DEIM Forum 2018 F5-2

組合せ依存型ランキング学習 – 競馬への応用事例

三枝 守輝　　加藤　　誠　　浅野 泰仁

京都大学大学院情報学研究科〒 606–8501 京都府京都市左京区吉田本町

E-mail: †saigusa@db.soc.i.kyoto-u.ac.jp, ††kato@dl.kuis.kyoto-u.ac.jp, † † †asano@i.kyoto-u.ac.jp

あらまし　本論文では，ランク付けの候補となるエンティティ同士が互いに影響を与え合うことでエンティティの順位が決定される領域を対象とした，組合せ依存型のランキング学習の手法を提案する．競馬・競輪などのレースや選挙といった競争では，エンティティの順位はエンティティ自身だけではなく，競争相手となるエンティティの特徴からも影響を受けて決定される．このような領域において，Factorization Machine を応用した，組合せを考慮したランキング手法を提案するとともに，その応用事例として競馬の勝ち馬予測に適用し，従来の手法と比較する．

キーワード　ランキング学習，エンティティランキング，Factorization Machine

1. は じ め に

ランキングは人気の高いコンテンツであると同時に，意思決定の基準となる重要な情報である．例えば，商品を購入する際に，売れ筋のランキングはどの商品を実際に購入するかを決めるための重要な手がかりとなる．そのため，ランキングはコンテンツとしての人気とその有用性から，多くの Web サイトで公開されている．

エンティティに関連する様々な特徴量を利用してランキングを生成する手法は数多く提案されてきたが，近年では，ランキングを機械学習のアプローチを活用して推定する手法が研究されてきている．一般に，エンティティの順位付け問題を解決するために機械学習の手法を利用したものをランキング学習 (Learning to Rank) と呼ぶ [1].

ランキング学習は主に情報検索の分野において，Web 上に存在する無数の文書の中から，ユーザの検索要求を表現したクエリに関連するものを選出することに利用されている．クエリに対して関連性があることを適合性といい，その関連度合いは適合度というスコアで表現される．情報検索の分野でのランキング学習では，実際に観測された文書の選好の順序から，文書の適合度の学習と推定を行なっている．

情報検索の分野では，文書の適合性について，適合性の独立性が成立することを前提としている．適合性の独立性とは，与えられた検索要求に対する個々の文書の適合性は，その検索要求およびその文書のみに依存する [2]，という性質を指す．言い換えれば，文書を順位付ける際には，個々の文書の適合度はその他の文書から影響を受けず，その文書と検索要求だけに依存して決定されることを意味している．

しかし，Web 上の文書以外のエンティティを順位付ける場合，適合性の独立性のように，エンティティの順位を決める適合度のようなスコアが，その他のエンティティから独立に決定されることを前提とすることは難しい場合もある．例えば競馬や選挙のような競争では，エンティティの順位はエンティティ自らの能力だけではなく，その他のエンティティの能力からも大きく影響を受けて決定される．このような領域において，情報検索の分野で利用されてきたランキング学習の手法をそのまま用いては，予測精度が芳しくないことが予想される．

情報検索において，文書のランキングに対する評価指標の中には，Expected Reciprocal Rank (ERR) のように，ある文書がユーザに与える効用がその他の文書の適合度に影響を受けるという限界効用の逓減を取り込んでいる指標も存在する [3]．この指標を利用してランキング学習のモデルを評価することで，個々の文書が与える効用がその他の文書により影響を受けて決定されることを考慮することが可能ではあるものの，個々の文書の適合度そのものがその他の文書から影響を受けることを考慮できているわけではない．したがって，文書の適合度そのものは，文書そのものと情報要求の二つのみにより決定されていることは変わらない．

本論文では，情報検索の分野では考慮されなかった，順位付けの対象となるエンティティ同士が互いに影響を及ぼしあうことでランキングが決定される領域を対象とした，組合せ依存型のランキング学習の手法を提案する．組合せ依存型ランキング学習では，エンティティ同士が互いに影響を及ぼし，順位が決定されることを考慮するために，Factorization Machine (FM) を応用する．FM は複数の特徴量の交互作用を推定可能なモデルであり，エンティティやエンティティを取り巻く環境を表す特徴量同士の影響を考慮したランキングの学習が可能なことが期待される．また，本論文で提案する手法が有効であることを示すため，エンティティ同士が互いに影響を及ぼすことで順位が変動する一つの応用先として競馬を対象とし，組合せ依存型ランキング学習の実験を行なった．

