の性能を比較している。クリックログから得られた訓練データを用いたランキング学習により、人手による適合性評価から得られた訓練データを用いた結果に比べて高い精度が得られることを示した。ランキング評価に用いるテストデータには、人手による適合性評価に基づく評価データを用いているため、一般的には人手による適合性評価から得られた訓練データを用いる方が精度が高いランキングモデルを生成できると考えられる。予想に反する結果が得られたことに対して、Douらは、クリックログを用いることで、人手による適合性評価を用いる場合に比べて大量の訓練データを用いることができるため、このような結果が得られたと考察している。Kampsら [37] は、人手による適合性評価とクリックログの相関を分析し、弱い相関とある程度の一致があると結論付けている。これらの知見より、クリックログから得られた訓練データには必ずしも人手による適合性評価と一致しないものの、膨大な量を利用可能であるため、ランキング学習において一定の有用性があると考えられる。

半教師あり学習の枠組みでランキング学習を行う研究もある [21, 38, 59]. 半教師あり学習では, ラベルなしデータに着目する. 例えば Duh ら [21] は, ランキング対象のラベルなしデータを用いることで, 当該データをランキングするのに適切な素性を選択してランキング学習を行うトランスダクティブ手法を提案している. 本稿で提案する手法は, 人手による適合性評価から得られた訓練データ以外に利用する, 訓練データに付与されたラベル情報を用いる点で半教師あり学習の枠組みとは異なる.

複数の訓練データを用いて, ドメイン適応 (domain adaptation) の枠組みによって高い精度のランキング関数を生成する試みもある. Chen ら [13] は, 別の国など異なるドメインから得られる訓練データを用いることを提案している. Chen らの方法では, まず別ドメイン, すなわち別の国の訓練データを用いて Gradient Boosting Tree (GBT) を構築し, 目標ドメインの訓練データを用いて GBT の更新を行う. 別の Chen ら [12] は, 設定された効用関数によって, 別ドメインの訓練データの中から目標ドメインの学習に適切なクエリを選択し, 訓練データに加える TransRank と呼ばれる手法を提案している. これらの研究はいずれも素性空間における訓練データの分布の違いを考慮しているものの, 適合性分布の違いを考慮していない.

クリックログから適切な適合性評価を抽出する研究がある [2, 10, 27, 34, 35, 36, 52]. Joachims [34] の方法では, クリックログにおけるクリック位置に着目し, ランキング 2 位の文書をクリックせずに 3 位の文書をクリックした場合, 3 位の文書が 2 位の文書よりも好まれるという情報を抽出する. Gao ら [27] は, 人手による適合性評価を用いることで大量のクリックログからランキングバイアスを取り除くスムージング手法を提案している. これらの研究では, クリックログのみを用いてランキング学習を行っている.

2.3.3 オンラインランキング学習

オンラインランキング学習アルゴリズムもいくつか [22][55] 提案されている. オンライン学習においては計算コストの問題からペアワイズ手法を用いることが一般的である.

線形モデルを仮定すると, 順序誤りの最小化を二値分類問題として定式化できることを既に述べた. そのため, 1 事例をサンプリングし, 1 事例に基づいて計算さ

れた損失関数の勾配によって損失関数の最小化を行う Stochastic Gradient Descent (SGD) をペアワイズ手法に拡張したものが Stochastic Pairwise Descent (SPD) [55] である。SPD はすべてのクエリ・文書ペア集合の中から同じクエリに含まれる文書ペアを2つサンプリングし、損失関数の勾配を計算し、重みの更新処理を行う。重みの更新処理には Perceptron 学習アルゴリズム、Passive Aggressive アルゴリズム または online SVM などを用いることが可能である。

2.4 自己組織化マップの関連研究

SOMの応用例として、1次元SOMを用いて順位を再現する方法がある [40]。ただし、この方法では従来の教師なし学習であるSOMであり、明示的な順序ラベルを利用しない。そのため、各順位の事例集合が特徴空間上で順位に沿って存在しない限り、適切な順位の再現ができない。

教師ありSOMはいくつかの方法が提案されている [40, 24, 30, 54]。SOMの最も一般的な教師あり学習への拡張方法としては、特徴ベクトルにラベルを加えた拡張ベクトル $\mathbf{x}' = (y, \mathbf{x}^T)^T$ を用いる方法である。ここで y はクラスを表し、 \mathbf{x} は元の特徴ベクトルを表す。これを用いてSOMを学習することによってラベルが未知のデータについても予測が可能となる。ただし、この方法ではラベルとして明示的な順位の情報が必要であるため、本稿で扱う相対的な順位のみから順位関数を推定する問題を扱うことができない。

本稿で提案する更新ノードの修正に関連する研究として、ノードを交換することで効率的な学習を実現する Miyoshi の手法 [46] が挙げられる。しかしながら、この方法ではノードの交換すなわち、重みベクトルそのものの交換を行い、教師情報を利用しないという点で提案手法と異なる。

SOMにおいて競合層のノードに対して隣接するノードが持つ重みベクトルとの整合性に基づいて正則化を行う Goppert ら [29] の研究がある。OrderSOMは相対的な順位に基づいて更新すべきBMUを修正する方法であるため、何かしらの情報に基づいて競合層のトポロジーを修正する方法という点で関連するが、Goppert らの方法では教師ラベルを競合層のトポロジーに利用しない点で異なる。

相対順位から順位関数の予測を行う順序学習は情報検索分野においても広く研究されている。本稿で対象とする順序ラベルを持つ事例ペアから順位予測を可能

とする枠組みはペアワイズ手法と呼ばれ，RankingSVM [34] が代表的な手法として知られている．しかしながら，順序学習においてSOMを適用した例は著者の知る限り存在しない．

第3章 適合性分布が異なる情報源を用いたランキング学習

3.1 概要

ランキング学習においては、通常、人手によって構築された訓練データを用いてランキング関数の生成を行う。しかしながら、人手による訓練データ生成はコストが高いため、作成できる量が限定されているという問題がある。本章では複数の訓練データを用いる、転移学習の枠組みを用いたランキング学習手法を提案し、人手評価に基づく訓練データとクリックログに基づく訓練データを用いて、単純に訓練データを混合する方法に比べて高精度なランキングを実現可能であるか評価を行う。

Douら [20] は、人手による適合性評価を用いた場合と、クリックログを用いた場合のランキング学習の性能比較を行った。実験結果より、クリックログを訓練データに用いることで、より高性能のランキング学習が可能であることを示した。また、Douらは人手による適合性評価を用いたランキング学習が劣る理由として、訓練データ数の不足を理由に挙げている。このように、訓練データ数が不足しているため、与えられた訓練データに対して過剰に適合し、未知のデータに対して汎化性能を持たないモデルを学習する現象を過学習 (over fitting) と呼び、機械学習を適用する際にしばしば問題となる。

ランキングの有効性を表す評価指標は、人手による適合性評価を用いて計算されるため、目標とすべき適合性評価の分布 (適合性分布) は人手による適合性評価における適合性分布である。本稿では人手による適合性評価が十分多量に用意されれば、過学習の問題が解消され、高精度なランキング学習が可能であると考えている。しかしながら、訓練データの作成コストは非常に高いため、十分な量の適合性評価を人手で用意することは困難である。そこで本稿では人手による適合性評価に加えて、クリックログなどの情報源から得られる訓練データを適切に活

用することにより、高性能なランキング関数が実現可能だと考えた。

クリックログはユーザのクリック行為に基づく暗黙的な適合性評価のため、人手によって作成された明示的な適合性評価とは異なる適合性分布であることが予想される。既存の教師あり機械学習の枠組みでは、訓練データに含まれる各事例が同じ適合性分布から得られるという仮定を置いているため、クリックログから得られた訓練データを単純に混ぜる方法では適切に学習ができないと考えられる。したがって、人手による適合性評価に適合性分布が異なる訓練データを加えてランキング学習を行うために、適合性分布の異なりを考慮しなければならない。

そこで本稿は、目標となる分布の訓練データに分布が異なる訓練データを加えて高精度な学習を行う転移学習の枠組みをランキング学習に適用することで、本課題の解決を目指す。本章では、転移学習を用いたランキング学習手法である TRankBoost を提案し、実データを用いた評価実験を通じて提案手法の有効性を検証する。本章の貢献は以下のとおりである。

1. 転移学習を適用する、適合性分布が異なる複数の訓練データを用いたランキング学習の枠組みを提案する。
2. 転移学習に基づく新しいランキング学習手法である TRankBoost を提案する。
3. 評価実験を通じて提案手法の有効性を検証する。また、適合性分布が異なる訓練データに対して従来手法がうまく働かないことを検証する。

本章の構成を以下に示す。3.2 章では複数情報源を用いたランキング学習の枠組みと、転移学習を用いたランキング学習について述べ、提案手法の詳細について述べる。3.3 章では、評価実験の内容と結果を述べ、3.4 章で結果を踏まえた考察を述べる。最後に本論文のまとめと今後の課題について述べる。

3.2 複数情報源を用いたランキング学習

本章では、複数情報源を用いたランキング学習の動機と課題について述べ、転移学習を用いたランキング学習を提案する。最後に提案手法である TRankBoost について詳しく述べる。

3.2.1 動機と課題

本節では，複数情報源を用いる動機と課題について述べる．

3.1 章で述べたように，人手による適合性評価は，目標となる適合性分布の情報を保持しており，訓練データを十分に用意することで，高精度なランキング学習を実現できると考えられる．しかし，人手による適合性評価はコストが高く，十分な訓練データを用意することは現実的には難しい．そこで，追加情報源を用いて人手による適合性評価を補完することによって性能を向上することを試みる．ここで追加情報源とは，情報源から得られる訓練データが部分的に目標とする適合性分布に類似しているものを表し，たとえばクリックログやソーシャルブックマークが挙げられる．本稿では，先行研究により適合性分布が類似していることが示されているクリックログを用いる．最も簡単には，人手による適合性評価とクリックログから得られた訓練データをひとつの訓練データとみなして，既存のランキング学習手法を用いる方法が考えられる．しかしながら，既存のランキング学習手法に用いられる教師あり機械学習では，訓練データが同一分布から得られることを仮定しているため，単純に混合された訓練データを用いて適切に学習できないことが予想される．

実際に人手による適合性評価とクリックログ間における適合性分布の異なりを確認するため，あるクエリの検索結果に対して付与された人手による 4 段階の適合性評価と，同じ文書に対するクリックログを用意した．クリックログは Dou らの方法 [20] を用いて適合性評価に変換し，人手による適合性評価と合わせるために，4 段階への正規化を行った．人手による適合性評価とクリックログそれぞれについて，BM25 スコア [53]，入力リンク数 [44] という 2 次元の素性空間上に文書の評価点数をプロットしたものを図 3.1 に示す．図 3.1(a) と (b) を比較すると，人手による適合性評価において，ほぼ全ての文書に対する評価点数が 2 点であるのに対して，クリックログでは評価点数に散らばりがある．これより，人手による適合性評価とクリックログから得られた適合性評価の分布が異なることがわかる．

もし人手による適合性評価とクリックログが完全な相関を示した場合（すなわち，図 3.1 の (a) と (b) が完全に一致する場合），これらの訓練データを混合したものに対して教師あり機械学習を利用することで効果が得られることが予想できる．一方，これらの間に相関が見られない場合には，人手による適合性評価にクリックログを加えても効果が得られないと考えられる．先行研究 [20][37] によって，人

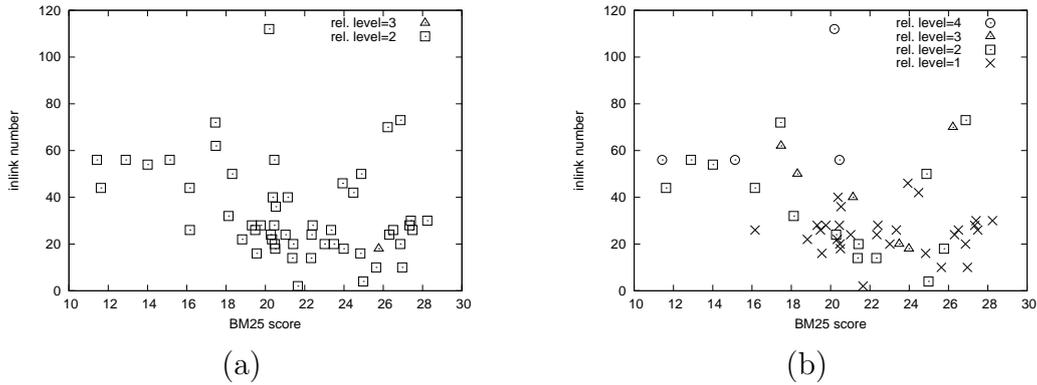


図 3.1: 同一クエリに対する適合性分布: (a) 人手による適合性評価 (b) クリックログ

Algorithm TRankBoost

Input Data sets \mathbf{x}^d , \mathbf{x}^t , weak learn algorithm **Learner** and iteration number N

Initialize $D_1 = (1, 1, \dots, 1)$, β

For $t = 1, \dots, N$:

1. $D_t(x_0, x_1) = D_t(x_0, x_1) / \sum_{x_0, x_1 \in \mathbf{x}^t \cup \mathbf{x}^d} D_t(x_0, x_1)$
2. Train weak learner with **Learner** using distribution \mathbf{D}_t .
4. Set α_t (Eq.3.1).
5. Update the weight vector:

$$D_{t+1}(x_0, x_1) = \begin{cases} D(x_0, x_1)\beta & \text{if } x_0, x_1 \in \mathbf{x}^d \wedge h_t(x_0) - h_t(x_1) < 0 \\ D(x_0, x_1)e^{\alpha_t(h_t(x_0) - h_t(x_1))} & \text{otherwise} \end{cases}$$

Output $H(x) = \sum_{t=1}^N \alpha_t h_t(x)$

図 3.2: TRankBoost アルゴリズム

手による適合性評価とクリックログの間には弱い相関があることが示されている。これより、クリックログは適合性分布が目標分布と異なるものの、部分的に目標分布の情報を保持していると考えられる。

以上より、本稿における複数情報源を用いたランキング学習の目標は、人手による適合性評価に加えて、クリックログなどの適合性分布が異なる訓練データを用いて、より高精度なランキング学習を実現することである。

3.2.2 転移学習によるランキング学習

はじめに転移学習の説明を行った後に、本稿で提案する転移学習によるランキング学習の枠組みについて述べる。

転移学習とは、目標分布の訓練データが不十分、かつ分布が異なるデータが大量に存在する場合に、それらの訓練データを用いて分類器を学習する問題である [49]。ここで目標分布とは、評価対象となるテストデータの分布である。すなわち、目標分布の訓練データは、テストデータと同じ分布であると仮定する。本稿では、分布が目標分布と異なる訓練データのことを別分布の訓練データと呼ぶ。

X_t を目標分布の事例空間、 X_d を別分布の事例空間とする。 $Y = \{0, 1\}$ を、分類問題におけるクラスラベルとする。テストデータを $S = \{x_i^s\}_{i=1}^k$ (ただし $x_i^s \in X_t$) で表す。 s はテストデータを表す添字である。 X を特徴空間とすると、訓練データ集合 $T \subseteq X \times Y$ は、目標分布の訓練データ T_t と別分布の訓練データ T_d の 2 つに分けられる。目標分布の訓練データは $T_t = \{(x_i^t, y_i^t)\}_{i=1}^n$ (ただし $x_i^t \in X_t, y_i^t \in Y_t$)、別分布の訓練データは、 $T_d = \{(x_j^d, y_j^d)\}_{j=1}^m$ (ただし $x_j^d \in X_d, y_j^d \in Y_d$) である。解くべき問題は、 T_t, T_d と S が与えられた際に、分類器 $\hat{c}: X \rightarrow Y$ を学習することである。

この問題をランキング学習の問題に適用する。 $Q_t = \{q_1^t, q_2^t, \dots, q_n^t\}$ と $Q_d = \{q_1^d, q_2^d, \dots, q_m^d\}$ が与えられたとする。ここで、 Q_t は、目標分布のクエリ集合、 Q_d は別分布のクエリ集合である。各クエリ q_i^t と q_j^d の検索結果である文書集合を、 $\mathbf{d}_i^t = (d_{i,1}^t, d_{i,2}^t, \dots, d_{1,n_i^t}^t)$ と、 $\mathbf{d}_j^d = (d_{j,1}^d, d_{j,2}^d, \dots, d_{1,n_j^d}^d)$ (ただし n_i^t は q_i^t の検索結果に含まれる文書数、 n_j^d は q_j^d に対する文書数を表している) とする。すると訓練データは、 $T_t = \{(q_i^t, \mathbf{d}_i^t, y_i^t)\}_{i=1}^n$ と $T_d = \{(q_j^d, \mathbf{d}_j^d, y_j^d)\}_{j=1}^m$ で表現することができる。ここで $x_i^t = (q_i^t, \mathbf{d}_i^t)$ 、 $x_j^d = (q_j^d, \mathbf{d}_j^d)$ と考えると、ランキング学習の問題は転移学習の問題と見なすことができる。

本稿においては、目標分布の訓練データ T_t は人手による適合性評価、別分布の訓練データ T_d はクリックログを表している。

例えば、Dou らによる方法 [20] のように、クリック頻度を評価点数と見なす場合、検索結果の適合性評価における評価点数は、クリック頻度の異なり数だけ存在することになる。人手による適合性評価は、あらかじめ設定された多段階評価であるため、クリックログから得られた訓練データを活用するためには、適合性評価の尺度の違いをどのように補正するか、という課題が存在する。この課題に対して、ペアワイズ手法のランキング学習に、異なる分布の訓練データから役立つ事例を転用することで分類器の精度向上を目指す事例転移を適用することにより、この問題を解消しつつ、適合性分布が異なる訓練データを用いた転移学習に

よるランキング学習を実現する。

本章での提案の特長は2つ挙げられる。まず、ペアワイズ手法を用いることにより、絶対的な適合性評価を相対的な順序に変換するため、人手による適合性評価から得られた訓練データと、クリックログから得られた訓練データにおける適合性評価のスケールの違いによる悪影響を排除することが可能となる。2つ目に、本稿では転移学習の中から事例転移のアプローチを用いる。別分布の訓練データの中から適切な事例集合を目標分布の学習に用いる事例転移アプローチ [49] を用いることにより、人手による適合性評価による順序ペアと矛盾するような順序ペアを排除しながら学習が可能とした。

3.2.3 TRankBoost

本章では、転移学習に基づくランキング学習を実現するため、ブースティングに基づく転移学習手法である TrAdaBoost[18] を元に、新しいランキング学習手法 TRankBoost を提案する。まず手法の基本となる TrAdaBoost について述べたのちに、ランキング学習へ適用した TRankBoost の詳細を述べる。

TrAdaBoost は、少量の目標分布の訓練データと大量の別分布の訓練データが与えられた際に、事例転移の枠組みで目標分布の事例に対して高い性能を持った2値分類器を学習するブースティング手法 [26] である。ブースティング手法は、重み付き訓練データから問題に対して相対的に性能がよくない弱学習器を多数生成し、最終的に生成された弱学習器の多数決で分類を行う学習手法である。

TrAdaBoost は以下の3つの点で AdaBoost[26] と異なる。1つ目に、TrAdaBoost は目標分布と別分布の事例に対して、異なる重み変更方法を行うことが挙げられる。TrAdaBoost の各試行において、生成された弱学習器によって誤分類された目標分布の事例の重みは AdaBoost と同様に増大させる。しかし、誤分類された別分布の事例については、逆に重みを小さくする。これは、目標分布を予測するモデルを構築する上でノイズとなるような事例の影響を小さくするという解釈をすることができる。2つ目は、最終的な分類器を構築する上で各弱学習器の重みとなる重要度の推定方法に特徴がある。AdaBoost は、全事例の誤り率を元に弱学習器の重要度を決定しているが、TrAdaBoost は、目標分布の事例のみを用いて重要度を決定することである。最後に、生成された弱学習器のうち、全イテレーションの後半において生成された弱学習器のみを用いて分類器を構築する点が異なる。

以上の枠組みにより，TrAdaBoost では目標分布の予測誤差と，別分布の事例による訓練誤差を同時に最小化することを理論的に保証する [18]．しかしながら，TrAdaBoost は目標分布とは異なる別分布の事例に敏感であることが報告されている [18]．これは，誤分類された別分布の事例に対して重みを小さくするため，別分布の事例をうまく分類できなかった場合には，追加された別分布の訓練データによる学習効果があまり得られず，目標分布の訓練データに過学習してしまうためだと考えられる．

3.2.2 節で述べたように，TrAdaBoost の枠組みは容易にペアワイズ手法のランキング学習手法に適用することができる．具体的には RankBoost と同様にランキング学習を実現する．RankBoost では，順序ペアを事例と見なし，全ての順序ペアに対して重みを保持し，各試行において重み付け順序誤差を最小にするような弱学習器を生成する．重みの更新方法は基本的に AdaBoost と同様に行い，最終的なランキング関数は，弱学習器の重み和で実現する．各弱学習器の重みは，それぞれの重要度を表しており，当該弱学習器によって訓練データ中に含まれる順序ペアに対する正解率を元に計算する．

本稿では，TrAdaBoost が別分布の事例の性質に敏感であることに着目し，敢えて理論的妥当性を捨てて，より一般的な転移学習によるランキング学習手法への変更を行う．TrAdaBoost では，目標分布と異なる別分布の事例の影響を排除するために，誤分類された別分布の事例に対して大きく重みの減少を行う．また，生成された弱学習器のうち，全イテレーションの後半において生成された弱学習器しか利用しないため，別分布の訓練データの学習効果が得られていると考えられる前半の弱学習器を捨ててしまう．このため，TrAdaBoost における別分布の訓練データの学習効果が得られない．この問題を防ぐために，アルゴリズムを改善する必要があると考えた．

具体的には，上記に挙げた 3 つの点を調整可能なモデルパラメータとすることで，別分布の訓練データの学習効果が得られない問題の回避を試みる．

- (a) TRankBoost は， N 回の試行で生成された弱学習器のうち， M 試行目から N 試行目の弱学習器を用いて最終的なランキング関数を構築する．すなわち，RankBoost は $M = 1$ ，TrAdaBoost は $M = \lceil N/2 \rceil$ とみなすことができる．
- (b) 別分布の事例の重み更新には，重み全体の更新係数に関わるハイパーパラメータ β を用いる．

- (c) 試行 t で生成された弱学習器の重要度である α_t の計算に目標分布の順序ペア \mathbf{x}^t のみを用いるか、または全ての順序ペアを用いる。

最終的な TRankBoost のアルゴリズムを図 3.2 に示す。各試行において、現在の順序ペアの重みを用いて構築された弱学習器 h_t によって各文書のスコアを出力し、スコアに基づいて順序付けを行う。この際、 h_t の構築には、任意の教師あり機械学習手法を用いることができる。例えば、RankBoost を提案した Freund と Schapire [25] は、重み付け誤り率が最小になるような素性と閾値を選択し、閾値によって $\{0,1\}$ を出力するような WeakLearn アルゴリズムを弱学習器生成に用いている。この際、弱学習器の重み α_t は

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1+r}{1-r} \right), \quad r = \sum_{x_0, x_1} D(x_0, x_1) (h_t(x_1) - h_t(x_0)) \quad (3.1)$$

によって計算される [25]。

本研究では、最初のステップとして、TRankBoost の 2 つの実装を用意した。与えられたデータセットによって、適切なモデルパラメータを設定することは重要な課題ではあるが、本研究ではモデルパラメータは推定せず、固定値を用いる。

1 つ目の実装は、TrAdaBoost と同じモデルパラメータを利用する。すなわち、 $M = \lceil N/2 \rceil$, $\beta = 1/(1 + \sqrt{2 \ln m/N})$ (ただし N は試行回数、 m は別分布の順序ペア総数である) とし、目標分布の順序ペア \mathbf{x}^t のみを用いて α_t を計算する。これ以降、この手法を TRankBoostI (TRB_1) と呼ぶ。

2 つ目の実装は、追加された別分布の訓練データを重要視する。例えばクリックログのように、目標分布と類似性が高いと考えられる訓練データからの学習効果を得るため、より目標分布に近いデータとみなすように設定を行う。具体的には、 $M = 1$, $\beta = 1$ とし、 α_t を計算するために \mathbf{x}^t と \mathbf{x}^d の両方を用いるようにする。そのため、RankBoost に非常に近いアルゴリズムになる。ここで、 $\beta = 1$ は、別分布の順序ペアが誤ってランキングされた際に、重みを変更せずに今の重みを維持することを表している。この設定により、弱学習器が誤ってランキングした別分布の順序ペアの重みを小さくしないため、別分布の訓練データから転移学習できないという TrAdaBoost が持つ問題点を解消し、別分布の訓練データからも学習効果が得られることを期待する。この手法を TRankBoostII (TRB_2) と呼ぶ。RankBoost に目標分布、別分布の訓練データを与えた場合、目標分布、別分布の

	RankBoost	TRankBoost I	TRankBoost II
training data			
parameter update for diff-distribution	—	$\frac{1}{1 + \sqrt{2 \ln m / N}}$	1
how to calculate α_t	use	use	use
Index of weak learners used for ranking function	1 to N	N/2 to N	1 to N

instance in the target-distribution
 instance in the diff-distribution

図 3.3: RankBoost, TRankBoostI, TRankBoostII の比較

区別なく誤ってランキングした順序ペアの重みを大きくするため、この点において TRB_2 は RankBoost と異なる。

RankBoost (RB), TRB_2, TRB_2 の違いを図 3.3 に示す。3 つの手法について、上から順番に比較している。目標分布と別分布の訓練データを用いた場合の訓練データの利用方法、別分布に対する重み変更パラメータ β の値、弱学習器に対する重要度の計算方法、最終的に利用する弱学習器の個数を表している。

3.3 評価

提案手法の有効性を検証するため、提案手法と既存のランキング学習手法との比較を行った。ベースラインとしては、ペアワイズ手法である RankingSVM (RSVM), RankBoost (RB) を用いた。RSVM の実装には、`svm_rank`¹ を用いた。RSVM のカーネルは線形カーネルを用いた。RB は文献 [25] にしたがって、C++ で実装した。RB と TRB の弱学習器には WeakLearn アルゴリズム [25] を用いた。

¹http://www.cs.cornell.edu/People/tj/svm_light/svm_rank.html

3.3.1 実験条件

データセット

ランキング学習の有効性を評価するため，商用モバイルウェブ検索エンジンのインデクスを元に3つのデータセットを作成した．

1つ目は人手による適合性評価データセット (human) である．あらかじめ用意された49クエリそれぞれについて得られた約300件の検索結果に対して，3人の評価者によって4段階の評価を付与した．この際，評価不能という判定が付与された文書は取り除いた．humanは，ランキング学習の評価にも用いられるため，実験では5分割交差検定を行った．49クエリを5分割し，3ブロックを訓練データ，1ブロックをパラメータの設定に用いる検証データ，残りの1ブロックを評価に用いるテストデータに利用した．

2つ目はクリックログデータセット (click) である．同じ検索エンジンの3ヶ月分のクリックログの中から5,000クエリを選択した．テストデータを学習してしまうことを防ぐため，humanに含まれるクエリは含まない．文献[20]にしたがい，クリック情報をクエリ-文書ペアのクリック頻度という情報に集約し，これを適合性評価とする訓練データを作成した．

3つ目は，humanとclickを単純に混合²したデータセット (both) である．bothについてもhumanと同様の方法で5分割交差検定を用いて評価を行った．

ベースライン手法については，これら3つの訓練データ (human, click, both) を用いて評価を行った．提案手法は，ふたつの異なる訓練データを同時に利用するため，humanとclickを用いた．

データセットの概要を表3.1に示す．#queryは，クエリ数，#doc/#queryはクエリあたりの文書数，#pairは，訓練データから生成される順序ペアの数を表している．各クエリ-文書ペアについて，12個の素性を抽出した．抽出した素性を表3.2に示す．

²この場合には，特徴空間を一致させる必要があり，本実験では全く等しい特徴ベクトルを用いた．

表 3.1: データセットの概要

dataset	#query	#doc/#query	#pair
human	49	331.9	1,192,707
click	5,000	21.8	704,188

表 3.2: 実験に用いた素性集合

Feature	Description
BM25	Okapi BM25 score [53]
BM25_log	log of Okapi BM25 score
within_site_inlink	inlink number from inside the site
between_site_inlink	inlink number from outside the site
refer_num	inlink number
refer_num_log	log of inlink number
PageRank	PageRank score [4]
URL_length	URL length
URL_slash_num	slash number in URL
title_length	title length
query_in_title	query is in title
is_index	is index page

パラメータ選択

RB と TRB は試行回数のパラメータ N , RSVM はモデルの複雑さと訓練誤差のトレードオフを表すパラメータ c を持っている . 実験では , それぞれのパラメータについて , $N \in \{10, 20, 30, 40, 50\}$, $c \in \{0.01, 0.05, 0.1, 0.5, 1.0\}$ の中から検証用データにおいて誤差を最小にする値を選択した .

評価指標

人手による適合性評価データに対するランキング学習の有効性を検証するため , 情報検索において評価指標として広く用いられている Normalized Discounted Cumulative Gain [33] (NDCG) を用いた . NDCG は検索結果上位 k 位において , 理想的なランキングへの近さを表す評価指標と解釈することができる . k 位における NDCG の値は ,

$$\text{NDCG}_q@k = Z_q \left(\text{rel}_1 + \sum_{i=2}^k \frac{\text{rel}_i}{\log_2 i} \right) \quad (3.2)$$

によって求められる . rel_i は , i 位になった文書の評価点数を表しており , Z_q は , クエリ q に対する理想的なランキングにおいて NDCG の値が 1 となるように設定された正規化項である .

モバイルや PC ウェブ検索エンジンでは , 検索結果を 5 件または 10 件表示するものが一般的であるため , 評価実験では $k = 5, 10$ を用いて手法の有効性の検証を行った . また , それぞれの手法の特徴を把握するため , 各手法における $k = 1, 2, \dots, 50$ の NDCG 値の比較も行った .

3.3.2 結果

表 3.3 に , 3 つのデータセットを用いた 2 つのベースライン手法と提案手法の NDCG@5 , 10 の値を示す . 表 3.3 より , 以下の結果を確認した .

- 全ての手法において TRB.2 が NDCG@5 , 10 において最良の結果を示した .
- TRB.1 は , RSVM_human , RB_human を上回る精度を示したものの , その他の手法に比べて低い NDCG 値を示した .

表 3.3: 実験結果

method	NDCG@5	NDCG@10
RSVM_human	.6161	.6186
RSVM_click	.7060	.6956
RSVM_both	.7072	.7075
RB_human	.6407	.6385
RB_click	.7104	.7188
RB_both	.6957	.7129
TRB_1	.6789	.6642
TRB_2	.7223	.7217

- RSVM_human と RSVM_click , RB_human と RB_click を比較すると , 同じランキング学習手法においては , click が human よりも高い値を示した .
- RSVM においては RSVM_both が最大の値を示しており , RB においては RB_both が RB_click に比べて低い値を示した .

t 検定により , NDCG@5 , 10 の平均の差を評価したところ , TRB_2 と RSVM_human の間に有意差が確認された (p 値 < 0.05) . しかしながら , それ以外の組み合わせについては有意差が見られなかった .

また , 図 3.4 に NDCG@ k ($k = 1, 2, \dots, 50$) の結果を示す . 図 3.4 より , 以下の結果を確認した .

- TRB_2 は , ベースライン手法の click に対して NDCG@ k ($k = 4, \dots, 50$) において上回る精度 , both に対して NDCG@ k ($k = 3, \dots, 50$) において上回る精度を示した .
- click を用いたベースライン手法は , 検索結果上位 ($k = 1, 2, 3$) における NDCG 値が高く , それ以降で急激に値が小さくなっていることが確認できた .

3.4 考察

本章では , 3.3.2 節から得られた結果の考察を行う . 本実験において TRB_1 が低い精度を示した理由については 3.4.1 節で述べる .

表2の実験結果より、TRB_2は、NDCG@5、10における全ての手法において最大の値を示していることから、TRB_2の有効性を検証することができた。ここで、TRB_2は単純な混合であるbothに比べて高い精度のランキング学習を実現していることから、目標分布の訓練データが少数であることに起因する過学習を防ぎ、より適切に別分布の訓練データを活用して汎化性能の高いランキング関数を生成していることがわかる。また、NDCG@5、NDCG@10以外のNDCG値を比べると、TRB_2は、NDCG@1,2,3においてRSVM.clickとRB.clickに劣る以外は、他のすべての手法に対してより高い値を示しており、ランキング下位の精度向上にも効果があることがわかる。以上より、適合性分布が異なるデータセットを用いる提案手法が有効に働いていることがわかる。

bothの結果より、単純に混合した訓練データを用いて従来のランキング学習手法では、精度を向上することが難しいことが示唆された。RSVMにおいてわずかに精度が向上している理由としては、clickが単純な混合でもある程度過学習を抑えることが可能なほど、分布が類似していることが考えられる。RBに関しては、単純な混合でclickを用いた場合に比べて低い精度を示しており、手法によって傾向が異なることがわかる。

ベースライン手法において、単体の訓練データを用いた方法では、clickがhumanに比べて高い精度を示し、Douら[20]による実験と同様の傾向が得られた。これは、人手による適合性評価によって得られた訓練データが不足しているため、humanでは過学習を起こしているものだと考えられる。本実験においては5分割交差検定を用いたため、訓練データにはおよそ30クエリ分の訓練データしか含まれない。この結果より、訓練データに含まれるクエリ数が汎化性能に影響を与えていることがわかる。また、clickがhumanより高い精度を示したことにより、本データセットにおいて、クリックログから得られた適合性分布は、目標分布である人手による適合性分布に類似していることが伺える。

図3.4の結果から、clickがNDCG@1,2,3においてhumanに比べて高い精度を示す理由としては、クリックという行動が検索結果上位を中心に行われていることが挙げられる。このため、クリックログから得られる訓練データの適合性評価が、検索結果上位における順序の差を明確に分けるような訓練データとなっており、NDCG上位の評価値に反映されたと考えられる。

以上より、適合性分布が異なる複数情報源を訓練データとして用いる場合には、

従来手法では複数情報源の訓練データを適切に用いたランキング学習の実現が困難であることが示唆され、提案手法の枠組みによって高精度なランキング学習が実現可能なことを検証した。

3.4.1 TRankBoostI の性能が低い原因

TRB₁ が低い精度を示した原因は2つ考えられる。1つ目に、利用する弱学習器の数が影響していると考えられる。TRB₁ の各試行における α_t の値を眺めたところ、試行 t に関してほぼ単調に減少していることを確認した。TRB₁ では、試行全体の後半の弱学習器のみを用いてランキング関数を実現するため、効果的な弱学習器を捨ててしまっていると考えた。しかしながら、実際に TRB₁ の M のパラメータを1に変更した手法の評価を行ったところ、大きな変化は見られなかった。これより、TRB₁ の性能については M 以外の影響が大きいことが推測できる。

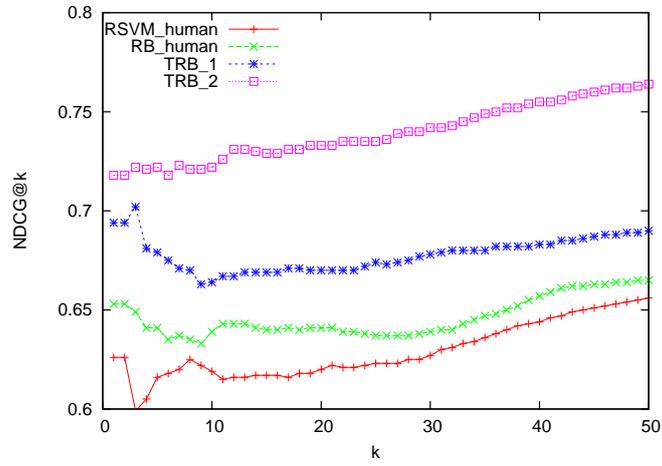
そこで2つ目の原因として、誤った別分布の順序ペアに対する重み変更のパラメータ β の値が大きいということが推測される。TrAdaBoost では、誤ってランキングした順序ペアについて、一定の割合で重みを小さくする。各試行において別分布の順序ペアが一定の割合で誤ると仮定すると、試行を繰り返すごとに別分布の訓練データの影響がなくなり、最終的には目標分布の訓練データが弱学習器を生成する際の支配的な要因となる。更に TrAdaBoost では後半の弱学習器のみを用いるため、目標分布の訓練データに対して過学習した弱学習器をより多く利用することになる。そのため、試行回数を適切に設定しなければ、先述した過学習の問題を解決できない。human よりも高い精度を示しているものの、both に劣るのはこのためだと考えられる。TRB₂ では $\beta = 1$ かつ $M = 1$ に設定することで、別分布の訓練データに対してより適合するようなランキング関数を生成する。評価実験において最良の結果を得ていることから、人手による適合性評価の過学習することなく、より高い汎化性能を持ったランキング関数を生成していることがわかる。

3.5 まとめ

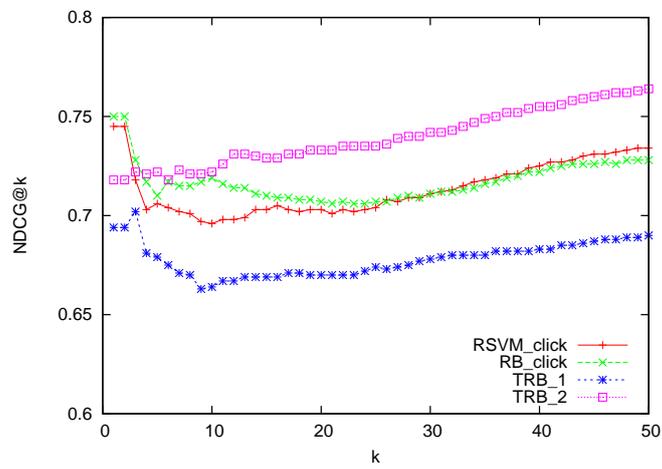
本章では、転移学習の枠組みを用いることにより、適合性分布が異なる訓練データを用いてランキング関数を生成するランキング学習手法の枠組みを提案した。具

体的には、学習に役立つ事例を選択的に用いる事例転移の方法とペアワイズ手法を組み合わせることによって転移学習によるランキング学習を実現する。本章では、この枠組みを用いた新しいランキング学習手法である TRankBoost を提案した。

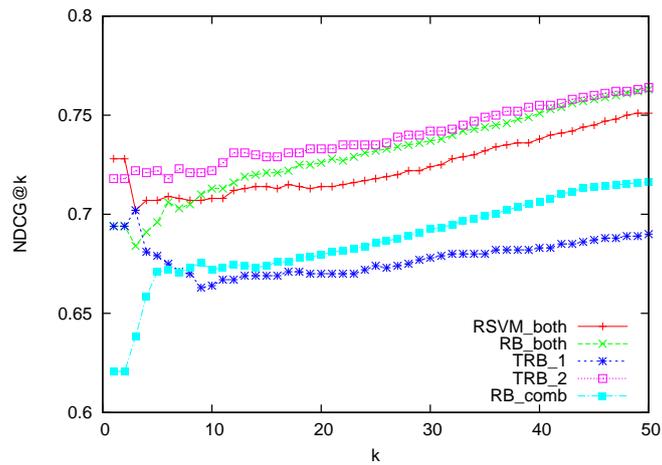
RankingSVM や RankBoost のような従来のランキング学習手法では、教師あり機械学習の枠組みを用いるため、文書ペアが同一の分布から抽出されるという仮定を置いているため、適合性分布が異なる訓練データを単純に混合するだけでは、従来法によって適切にランキング学習ができないと考え、検証を行った。商用モバイル検索エンジンのインデクスとクリックログを元に作成されたデータセットを用いた評価実験を通じて、提案手法 TRankBoostII によって、NDCG@5, 10 の値でベースライン法を上回る精度でランキング学習が可能であることを確認した。また、ベースライン手法については、単純な混合では精度を向上することが困難であることが示唆された。これにより、提案手法を用いることにより、従来手法では達成が困難であった目標分布と適合性分布が異なる訓練データを同時に用いて、より高精度なランキング学習が実現可能なことを示した。



(a)



(b)



(c)

図 3.4: NDCG@ k 値の比較結果: (a) 提案手法と human , (b) 提案手法と click , (c) 提案手法と both

第4章 評価指標を考慮した選択的ペアワイズ手法によるオンラインランキング学習

4.1 概要

本章では (1) 大規模データに対して学習可能であること, (2) 高速かつ逐次学習可能であること, の2つの要請がある. この観点から, オンライン学習を用いたランキング学習に焦点を当てる. 1つ目の理由については, Amazon Mechanical Turks¹⁾をはじめとするクラウドソーシングサービスの普及に伴って大規模の人手評価データが利用可能になった背景がある. 機械学習の枠組みにおいては大規模な訓練データを用いた方がより高精度な予測を達成可能であることが知られているため, より大規模な訓練データを利用可能であることは大きな利点となる. 2つ目の理由については, ユーザのリアルタイムフィードバックを用いてランキング関数を調整する状況が近年注目を集めている背景がある [19, 32, 47]. こうした状況を受けて, 高速かつ逐次学習可能なランキング学習アルゴリズムを用いることができれば, ユーザの嗜好に素早く適応することが可能となる.