本論文の構成は以下の通りである．2. 節ではエンティティのランキング学習にまつわる関連研究，および，組合せ依存型ランキング学習との関係について述べる．3. 節では対象となる領域と問題設定を説明し，組合せ依存型ランキング学習の適用方法について述べる．4. 節では応用事例として競馬の順位予測を応用事例として，実験結果を示す．5. 節では今後の課題とともに本論文の結論を述べる．

2. 関連研究

本節では，順位付けの対象となるエンティティの特徴同士の交互作用を考慮したランキング推定を行うことが可能な Factorization Machine (FM) の説明を行う．

Factorization Machine (FM) は，サポートベクトルマシン (SVM) のような汎用的な予測器であり，全ての特徴量同士の交互作用による影響をモデルに落とし込んだものである [4]．

一般に，予測器の学習とは，実数値からなる n 次元のベクトル $\mathbf{x} \in \mathbb{R}^n$ を入力とし，定義域がタスクに応じて設定される T を出力とするような関数 $y : \mathbb{R}^n \to T$ を推定することである．例えば，タスクが分類であれば $\{+, -\}$ が，回帰であれば $\mathbb{R}$ が T として設定される．

FM では，入力変数 $\mathbf{x} \in \mathbb{R}^n$ に含まれる d 個の要素同士の交互作用による影響を考慮することが可能で，特に $d = 2$ である場合のモデル式は次のように表される．

$$\hat{y}(\mathbf{x}) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j \tag{1}$$

モデル式 (1) において推定されるべきパラメータは以下である．

$$w_0 \in \mathbb{R}, \mathbf{w} \in \mathbb{R}^n, \mathbf{V} \in \mathbb{R}^{n \times k} \tag{2}$$

w_0 は，各事例において共有されるグローバルバイアスであり，w_i は，入力変数の i 次元目の要素による影響をモデル化している．また，$\langle \mathbf{v}_i, \mathbf{v}_j \rangle$ は，入力変数の i 次元目と j 次元目の要素同士の交互作用による影響の強さをモデル化している．モデル式 (1) における $\langle \mathbf{v}_i, \mathbf{v}_j \rangle$ は，二つの実数ベクトル $\mathbf{v}_i, \mathbf{v}_j \in \mathbb{R}^k$ の内積を表しており，次のように計算される．

$$\langle \mathbf{v}_i, \mathbf{v}_j \rangle = \sum_{f=1}^{k} v_{i,f} v_{j,f} \tag{3}$$

式 (2) における $\mathbf{V}$ の行方向の次数 $k \in \mathbb{R}^+$ は，FM におけるハイパーパラメータである．

モデルの出力に対して，各パラメータで偏微分を行った結果は以下の式 (4) となる．

$$\frac{\partial}{\partial \theta} \hat{y}(\mathbf{x}) = \begin{cases} 1, & \text{if } \theta = w_0 \\ x_i & \text{if } \theta = w_i \\ x_i \sum_{j=1}^{n} v_{j,f} x_j - v_{i,f} x_i^2, & \text{if } \theta = v_{i,f} \end{cases} \tag{4}$$

FM のモデル式はパラメータについて微分が可能であるため，タスクに応じて設定される目的関数が微分可能であれば，パラメータの更新が可能である．そのため，FM は様々なタスクに応用することが可能であり，特にレーティング予測や商品推薦，情報検索といったタスクで利用されている．FM を提案した Rendle は，ユーザの商品に対するレーティングを，データセットに対する平均平方二乗誤差 (RMSE) を最小とするような回帰のタスクとして FM を利用できることを示した [4]．また，Yuan らは，上位 N 件の商品推薦タスクに対して，FM の手法において，ブースティングを応用した BoostFM を提案している [5]．情報検索のタスクにおいては，Qiang らは Twitter といったマイクロブログサービスでの検索において，FM をランキング推定に応用する手法を提案している [6]．

FM の特徴の一つとして，特徴量同士の交互作用項の推定方法があげられる．一般的なモデルのように，交互作用項 $x_i x_j$ に対する重みを w_{ij} として推定するのではなく，$\hat{w}_{ij} = \langle \mathbf{v}_i, \mathbf{v}_j \rangle$ のように二つのベクトルの内積として推定することは，与えられた事例において観測されない交互作用項に対する重みの推定が可能であるといった利点を生む．例えば，入力変数の次元数が 3 であり，入力変数に含まれる 2 つの要素同士の交互作用に対する重みを推定したい場合を考える．与えられた事例が以下の

$$\mathbf{x}^{(1)} = (1.5, 0.0, 0.0)$$
$$\mathbf{x}^{(2)} = (0.5, 0.1, 0.0)$$
$$\mathbf{x}^{(3)} = (0.0, 0.3, 0.2)$$