オンラインランキング学習の既存手法は, パーセプトロンに基づく方法 [17, 22], 文書ペアをランダムサンプリングして重み更新を行う確率的勾配法ベースの方法 [55] など, いくつか提案されている. 既存のオンラインランキング学習の利点は, 文書ペアに基づく損失関数を用いるため, 重み更新処理を高速に行うことができる点にある.

本稿では既存のオンラインペアワイズ手法を拡張し, 選択的ペアワイズアプローチ (*selective pairwise approach*) を提案する. 選択的ペアワイズ手法では, 高精度なモデル生成が可能なリストワイズアプローチの長所と低い計算コストであるペ

¹<https://www.mturk.com/mturk/>

アワイズアプローチの長所を併せ持つ。本稿では、情報検索分野で広く用いられる NDCG の値を近似的に最適化するオンラインランキング学習手法である PARank-NDCG を提案する。PARank-NDCG では、文書ペアに対して優先度を与えるため、NDCG に基づくマージンサイズを設定する。最大損失を与える文書ペアを選択して Passive-Aggressive (PA) アルゴリズム [16] に基づいて重みベクトル更新を行う。PA をベースアルゴリズムとして用いる理由は (1) マージンサイズという形式で優先度を導入可能であること、(2) 重み更新式が解析的に解けるため、与えられた文書ペアに対する重み更新処理が 1 つの処理で可能である、の 2 つが挙げられる。また、本稿では訓練データに含まれるノイズの影響を排除するため ramp loss の導入を検討する。ペアワイズアプローチにおいては、1 つの文書に対して適合度判定の誤りが存在すると組み合わせ効果で誤りが伝播してしまうため、訓練データに含まれるノイズに影響されやすいことが知られている [11]。ロバストなランキング関数の生成は、ランキング学習の課題のひとつ [11] であり、これを解決することによって PARank-NDCG のさらなる精度向上を目指す。

本章では、大規模なベンチマークデータセットである MSLR-WEB データセットを用いた実験を行い、Stochastic Pairwise Descent [55] と Committee Perceptron [22] を含むベースライン手法の評価を通じて、提案手法の有効性を検証した。

本章の主要な貢献は以下のとおりである:

- 情報検索評価指標に基づいて事例選択の優先度を設定し、優先度の観点で最大損失を与えるペアを選択して更新を行う選択的ペアワイズアプローチを提案する。
- ペアの要素の 1 つはもう一方の要素とは異なる適合度を持つようなペアそれぞれに対して、評価指標を考慮してペアごとにマージンを設定し、設定されたマージンを鑑みた際に、最大の損失を与えるペアを線形対数時間で探索可能なオンラインランキング学習アルゴリズム PARank-NDCG を開発する。
- 訓練データに含まれるノイズの影響を排除するため、提案手法に ramp loss を導入し、ランキング学習タスクにおける ramp loss の有効性を検証する。

本章の構成を以下のとおりである: 4.2 において提案手法である PARank-NDCG の詳細とアルゴリズムについて述べる。4.3 において評価実験の結果を述べ、4.4 で考察を行い、4.5 でまとめる。

4.2 PARank-NDCG

本稿では、文書ペアに対して異なる重要度を付与し、最大損失を与える文書ペアを選択して重み更新を行うオンラインランキング学習アルゴリズムを開発する。本稿ではNDCGを最大化する評価指標とし、これを近似的に最大化するため、提案手法をPARank-NDCG (Passive-Aggressive Rank based on NDCG) と呼ぶ。PARank-NDCGでは、各クエリにおける各文書ペアに対して異なるマージンサイズに基づく損失関数を設定し、それに基づいて学習を行う。既存のペアワイズ手法では適合度の差によらず、文書ペアに対して等しいマージンサイズが設定されていたことに相当する。PARank-NDCGにおいて文書ペアに応じて異なるマージンサイズを設定することによって、ペアワイズ手法においても目標とする評価指標を最大化することを目指すためである。たとえば、既存のペアワイズ手法においては、適合度 y_a である文書 a と適合度 y_b である文書 b で構成される文書ペア $(y_a, y_b) = (5, 1)$ の誤りと文書ペア $(y_a, y_b) = (2, 1)$ の誤りが等しい損失を持つことを想定している。しかしながら、NDCGのような多段階の適合度を考慮する情報検索指標においては、これは不適切である。

本節では最初に Passive-Aggressive アルゴリズム (PA) [16] をランキング学習に適用する方法を述べる。次にNDCGをマージンサイズに反映する方法を説明し、最大損失に基づく更新方法とその効率的な計算方法について述べる。次に Ramp loss について述べ、最後に PARank-NDCG アルゴリズムを解説する。

4.2.1 PA に基づくランキング学習

提案手法のベースとなる PA を用いてペアワイズランキング学習の実現方法について述べる。まず PA について説明を行ったのちに、ランキング学習への適用について述べる。PA では t 回目の試行において重みベクトル \mathbf{w}_t の更新を以下の最適化問題として定式化する:

$$\mathbf{w}_{t+1} = \operatorname{argmin}_{\mathbf{w} \in \mathbb{R}^n} \frac{1}{2} \|\mathbf{w} - \mathbf{w}_t\|^2 \quad \text{s.t.} \quad \ell_t(\mathbf{w}_t; \mathbf{x}_t, y_t) = 0. \quad (4.1)$$

ここで ℓ_t は以下の hinge loss を用いる:

$$\ell_t(\mathbf{w}_t; \mathbf{x}_t, y_t) = \begin{cases} 0 & y_t(\mathbf{w}_t \cdot \mathbf{x}_t) \geq 1 \\ 1 - y_t(\mathbf{w}_t \cdot \mathbf{x}_t) & \text{otherwise.} \end{cases} \quad (4.2)$$

ここで $y_t(\mathbf{w}_t \cdot \mathbf{x}_t)$ がマージンサイズである 1 よりも大きければ, $\ell_t = 0$ となり, 重みの更新は行わない. このようにマージンサイズとは損失を 0 にするために必要とされる, 当該事例と超平面の距離と解釈することができる. $\ell_t > 0$ 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 (4.1) 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, 以下のように閉じた式で \mathbf{w}_{t+1} を表すことができる:

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \tau_t y_t \mathbf{x}_t \quad \text{ただし} \quad \tau_t = \frac{\ell_t}{\|\mathbf{x}_t\|^2}. \quad (4.3)$$

ここで, \mathbf{w}_{t+1} は, 入力 \mathbf{x}_t に対して正しく分類するような重みになっている. このように, 選択されたペアに対して一度の更新で誤りを訂正するような更新を行う.

訓練データに対する制約の誤りを許容するためにスラック変数 ξ を導入した場合も, 同様に閉じた式で重みベクトルの更新式を表すことが可能である. 一次のスラック変数 ξ を用いる手法は PA-I と呼ばれ [16], aggressiveness parameter と呼ばれるパラメータ C を用いて,

$$\tau_t = \min \left\{ C, \frac{\ell_t}{\|\mathbf{x}_t\|^2} \right\} \quad (4.4)$$

を用いて更新する. PA では損失が大きいほどパラメータ更新の値が大きくなるが, この更新に対して上限 C で抑えていることになる.

与えられた順序ペアの特徴表現を $\mathbf{x} = \mathbf{x}_a - \mathbf{x}_b$ とし, 適合性評価をラベル $y = \text{sign}(y_a - y_b)$ とすることで, 式 (4.2) にしたがって損失を計算し, 式 (4.3) によって重みを更新することで, PA を用いたランキング学習を行うことができる.

4.2.2 NDCG に基づくマージン設定

本稿では, 各適合度のペアに対して NDCG を考慮して異なるマージンサイズを設定する方法を提案する. PA をそのままランキング学習に適用する場合, すべて

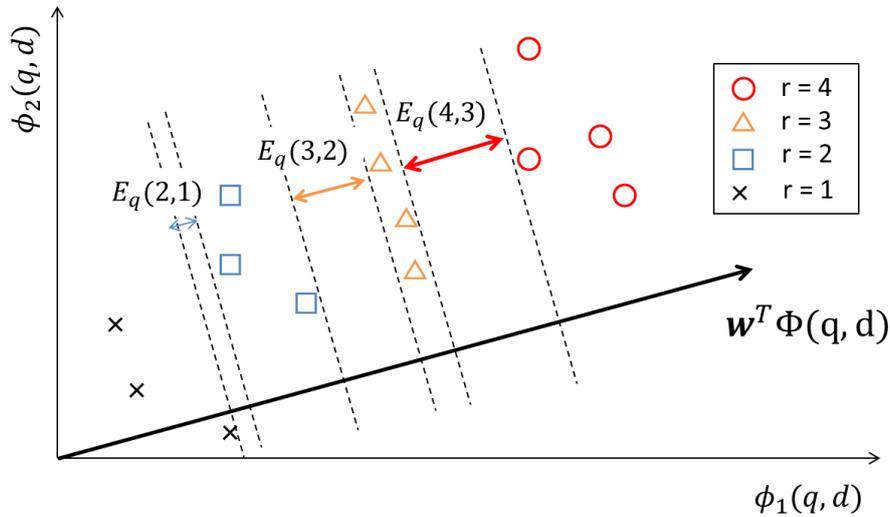


図 4.1: 異なる適合度間のマージンサイズ.

の文書ペアに対して等しいマージンサイズ (すなわち 1) を想定して重みの更新を行う。異なるサンプルに対して異なるマージンサイズを設定する方法はコスト考慮型学習でもしばしば用いられている方法である [56]。検索評価指標の観点で文書ペアに対してマージンサイズを設定することにより、提案手法の精度向上を目指す。

式 (4.2) で利用されている固定値のマージンの代わりに、クエリ q と適合度のペア (y_a, y_b) を入力とし、これらの適合度ペアの文書を誤ってランキングした場合の評価値の減少値に基づく適応型マージンサイズ $E_q(y_a, y_b)$ を利用する。そのため、提案手法では以下の損失関数を用いる:

$$\ell_t^{hinge}(\mathbf{w}_t; \mathbf{x}_a, \mathbf{x}_b, y_a, y_b, q) = \begin{cases} 0 & \text{if } E_q(y_a, y_b) \leq \mathbf{w}_t \cdot \mathbf{x}_t \\ E_q(y_a, y_b) - \mathbf{w}_t \cdot \mathbf{x}_t & \text{otherwise.} \end{cases} \quad (4.5)$$

異なる適合度ペアに対して異なるマージンサイズを用意する方法の概略図を図 4.1 に示す。この例ではクエリに対する検索結果について 4 段階の適合度評価が付与されている。適合度ペアに対して異なるサイズのマージンが設定されている。提案手法では、このマージンサイズを NDCG のような評価指標を反映するように適切なサイズに設定することを目指す。

$E_q(y_a, y_b)$ の計算方法について述べる。クエリ q における適合度 y_a と y_b (ここで $y_a > y_b$ とする) に対するマージンサイズ $E_q(y_a, y_b)$ は、適合度が y_a である文書と

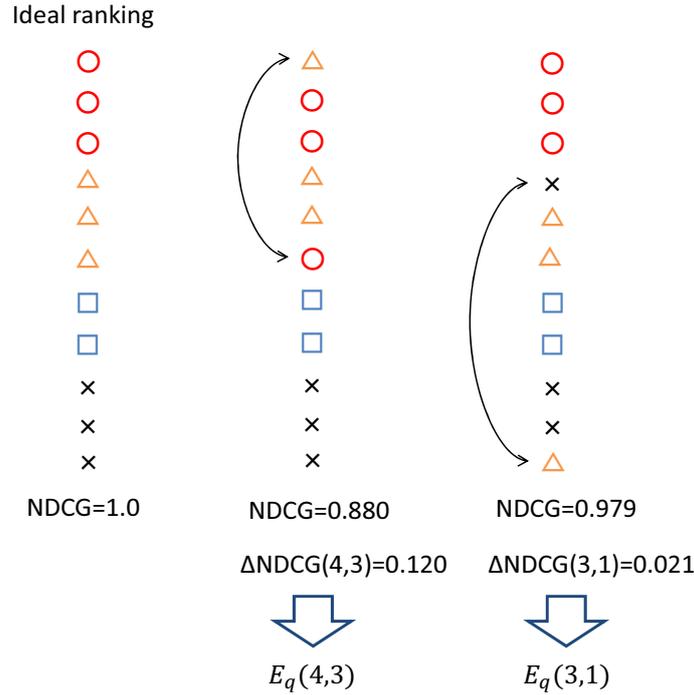


図 4.2: マージンの計算方法

適合度が y_b である文書を交換した際の NDCG の減少値にもとづいて計算する。

クエリ q において適合度が $(4, 3, 2, 1)$ である文書がそれぞれ $(3, 3, 2, 3)$ 件ずつ存在する例を考える。この例では、適合度 4 と適合度 3 の文書の交換パターンは合計 $3 \cdot 3 = 9$ パターン存在する。

提案手法では、理想のランキングから文書ペアを交換した際に NDCG の値が最も減少する組み合わせのみを考える。すなわち、この例では適合度 4 の最上位の文書と、適合度 3 の最下位の文書を交換した際の NDCG の減少値を利用する。この値を $\Delta\text{NDCG}_q(4, 3)$ と表現する。この例では、文書の位置を交換した際の NDCG の値は 0.88 となり、 $\Delta\text{NDCG}_q(4, 3) = 1 - 0.88 = 0.12$ である。一般化すると、 ΔNDCG は以下の計算式で求めることができる:

$$\Delta\text{NDCG}_q(y_a, y_b) = 1 - \text{NDCG}_q(y_a, y_b),$$

$$E_q(y_a, y_b) = E_{\text{scale}} \Delta\text{NDCG}_q(y_a, y_b).$$

ここでスケールパラメータ E_{scale} は最小のマージンを 1 にするように設定する。

訓練データに含まれる適合度の分布はクエリによって異なることが知られている

ため [45], 提案手法ではマージンサイズをクエリごとに設定する. そのため, マージンサイズ $E_q(y_a, y_b)$ はクエリ q に含まれる適合度のペアに対して用意される. このマージン設定方法を Δ NDCG と呼ぶ.

4.2.3 最大損失更新と高速選択手法

本稿で提案する選択的ペアワイズアプローチは, NDCG の値に影響を与える文書ペアを優先的に選択する機能を持つ. このような機能を実現するために, 従来の PA でも利用されている最大損失更新 (max-loss update) 戦略 [16] を導入する. Stochastic Pairwise Descent (SPD) [55] においては文書ペアのランダムサンプリングが利用されていたのに対し, max-loss update はクエリに対応する文書集合内の文書ペアにおいて, 適合度ペア毎に設定されたマージンサイズを考慮して損失 ℓ_t が最大になる文書ペアを選択し, それを用いて PA と同様の方法で重みを更新する. これは, 最大損失を与える文書ペアが正しくランキングされるように修正することは, 全体の損失を最小化するための効率的な方法であるという直感にもとづいている. 損失を計算するために式 (4.5) を用いているため, NDCG を最も減少させるように誤った文書ペアを最大損失ペアとして選択する. 選択された文書ペアを正しくランキングするように, PA に基づいて重みベクトルの更新を行う.

クエリ q の検索結果に含まれる文書数を n_q とすると, クエリの検索結果に含まれる全文書の組み合わせの数は $O(n_q^2)$ である. 最悪の場合には, 最大損失を与えるペアを選択するために, 単純な方法で $O(n_q^2)$ の計算コストがかかる. 提案手法のマージン設定方法によって, 文書数に対して線形対数時間のコストで厳密な最大損失ペアを発見可能であることを示す. マージンは文書ペアそのものではなく, 2つの適合度に対して設定されるため, 適合度の組み合わせが同じ文書ペアについては同じマージンサイズが用意される. ここでは, 上位の適合度を r_h , 下位の適合度を r_l ($r_l < r_h$) とする. 各文書ペアに対する損失関数は $w_t \cdot x_t$ の値によって異なる. この値は $w_t \cdot x_a - w_t \cdot x_b$ に分解することができる. そのため, ある適合度ペアにおける最大損失ペア (あるクエリにおける max-loss pair ではなく, あるクエリの適合度ペアにおける最大損失を与えるペアであるため, local max-loss pair と呼ぶ) は容易に発見することが可能である. 具体的には, 上位の適合度 (すなわち $y_a = r_h$) における $w_t \cdot x_a$ が最小の文書と, 下位の適合度 (すなわち $y_b = r_l$) における $w_t \cdot x_b$ が最大となる文書の組み合わせを選択すればよい. クエリ q 全体

における最大損失ペアはすべての適合度の組み合わせにおける local max-loss pair 集合に含まれている．この local max-loss pair 集合の大きさは適合度のペア数合計に等しい．

この方法の計算コストについて述べる．特徴ベクトルの次元数を m とし， R が適合度が取りうる値の総数とする．まず，現在の重みベクトルと各文書の特徴ベクトルの内積に基づいて検索スコアを計算する．計算時間は文書数に線形である．すなわち， $O(n_q \cdot m)$ ．各適合度について，最大のスコアと最小のスコアの文書がどれだあるかという情報が必要であり，その計算コストは $O(n_q)$ である．適合度が最上位の文書集合については最大スコアの文書，適合度が最下位の文書集合については最小スコアの文書を考慮する必要がないことに注意されたい．そのため， $\frac{1}{2}R(R-1)$ 組の文書ペアを考慮すればよい．その結果，全体の計算コストは $O(n_q \cdot m + n_q + \frac{1}{2}R(R-1))$ となる．

データセットに含まれる適合度順位の数はたいてい3から5であり，最大でも10程度であるため，この $\frac{1}{2}R(R-1)$ は $n_q \cdot m$ に対して定数と見なすことができる．そのため，PARank-NDCG における最大損失ペアの選択処理は文書数に対して線形時間で発見可能である．これにより，PARank-NDCG がスケールし，大規模なデータセットにおいても実行可能なことを示唆している．

評価実験で用いる MSLR-WEB10K データセットを紹介する．このデータセットでは1クエリあたり平均120文書(5,264文書ペア)存在し，5段階の適合度評価が付与されている．単純な方法では最大5,264回の損失計算が必要なのに対し，PARank-NDCG ではたった10回の損失を計算すれば厳密な最大損失ペアを見つけることができる．

4.2.4 Ramp loss を用いたロバストな学習

ランキング学習タスクにおいては，訓練データに含まれるノイズの影響によって，生成されるランキング関数の性能が低下することが知られている [11, 62]．我々は選択的ペアワイズアプローチを採用しているため，PARank-NDCG も同様にノイズの影響を受けると予想される．提案手法によって，誤って適合度判定された文書などノイズとなる文書を含む文書ペアが選択されることで，それに基づいて更新された重みベクトルによる最終的な出力の品質が低下すると考えられる．

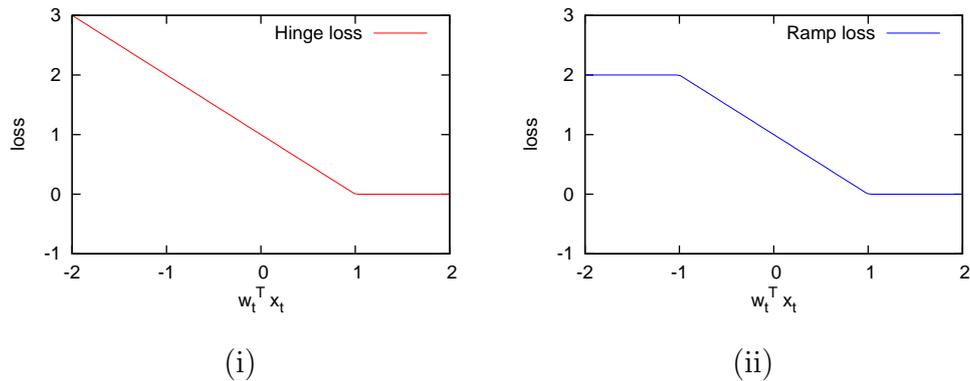


図 4.3: 損失関数: (i) hinge loss, (ii) ramp loss.

この問題を解決するために、損失関数に ramp loss [15] を導入することを提案する。Ramp loss は hinge loss において損失の値が一定の値を超えるサンプルについては一定の損失を与える損失関数である。Ramp loss は元々SVMの学習においてサポートベクタを減らすために提案されたものであるが、ノイズを含むデータに対してロバストな学習を実現できる損失関数としても知られている [15, 61]。

Hinge loss と ramp loss の違いを図 4.3 に示す。この図では、 x 軸は $w_t \cdot x_t$ を表し、 y 軸は損失の値を表す。この例では、それぞれの損失関数のマージンサイズは 1 に設定されている。スコアがマージンサイズ以上の値、すなわち $w_t \cdot x_t \geq 1$ である場合には、いずれの損失において損失の値は 0 となる。Hinge loss においては分類が誤った場合に、損失の値が分離超平面からの距離に比例して大きくなる。対照的に、ramp loss は分離超平面から一定の距離（この例では $w_t \cdot x_t \leq -1$ ）を超えると一定の損失を与えている。そのため、ramp loss は大きく誤った文書ペアなどに対して過剰に大きな損失を出力することを防ぐ効果がある。

Ramp loss を用いる場合においてもマージンサイズに $E_q(y_a, y_b)$ を利用し、損失関数が一定になる閾値として定数 (-1) を採用する。そのため、 t 試行目における重みベクトル w_t を用いて、適合度が y_a である文書 a 、適合度が y_b である文書 b の

損失は以下のとおり計算できる:

$$\ell_t^{ramp}(\mathbf{w}_t; \mathbf{x}_a, \mathbf{x}_b, y_a, y_b, q) = \begin{cases} 0 & \text{if } E_q(y_a, y_b) \leq \mathbf{w}_t \cdot \mathbf{x}_t \\ E_q(y_a, y_b) - \mathbf{w}_t \cdot \mathbf{x}_t & \text{if } -1 < \mathbf{w}_t \cdot \mathbf{x}_t < E_q(y_a, y_b) \\ 1 + E_q(y_a, y_b) & \text{if } \mathbf{w}_t \cdot \mathbf{x}_t \leq -1. \end{cases} \quad (4.6)$$

文献 [61] により, 更新式は

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \tau_t \mathbf{x}_t, \text{ where } \tau_t = \begin{cases} \min \left\{ C, \frac{\ell_t^{ramp}}{\|\mathbf{x}_t\|^2} \right\} & \text{if } -1 < \mathbf{w}_t \cdot \mathbf{x}_t < E_q(y_a, y_b) \\ 0 & \text{otherwise.} \end{cases} \quad (4.7)$$

となる.

Ramp loss を用いる場合には, max-loss update は $-1 < \mathbf{w}_t \cdot \mathbf{x}_t < E_q(y_a, y_b)$ を満たす最大損失を与える文書ペアを選択する. これは $\mathbf{w}_t \cdot \mathbf{x}_t \leq -1$, or $E_q(y_a, y_b) \leq \mathbf{w}_t \cdot \mathbf{x}_t$ を満たす最大損失を与える文書ペアが重み更新に影響を与えないためである. このため, 最大損失ペアの高速選択手法が適切に働かない可能性がある. たとえば, スコア $\mathbf{w}_t \cdot \mathbf{x}_t$ を与える文書ペア (\mathbf{x}_a と \mathbf{x}_b) が選択されたと仮定する. ここで文書 \mathbf{x}_a は適合度 y_a において最小スコア $\mathbf{w}_t \cdot \mathbf{x}_a$ であり, 文書 \mathbf{x}_b は適合度 y_b において最大スコア $\mathbf{w}_t \cdot \mathbf{x}_b$ であるとする. もし $\mathbf{w}_t \cdot (\mathbf{x}_a - \mathbf{x}_b)$ が $-1 < \mathbf{w}_t \cdot (\mathbf{x}_a - \mathbf{x}_b) < E_q(y_a, y_b)$ を満たさない場合には, 重み更新が行われなため, ramp loss に基づく更新を行うためには無駄な選択となってしまう.

Ramp loss においても高速選択手法を利用可能にするためには, 各適合度の文書群をスコア $\mathbf{w}_t \cdot \mathbf{x}$ に基づいてソートをすればよい. これにより, $\mathbf{w}_t \cdot \mathbf{x}_t \leq -1$ を満たす最大損失ペアを簡単に見つけることができる. 具体的には, クエリ q の検索結果に含まれる文書集合のうち, i 番目の適合度の文書数が $n_{q,i}$ とすると (すなわち $\sum_i n_{q,i} = n_q$ である), ソートのコストは $\sum_{i=1}^R n_{q,i} \log n_{q,i}$ である. $\sum_{i=1}^R n_{q,i} \log n_{q,i} < n_q \log n_q$ より, これは $O(n_q \log n_q)$ である. 全体の計算コストは $O(n_q \cdot m + n_q \log n_q + \frac{1}{2} R(R-1))$ となり, $n_q \cdot (m + \log n_q)$ に対して $\frac{1}{2} R(R-1)$ を定数と見なすことができる. よって, ramp loss を用いる PARank-NDCG の最大損失ペア選択は文書数に対して線

形対数時間のアルゴリズムといえる。

同じ適合度である文書集合が $w_t \cdot x_a$ ($a = 1, \dots, n_q$) によってソートされているため、この処理では、 $w_t \cdot x_t < E_q(y_a, y_b)$ を満たす文書ペアが見つかるまで、文書ペアの交換と $w_t \cdot x_a - w_t \cdot x_b$ の評価を繰り返し行うことになる。最悪の場合 $O(n_q)$ の計算コストが必要となるが、たいていの場合には処理の早い段階で $w_t \cdot x_t < E_q(y_a, y_b)$ を満足する文書ペアを見つけることができる。そのため、すべての文書ペアについて損失を計算し、 $w_t \cdot x_t < E_q(y_a, y_b)$ を満たしつつ最大損失を与える文書ペアを発見する単純な方法に比べて、高速選択手法は十分に速いといえる。4.3.3において、実データを用いたモデルの学習にかかった計算時間に基づいて、高速選択手法の効率性を検証する。

4.2.5 アルゴリズム

PARank-NDCG のアルゴリズムを図 4.4 に示す。PARank-NDCG は訓練データ D 、繰り返し回数 T 、トレードオフパラメータ C を入力として受け取る。まず、訓練データに含まれるすべてのクエリについて $E_q(y_a, y_b)$ を計算する。次に選択されたクエリについて重みベクトルを更新する。この更新フェーズにおいては、最大損失を与える文書ペアを選択し、重み更新を行う。なお、

$$Z = \{(x_a, x_b, y_a, y_b) \mid x_a, x_b \in X, y_a, y_b \in Y, y_a > y_b\}.$$

これはクエリ q において $y_a > y_b$ であるような文書ペア集合を表す。上記の処理をすべてのクエリについて行い、 T 回繰り返しを行う。ここで $|Q|$ は訓練データに含まれるクエリ数を表す。最後に、すべての重みベクトルの平均を出力とする。これは、Averaged Perceptron[14] で用いられている方法である。

4.3 評価

本節ではベースライン手法との比較実験を通じて、PARank-NDCG の有効性を検証する。まず、大規模データセットを用いた実験を通じてオンラインランキング学習の既存手法に対する優位性を検証する。次に、学習曲線と学習時間の比較実験を通じて PARank-NDCG の学習効率のよさの評価を行う。

表 4.1: MSLR-WEB10K と MSLR-WEB30K における実験結果 .

(a) MSLR-WEB10K					
	NDCG				
	@1	@2	@3	@4	@5
PARank-NDCG	0.3182	0.3211	0.3276	0.3330	0.3380
SPD-SVM	0.2871	0.2947	0.3039	0.3104	0.3161
SPD-PA	0.2830	0.2987	0.3061	0.3131	0.3193
ComP	0.2409	0.2515	0.2620	0.2697	0.2760
RSVM	0.2990	0.3118	0.3214	0.3281	0.3339
	NDCG				
	@6	@7	@8	@9	@10
PARank-NDCG	0.3423	0.3471	0.3503	0.3541	0.3572
SPD-SVM	0.3216	0.3267	0.3310	0.3350	0.3384
SPD-PA	0.3246	0.3297	0.3338	0.3379	0.3415
ComP	0.2820	0.2876	0.2923	0.2969	0.3008
RSVM	0.3390	0.3442	0.3483	0.3527	0.3564
(b) MSLR-WEB30K					
	NDCG				
	@1	@2	@3	@4	@5
PARank-NDCG	0.3208	0.3247	0.3304	0.3357	0.3408
SPD-SVM	0.2845	0.2929	0.2997	0.3061	0.3123
SPD-PA	0.2869	0.2997	0.3075	0.3147	0.3283
ComP	0.2461	0.2585	0.2674	0.2752	0.2814
RSVM	0.3008	0.3118	0.3212	0.3283	0.3344
	NDCG				
	@6	@7	@8	@9	@10
PARank-NDCG	0.3448	0.3490	0.3529	0.3567	0.3598
SPD-SVM	0.3173	0.3218	0.3262	0.3304	0.3339
SPD-PA	0.3344	0.3397	0.3445	0.3491	0.3533
ComP	0.2869	0.2922	0.2971	0.3014	0.3056
RSVM	0.3397	0.3445	0.3491	0.3533	0.3571

Algorithm PARank-NDCG algorithm

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

- 1: Calculate $E_q(y_a, y_b)$ for all queries q in Q .
 - 2: $\mathbf{w}_0 \leftarrow \mathbf{0}$
 - 3: **FOR** $i = 1$ to T
 - 4: **FOR** all queries q in Q
 - 5: $(\mathbf{x}_a, \mathbf{x}_b, y_a, y_b) = \operatorname{argmax}_{(\mathbf{x}_a, \mathbf{x}_b, y_a, y_b) \in \mathbf{Z}} \ell_t^{\text{ramp}}(\mathbf{w}_t; \mathbf{x}_a, \mathbf{x}_b, y_a, y_b, q)$
 - 6: $\mathbf{x}_t = \mathbf{x}_a - \mathbf{x}_b$
 - 7: $\tau_t = \min \left\{ C, \frac{\ell_t^{\text{ramp}}}{\|\mathbf{x}_t\|^2} \right\}$
 - 8: $\mathbf{w}_{t+1} = \mathbf{w}_t + \tau_t \mathbf{x}_t$
 - 9: **ENDFOR**
 - 10: **ENDFOR**
 - 11: $\mathbf{w}^* = \frac{1}{T|Q|} \sum_{t=1}^{T|Q|} \mathbf{w}_t$
 - 12: **RETURN** \mathbf{w}^*
-

図 4.4: PARank-NDCG アルゴリズム .

表 4.2: データセットの概要 .

dataset	#query	#doc/#query	#feature
MSLR-WEB10K	10,000	120.0	136
MSLR-WEB30K	30,000	125.7	136

4.3.1 データセット

評価実験は Microsoft によって公開されているランキング学習用データセットである Microsoft Learning to Rank (MSLR-WEB)² を用いて行った . MSLR-WEB にはデータセットのサイズによって MSLR-WEB10K と MSLR-WEB30K が存在する . 評価実験でこのデータセットを利用したのは以下の 2 つの理由である: (1) 十分に大規模であること , (2) ウェブ検索エンジンに基づいて作成されたデータセットであるため実用性評価の観点で適していること .

データセットの概要を表 4.2 に示す . #query はデータセットに含まれる総クエリ数 , #doc/#query はクエリの検索結果に含まれる平均文書数であり , #feature が特徴次元数を表している . MSLR-WEB に含まれているすべてのクエリ・文書

²<http://research.microsoft.com/en-us/projects/mslr/>

ペアには、人手によって5段階の適合性判定が付与されている。データセットに添付されている評価用ツールにおいてNDCG計算のための適合度のとりうる値は(15, 7, 3, 1, 0)となっている。MSLR-WEBデータセットの各特徴の詳細はウェブで公開されている³。たとえば、クエリに含まれる単語がタイトルに出現する数、本文に出現する数、BM25スコアや言語モデルに基づくスコアなどである。特徴値は線形スケールによって[0, 1]に正規化を行った。

4.3.2 MSLR-WEBを用いた評価

オンラインランキング学習のベースライン手法として、Committee Perceptron (ComP) [22], Stochastic Pairwise Descent (SPD) [55]を用いた。ComPはC++で実装し、SPDの実装にはsofia-ml⁴を用いた。ComPについてはweighted averaging strategy [22]を採用した。ComPについては、検証セットを用いてNDCGの観点で各仮説の重み平均を用いる戦略を用いた⁵。また、SPDの学習手法としてはPegasos-SVM (SPD-SVM) と PA-I (SPD-PA) を選択し、サンプリング方法には予備実験においてquery-norm-rankよりも高い性能を示したrankを選択した。バッチ手法のランキング学習としてRSVMを用いた。RSVMの実装にはsvm_rank⁶を用いた。

MSLR-WEB10K と MSLR-WEB30K は既に交差検定のために5分割されており、各foldにおいてラベルありデータが訓練データ、検証データ、評価データの3つに分割されている。本章の実験においても、この分割をそのまま利用し、5分割交差検定を用いて評価を行った。PARankにおける T の値はMSLR-WEB10K、MSLR-WEB30K両方のデータセットにおいて10に設定した。各foldにおいては、MSLR-WEB10Kにおいては約6,000クエリ、MSLR-WEB30Kにおいては約18,000クエリが含まれているため、この設定によってそれぞれ60,000、180,000試行となる。ComP、SPD-SVM、SPD-PAにおける試行回数は文献[55]に従い、MSLR-WEB10Kにおいては100,000、MSLR-WEB30Kにおいては300,000とした。PARank-NDCG、SPD-SVM、SPD-PA、RSVMにおけるトレードオフパラメータ C は{0.0001, 0.001, 0.01, 0.1, 1.0, 10.0}の中から検証データにおいて、

³<http://research.microsoft.com/en-us/projects/mslr/feature.aspx>

⁴<http://code.google.com/p/sofia-ml/>

⁵文献[22]においてはBordaFuseを用いる方法も提案されているが、BordaFuseの場合にはすべての仮説を保持しておく必要があるため、本稿では仮説の重み平均を用いる戦略を選択した。

⁶http://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html

NDCG@10 が最大となる値を選択した．ComP における Committee の数は {10, 30, 50, 100} の中から検証データにおいて NDCG@10 が最大となる値を選択した．

本実験においては，各手法について NDCG@1 ~ NDCG@10 で評価を行い， t 検定によって各手法の平均の差を評価した．実験結果を表 4.1 に示す．各評価指標について，最大の値を示した手法の結果をボールド体で表示している．表 4.1 より，以下の結果を確認した．

- MSLR-WEB10K については，PARank-NDCG が SPD-SVM , SPD-PA , ComP に対して NDCG@1 ~ NDCG@10 において有意に高い値を示し ($p < 0.001$) , RSVM に対しては NDCG@1 ~ NDCG@4 ($p < 0.01$) , NDCG@5 ~ NDCG@7 ($p < 0.05$) において有意に高い値を示した．
- MSLR-WEB30K については，PARank-NDCG が SPD-SVM , SPD-PA , Comp , RSVM に対して NDCG@1 ~ NDCG@10 ($p < 0.001$) において有意に高い値を示した．

この結果より，PARank-NDCG を用いることで従来のオンラインランキング学習手法にとどまらず，バッチ学習を行うペアワイズ手法である RSVM に比べても高精度なランキングモデル生成が可能であることがわかった．

4.3.3 学習効率性の比較

PARank-NDCG によって効率よく学習できることを示すため，PARank-NDCG , SPD-PA , ComP の学習曲線を比較した．本評価では，MSLR-WEB10K の Fold1 をデータとして用いた．訓練データを用いてモデルを構築し，テストデータにおける NDCG@10 の値で評価を行った．各手法のパラメータ選択には，検証データにおいて NDCG@10 が最大の値を示すものを選択した．その結果，PARank-NDCG の $C = 10$, SPD-PA の $C = 0.01$, ComP の committee 数は 10 と設定した．ここで PARank-NDCG においては各試行で 1 クエリに関わる文書ペアの情報が与えられることに注意されたい．PARank-NDCG は与えられた文書ペア集合の中から最大損失を与える文書ペアを選択し，重みベクトルの更新に用いる．そのため重みベクトルの更新回数という観点では他の 2 手法と同じであり，本実験は公平であると考えた．

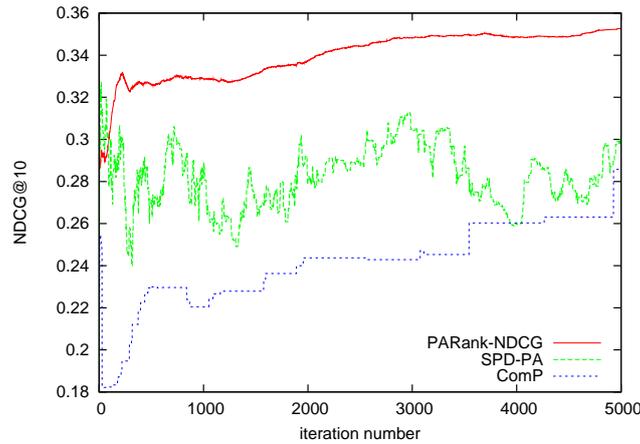


図 4.5: MSLR-WEB10K データセットの Fold1 における学習曲線 .

サンプリングごとのテストデータに対する NDCG@10 の遷移を図 4.5 に示す . この結果より , SPD-PA と ComP に対して PARank-NDCG を用いることで急速に精度向上していることがわかる . また , PARank-NDCG は約 300 という少ないイテレーション回数で , 安定した精度を示している . PARank-NDCG は与えられた訓練データについてクエリ単位で学習対象の文書ペアを選択するため , イテレーション回数が 300 であることから 300 クエリを学習に利用したことがわかる . MSLR-WEB10K において 300 クエリは全訓練データの 1/20 であり , PARank-NDCG においては学習を途中停止しても最終的な結果にそれほど影響を与えないといえる .

また , 図 4.5 より PARank-NDCG における 3,000 試行の学習結果が , SPD-PA における 100,000 試行の学習結果と同等の性能を示していることがわかる . この結果より , PARank-NDCG は与えられた訓練データを一度だけ利用して学習するワンパスアルゴリズムとしても用いることができるといえる . ワンパスアルゴリズムとして利用できることから , 訓練データが逐次的に与えられ , 学習に利用した訓練データを廃棄するストリーミング学習の状況においても PARank-NDCG が有用であることが示唆された . この場合 , PARank-NDCG は全訓練データを保持する必要がなく , 対象のクエリに含まれる文書ペアに関わる情報のみを保持しておけばよい . また , メモリ効率の観点からも有用である .

PARank-NDCG の学習にかかる計算時間について評価を行った . 本章では既に PARank-NDCG をワンパスアルゴリズムとして用いることによって , SPD-PA と ComP よりも高い精度を示すことを確認したため , PARank-NDCG においては訓練データに含まれるクエリ数だけ試行を行った . 計算時間は MSLR-WEB10K の

表 4.3: MSLR-WEB10K データセット (Fold1-5) における平均訓練時間 .

method	training time [sec.]
PARank-NDCG	6.94
PARank-NDCG_naive	44.73
SPD-SVM/SPD-PA	1.16
ComP	37.06
RSVM	442.49

Fold1-5 に含まれる訓練データに対する学習時間で計測した . 本章で提案した高速選択手法の有効性を確認するため , PARank-NDCG において高速選択手法を利用せず , 各クエリについてすべての文書ペアの損失を計算して最大損失を与えるペアを選択する手法 (PARank-NDCG_naive) も比較した . この際 , PARank-NDCG_naive は内積のキャッシュを利用することによって , $w_t \cdot x_t$ の冗長な計算を省くことができることに注意されたい . 本実験では PARank-NDCG における評価指標を考慮したマージンサイズ計算処理も学習時間に含めた .

結果を表 4.3 に示す . 表 4.3 より , PARank-NDCG は , ComP や RSVM に比べて著しく高速に学習可能であることを確認した . また , 高速選択手法を用いることによって学習時間は 44.73 秒から 6.94 秒へと減少した . なお本実験において PARank-NDCG におけるマージンサイズ計算処理の平均時間は 5.7 秒であった . これより , 高速選択手法を用いた PARank-NDCG は , 文書ペアのランダムサンプリングで事例選択する SPD に匹敵する速度で重みベクトル更新処理を実行し , かつ , SPD-SVM や SPD-PA より高精度に学習可能であることがいえる .

4.4 考察

MSLR-WEB に対する実験結果より , PARank-NDCG を用いることで SPD-SVM , SPD-PA , RSVM に比べて高精度なランキングを実現可能であることを示した . 特に NDCG@ k の k が大きい状況に比べて , 小さい状況においてベースライン手法に対する差が大きかったことから , PARank-NDCG においては , 検索結果上位に焦点をあてた学習を実現していることがわかる . 本節では , PARank-NDCG が他手法に対して高い性能を示す原因を (1) 評価指標に基づくマージン設定 , (2) Ramp loss の利用の 2 つの観点から分析を行う .