3 つである場合，考えうる交互作用の総通り数は ${}_3\mathrm{C}_2 = 3$ であり，それぞれの重みは w_{12}, w_{13}, w_{23} と表現できる．SVM のような一般的なモデルの場合，この 3 通りの事例では，$\mathbf{x}^{(2)}, \mathbf{x}^{(3)}$ からそれぞれ w_{12}, w_{23} を推定することができるが，どの事例についても $x_1^{(i)} x_3^{(i)} = 0$ であるため，w_{13} は推定されない (つまり $w_{13} = 0$ となる)．一方，FM の場合は，$\hat{w}_{12} = \langle \mathbf{v}_1, \mathbf{v}_2 \rangle, \hat{w}_{23} = \langle \mathbf{v}_2, \mathbf{v}_3 \rangle$ を推定するために，因子ベクトル $\mathbf{v}_1, \mathbf{v}_3$ が求められている．したがって，$\langle \mathbf{v}_1, \mathbf{v}_3 \rangle$ も求めることができ，交互作用項 $x_1 x_3$ に対する重み $\hat{w}_{13}$ を推定することが可能である．

上述の例では，入力変数の次元数が 3 であったため，交互作用項が 0 となる組合せは少なくて済む．しかし，入力変数の次元が大きく，かつ疎である場合には，交互作用項が 0 となる場合が多い．FM はこのように入力変数が疎である場合にも，交互作用項の推定が可能であるといった特徴を持つ．また，入力変数が密である場合においても，汎化能力が高く，訓練データに対して過学習を引き起こしにくいといった利点が存在する．

我々が提案する組合せ依存型ランキング学習では，FM の入力変数を拡張し，順位付けの対象となるエンティティの組合せを表現するベクトルを組み込むことで，組合せに依存したランキングの推定を可能にしている．

3. 手　　法

本節では，まず順位付けの対象となるエンティティの組合せにより各エンティティの特徴量が互いに影響を与え合うことで順位が決定される領域に対する問題設定を行う．次に，そのような問題設定における特徴量の抽出方法について述べ，最後に，組合せ依存型ランキング学習の性質についての説明を行う．

3.1 問題設定

E をあるクラスのエンティティ集合とし，E に属するエンティティ数を M で表す．また，集合 E に属するエンティティを $e_i \in E, i \in [1, M]$ とする．E に属する複数のエンティティにより決定される j 番目の組合せを $c^{(j)}$ とし，$c^{(j)}$ に含まれるエンティティ数を $m_j = |c^{(j)}|$ と表す．さらに，エンティティ順序 $\preceq_j$ を $c^{(j)}$ 上の全順序として定義する．

この時，$c^{(j)}$ について観測された順序集合 $O_{\preceq_j}$ から，関数 $y(\mathbf{x_i^{(j)}})$ を学習することが我々の主目的である．ただし，ここでの y は，式 (1) と同様であり，入力変数である $\mathbf{x}_i^{(j)}$ は次節において説明される組合せ依存型ランキング学習の特徴量である．我々は y を，エンティティ順序 $\preceq_j$ を保つ，つまり $e_i^{(j)}, e_k^{(j)} \in c^{(j)}$ について $e_i^{(j)} \preceq_j e_k^{(j)} \Rightarrow y(\mathbf{x}_i^{(j)}) \leqq y(\mathbf{x}_k^{(j)})$ となるように推定する．

3.2 組合せ依存型特徴量の作成

ここで，組合せ依存型ランキング学習の入力となる特徴量を定義するに当たって次の 3 つのベクトルを定義する．

（1） エンティティインデックスベクトル

（2） 組合せインデックスベクトル

（3） 文脈ベクトル

エンティティインデックスベクトルは，組合せ $c^{(j)}$ において，エンティティe_i が順位推定の対象であることを識別するためのベクトルであり，以下の (5) で表現されるような $\mathbf{x}_i^e \in \{0,1\}^M$ として定義される．

$$\mathbf{x}_i^e = (\underbrace{0,\dots,0}_{i-1\text{ 個}}, 1, \underbrace{0,\dots,0}_{M-i\text{ 個}}) \tag{5}$$