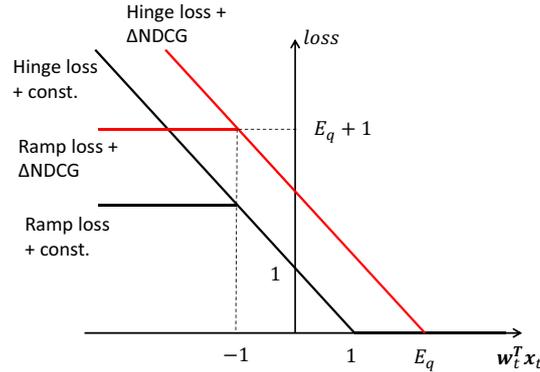


図 4.6: 損失関数の比較 .

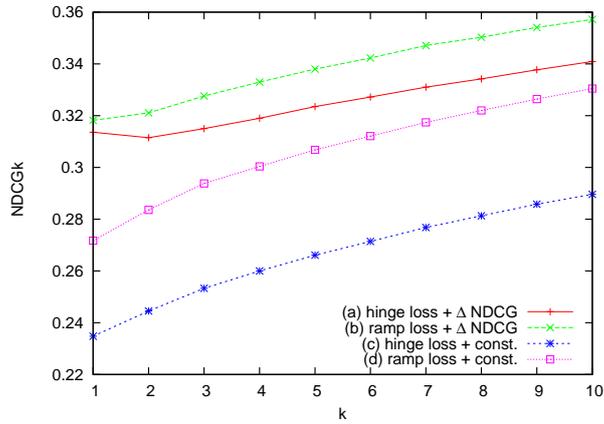
4.4.1 マージンサイズ導入の効果

Δ NDCG によるマージンサイズ導入の効果について検証を行う . Δ NDCG マージン (図 4.7 中 Δ NDCG) と一定マージン (図 4.7 中 const.) の比較を行う . ここで一定マージンは SPD-PA のような従来手法で用いられている方法である . 有効性を明確にするため , マージンサイズ設定方法の違いを hinge loss と ramp loss を用いた両方の場合で検証を行う . これらの実験条件毎の損失関数を図示したものを図 4.6 に示す . その他の実験条件は 4.3.2 と同様に設定した .

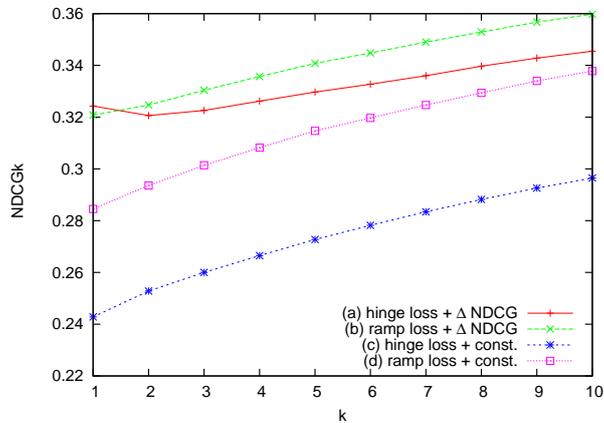
4 つの比較手法の NDCG@1-10 の実験結果を図 4.7 に示す . 図 4.7 より , MSLR-WEB10K , MSLR-WEB30K 両方のデータセットにおいて損失関数によらずマージンサイズの設定に Δ NDCG を用いた手法が , 一定マージンを用いる手法に比べて高い精度を示した ($p < 0.001$) . これにより , 本章で提案したマージン設定方法により , ランキング精度向上が可能であることを示した .

4.4.2 Ramp loss 導入の効果

Ramp loss 導入の効果について検証を行う . 図 4.7 の (a) と (b) の比較を通じて , Ramp loss を用いた (b) が , Hinge loss を用いた (a) に比べて NDCG@2-10 において MSLR-WEB10K , MSLR-WEB30K 両方のデータセットにおいて高い値を示した ($p < 0.001$) . また , 図 4.7 の (c) と (d) の比較を通じて Ramp loss が一定マージンを用いる場合においても精度向上していることを確認した ($p < 0.001$) . これより , 損失関数に Ramp loss を用いることで PARank-NDCG によるランキング精度向上が可能であることと , Ramp loss がランキング学習において有効であることを確



(i)



(ii)

図 4.7: マージンサイズ設定の比較 (i) MSLR-WEB10K データセット, (ii) MSLR-WEB30K データセット.

認した.

4.5 まとめ

本章では, ランキング評価指標を考慮した損失関数の観点から, 損失が最大となる文書ペアを選択する選択的ペアワイズアプローチを提案した. また, NDCG の近似値に基づいて適合度ペア毎に異なるマージンサイズを設定するオンラインランキング学習 PARank-NDCG を開発した. PARank-NDCG は NDCG の減少値に基づいて設定された損失関数において, 最も優先度が高い文書ペアを選択する最大損失更新戦略を採用し, 訓練データに含まれるノイズの影響を排除するため, Ramp loss を損失関数に採用している.

MSLR-WEB10K と MSLR-WEB30K による大規模データセットの実験を通じて、オンラインランキング学習の従来手法である Stochastic Pairwise Descent , Committee Perceptron , バッチ学習である RankingSVM に比べて、PARank-NDCG によって高精度なランキングが可能であることを示した。効率性の分析を通じて、高速選択手法を用いた PARank-NDCG の重み更新処理は、ランダムサンプリングに基づく方法と同等の速度であり、かつ PARank-NDCG によってランダムサンプリングに基づく手法である SPD-PA , SPD-SVM に比べて NDCG@1 ~ NDCG@10 の値でより高い精度を示した。

上記に加えてランキング学習における Ramp loss の有効性を示したことが本研究の貢献である。著者の知る限り、本研究がランキング学習に Ramp loss を採用した初めての研究である。

第5章 順序ラベルを用いた教師あり 自己組織化マップ

5.1 概要

近年、情報検索の分野で教師あり学習の枠組みを用いて最適なランキングモデルを構築する技術が盛んに研究されている [43]。教師あり学習を用いたランキング問題においては、あらかじめ評価ラベルが付与された訓練事例集合をもとに予測モデルを構築し、未知の事例集合に対して評価ラベルと整合する順位を予測する。この際、評価ラベルとしては順位そのものや、順位グループを表すラベルなどを用いる。なお、本問題においては真の順位そのものを予測する必要はなく、真の順位と整合的な順位を予測すればよい。

教師あり学習の枠組みにおいて、事例ペア間の順序が教師信号として与えられたとき、事例空間内の順位関数を推定し、予測の場合には事例に対してその順位を予測する問題を考える。たとえば情報検索においては、ウェブ検索エンジンのクリックログをもとに、入力クエリに対して文書の相対的な優先順位が得られる。そして、これに対してSVMを用いて教師あり学習の枠組みで予測器を生成する研究がある [34]。本章における問題設定では、教師信号として各事例の順位そのものではなく、事例ペアに対する順序が与えられると想定する。本章では以下、順序を表す教師信号を順序ラベルと呼ぶことにする。ここで順序には絶対値が付与されないことに注意されたい。すなわち、たとえば2段階上位である、3段階下位であるといった順位に対する差の情報を一切持たない。

情報検索の分野では順位ラベル作成のために、多値の適合度を被験者が付与する方法が一般的に用いられている [3]。しかしながら、各被験者によって基準が異なったり、長い期間で安定した基準を用いるのが難しいため、各事例に整合的に適合度を付与するのは難しい。一方、2つの事例が与えられた場合にその相対関係を教師信号として付与することは容易であり、正確なアノテーションが可能であ

ることが知られている [48] . 近年 , Amazon Mechanical Turk¹ のようなクラウドソーシングサービスの普及に伴い , 難易度の低いアノテーション作業を大量に行うコストが低くなっている . このように , 実用面を考えても全事例に順位を与えるのではなく , 相対的な順位を表す順序ラベルが教師ラベルとして与えられる状況が自然である .

人工ニューラルネットワークの一種である自己組織化マップ (Self-Organizing Maps; SOM) [40] は , 入力データの次元削減 , クラスタリング , データ可視化など , 幅広い分野で利用されている . SOM は入力データに正解ラベルを必要としない教師なし学習の一種であるが , SOM の教師あり学習への拡張も提案されている [40, 24, 30, 54] . 訓練データに含まれる予測ベクトルのラベルを特徴ベクトルに含めるように拡張し , 従来の SOM と同じ枠組みで更新処理を行うことで SOM を教師あり学習として利用可能である .

本稿では , SOM を順序学習に拡張することを検討する . しかしながら , 従来の教師あり SOM の枠組みでは , 教師ラベルを明示的に特徴ベクトルに埋め込む必要があり , 2 つの特徴ベクトルとその相対的な順位 (順序ラベル) が与えられる問題においては , そのままの適用ができないという課題がある . そこで本稿では順序ラベルという情報を用いて , 入力データの特徴ベクトルに対して最もよく一致したユニット (Best Matching Unit; BMU) を修正したのちに重みベクトルの更新処理を行う順序型 SOM を提案する . 順序型 SOM は事例ペア間の順序情報から , 事例集合に対する全順序の情報を復元することを可能とする .

人工データと実データを用いた評価実験を通じて , 本稿で提案する順序型 SOM の有効性を検証し , 相対的な順位関係が教師ラベルとして与えられる状況において順序型 SOM が適切に順位を学習することを確認する .

本章における貢献は以下のとおりである :

- 事例ペアとその順序を訓練データとして用いて学習することで入力事例に対して順位を予測する順序型 SOM を提案する .
- 人工データと実データを用いた評価実験を通じて順序型 SOM の有効性を検証し , 特に低次元特徴空間で複雑な分布をしている状況において , 順序型 SOM が有効に働くことを示す .

¹<https://www.mturk.com/mturk/>

本章の構成は以下のとおりである．5.2 で従来の SOM，順序学習について述べたのちに 5.3 で提案手法とアルゴリズムについて述べる．5.4 で提案手法の有効性を検証するための評価実験について述べ，5.5 で実験結果から得られた知見について考察を行い，最後に 5.6 でまとめる．

5.2 自己組織化マップと順序学習

本章が対象とする順序学習の概要について述べたのち，既存の SOM とその特長と順序学習への適用妥当性について述べる．

5.2.1 順序学習

本章で扱う順序学習においては，あらかじめ M 次元の特徴ベクトルで表現された訓練事例集合 $\{\mathbf{x}^{(i)}\}_{i=1}^N$ ($\mathbf{x} \in \mathcal{R}^M$) と事例の添え字ペアとそれらに対する順序情報から構成される順序ラベル $\{i, j, y\}$ ($i, j \in \{1, \dots, N\}, i \neq j, y \in \{+1, -1\}$) を用いて予測モデルを構築し，同様に M 次元ベクトルで表現され，順位が未知であるテスト事例集合 $\{\mathbf{x}^{(i)}\}_{i=N+1}^L$ ($\mathbf{x} \in \mathcal{R}^M$) の順位を予測する．この際，順序ラベルには訓練事例集合に含まれる 2 つの事例に対してペアの第一要素の事例が相対的に上位である ($y = +1$) か，相対的に下位である ($y = -1$) というラベルが付与されているものとする．

このような枠組みにおいて，たとえば，順位関係が線形であれば線形識別器を用い，二値分類の教師あり学習の枠組みを用いて予測器を構築することが可能である．具体的には，順序ラベルが与えられた事例集合について，事例ペアの特徴ベクトルの差分からひとつの事例を生成し， $+1$ または -1 のラベルを付与する．すなわち， $\mathbf{x}^{(i,j)} \equiv \mathbf{x}^{(i)} - \mathbf{x}^{(j)}$ を順序ラベル i, j, y に対応する事例の特徴ベクトルとし， y をクラスラベルとして用いることで，既存の二値分類アルゴリズムを用いればよい．情報検索の分野では教師あり学習である SVM をこのように順序学習に適用した Ranking SVM (RSVM) [34] が用いられる．

5.2.2 自己組織化マップ

SOM は入力層と競合層から成る人工ニューラルネットワークの一種である [40] . 競合層は 1 次元鎖状や 2 次元格子のようにあらかじめ決められた構造でノードとエッジを配置する . 入力層は SOM が扱う事例集合が持つ特徴ベクトルと同次元のノード数から成り , 各ノードから競合層の全てのノードに対してエッジが張られている . 競合層を中心に考えると , 競合層の各ノードはそれぞれ事例が持つ特徴ベクトルと同次元の重みベクトルを持っていると解釈することができる .

ラベルなし事例集合を用いた SOM の訓練について述べる . 事例が入力されると , 競合層の中から事例が持つ特徴ベクトルに最も近い重みベクトルを持つノード (BMU) がひとつ選択される . この距離計算には L2 距離が一般的に用いられる . BMU が選択されると , 当該ノードが持つ重みベクトルを入力事例が持つ特徴ベクトルに近づけるための更新処理を行う . t 試行目の入力事例を $\mathbf{x}^{(t)}$, 選択された BMU に対応する重みベクトルを $\mathbf{w}^{(t)}$ とし , 更新後の重みベクトルを $\mathbf{w}^{(t+1)}$ とすると ,

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + \theta(t)\alpha(t)(\mathbf{x}^{(t)} - \mathbf{w}^{(t)}) \quad (5.1)$$

という規則により更新する . ここで $\alpha(t)$ は学習率を表す . また , $\theta(t)$ は近傍関数であり , BMU の近傍に対して減衰した更新を行う際の減衰項であり , いずれも試行 t の増加に応じて減衰する関数を用いる .

SOM においては , 入力事例に対して各重みベクトルとの距離を計算し , 最近傍の重みベクトルに対応するノードにデータを配置することによって高次元特徴ベクトルを持つデータを 2 次元に配置することが可能となる . このように SOM は教師なし学習としてしばしば利用される .

教師あり SOM の場合には , 入力事例の特徴次元数に加えて事例に与えられたラベル y を特徴ベクトルに加えて SOM と同様の更新処理を行う . これにより , 予測の際に与えられた入力事例の BMU が持つ , ラベルに相当する重みパラメータを予測として出力することによってラベルの推定を実現する . しかしながら , 今回の問題設定では相対的な順位情報のみが与えられるため , 教師情報として順位そのものを利用できず , 教師あり SOM の枠組みそのものが利用できない .

SOM の長所として , 入力事例の特徴空間における分布の形によらず分布の形に合わせた多様体の学習が可能であることが挙げられる . たとえば , 5.4.1 節で利用

する人工データのように特徴空間において順位を線形モデルで再現できない場合において、適切に SOM の訓練を行うことができれば、ノードに対応する重みベクトルを各順位の分布付近に配置することが可能だと考えられる。このため、相対的な順序ラベルをどのように SOM の学習に反映し、その結果を予測にどのように利用するかという点が課題となる。

5.3 順序型自己組織化マップ

従来の教師なし SOM においても、1次元鎖状の SOM のノード番号を用いてデータに対する順位づけの解釈が可能である [40]。しかしながら、教師なし SOM では訓練データが持つ順序に関する情報を扱うことができないため、そのままでは順序の学習が一般に困難である。また、教師あり SOM の枠組みと RSVM における事例ペアの特徴ベクトルの差分を利用するアプローチの組み合わせを考えると、差分ベクトルを利用して学習を行うため、予測の際に与えられた事例に対する BMU 選択の妥当性が失われ、適切な予測ができない。このため、従来の教師あり SOM では事例ペアに対して順序ラベルが与えられるという本稿の問題設定において学習と予測を行うことができない。

SOM の特長を活かしつつ順序学習を実現するため、本稿では SOM の拡張を試みる。具体的には順序ラベルと事例ペアが与えられた際に、それらの事例に対応する BMU のノード位置が順位を表すと解釈し、その位置が与えられた順序ラベルと整合性が取れていない場合には、更新対象のノード位置を修正してから重みベクトルの更新処理を行う順序型 SOM (OrderSOM) を提案する。

OrderSOM の特徴を以下にまとめる：

- OrderSOM においては、1次元または多次元に配置された競合層のうち、事前に定めた 1次元を用いて順位を表現する。この次元を順位次元と呼ぶ。
- 更新処理においては、与えられた順序ペアの各事例の BMU をそれぞれ取得し、順位次元における BMU の位置情報を比較する。順序ラベルと整合性が取れていない場合には、更新対象のノード位置を修正し、入力事例の特徴ベクトルを用いて重みベクトルの更新を行う。

本章ではこれより 5.3.1 節で OrderSOM の概要について述べ、5.3.2 節で 1 次元 OrderSOM、5.3.3 節で多次元 OrderSOM について述べる。

5.3.1 OrderSOM の概要

OrderSOM においては、SOM の競合層のある 1 次元を順位次元として利用することで、入力事例に対する BMU のノード位置に基づいて全順序出力が可能となる。そのため、ある 1 次元について鎖状に配置された競合層を持つ SOM であれば、順序の学習と予測を行うことができる。本稿においては説明のため、1 次元鎖状と 2 次元格子状の競合層の例で説明する。

OrderSOM の更新と予測方法について図 5.1 を用いて説明する。ここでは競合層に 1 次元鎖状の 5 つのノードを持つ SOM の例を表している。この例では競合層の構造は 1 次元で表現されているため、ノードの番号そのものを順位として解釈する。本稿ではノード番号が大きい方が高い順位を表すものとする。

各試行において、順序ラベルと共に 2 つの事例の特徴ベクトルが与えられる。ここで x_{l_i} と x_{l_j} を 2 つの事例に対応する特徴ベクトルとし、 x_{l_i} が x_{l_j} に比べて相対的に高い順位である (すなわち $y = 1$) という情報が与えられているとする。図の例では、 x_{l_i} に対して 2 番目のノード、 x_{l_j} に対して 4 番目のノードが BMU として選択されている。すなわち、この場合には x_{l_i} に対応する BMU の添え字番号は x_{l_j} に対応する BMU の添え字番号に比べて小さいため、順位ラベルを鑑みると望ましい状態ではない。そこで、従来の SOM と同様にそのまま BMU に対して更新処理を行うのではなく、適切な BMU を修正するという処理を行う。それぞれのノード番号について x_{l_i} についてはより上位ノードになるよう、 x_{l_j} についてはより下位ノードになるように変更する。この例では、3 ノード分の修正を行い、 x_{l_i} に対応する BMU を 5 番目のノード、 x_{l_j} に対応する BMU を 1 番目のノードに変更している。そして、変更された BMU を用いて重みベクトルの更新処理を行う。この試行を繰り返すことによって順序ラベルのみから正しい順序を満たすような SOM を実現する。

図 5.1 では、順序関係に誤りがあった場合にノード番号を 2 つ修正する例を示している。この例では説明のため、修正幅を 2 と設定しているが、真の適合順位数に対して競合層のノード数が小さい場合やデータセットの性質においては、学習を繰り返しても順序誤りの発生を解消できず、重みベクトルの更新が収束しない場

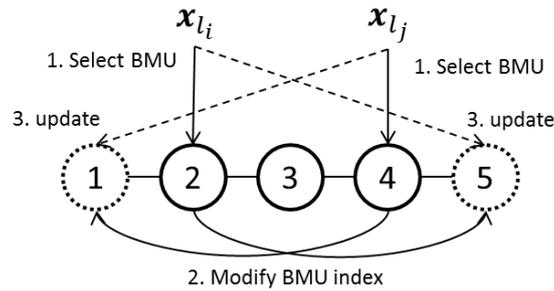


図 5.1: 1次元 OrderSOM の更新アルゴリズムの概要．順序ラベルと現在の OrderSOM による順位が整合していない場合．

合が考えられる．そのような場合においては，更新幅を小さく設定することで更新後の重みベクトルの値を安定させることができると考えられる．そのため，本稿では順序誤りに基づく更新幅を最小の 1 と設定し，学習の試行回数を増やすことで上記問題を可能な限り解消することを目指す．

また，OrderSOM では，順序ラベルを用いた上記更新処理に加えて，ラベルなしデータを用いて従来の SOM と同様の更新処理を行う半教師あり学習の適用も可能であるが，本稿ではラベルありデータのみを利用する教師あり学習の OrderSOM のみを対象とする．

5.3.2 1次元 OrderSOM

ノードを 1次元鎖状に並べた 1次元 OrderSOM (1d-OrderSOM) について述べる．図 5.1 の例と同様に N 個のノードを鎖状に用意する．

1d-OrderSOM の学習処理のアルゴリズムを図 5.2 に示す．1d-OrderSOM は入力として，事例集合 D とそれらの添え字組と順序ラベルである訓練ラベル集合 L ，アルゴリズムの試行回数 T ，ノード数 N と修正幅 s を入力パラメータとして利用する．先述のとおり，本稿の OrderSOM では順序ラベル付きデータのみを用いて更新を行うため，事例集合 D に含まれる事例のペアに対するラベルが訓練ラベル集合 L に含まれていることを想定していることに注意されたい．

まず， N 個のノードから成る鎖状 SOM を構築する．そして，各ノードに対応する重みベクトル w をランダムに初期化し，全ノードの重みベクトルを保持した重み行列 $W = (w_1, \dots, w_N)^T$ を構築する (ステップ 1)．各試行においては，ランダムに順序ペアを取得する (ステップ 3)．取得された順序ペアにおける 1 つ目の事例

の添え字を l_i , 2つ目の事例の添え字を l_j とし, 順序ラベルを l_y とする. 事例集合 D から順序ペアに含まれる事例の特徴ベクトルを取得し (ステップ 4), それぞれの事例に対応する BMU を選択する (ステップ 5). BMU の選択には, 各事例の特長ベクトルとノードに対応する重みベクトルの L2 距離などを用いる. L2 距離を用いる場合には,

$$BMUidx(\mathbf{x}) = \operatorname{argmin}_{n \in \{1, \dots, N\}} \|\mathbf{w}_n - \mathbf{x}\|_2^2 \quad (5.2)$$

によって, L2 距離が最小の重みベクトルを持つノードを BMU として選択できる. 次に, 順序ラベルが正順である ($l_y = 1$) 場合には, 取得された BMU の添え字番号を比較して, 逆順になっている場合に更新するノードの修正を行う (ステップ 10, 11). ここで $idx(\cdot)$ は BMU の添え字番号を返す関数とする. $l_y = 1$ の場合には, 事例 l_i に対応する BMU の添え字番号が事例 l_j に対応する BMU の添え字番号に比べて大きい値であることが望ましい. そのため, $idx(BMU_{x_{l_i}}) \leq idx(BMU_{x_{l_j}})$ となる場合には, 事例 l_i の BMU であるべきノードと, 事例 l_j の BMU であるべきノードを修正する必要がある. そこで, 修正幅 s を用いて $BMUidx_{x_{l_i}}^{(new)}$ と $BMUidx_{x_{l_j}}^{(new)}$ を修正する (ステップ 10, 11). 順序ラベル $l_y = -1$ の場合には, 逆の処理を行う (ステップ 15, 16). 各試行の最後に修正された BMU の重みベクトルを, 入力事例が持つ特徴ベクトルを用いて式 (5.1) に従って更新する (ステップ 19). 重みベクトルの更新処理は通常 SOM と同様の更新を行う. すなわち, 更新対象ノードの近傍に対しても減衰した値を用いて更新を行う (式 (5.1)). 上記の処理を T 試行行い, 更新処理を終了する. 網羅的にラベル情報を利用するためにはラベル集合 L に含まれる順序ラベルを順番に利用する方法が考えられるが, ランダムサンプリングにおいても試行回数 T を十分に大きく設定することにより, 確率的にすべてのラベル情報を利用することが可能であり, また経験的に事例の適用順序を固定しない方が安定した学習が可能であるため, 本稿ではラベル集合 L からランダムサンプリングを行う方法を採用する.

1d-OrderSOM の予測処理のアルゴリズムを図 5.3 に示す. 予測処理においては, 予測対象の各事例について選択された BMU の添え字を順位と解釈して出力する.

Algorithm 1d-OrderSOM Update

Input: $\{\mathbf{x}_n\} \in D, \{(i, j, y)\} \in L, T, N, s$
Output: \mathbf{W}

- 1: **Initialize:** Create \mathbf{W} using random vectors.
- 2: **FOR** t in 1 to T
- 3: Obtain random pair label l from L
- 4: Obtain \mathbf{x}_{l_i} and \mathbf{x}_{l_j} from D
- 5: Select BMU for \mathbf{x}_{l_i} ($\text{BMU}_{x_{l_i}}$) and \mathbf{x}_{l_j} ($\text{BMU}_{x_{l_j}}$)
- 6: $\text{BMUidx}_{x_{l_i}}^{(new)} \leftarrow \text{idx}(\text{BMU}_{x_{l_i}})$
- 7: $\text{BMUidx}_{x_{l_j}}^{(new)} \leftarrow \text{idx}(\text{BMU}_{x_{l_j}})$
- 8: **IF** $l_y = 1$ **THEN**
- 9: **IF** $\text{idx}(\text{BMU}_{x_{l_i}}) \leq \text{idx}(\text{BMU}_{x_{l_j}})$ **THEN**
- 10: $\text{BMUidx}_{x_{l_i}}^{(new)} \leftarrow \text{BMUidx}_{x_{l_i}}^{(new)} + s$
- 11: $\text{BMUidx}_{x_{l_j}}^{(new)} \leftarrow \text{BMUidx}_{x_{l_j}}^{(new)} - s$
- 12: **ENDIF**
- 13: **ELSE**
- 14: **IF** $\text{idx}(\text{BMU}_{x_{l_i}}) \geq \text{idx}(\text{BMU}_{x_{l_j}})$ **THEN**
- 15: $\text{BMUidx}_{x_{l_i}}^{(new)} \leftarrow \text{BMUidx}_{x_{l_i}}^{(new)} - s$
- 16: $\text{BMUidx}_{x_{l_j}}^{(new)} \leftarrow \text{BMUidx}_{x_{l_j}}^{(new)} + s$
- 17: **ENDIF**
- 18: **ENDIF**
- 19: Update \mathbf{W} with $\text{BMUidx}_{x_{l_i}}^{(new)}$ and $\text{BMUidx}_{x_{l_j}}^{(new)}$
 in the manner determined by Eq.(5.1).
- 20: **ENDFOR**
- 21: **RETURN** \mathbf{W}

図 5.2: 1d-OrderSOM の更新アルゴリズム

5.3.3 多次元 OrderSOM

本節では、競合層が2次元以上の格子状から成る多次元 OrderSOM における学習と予測アルゴリズムについて述べる。5.3.2 節で述べた方法と同様に、順位次元を持つ構造であれば2次元以上の構造についても同様の更新手続きで学習を行うことが可能である。2次元格子状 OrderSOM (2d-OrderSOM) の更新手続きの概略図を図 5.4 に示す。

2d-OrderSOM においては、順位次元 (図 5.4 横軸方向) とクラスタ次元 (図 5.4 縦軸方向) に分けられる。1d-OrderSOM においてはノード数に関するパラメータ N を1つ設定していたが、2d-OrderSOM においては順位次元のノード数 N_x 、クラスタ次元のノード数 N_y という2つのパラメータによって競合層の構造を決定する。すなわち、2d-OrderSOM の競合層には合計 $N_x \cdot N_y$ ノードが存在する。ある順位のデータが特徴空間上に広く分布している場合においても、ひとつの順位の事例分布に対して複数ノードを対応づけることが可能となり、クラスタ次元を利

Algorithm OrderSOM (1-dimension) Prediction

Input: $\{\mathbf{x}_n\} \in D^{test}$, \mathbf{W} **Output:** \mathbf{y}

- 1: **FOR** n in 1 to $|D^{test}|$
 - 2: Select BMU for \mathbf{x}_n
 - 3: $y_n \leftarrow idx(\text{BMU}_{x_n})$
 - 4: **ENDFOR**
 - 5: **RETURN** \mathbf{y}
-

図 5.3: 1d-OrderSOM の予測アルゴリズム

用してデータのクラスタリングを実現する．このように，多次元 OrderSOM では 1 次元 OrderSOM よりも複雑な分布のデータに柔軟に対応することが可能である．

2d-OrderSOM における更新手続きは 1 次元 OrderSOM と同様に，入力ペアの各事例に対して BMU を選択し，BMU のインデクスを取得する (図 5.4 ステップ 1)．この際，順位次元のインデクスのみにより順序が正しく表現されているかを判定し，インデクスの修正を行う (図 5.4 ステップ 2)．この際，クラスタ次元のインデクスは一切変更しない．変更したインデクスに対応するノードを各事例に対応する BMU とみなして更新処理を行う (図 5.4 ステップ 3)．更新の際には，順位次元とクラスタ次元双方における近傍に対しても減衰した更新値を適用する．これにより，1 次元 OrderSOM と同様に順序情報から順位を再現する SOM の学習と同時に，クラスタ次元の隣り合うノード同時が近い特徴ベクトルを持つことで，同順位の事例集合が特徴空間に広く分布しているような状況においても適切に順序学習が可能となる．

3 次元以上の多次元 OrderSOM においては，クラスタ次元に利用されるノード数が多くなるため表現力が高くなる一方，指数的にノード数が増加する．競合層におけるノード数増加はそのまま BMU 選択コスト増加につながり，また重み更新操作においても近傍ノードが増えるため，ノード数増加の影響を受ける．このように次元数を高く設定することで学習コストが高くなる欠点がある．また，新たな次元に対応するノード数パラメータも増えるため，実際に適用する場合にはパラメータ探索の観点からもコストが大きくなるという問題がある．そのため，本稿では多次元 OrderSOM として 2d-OrderSOM に限定してその有効性を検証する．

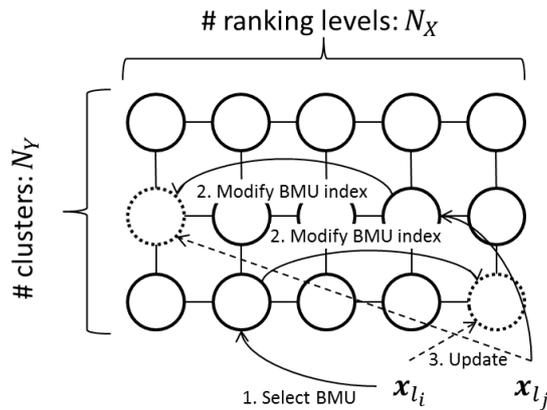


図 5.4: 2d-OrderSOM の更新アルゴリズムの概要

5.4 評価

本稿では，人工データと実データを用いた実験を通じて OrderSOM が順序ペアの教師信号を利用して，適切に順位を再現すること，および既存手法との比較を通じて OrderSOM の有効性を検証する．

5.4.1 実験 1: 人工データを用いた評価

人工データを用いた評価では，まず 1d-OrderSOM の有効性検証を行い，次に 1d-OrderSOM では対応が困難な人工データにおける 2d-OrderSOM の有効性を検証する．

実験 1a: 1d-OrderSOM の有効性評価

2次元特徴空間において5段階の順位を持つ事例集合の真の分布が平均 $((1, 4), (0, 2), (3, 3), (4, 0), (2, 0))$ であり，分散が 0.1 の等方ガウス分布と仮定する．これらの分布から各順位につき 100 事例サンプリングを行い，合計 500 事例生成した．得られた 500 事例のうち，ランダムに選択した 250 事例を訓練データとして利用し，残りをテストデータとして用いた．

ベースライン手法として RSVM を採用し， C パラメータは 1.0 を用いた．RSVM の実装には `svm_rank`² を用いた．1d-OrderSOM の試行回数 T は 5000 とし，ノ

²http://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html

表 5.1: 実験 1a の結果

RSVM	1d-OrderSOM (N)						
	2	3	4	5	10	20	30
.786	.845	.942	.956	1.0	.993	.997	.997

ド数 N_x が真の順位数と異なる場合のアルゴリズムの挙動を確認するため $N_x \in \{2, 3, 4, 5, 10, 20, 30\}$ を比較した。5.3.1 で述べたとおり，重みベクトル更新の結果を安定させるため，更新幅 s は 1 とした。学習率 $\alpha(t)$ は初期値 1.0，近傍関数 $\theta(t)$ は初期値 0.1 とし，いずれも試行回数で減衰する係数 $\frac{1000}{t+1000}$ をかけた値を用いた。

テストデータにおける事例集合の順位を適切に再現しているかの評価指標には順位相関指標である Kendall's τ を利用した。

評価結果を表 5.1 に示す。結果より，ノード数によらず OrderSOM が RSVM に比べて高い順位相関を示した。特にノード数が真の順位数と一致した場合には，テストデータにおいて完全な順位の再現を達成した。 N_x を大きく設定しても順位相関が大きく下がることはなかった。

本実験においては，RSVM の結果は C パラメータの値に大きく依存しなかった。 $C \in \{0.001, 0.01, 0.1, 1.0, 10.0\}$ の値を比較したところ，Kendall's τ の分散が $3 \cdot 10^{-6}$ であったため， C パラメータに依存しないと判断した。そのため， C パラメータについては検証セットを用意して探索を行わず， $C = 1.0$ と固定した値で実験を行った。

人工データに対する 1d-OrderSOM の適用結果を図 5.5 に示す。図 5.5 に実験に利用した 500 事例を全てプロットしており，色とマーカーで順位を表している。実線で結ばれた丸マーカー（黒色）のプロットで競合層のノード位置を，ノード間を結ぶ実線でエッジを表している。破線で示された直線は RSVM によって学習された分離直線 $-0.813x_1 + 2.011x_2 = 0$ を表している。この破線からの距離が離れれば離れるほど絶対値が大きいスコアが与えられ，破線上部では正の値，破線下部では負の値が与えられる。すなわち，図中左上方向が高い順位，右下方向が低い順位を表している。

実験 1b: 2d-OrderSOM の有効性評価

実験 1b では，1d-OrderSOM による学習が困難な人工データにおいて，異なる N_y の 2d-OrderSOM の精度を確認することで 2d-OrderSOM の有効性を検証する。

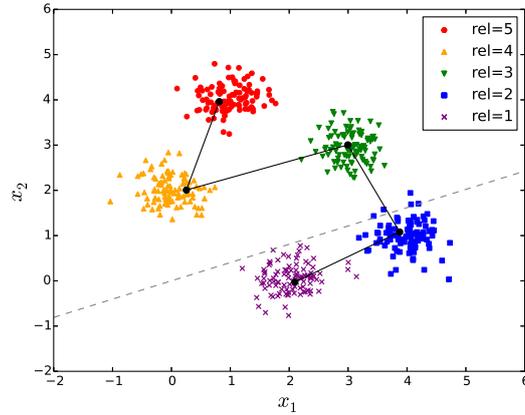


図 5.5: 実験 1a における 1d-OrderSOM と RSVM の学習結果

2次元特徴空間において5段階の順位を持つ事例集合の真の分布が平均 $((0, 3), (4, 3), (1, 3), (3, 3), (2, 3))$ であり, すべての分布の x_1 方向の分散が 0.01, x_2 方向の分散が 1.0 であり, 共分散が 0 のガウス分布と仮定する. 実験 1a と同様にこれらの分布から各順位につき 100 事例サンプリングを行い, 合計 500 事例生成した. 得られた 500 事例のうち, ランダムに選択した 250 事例を訓練データとして利用し, 残りをテストデータとして用いた. 各分布が棒状に広く分布しているため, 各順位の分布に対して 1 つのノードを用意するだけでは不十分である状況を想定している. また, 各分布が順位に対して順番に並んでいないため, 1d-OrderSOM での順位の学習は困難であると予想される. これらの分布から生成されたサンプルに対して教師なし SOM を適用した場合には, 順位に関する情報を利用することができないため, 特徴空間における位置関係に基づく競合層ノードの配置が行われ, その結果, x_1 方向に対して降順または昇順にノードが配置されるように学習される. この人工データを用いる実験を行うことで, 教師なし SOM では学習が困難な状況における OrderSOM の有効性を確認することができる.

2d-OrderSOM の試行回数 T は 10,000 とし, ノード数 N_x は真の順位数と同様の 5 とし, 近傍関数の初期値を 0.01 とした以外は実験 1a と同じ条件で行った. 実験 1b では, クラスタ次元数の変化による精度の変化を確認するため, $N_y \in \{1, 2, 4, 6, 8, 10\}$ の組み合わせを比較した. 実験 1a と同様に RSVM をベースラインとし, 評価指標は Kendall's τ を用いた.

実験 1b の各手法の Kendall's τ の値を表 5.2 に示す. RSVM の τ が比較手法の中で最小の値を示した. 今回のデータでは 1d-OrderSOM ($N_y = 1$) では τ の値が

表 5.2: 実験 1b の結果

RSVM	1d-OrderSOM	2d-OrderSOM (N_y)				
		2	4	6	8	10
.150	.280	.884	.877	.910	.813	.727

低かった。 $N_y > 1$ の結果では、 $N_y = 6$ が最大の値を示しており、それ以外の N_y についても τ が 0.7 以上の値を示している。

実験 1b における $N_y = 1$ と $N_y = 6$ の OrderSOM の学習結果を図 5.6 に示す。図中の破線は実験 1a と同様に RSVM の学習で得られた分離直線を表している ($-0.391x_1 - 0.079x_2 = 0$)。

図 5.6 (a) では 1d-OrderSOM の学習結果を示している。表 5.2 の結果からもわかるとおり、1d-OrderSOM は今回のデータに対しては適切に学習できていないことがわかる。

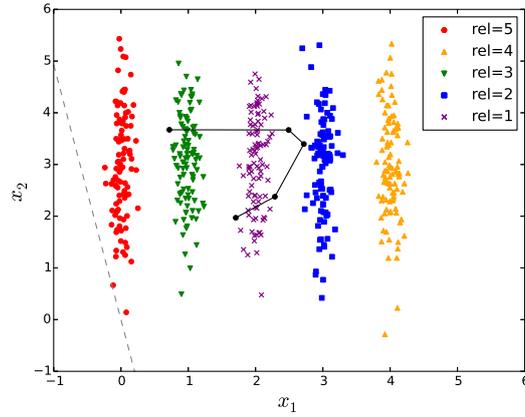
図 5.6 (b) では 2d-OrderSOM の学習結果を示している。この結果から、 $N_y = 6$ の 2d-OrderSOM においては、教師なし SOM では再現が困難である順位分布に対応した競合層の形状を再現し、特にクラスタ次元に用意された複数のノード集合が、広く分布している同一順位の事例集合に対応している様子がわかる。

5.4.2 実験 2: 検索ランキングデータを用いた評価

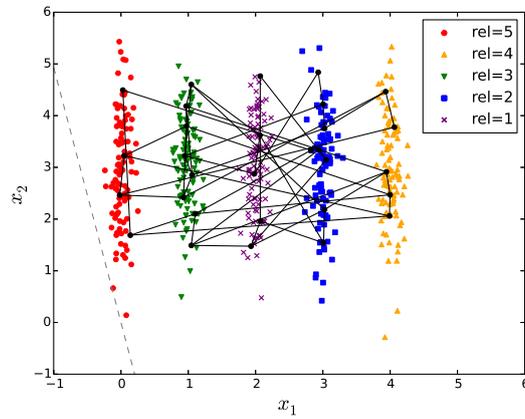
情報検索分野で利用される検索ランキングデータ LETOR4.0 MQ2007 データセット³ を用いて評価実験を行った。データセットはクエリ集合とクエリ集合に含まれる各クエリの検索結果に対する適合度評価によって構成される。具体的には、各クエリ・文書ペアに対して TF-IDF [44] などの検索ランキングに利用されるスコアが特徴として 47 次元付与されており、また各ペアに対する順位ラベルが 3 段階で付与されている。これを用いることで本稿の問題設定における提案手法の有効性の検証が可能である。

各データセットは交差検定のためにあらかじめ 5 分割され、3 つを訓練事例、1 つを検証事例、1 つをテスト事例としたセットが 5 つ構築されている。本実験ではこのセットをそのまま利用し、訓練事例をモデル構築、検証事例をパラメータ選択に、テスト事例を評価に利用した。

³<http://research.microsoft.com/en-us/um/beijing/projects/letor/letor4dataset.aspx>



(a)



(b)

図 5.6: 実験 1b における OrderSOM (a) $N_x = 5, N_y = 1$ (b) $N_x = 5, N_y = 6$ の学習結果

情報検索ランキングに利用される特徴は，各次元同士の相関が高いことが知られており [28]，特徴空間における距離を用いて BMU を選択する OrderSOM が適切に動作しない可能性が考えられる．そこで，特徴ベクトルを全次元を利用する条件 (All features) と，特徴を k 次元 ($k \in \{2, 3, \dots, 10\}$) に削減した条件 (Small feature set) における 2 つの条件で実験を行った．次元削減には，適合性評価に対して順位相関が高い上位 k 件の特徴ベクトルをグリーディに選択する方法を用いた．

評価指標には，情報検索で用いられる NDCG@1，MeanNDCG [33] と Kendall's τ を用いた．

ベースライン手法はランキング学習において標準的に利用される RSVM を採用し，RSVM は線形カーネルを利用し， C パラメータは $\{0.0001, 0.001, 0.01, 0.1, 1.0, 10\}$

表 5.3: 実験 2: LETOR データに対する評価結果 . Small feature set 括弧内の数字は特徴次元数を表す .