組合せインデックスベクトルは，組合せ $c^{(j)}$ に含まれるエンティティ集合を識別するためのベクトルであり，以下の (6) で表現されるような $\mathbf{x}^{c^{(j)}} \in \{0,1\}^M$ として定義される．

$$\mathbf{x}^{c^{(j)}} = (\underbrace{0,1,1,0,\dots,1,0,\dots}_{M\text{ 個}}), \bar{z}(\mathbf{x}^{c^{(j)}}) = m_j \tag{6}$$

ただし，$\bar{z}(\mathbf{x})$ は，ベクトル $\mathbf{x}$ の要素の中で，0 でないものの数を出力する関数である．

文脈ベクトルは，モデルの利用者に応じて自由に設定されるベクトルであり，以下の (7) のように定義される．

$$\mathbf{x}_i^f \in \mathbb{R}^n \tag{7}$$

具体的には，エンティティe_i の特徴量や，組合せに含まれる m_j 個 のエンティティの特徴量を並べたもの，それらに加えて組合せにおいて共有される周囲の環境を加えたものといったものが利用できる．

上述の (5)，(6)，(7) を利用して，組合せ $c^{(j)}$ における i 番目のエンティティの特徴ベクトル $\mathbf{x}_i^{(j)} \in \mathbb{R}^{2M+n}$ は次の式 (8) ように設定される．

$$\mathbf{x}_i^{(j)} = \mathbf{x}_i^e \frown \mathbf{x}^c \frown \mathbf{x}_i^f \tag{8}$$

ただし，演算子 $\frown$ はベクトルの結合を意味する．例えば，二つの実数ベクトル $\mathbf{p} = (p_1, p_2), \mathbf{q} = (q_1, q_2)$ について，$\mathbf{p}\frown\mathbf{q} = (p_1, p_2, q_1, q_2)$ である．

例えば，$M = 5$ であり，1 番目の組合せ $c^{(1)}$ においてエンティティe_1, e_3, e_4 を順位付けているとする．このとき，エンティティインデックスベクトルはそれぞれ

$$\mathbf{x}_1^e = (1,0,0,0,0) \quad \mathbf{x}_3^e = (0,0,1,0,0) \quad \mathbf{x}_4^e = (0,0,0,1,0)$$

であり，組合せインデックスベクトルは

$$\mathbf{x}^{c^{(1)}} = (1,0,1,1,0)$$

となる．それぞれのエンティティの文脈ベクトルが

$$\mathbf{x}_1^f = (x_{11}^f, x_{12}^f) \quad \mathbf{x}_3^f = (x_{31}^f, x_{32}^f) \quad \mathbf{x}_4^f = (x_{41}^f, x_{42}^f)$$

のように 2 次元で表現されているとすれば，以下のベクトルがモデルの入力特徴量となる．

$$\mathbf{x}_1^{(1)} = (1,0,0,0,0,1,0,1,1,0,x_{11}^f,x_{12}^f)$$
$$\mathbf{x}_3^{(1)} = (0,0,1,0,0,1,0,1,1,0,x_{31}^f,x_{32}^f)$$
$$\mathbf{x}_4^{(1)} = (0,0,0,1,0,1,0,1,1,0,x_{41}^f,x_{42}^f)$$

3.3 組合せ依存型ランキング学習

この節では，入力変数を前節のように作成することにより，エンティティ同士の特徴量の交互作用を考慮し，その他のエンティティから影響を受けてスコアが決定されることを説明する．

前節において説明した特徴量 $\mathbf{x}_i^{(j)}$ を入力とし，組合せ $c^{(j)}$ における i 番目のエンティティの順位を推定する，FM を利用した回帰問題を解く．その結果として推定されるパラメータは，組合せ依存型ランキング学習においては次のようになる．

$$w_0 \in \mathbb{R}, \mathbf{w} \in \mathbb{R}^{2M+n}, \mathbf{V} \in \mathbb{R}^{(2M+n)\times k} \tag{9}$$

(9) における交互作用項に対する重みの因子ベクトル $\mathbf{v}_i, i \in [1, (2M+n)\times k]$ について考えれば，1 行目から M 行目までの因子ベクトルは i 番目のエンティティ固有の性質を表す因子ベクトルであり，$M+1$ 行目から $2M$ 行目までの因子ベクトルは，組合せにおける i 番目のエンティティの性質を表す因子ベクトルとして解釈することができる．これら因子ベクトルを利用して，同じ組合せに含まれるエンティティが互いに影響を与え合うことを表現可能であることを示す．

j 番目の組合せに含まれる 2 つのエンティティ$e_i^{(j)}, e_k^{(k)}$ の関係について，次の 4 つを考えることができる．

（1） $e_i^{(j)}, e_k^{(k)}$ は共にパフォーマンスを向上させ合う．

（2） $e_i^{(j)}, e_k^{(j)}$ は共にパフォーマンスを低下させ合う．

（3） $e_i^{(j)}$ は $e_k^{(j)}$ に対して『得手』であり，$e_i^{(j)}$ のパフォーマンスは向上し，$e_k^{(j)}$ のパフォーマンスは低下する．

（4） $e_i^{(j)}$ は $e_k^{(j)}$ に対して『不得手』であり，$e_i^{(j)}$ のパフォーマンスは低下し，$e_k^{(j)}$ のパフォーマンスは向上する．

いま，エンティティ$e_i^{(j)}$ についてのモデルの特徴量である $x_i^{(j)}$ について，$\hat{y}(x_i^{(j)})$ の値が小さいほど，そのエンティティは高順位で，より良いパフォーマンスをすると推定されることを表現していると考える．$x_i^{(j)}x_{M+k}^{(j)} > 0$ かつ $x_k^{(j)}x_{M+i}^{(j)} > 0$ であるから，1 のような関係性を表現するためには，式 (1) より $\langle \mathbf{v}_i, \mathbf{v}_{M+k}\rangle$ と $\langle \mathbf{v}_k, \mathbf{v}_{M+i}\rangle$ が負の値を取ればよく，お互いのパフォーマンスを向上させ合い，より良い順位が推定されるように作用する．また，$\langle \mathbf{v}_i, \mathbf{v}_{M+k}\rangle < 0$ かつ $\langle \mathbf{v}_k, \mathbf{v}_{M+i}\rangle > 0$ であれば，3 の関係性を表現していると言える．このような因子ベクトルの組合せは実現可能であり，因子ベクトルの次元数が 2 である場合は，$\mathbf{v}_i = (1,1), \mathbf{v}_{M+k} = (-1,-1)$ や

$\mathbf{v}_k = (-1, 1), \mathbf{v}_{M+k} = (-1, 1)$ がその具体例である．2，4 の関係性については，それぞれ 1,3 の場合の議論と同様である．

このように FM と組合せ依存型の特徴量を利用することにより，組合せ $c^{(j)}$ が決まることにより変動するエンティティのパフォーマンスをモデルに落とし込むことができ，その結果として変動する順位を推定することが可能となる．

4. 実　　験

本節では，組合せ依存型ランキング学習の応用事例として競走馬の順位予測を取り上げる．我々のタスクには公開されたデータセットが存在しないため，まずデータセットの作成方法について説明するとともに，特徴量の作成方法と提案手法のモデルの学習方法について述べる．つぎに，競馬の勝ち馬予測の研究においてベースラインとしてよく利用される手法を紹介する．最後には，データセットについて提案する組合せ依存型ランキング学習による結果とベースラインによる結果を評価し，比較を行う．

4.1 データセット

本論文では，2010 年から 2016 年の間に JRA の 10 の競馬場(注1)で開催されたレースを対象としてデータの収集を行った．対象レースの情報は，競走馬については競馬新聞＆スピード指数(注2)において公開されている情報から抽出し，競走馬に騎乗した騎手については netkeiba(注3)において公開されている情報から抽出を行った．表 1 にデータセットについての統計情報を示す．

モデルで利用される特徴量については，Lessmann らによる特徴量の抽出にしたがった [7]．この特徴量は，提案手法においては文脈ベクトル，ベースラインとなる手法においてはモデルの入力変数として利用する．抽出される特徴量の中には，競走馬や騎手の過去のレースにおけるパフォーマンスを利用したものが存在するため，表 1 に示される全レースのデータに含まれる一般競争(注4)，特別競争のうち，各出走馬が過去に比較的多くの出走を経験している 1600 万円以下とオープン戦のデータ，合計 2639 レースを対象として次節に示す方法で特徴量の抽出を行った．

4.2 提案手法の競馬への応用方法

この節では，提案手法を競走馬の着順予測に応用するための具体的な方法について述べる．まず，前節で説明したデータセットを対象に，提案手法に適切な形式の特徴量を作成する手法について説明し，次に，モデルをどのように学習させるかについて説明する．

4.2.1 特徴量の作成

M をデータセットに含まれているユニークな競走馬の総数とする．まず，特徴量抽出の対象となるデータに含まれる競走馬名から各競走馬に対して識別子を割り振り，各競走馬の識別子を M 次元のベクトルの各次元に 1 対 1 に対応させる．各事例において，着順の推定の対象となる競走馬に対応する次元が 1 であり，その他の要素が 0 であるような one-hot ベクトルをエンティティインデックスベクトルとし，競争相手として同じレースに出走する競走馬に対応する次元が 1 であり，その他の要素が 0 であるようなベクトルを組合せインデックスベクトルとする．文脈ベクトルについては，前節で述べた Lessmann らの先行研究で提案された特徴量を各競走馬の過去の出走成績から抽出する．このようにして各事例に対して得られる 3 のベクトルを結合し，モデルの入力とする．