Method	All features			Small feature set (2)		
	NDCG@1	MeanNDCG	τ	NDCG@1	MeanNDCG	τ
RSVM	.4156	.4975	.1872	.3551	.4774	.1782
OrderSOM	.3228	.4477	.1588	.3651	.4810	.1836
Method	Small feature set (5)			Small feature set (10)		
	NDCG@1	MeanNDCG	τ	NDCG@1	MeanNDCG	τ
RSVM	.3545	.4773	.1733	.3517	.4772	.1737
OrderSOM	.3639	.4790	.1812	.3479	.4684	.1702

から検証データにおいて最大精度のものを選択した . 提案手法においては , $N_x \in \{10, 20, 30\}$, $N_y \in \{5, 10, 30, 50\}$ を探索パラメータとして , いずれも検証セットにおいて精度が最大になるものを選択した . BMU 選択の距離には L1 距離を利用し , 学習率の初期値は 0.025 , 近傍関数の初期値は 0.5 を用いた .

結果

All features および $k = 2, 5, 10$ の Small feature set の結果を表 5.3 に示す . 表 5.3 より , 全ての特徴次元を利用した All features においては NDCG@1 , MeanNDCG , τ の値において RSVM が OrderSOM に比べて高い値を示した (paired t -test $p < 0.01$) . 一方で特徴次元を少なくした Small feature set ($k = 2, 5$) においては , 全ての評価指標において OrderSOM が RSVM に比べて高い値を示し (paired t -test n.s.) , $k = 10$ においては全ての評価指標において RSVM が OrderSOM に比べて高い値を示した (paired t -test n.s.) .

また , すべての k について OrderSOM と RSVM の Kendall's τ の値をプロットしたグラフを図 5.7 に示す . これより , $k = 2, 3, 4, 5, 8$ において OrderSOM が RSVM より高い τ の値を示し , $k = 6, 7, 9, 10$ においては RSVM が OrderSOM より低い τ の値を示した .

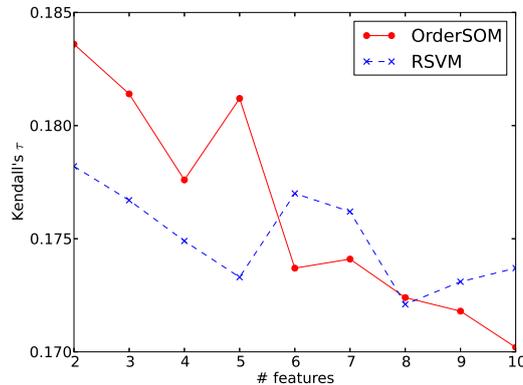


図 5.7: 実験 2: 特徴次元数毎の τ の値

5.5 考察

実験 1a, 1b の結果より, 線形モデルで順位を表現できない特徴空間および事例分布においては RSVM が順位再現に適切に働かないことがわかった.

1d-OrderSOM は実験 1a において有効に働き, 実験 1b においては適切に働かなかったことから, 各順位的事例分布がまとまっており, 分布毎の距離が同一分布内の分散に比べて大きい場合に有効に働くことを確認した. また, その際の N_x の値は真の値が事前にわかる場合には真の値を利用し, そうでない場合には大きめの値を設定しておくことで有効に働くことを確認した.

実験 1b の結果より, 異なる順位の分布の平均距離に比べて同一順位の分散が大きい場合においては 1d-OrderSOM ($N_y = 1$) が適切に働かないことを確認した. また, このような状況においても N_y の値を増やすことで OrderSOM が有効に働くことを確認した. 実験 1b に利用したデータは特徴空間上に異なる順位に対応する分布が並んで存在しないため, このようなデータに対して教師なし SOM を適用すると, 順位ラベルどおりの順序関係を維持した競合層ノードの配置が困難である. 図 5.6 より, N_y を大きく設定することにより, 教師なしの SOM では順位ラベルにしたがってフィットさせることが困難な状況においても, 順位相関の観点から順位を適切に再現することを確認した.

実験 2 においては, 全ての次元を利用 (All features) した実験条件においては OrderSOM に比べて RSVM が高い精度を示し, 2次元と5次元を利用 (Small feature set (2), (5)) した実験条件においては OrderSOM が RSVM に比べて高い精度を示した. これにより, 人工データを用いた実験 1 と同様に, 低次元特徴空間にお

いてRSVMによってうまくランキングできないような状況において，OrderSOMがRSVMに比べて有効に働くことが示唆された．しかしながら，Small feature set (10)においては，RSVMがOrderSOMに比べてすべての評価指標で高い値を示しており，今回の実験条件においては5次元程度の低次元特徴空間におけるOrderSOMの有効性を確認した．この傾向は，図5.7においても確認することができる．OrderSOMは $k = 5$ 以下においては安定してRSVMより高い値を示していることから，個別の評価値では有意差を確認できなかったものの，低次元特徴空間におけるOrderSOMの有効性が示唆された．また， $k = 6$ 以上においてはほぼ同じ値を示した $k = 8$ を除いて低い値を示しており，OrderSOMはAll feature setにおいては $\tau = 0.1588$ であることから， k の値を増やし，特徴次元数が増えるにつれてOrderSOMは順位相関が低くなる傾向があることを確認した．一方でRSVMはAll featuresにおいては $\tau = 0.1872$ であり， $k = 10$ 以下の特徴次元では達成できなかった順位相関を達成しており，特徴次元数を増やすことで高い順位相関を達成することができることが示唆された．

いずれもKendall's τ の値が0.2を超えない結果であり，実験1で用いた人工データのように各順位の事例集合が分離可能な分布をしていないと予想される．このような状況においては，OrderSOMはRSVMのような汎化性能を示すことができないといえる．高次元かつ各順位の分布がきれいに分かれていないような状況においては，何かしらの距離に基づいてBMUを選択するOrderSOMのアプローチでは汎化性能に限界があることがわかった．

5.6 まとめ

本章では，自己組織化マップを拡張し，事例ペアの順序ラベルから順位を学習する教師ありSOMであるOrderSOMを提案した．OrderSOMでは，競合層のノードの添え字を予測順位と見なし，入力事例ペアの各事例に対するBMUの位置を順序ラベルに基づいて修正することにより，SOMにおいて順位学習を可能にする手法である．本稿では，鎖状の構造を持つ1d-OrderSOMと，順位次元とクラスタ次元を持つ2d-OrderSOMにおける学習と予測アルゴリズムを提案した．人工データを用いた評価実験を通じて，既存手法であるRSVMに比べてOrderSOMが順位相関の観点で高い汎化性能を示した．実データを用いた評価実験を通じて，RSVM

が有効に働かない特徴次元が少ない場合においても OrderSOM によって高い精度でランキングを再現することを確認した。

第6章 まとめ

本章では，本論文のまとめと今後の課題について述べる．

6.1 結論

本論文では，3つのランキング学習アルゴリズムの開発を通じて，以下の4つの課題を達成した．

1. 複数の異なる性質の訓練データを用いて，より高精度なモデルを構築すること (課題 1)．
2. 大規模データに対して学習アルゴリズムがスケールし，逐次学習が可能であること (課題 2)．
3. 学習データに含まれるノイズに強い学習アルゴリズムであること (課題 3)．
4. 低次元特徴空間においても有効な予測モデルを構築可能なこと (課題 4)．

第3章では，人手により作成した評価データとクリックログデータという適合性分布が異なる複数情報源を用いる，ブースティングに基づくランキング学習アルゴリズムである TRankBoost を提案し，実データを用いた評価実験を通じて有効性を検証した．第4章では，ランキング評価指標の損失関数への導入，ノイズに頑健な損失関数の採用，最大損失を与える事例の選択といった特徴を持つ選択的ペアワイズ手法によるオンラインランキング学習である PARank-NDCG を提案し，従来のオンラインランキング学習に対して優れた性能を示した．第5章では，ランキングに用いる多数の要因を抽出できない状況において高精度なランキングを実現するため，自己組織化マップを教師あり順序学習に拡張した OrderSOM を提案し，その実験的評価を通じて有効性を示した．

6.2 今後の課題

本論文で取り組んだ研究について，トピックごとに今後の課題を述べる．

適合性分布が異なる情報源を用いたランキング学習に関する今後の課題は以下のとおりである：

- TRankBoost のモデルパラメータ (α_t, β, M) を自動的に推定する方法について検討したい．
- 本研究では，人手による適合性評価と異なる情報源から得られた訓練データにおける適合性分布の尺度の違いを補正するために，ペアワイズ手法に着目した．それ以外のポイントワイズ手法やリストワイズ手法にも転移学習を適用したい．

ランキング評価指標を考慮した選択的ペアワイズ手法によるオンラインランキング学習に関する今後の課題は以下のとおりである：

- NDCG 以外のランキング評価指標への拡張．
- 全順序ペアではなく部分集合に対して最大損失を与えるペアを重み更新に用いる Max-loss update 戦略の拡張．

順序ラベルを用いた教師あり自己組織化マップに関する今後の課題は以下のとおりである：

- データに合わせてノード数を自動決定するような拡張．
- 学習と予測において，入力事例とノードの何らかの意味での距離に基づいて BMU 選択を行うため，特徴空間上に異なる順位の事例が集中して存在するような状況においては適切に学習できない問題の解決．

多様な情報源を用いたランキング学習はまだ発展途上の研究テーマであり，新たな手法の検討も含め，提案法の改善に取り組みたい．

参考文献

- [1] Eugene Agichtein, Eric Brill, and Susan Dumais. Improving web search ranking by incorporating user behavior information. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 19–26, New York, NY, USA, 2006. ACM.
- [2] R. Agrawal, A. Halverson, K. Kenthapadi, N. Mishra, and P. Tsaparas. Generating labels from clicks. In *WSDM '09: Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pp. 172–181, New York, NY, USA, 2009. ACM.
- [3] Azzah Al-Maskari, Mark Sanderson, and Paul Clough. Relevance judgments between trec and non-trec assessors. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 683–684, New York, NY, USA, 2008. ACM.
- [4] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. In *WWW7: Proceedings of the seventh international conference on World Wide Web 7*, pp. 107–117, Amsterdam, The Netherlands, The Netherlands, 1998. Elsevier Science Publishers B. V.
- [5] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pp. 89–96, New York, NY, USA, 2005. ACM.
- [6] Christopher J. C. Burges, Robert Ragno, and Quoc Viet Le. Learning to rank with nonsmooth cost functions. In *NIPS '06: Proceedings of the Twentieth*

- Annual Conference on Neural Information Processing Systems*, pp. 193–200, 2006.
- [7] Stefan Büttcher, Charles L. A. Clarke, and Gordon V. Cormack. *Information Retrieval: Implementing and Evaluating Search Engines*. MIT Press, 2009.
- [8] Yunbo Cao, Jun Xu, Tie-Yan Liu, Hang Li, Yalou Huang, and Hsiao-Wuen Hon. Adapting ranking svm to document retrieval. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 186–193, New York, NY, USA, 2006. ACM.
- [9] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pp. 129–136, New York, NY, USA, 2007. ACM.
- [10] Ben Carterette and Rosie Jones. Evaluating search engines by modeling the relationship between relevance and clicks. In *NIPS '07: Proceedings of the conference on neural information processing systems*, pp. 217–224. MIT Press, 2007.
- [11] Olivier Chapelle, Yi Chang, and Tie-Yan Liu. Future directions in learning to rank. In *Proceedings of the ICML '10 Workshop on Yahoo! Learning to Rank Challenge*, Vol. 14 of *JMLR Workshop and Conference Proceedings*, pp. 91–100, 2011.
- [12] Depin Chen, Jun Yan, Gang Wang, Yan Xiong, Weiguo Fan, and Zheng Chen. Transrank: A novel algorithm for transfer of rank learning. In *Proceedings of the 8th IEEE International Conference on Data Mining Workshops*, pp. 106–115. IEEE Computer Society, 2008.
- [13] Keke Chen, Rongqing Lu, C. K. Wong, Gordon Sun, Larry Heck, and Belle Tseng. Trada: tree based ranking function adaptation. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pp. 1143–1152, New York, NY, USA, 2008. ACM.

- [14] Michael Collins. Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10*, EMNLP '02, pp. 1–8, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.
- [15] Ronan Collobert, Fabian Sinz, Jason Weston, and Léon Bottou. Trading convexity for scalability. In *Proceedings of the 23rd international conference on Machine learning*, ICML '06, pp. 201–208, New York, NY, USA, 2006. ACM.
- [16] Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online passive-aggressive algorithm. *Mach. Learn.*, Vol. 7, pp. 551–585, 2006.
- [17] Koby Crammer and Yoram Singer. Pranking with ranking. In *NIPS '01: Proceedings of the 14th annual conference on Neural Information Processing Systems*, pp. 641–647, 2001.
- [18] Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. Boosting for transfer learning. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pp. 193–200, New York, NY, USA, 2007. ACM.
- [19] Anlei Dong, Yi Chang, Zhaohui Zheng, Gilad Mishne, Jing Bai, Ruiqiang Zhang, Karolina Buchner, Ciya Liao, and Fernando Diaz. Towards recency ranking in web search. In *Proceedings of the third ACM international conference on Web search and data mining*, WSDM '10, pp. 11–20, New York, NY, USA, 2010. ACM.
- [20] Zhicheng Dou, Ruihua Song, Xiaojie Yuan, and Ji-Rong Wen. Are click-through data adequate for learning web search rankings? In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pp. 73–82, New York, NY, USA, 2008. ACM.
- [21] Kevin Duh and Katrin Kirchhoff. Learning to rank with partially-labeled data. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR*

- conference on Research and development in information retrieval*, pp. 251–258, New York, NY, USA, 2008. ACM.
- [22] Jonathan L. Elsas, Vitor R. Carvalho, and Jaime G. Carbonell. Fast learning of document ranking functions with the committee perceptron. In *WSDM '08: Proceedings of the international conference on Web search and web data mining*, pp. 55–64, New York, NY, USA, 2008. ACM.
- [23] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *J. Mach. Learn. Res.*, Vol. 9, pp. 1871–1874, June 2008.
- [24] Françoise Fessant, Patrice Aknin, Latifa Oukhellou, and Sophie Midenet. Comparison of supervised self-organizing maps using euclidian or mahalanobis distance in classification context. In *Proceedings of the 6th International Work-Conference on Artificial and Natural Neural Networks: Connectionist Models of Neurons, Learning Processes and Artificial Intelligence-Part I, IWANN '01*, pp. 637–644, London, UK, UK, 2001. Springer-Verlag.
- [25] Yoav Freund, Raj Iyer, Robert E. Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. *J. Mach. Learn. Res.*, Vol. 4, pp. 933–969, 2003.
- [26] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, Vol. 55, No. 1, pp. 119–139, 1997.
- [27] Jianfeng Gao, Wei Yuan, Xiao Li, Kefeng Deng, and Jian-Yun Nie. Smoothing clickthrough data for web search ranking. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pp. 355–362, New York, NY, USA, 2009. ACM.
- [28] Xiubo Geng, Tie-Yan Liu, Tao Qin, and Hang Li. Feature selection for ranking. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 407–414, New York, NY, USA, 2007. ACM.

- [29] Josef Goppert and Wolfgang Rosenstiel. Regularized som-training: a solution to the topology-approximation dilemma? In *IEEE International Conference on Neural Networks*, Vol. 1, pp. 38–43. IEEE, 1996.
- [30] Markus Hagenbuchner and Ah Chung Tsoi. A supervised training algorithm for self-organizing maps for structures. *Pattern Recogn. Lett.*, Vol. 26, No. 12, pp. 1874–1884, September 2005.
- [31] Ralf Herbrich, Thore Graepel, and Klaus Obermayer. Large margin rank boundaries for ordinal regression. In A.J. Smola, P.L. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pp. 115–132, Cambridge, MA, 2000. MIT Press.
- [32] Yoshiyuki Inagaki, Narayanan Sadagopan, Georges Dupret, Anlei Dong, Ciya Liao, Yi Chang, and Zhaohui Zheng. Session based click features for recency ranking. In *AAAI '10: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.
- [33] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, Vol. 20, No. 4, pp. 422–446, 2002.
- [34] Thorsten Joachims. Optimizing search engines using clickthrough data. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 133–142, New York, NY, USA, 2002. ACM Press.
- [35] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. Accurately interpreting clickthrough data as implicit feedback. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 154–161, New York, NY, USA, 2005. ACM.
- [36] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, Filip Radlinski, and Geri Gay. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Trans. Inf. Syst.*, Vol. 25, No. 2, p. 7, 2007.

- [37] Jaap Kamps, Marijn Koolen, and Andrew Trotman. Comparative analysis of clicks and judgments for ir evaluation. In *WSCD '09: Proceedings of the 2009 workshop on Web Search Click Data*, pp. 80–87, New York, NY, USA, 2009. ACM.
- [38] Kye-Hyeon Kim and Seungjin Choi. Incremental learning to rank with partially-labeled data. In *WSCD '09: Proceedings of the 2009 workshop on Web Search Click Data*, pp. 20–27, New York, NY, USA, 2009. ACM.
- [39] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, Vol. 46, No. 5, pp. 604–632, 1999.
- [40] Teuvo Kohonen, editor. *Self-organizing maps*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1997.
- [41] R. Lempel and S. Moran. Salsa: the stochastic approach for link-structure analysis. *ACM Trans. Inf. Syst.*, Vol. 19, No. 2, pp. 131–160, 2001.
- [42] Ping Li, Christopher J. C. Burges, and Qiang Wu. Mcrank: Learning to rank using multiple classification and gradient boosting. In *NIPS '07: Proceedings of the 21st annual conference on Neural Information Processing Systems*, 2007.
- [43] Tie-Yan Liu. *Learning to Rank for Information Retrieval*. Springer, 2011.
- [44] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*. Cambridge University Press, 2008.
- [45] Tom Minka and Stephen Robertson. Selection bias in the letor datasets. In *LR4IR '08: Proceedings of SIGIR2008 Workshop on Learning to rank for information retrieval*, 2008.
- [46] Tsutomu Miyoshi. Node exchange for improvement of som learning. In *Knowledge-Based Intelligent Information and Engineering Systems, 9th International Conference, KES 2005, Melbourne, Australia, September 14-16, 2005, Proceedings, Part III*, Vol. 3683 of *Lecture Notes in Computer Science*, pp. 569–574. Springer, 2005.

- [47] Taesup Moon, Lihong Li, Wei Chu, Ciya Liao, Zhaohui Zheng, and Yi Chang. Online learning for recency search ranking using real-time user feedback. In *Proceedings of the 19th ACM international conference on Information and knowledge management, CIKM '10*, pp. 1501–1504, New York, NY, USA, 2010. ACM.
- [48] Shuzi Niu, Jiafeng Guo, Yanyan Lan, and Xueqi Cheng. Top-k learning to rank: labeling, ranking and evaluation. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval, SIGIR '12*, pp. 751–760, New York, NY, USA, 2012. ACM.
- [49] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. Technical Report HKUST-CS08-08, Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong, China, November 2008.
- [50] John C. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. In Bernhard Schölkopf, Christopher J. C. Burges, and Alexander J. Smola, editors, *Advances in kernel methods*, chapter Fast training of support vector machines using sequential minimal optimization, pp. 185–208. MIT Press, Cambridge, MA, USA, 1999.
- [51] Filip Radlinski and Thorsten Joachims. Active exploration for learning rankings from clickthrough data. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 570–579, New York, NY, USA, 2007. ACM.
- [52] Filip Radlinski, Madhu Kurup, and Thorsten Joachims. How does click-through data reflect retrieval quality? In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pp. 43–52, New York, NY, USA, 2008. ACM.
- [53] S. Robertson, S. Walker, S. Jones, M.M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. In *Proceedings of the Third Text REtrieval Conference (TREC-3)*, pp. 109–126, 1994.

- [54] Jyri Saarikoski, Jorma Laurikkala, Kalervo Järvelin, and Martti Juhola. Self-organising maps in document classification: a comparison with six machine learning methods. In *Proceedings of the 10th international conference on Adaptive and natural computing algorithms - Volume Part I, ICANNGA'11*, pp. 260–269, Berlin, Heidelberg, 2011. Springer-Verlag.
- [55] D. Sculley. Large scale learning to rank. In *Proceedings of the NIPS '09 Workshop on Advances in Ranking*, 2009.
- [56] Libin Shen and Aravind K. Joshi. Ranking and reranking with perceptron. *Mach. Learn.*, Vol. 60, pp. 73–96, September 2005.
- [57] Michael Taylor, John Guiver, Stephen Robertson, and Tom Minka. Softrank: optimizing non-smooth rank metrics. In *WSDM '08: Proceedings of the international conference on Web search and web data mining*, pp. 77–86, New York, NY, USA, 2008. ACM.
- [58] Ming-Feng Tsai, Tie-Yan Liu, Tao Qin, Hsin-Hsi Chen, and Wei-Ying Ma. Frank: a ranking method with fidelity loss. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 383–390, New York, NY, USA, 2007. ACM.
- [59] Nicolas Usunier, Vinh Truong, Massih R. Amini, and Patrick Gallinari. Ranking with unlabeled data: A first study. In *NIPS2005 workshop: Learning to Rank*, 2005.
- [60] Hamed Valizadegan, Rong Jin, Ruofei Zhang, and Jianchang Mao. Learning to rank by optimizing ndcg measure. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22a*, pp. 1883–1891, 2009.
- [61] Zhuang Wang and Slobodan Vucetic. Online passive-aggressive algorithms on a budget. In *AISTATS '10: Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, pp. 908–915, 2010.

- [62] Jingfang Xu, Chuanliang Chen, Gu Xu, Hang Li, and Elbio Renato Torres Abib. Improving quality of training data for learning to rank using click-through data. In *WSDM '10: Proceedings of the third ACM international conference on Web search and data mining*, pp. 171–180, New York, NY, USA, 2010. ACM.
- [63] Jun Xu and Hang Li. Adarank: a boosting algorithm for information retrieval. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 391–398, New York, NY, USA, 2007. ACM.
- [64] Jun Xu, Tie-Yan Liu, Min Lu, Hang Li, and Wei-Ying Ma. Directly optimizing evaluation measures in learning to rank. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 107–114, New York, NY, USA, 2008. ACM.
- [65] Gui-Rong Xue, Hua-Jun Zeng, Zheng Chen, Yong Yu, Wei-Ying Ma, WenSi Xi, and WeiGuo Fan. Optimizing web search using web click-through data. In *CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pp. 118–126, New York, NY, USA, 2004. ACM.
- [66] Yisong Yue, Thomas Finley, Filip Radlinski, and Thorsten Joachims. A support vector method for optimizing average precision. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 271–278, New York, NY, USA, 2007. ACM.

学位論文に関連する論文および口頭 発表

定期刊行誌掲載論文（主論文に関連する原著論文）

1. 数原良彦, 櫻井彰人, “順序ラベルを用いた教師あり自己組織化マップ”, 日本知能情報ファジィ学会論文誌, Vol.26(5), 2014 (to appear).
2. Yoshihiko Suhara, Jun Suzuki, Ryoji Kataoka, “Robust Online Learning to Rank via Selective Pairwise Approach Based on Evaluation Measures”, 人工知能学会論文誌, Vol.28(1), pp.22–33, 2013.
3. 数原良彦, 宮原伸二, 植松幸生, 金田有二, 藤野昭典, 片岡良治, “異なる適合性分布を用いたランキング学習”, 情報処理学会論文誌データベース (TOD-47), Vol.3(3), pp.99–111, 2010.

定期刊行誌掲載論文（その他の論文）

1. 徳永陽子, 数原良彦, 戸田浩之, 鷲崎誠司, “単語の出現度合いを考慮した質問文マルチタスク分類”, 日本データベース学会論文誌, (to appear).
2. 徳永陽子, 数原良彦, 佐藤吉秀, 戸田浩之, 鷲崎誠司, “知名度の地理的広がりを考慮した実世界スポットの地域局所性推定”, 情報処理学会論文誌, Vol.55(9), pp.2203–2215, 2014.
3. 木村昭悟, 数原良彦, 高橋寛幸, 横山達彦, “画像検索でのユーザ行動を利用した大規模画像アノテーション”, 電子情報通信学会論文誌 D, Vol.J96-D(8), pp.1711–1723, 2013.

4. 数原良彦, 藤田尚樹, 廣嶋伸章, 片淵典史, 片岡良治, “地域特有の話題発見を支援するスマートフォン向けマップ型検索システム: 発見探地図エリアダス”, 電子情報通信学会論文誌, Vol.J96-D(5), pp.1300–1312, 2013.
5. 数原良彦, 鈴木潤, 鷺崎誠司, “不均衡データにおける偽陽性率を考慮したスパム判別器のオンライン学習”, 情報処理学会論文誌データベース (TOD-57), Vol.6(2), pp.51–60, 2013.
6. 数原良彦, 植松幸生, 井上孝史, 片岡良治, “ソーシャルブックマークにおけるタグ付与行動に基づくスパマー判別”, 日本データベース学会論文誌, Vol.7(4), pp.31–36, 2009.
7. 数原良彦, 戸田浩之, 櫻井彰人, “ブログ記事を用いた複数話題語間の動作関係抽出手法”, 電子情報通信学会論文誌, Vol.J91-D(3), pp.619–627, 2008.

国際会議論文 (査読付きの full-length papers)

1. Kiyosuke Nishida, Hiroyuki Toda, Takeshi Kurashima, Yoshihiko Suhara, “Probabilistic Identification of Visited Point-of-Interest for Personalized Automatic Check-in”, In *Proceedings of the 2014 ACM International Joint Conference on Pervasive, Ubiquitous Computing (UbiComp2014)*, pp.631–642, 2014.
2. Yoko Tokunaga, Yoshihiko Suhara, Hiroyuki Toda, Seiji Susaki, Yoshimasa Koike, “Multi-class Question Classification Based on Word Occurrence Rates”, In *Proceedings of the 6th International Workshop with Mentors on Databases, Web, Information Management for Young Researchers (iDB2014)*, 2014.
3. Yoko Tanaka, Yoshihiko Suhara, Nobuaki Hiroshima, Hiroyuki Toda, Seiji Susaki, “Snippet Generation by Identifying Attributed Associated Information”, In *Proceedings of the 9th Asia Information Retrieval Societies Conference (AIRS2013)*, pp.50–61, 2013.

4. Yoshihiko Suhara, Hiroyuki Toda, Shuichi Nishioka, Seiji Susaki, “Automatically Generated Spam Detection Based on Sentence-level Topic Information”, In *Proceedings of the 22nd International Conference on World Wide Web Companion (WebQuality2013)*, pp.1157–1160, 2013.
5. Yoshihiko Suhara, Hiroyuki Toda, Akito Sakurai, “Extracting Related Named Entities from Blogosphere for Event Mining”, In *Proceedings of the 2nd International Conference on Ubiquitous Information Management, Communication (ICUIMC2008)*, pp.242–246, 2008.
6. Yoshihiko Suhara, Hiroyuki Toda, Akito Sakurai, “Event mining from the Blogosphere using topic words”, In *Proceedings of the 1st International Conference on Weblogs and Social Media (ICWSM2007)*, 2007.
7. Yoshihiko Suhara, Akito Sakurai, “A Simple Computational Model for Classifying Small String Sets”, In *Proceedings of the 3rd International Conference on Brain-Inspired Information Technology (BrainIT2006)*, pp.270–273, 2006.
8. Yoshihiko Suhara, Akito Sakurai, “Generalization by Categorical Nodes in Recurrent Neural Networks”, In *Proceedings of the 2nd International Conference on Brain-Inspired Information Technology (BrainIT2005)*, pp.161–164, 2005.

その他の国際会議発表

なし

国内学会発表

査読付き

1. 羽田野真由美, 数原良彦, 戸田浩之, 小池義昌, “行動トピックベクトルと地域語特徴を用いた検索キーワードに対する行動タイプ付与”, FIT2014 査読付き論文, 2014.

2. 今井良太, 数原良彦, 戸田浩之, 鷺崎誠司, “Web 文書を利用した POI に関連する地名の抽出”, 情報アクセスシンポジウム (ショート発表), 2013.
3. 田中陽子, 数原良彦, 佐藤吉秀, 戸田浩之, 鷺崎誠司, “知名度の地理的広がりを考慮した実世界スポットの地域局所性推定”, FIT2013 査読付き論文. 2013.
4. 木村昭悟, 数原良彦, 高橋寛幸, 横山竜彦, “画像検索でのユーザ行動を利用した大規模画像アノテーション”, 画像の認識・理解シンポジウム (MIRU2012), 2012.
5. 数原良彦, 植松幸生, 藤野昭典, 片岡良治, “複数情報源を用いた転移学習によるランキング学習”, Web とデータベースに関するフォーラム (WebDB Forum 2009), 2009.
6. 数原良彦, 植松幸夫, 井上孝史, 片岡良治, “ソーシャルブックマークユーザのタグ付与行動に基づくスパマー判別手法”, Web とデータベースに関するフォーラム (WebDB Forum 2008), 2008.
7. 数原良彦, 戸田浩之, 櫻井彰人, “出来事を発掘するマイニング手法”, 第 8 回 AI 若手の集い (MYCOM2007), 2007.
8. 数原良彦, 戸田浩之, 櫻井彰人, “ブログにおけるイベントマイニングのための適切な関連語の抽出”, 第 18 回データ工学ワークショップ (DEWS2007), 2007.

査読なし

1. Brendan Cowan, *Yoshihiko Suhara, Hiroyuki Toda, Yoshimasa Koike, “Active Learning Based on Geographical Orientation for Automatic Transportation Mode Estimation”, IPSJ DBS159, 2014.
2. *数原良彦, 鈴木雅大, 戸田浩之, 鷺崎誠司, “少数の正解ラベルを用いた移動履歴からの移動手手段判定”, JSAI2014, 2014.
3. *清武寛, 徳永陽子, 数原良彦, 戸田浩之, 鷺崎誠司, “クエリログを利用した質問文カテゴリ分類”, 第 114 回 IFAT 研究会, 2014.

4. *徳永陽子, 数原良彦, 戸田浩之, 鷺崎誠司, “単語の出現度合いを考慮した質問文マルチタスク分類”, DEIM2014, 2014.
5. *数原良彦, 戸田浩之, 鷺崎誠司, “地理的損失を考慮したマルチクラス分類による地域推定”, NLP2014, 2014.
6. *数原良彦, 戸田浩之, 西川仁, 鷺崎誠司, “整数計画法を用いた GPS ログからの滞留点と訪問 POI の同時推定”, IPSJ DBS 研究会, 2013.
7. *数原良彦, 鈴木潤, 鷺崎誠司, “構造学習を用いたテキストからの地域イベント情報抽出”, JSAI2013, 2013.
8. *井元麻衣子, 数原良彦, 鷺崎誠司, “併合後の探索コストを考慮した木構造のクラスタリング手法”, JSAI2013, 2013.
9. *吉本暁文, 数原良彦, 船越要, 鷺崎誠司. 複数の移動平均における順序関係に基づくバースト検出. DEIM2013, 2013.
10. *西村駿人, 数原良彦, 鷺崎誠司, “地域特徴語選択を用いたマルチクラス分類による Twitter ユーザの居住地推定”, 第 4 回集合知シンポジウム, 2012.
11. *数原良彦, 鈴木潤, 片岡良治, “偽陽性率に着目したオンライン学習を用いたスパム判別”, FIT2012, 2012.
12. *藤田尚樹, 数原良彦, 片岡良治, “地域特有の話題発見につながるスマートフォン向け検索サービス: 発見探地図エリアダス”, DEIM2012, 2012.
13. *数原良彦, 鈴木潤, 安田宜仁, 小池義昌, 片岡良治. 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. 言語処理学会全国大会 (NLP2011) F3-5, pp.872-875. 2011.
14. *鈴木克典, 湯川高志, 戸田浩之, 数原良彦, 片岡良治. 文書構造を考慮した近接性スコアを用いた文書検索結果ランキング方式. FIT2010, 2010.
15. *宮原伸二, 数原良彦, 小長井俊介, 片岡良治, “検索ログを用いた検索クエリの困難度判定によるクエリ推薦方式”, JSAI2010, 2B3-4, 2010.

16. *数原良彦, 片岡良治, “素性推定器を用いたランキング学習”, JSAI2010, 2A1-4, 2010.
17. *高橋大和, 数原良彦, 小長井俊介, 片岡良治. Web リンクスパム集合検出とスパムリンクを排除したページ重要度. 第 16 回電子情報通信学会 WI2 研究会 (IEICE SIG-WI2-16), 2009.
18. *数原良彦, 植松幸生, 戸田浩之, 井上孝史, 片岡良治. ソーシャルブックマーク数を正解とした検索ランキング学習. 第 23 回人工知能学会全国大会 (JSAI2009), 2009.
19. *数原良彦, 植松幸生, 安田宜仁, 井上孝史, 片岡良治. 全文検索における複合語を考慮した転置リストの併合処理. データ工学と情報マネジメントに関するフォーラム (DEIM2009), C7-2, 2009.
20. *数原良彦, 篠沢佳久, 櫻井彰人, “Folksonomy タグを用いた個人の視点に基づくコンテンツ検索手法”, 第 22 回人工知能学会全国大会 (JSAI2008), 2008.
21. *数原良彦, 篠沢佳久, 櫻井彰人, “Folksonomy におけるタグ情報を用いたコンテンツ推薦手法の提案”, 第 70 回情報処理学会全国大会 (IPSJ2008), 論文集 (1), pp.641-642, 2008.
22. *数原良彦, 戸田浩之, 櫻井彰人, “話題語を手がかりとしたブログからのイベントマイニングの検討”, 情報処理学会研究報告, 2006-NL-176, pp.67-73, 2006.

その他

1. 木村昭悟, 数原良彦, 高橋寛幸, 横山達彦, “画像検索でのユーザ行動解析に基づく大規模画像アノテーション”, 画像ラボ 2014 年 3 月号, 2014.
2. 小池義昌, 片淵典史, 小長井俊介, 安田宜仁, 廣嶋伸章, 藤田尚樹, 数原良彦, 佐藤大祐, “地域情報の発見を支援する時空間マップ型 Web 検索技術”, NTT 技術ジャーナル, Vol.24, No.5, pp.24-28, 2012.

3. 中島悠, 数原良彦, “学生フォーラム 石塚満氏 インタビュー 「日本発の技術を世界へ」”, 人工知能学会誌, Vol.23 No.2, pp.302-304,, 2008.
4. 白鳥成彦, 数原良彦, “学生フォーラム 山西健司氏 インタビュー 「自分の「根っこ」を大切に」”, 人工知能学会誌, Vol.23 No.1, pp.146-148, 2008.

謝辞

本研究を行うにあたり，多くの方々のご指導，ご鞭撻を賜りましたことを，この場をお借りして心より感謝いたします。

指導教官の櫻井彰人先生には，卒業研究に始まり，修士研究，そして博士研究と，長いようであつという間の9年間，たくさんのことを学ばせて頂きました。鋭い洞察力と共に柔らかい人柄を備えた櫻井先生は，私の目指す人物像です。研究のことはもちろん，人としても大きな影響を受けました。深く感謝申し上げます。

お忙しい中，副査を快諾していただき，本論文の執筆にあたり有益なご助言およびご指導を頂いた，鈴木秀男先生，萩原将文先生，榊原康文先生に厚く御礼申し上げます。

管理工学科 COM 系研究室の山口高平先生，飯島正先生，篠沢佳久先生には，学部生時代からお世話になりました。飯島先生，篠沢先生には ae-angels の取り組みを通じて計算機管理の基礎知識，心構えについて学ばせて頂きました。

櫻井研究室，山口研究室の先輩方には大変お世話になりました。特に岡誠氏，岩田梓氏，森田武史氏，黒川茂莉氏には研究議論にとどまらず，精神的な辛さを軽減するためのストレス対処法について学ばせて頂きました。櫻井研究室，山口研究室同期，そして後輩のみなさまにも大変お世話になりました。研究室生活を振り返ると楽しい思い出ばかりなのは，みなさまのおかげです。

日本電信電話株式会社の皆様には大変お世話になりました。上司として大変お世話になった奥雅博氏，大久保雅且氏，片岡良治氏に感謝致します。研究を進める上で様々な後押しをして頂き，ときには厳しく，ときには優しく指導して頂き，企業研究者としての素養を身に付けることができました。インターン時代から入社以降も引き続きお世話になった戸田浩之氏，井上孝史氏，そして入社以降研究指導をしてくださった植松幸生氏，宮原伸二氏に感謝します。仕事はもちろんのこと，企業において研究を進めるお作法，それ以外のことについてもたくさんのことを学ばせて頂きました。井上さんとナーガ・ラジャでカレーを食べながら検

索エンジンについて語った日々が懐かしいです。

入社以降、厳しく、そして厳しく指導して頂いた安田宜仁氏には深く感謝申し上げます。どんなにくだらない質問にも回答して頂き、教科書には載っていない知識をたくさん学ぶことができました。また、本質的なことは何か、目先の瑣末なことに囚われないためのものの考え方を学ばせて頂きました。

日本電信電話株式会社時代に私の立てた研究テーマに応募して来てくださったインターン生のみなさまにも感謝申し上げます。研究者として半人前ながら指導者という立場を経験させて頂いて、研究者としても人間としても一回り大きくなることができました。

日本電信電話株式会社 H20 同期のみなさまにも感謝致します。特に山本千尋氏、甲谷優氏、玉木秀和氏、西川仁氏、山室健氏、西野正彬氏には公私ともにお世話になりました。優秀な同期の存在に刺激を受け、本論文の完成に至ったことと信じています。林寮のあの日々、そして三浦海岸で夜通し語り合った思い出は最高の青春です。

NTT コミュニケーション科学基礎研究所の永田昌明氏、藤野昭典氏、平尾努氏、鈴木潤氏、木村昭悟氏に感謝いたします。永田さんには、CS 研での訪問研究の機会を頂き、それ以降も様々なご支援を頂きました。ありがとうございます。藤野さんには、テーマ研究アドバイザーという立場で機械学習のいろはから教えて頂きました。識別学習は目的関数の設計と目的関数の最適化の2つから成る、ということを人前で語れるようになったのは藤野さんのおかげです。平尾さんの鋭いコメント、そして素晴らしい指導力を横から見て学ばせて頂きました。平尾さんは私が目指す理想の父親像です。鈴木さんにはCS 研滞在期間中の指導にとどまらず、その後も共同研究を通じて引き続き指導をして頂き、大変お世話になりました。ロケーションが離れているため、電話での議論が中心でしたが、ときに2時間を超える電話ごしの議論は、間違いなく自分の財産になっています。木村昭悟さんには共同研究者という立場で実験や論文執筆を通じて多くのことを学ばせて頂きました。また、ベイズ勉強会参加者のみなさまにもこの場をお借りして感謝申し上げます。ベイズ勉強会参加を通じて論文に書かれていない機械学習に関わるノウハウを多く学ばせて頂きました。

NTT メディアインテリジェンス研究所の平野徹氏、小林のぞみ氏、貞光九月氏にはお世話になりました。いつでも気軽に議論に応じて頂き、部署の垣根を超え

て研究を進めることができました。特に貞光さんには、機械学習に関する輪講を通じて貞光さんの持つたくさんのノウハウを教えて頂きました。また、NTT レゾナントの西田京介氏には共著論文の執筆を通じて多くのことを学ばせて頂きました。その他にも優秀な先輩、同僚、後輩たちの存在から多くの刺激を受けました。みなさまとの、ちょっとした立ち話の積み重ねが研究成果の創出、そして研究者としての基礎を構築したと信じています。

株式会社リクルートホールディングスの石山洸氏には博士研究にご理解を頂き、入社まもない身分にも関わらず業務遂行の裁量を与えて頂きました。石山さんのおかげで、博士研究の仕上げに時間を割き、本論文を完成することができました。感謝申し上げます。

研究を進める上では、所属を超えて多くの方にお世話になりました。金田有二氏には、検索エンジンのいろは、資料の作り方、仕様書の書き方、研究の進め方など、1 から 10 まで手厚い指導をして頂きました。比戸将平氏、徳永拓之氏、山田浩之氏、海野裕也氏、佐藤一誠氏とは、情報検索と機械学習に関する議論を通じて多くのことを学ばせて頂きました。年齢や境遇が近いことから、研究議論に留まらず、明るい家庭構築のための様々な意見交換をさせていただきました。

IIR 勉強会主催者の山下達雄氏と IIR 勉強会参加メンバーに感謝します。IIR 勉強会、そしてその後の MG 勉強会の参加経験が、自分の情報検索に関する専門知識の基礎になっています。生駒 PRML 読書会の参加者にも感謝致します。特に主催者である @naoya.t 氏、@Prunus1350 氏、そして勉強会において特に多くのことを学ばせて頂いた中谷秀洋氏、坪坂正志氏に感謝致します。私の機械学習に関する基礎知識の多くは本勉強会の議論を通じて身につけられたものと信じています。また、Tokyo.NLP 主催者の奥野陽氏、DSIRNLP 主催者の佐藤敏紀氏をはじめとする、勉強会参加メンバーのみなさまに感謝申し上げます。ここでは、全員の名前を挙げることはできませんが、勉強会での発表や議論に刺激を受け、研究活動を邁進できたと信じています。

最後に、私の人生を支えてくれた家族に感謝します。両親、兄、愛猫をはじめ、尊敬する祖父母、そして今は亡き祖父母と愛犬という家族があつての自分だと思えます。妻には心身ともに私のすべてを支えてもらっています。たくさんの負担をかけてしまい申し訳ありません。本当にありがとうございます。博士研究に時間を割いている間、息子と一緒に過ごす時間を十分に取れず、申し訳なく思っ

います。これからは、たくさん遊ぼう。