表 1　全競馬場，および，各競馬場ごとのレース数と競走馬数

競馬場名	レース数	出走数	平均出走馬数
札幌競馬場	1008	12783	12.68
函館競馬場	1248	15472	12.40
福島競馬場	1502	22330	14.87
新潟競馬場	2072	30788	14.86
東京競馬場	3517	52392	14.90
中山競馬場	3153	46968	14.90
中京競馬場	1575	24326	15,46
京都競馬場	3769	53598	14.12
阪神競馬場	3272	47918	14.64
小倉競馬場	1982	29449	14.86
全競馬場	23098	38631	14.51

4.2.2 モデルの学習

モデルの出力 $\hat{y}$ は各競走馬のレースでの着順である．学習の段階では次に示すような正規化を行った．

$$y_i^{(j)} = \frac{FP_i^{(j)} - FP_{min}^{(j)}}{FP_{max}^{(j)} - FP_{min}^{(j)}} \tag{10}$$

ただし，式 (10) において $FP_i^{(j)}$ は j 番目のレースでの i 番目の競走馬の着順を表し，$FP_{min}^{(j)}$，$FP_{max}^{(j)}$ はそれぞれそのレースでの競走馬の着順の最小値と最大値を表している．また，$y_i^{(j)}$ は $FP_i^{(j)}$ に対応した，変換後の値を表現する．この変換により，各競走馬の変換後の着順は最小値が 0，最大値が 1 となるため，出走馬数が異なるレースに対しても同一のモデルで学習を行うことが可能になる．

これまでに述べた特徴量と着順のデータを利用し，変換した実際の着順とその予測値の 2 乗誤差の総和が，与えられたトレーニングデータに対して最小となるようにモデルの学習を行う．この際のパラメータの最適化手法としては，確率的勾配降下法 (SGD) を利用する．

4.3 2 段階条件付きロジスティック回帰モデル

競馬の勝ち馬予測のタスクにおいて，ベースラインとして利用される手法は Benter による条件付きロジスティック回帰を利用した 2 段階の予測モデルである [8]．このモデルの特徴は，勝ち馬予測を 2 段階に分け，各競走馬の過去のレースでの成績から得られる競走馬固有の競争能力と，将来出走するレースにおいて大衆が予想する各競走馬の勝ちやすさを表現するオッズの 2 つの情報から包括的に予測を行うという点である．

Edelman が指摘するように，競走馬のオッズは勝率との間に

(注1)：札幌，函館，福島，新潟，東京，中山，中京，京都，阪神，小倉競馬場
(注2)：http://jiro8.sakura.ne.jp/
(注3)：http://www.netkeiba.com/
(注4)：新馬，未勝利，500 万円以下，1000 万円以下，1600 万円以下

相関関係があるため，その他の入力変数と同時に利用することは，モデルの予測に過度な影響を与えてしまい，競走馬固有の競走能力を表す変数からの影響を隠してしまうといった問題がある [9]．そのため今回の実験においては，競走馬の特徴同士が互いに影響を与えて順位を変動させることを確かめるために，オッズを特徴量として利用しない．したがって，比較対象となる 2 段階条件付きロジスティック回帰モデルにおいても，オッズを利用しない第 1 段階の出力を利用する．

第 1 段階においては，競走馬の過去のレースにおけるパフォーマンスや騎乗予定の騎手の成績といった情報に基づいて，各出走馬の勝率を条件付きロジスティック回帰を利用して導出する．レース r において，$m^{(r)}$ 頭の競走馬が出走する時，このモデルの第 1 段階での入力は式 (11)，出力となる各競走馬の勝率は式 (12) で表される．ただし，k は各競走馬についての特徴量の次元数である．

$$\mathbf{X}^{(r)} = \begin{pmatrix} x_{11}^{(r)} & \cdots & x_{1k}^{(r)} \\ \vdots & \ddots & \vdots \\ x_{m^{(r)}1}^{(r)} & \cdots & x_{m^{(r)}k}^{(r)} \end{pmatrix} \tag{11}$$

$$\mathbf{p^{(r)}} = \left(p_1^{(r)}, p_2^{(r)}, \cdots, p_{m^{(r)}}^{(r)}\right)^{\mathrm{T}} \tag{12}$$

(11) を入力として (12) を推定するにあたり，レース r における各競走馬の勝率のロジットをリンク関数とし，線形予測子を $\mathbf{X}^{(r)}\beta$ として，(13) で表されるパラメータ β を推定する．(14) は，$\mathbf{p}^{(r)}$ について解くと (15) となる．

$$\beta = (\beta_0, \beta_1, \ldots, \beta_k)^{\mathrm{T}} \tag{13}$$

$$\ln\left(\frac{\mathbf{p}^{(r)}}{\mathbf{1} - \mathbf{p}^{(r)}}\right) = \mathbf{X}^{(r)}\beta \tag{14}$$

$$\mathbf{p}^{(r)} = \frac{\mathbf{1}}{\mathbf{1} + \mathrm{e}^{-\mathbf{X}^{(r)}\beta}} \tag{15}$$

パラメータ β は，(15) において，実際のレース r において勝利した競走馬に対して推定される勝率 $p_w^{(r)}$ の総乗である尤度 (16) がデータセットを通じてもっとも大きくなるように推定される．このようにして推定されたパラメータ β をもとに，レース r における競走馬 h の勝率は (17) により推定される．

$$l(\beta) \propto \prod_{r=1}^{R} p_w^{(r)} \tag{16}$$

$$p_h^{(r)} = \frac{\mathrm{e}^{\mathbf{X}_h^{(r)}\beta}}{\sum_{j\in m^{(k)}} \mathrm{e}^{\mathbf{X}_j^{(r)}\beta}} \tag{17}$$

第 2 段階においては，第 1 段階の出力である各競走馬の勝率と，出走するレースにおいて各競走馬に設定されるオッズの二つを入力として，再び各競走馬の勝率を条件付きロジスティック回帰を利用して導出する．オッズを各競走馬の「勝ちやすさ」として表現するために，(18) により変換する．最後に，(18), (19) を入力，各競走馬の勝率を出力として，第 1 段階と同様の推定を行う．

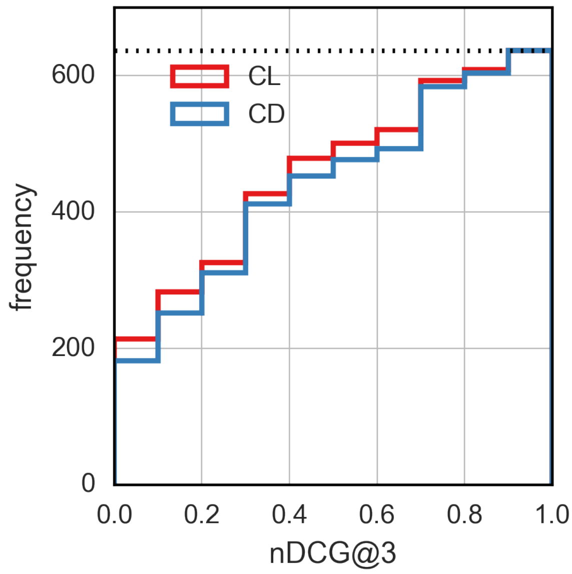

図 1　各手法の nDCG@3 の累積度数分布

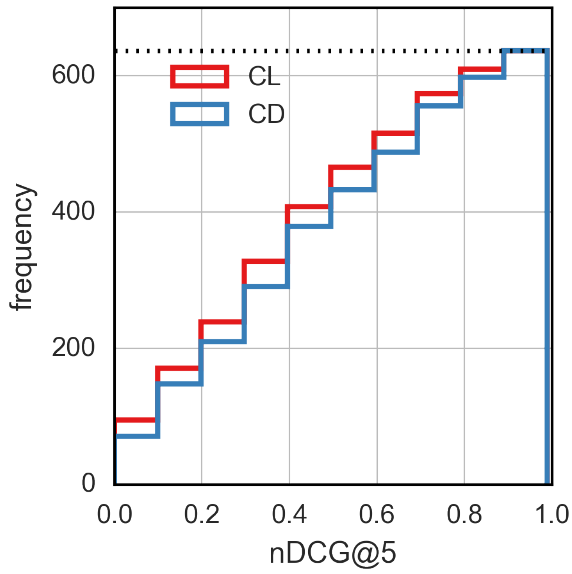

図 2　各手法の nDCG@5 の累積度数分布

$$\mathbf{q}^{(r)} = \left(\frac{1}{d_1^{(r)}}, \frac{1}{d_2^{(r)}}, \cdots, \frac{1}{d_{m^{(r)}}^{(r)}}\right) \tag{18}$$

モデルが最終的に出力するレース r における競走馬 h の勝率は (19) により求められる．

$$\mathbf{p}_h^{(r)\prime} = \frac{\mathrm{e}^{p_h^{(r)}\gamma_1 + q_h^{(r)}\gamma_2}}{\sum_{j\in m^{(r)}} \mathrm{e}^{p_j^{(r)}\gamma_1 + q_j^{(r)}\gamma_2}} \tag{19}$$

4.4　実験結果とモデルの比較

今回の実験において，提案手法では 4.2 で説明した特徴量を

表 2　各手法の評価指標の値

	ベースライン手法	提案手法
nDCG@3 の平均値	0.314	0.345
nDCG@5 の平均値	0.412	0.446

入力とし，各競走馬の着順を出力とするモデルの学習を行っている．その一方で，ベースラインの手法である条件付きロジスティック回帰モデルでは，その出力は各競走馬の勝率の推定値である．出力の意味が異なる両手法を比較する為に，提案手法については着順の予測値を昇順に，ベースラインの手法については勝率の推定値を降順に並び替えることで各モデルの出力と 1 対 1 に対応するランキングを作成した．このようにしてランキングを導出し，テストデータとして用意した 639 件のレースに対する nDCG@3 と nDCG@5 を評価指標として，各手法のパフォーマンスの比較を行った結果をそれぞれ図 1，図 2 に示す．また，各レースに対する nDCG@3，nDCG@5 の値の平均値を表 2 に示す．

図 1，図 2 はテストデータに含まれる各レースに対する nDCG の値の累積度数分布表であり，CL(Conditional Logit) はベースライン，CD(Conditional Dependent) は提案手法のものを表している．提案手法のモデルのハイパーパラメータである k の値は 8 とした．いずれの図からも，提案手法はベースラインと比較して全体的に評価指標が高い値を出していることが分かる．また，表 2 からも，提案手法はベースラインよりも評価指標の値の平均値が高く，より良いパフォーマンスを実現していることが分かる．

5. ま と め

本論文では，順位付けの対象となるエンティティの組合せに応じて各エンティティの特徴量同士が互いに与える影響を考慮したエンティティの順位付けの問題に取り組んだ．競馬や選挙のような，ランキングがエンティティ同士の特徴に応じて変動しうる領域において有効なエンティティランキングの手法として組合せ依存型ランキング学習を提案し，競馬を具体的な応用事例として実験を行った．実験では，組合せ依存型ランキング学習によって既存の勝ち馬予測の手法を上回るパフォーマンスが得られることが明らかになった．

今後の課題としては，ランキングにおいて特に重要となる上位 k 件 (例えば競馬では $k = 3$) の推定を重視したモデルの学習や，学習されるパラメータの一つである因子行列 V の値を利用したエンティティの性質の説明やクラスタリングの手法の提案などを挙げる．

謝辞　本研究は，JSPS 科研費 JP17H06099 JP15K00423 JP26700009 の助成を受けたものです．ここで心より感謝申し上げます．

文　　献

[1] Tie-Yan Liu, "Learning to Rank for Information Retrieval", Foundations and Trends in Information Retrieval, Vol. 3, No. 3, pp. 225–331, 2009.

[2] 酒井 哲也, "情報アクセス評価方法論", 2015.

[3] Chapelle Olivier and Metlzer Donald and Zhang Ya and Grinspan Pierre, "Expected reciprocal rank for graded relevance", Proceedings of the 18th ACM conference on Information and knowledge management, pp. 621–630, 2009.

[4] Rendle, Stefen. "Factorization machines", Data Mining (ICDM), 2010 IEEE 10th International Conference on, pp. 995–1000, 2010.

[5] Yuan, Fajie and Guo, Guibing and Jose, Joemon M and Chen, Long and Yu, Haitao and Zhang, Weinan, "BoostFM: Boosted Factorization Machines for Top-N Feature-based Recommendation", Proceedings of the 22nd International Conference on Intelligent User Interfaces, pp. 45–54, 2017.

[6] R. Qiang, F. Liang, J, Yang. "Exploiting ranking factorization machines for microblog retrieval", Proceedings of the 22nd ACM international conference on Conference on information & knowledge management, pp. 1783–1788, 2013.

[7] Lessman, Stefan and Sung, Ming-Chien and Johnson, Johnnie EV. "Adapting least-square support vector regression models to forecast the outcome of horseraces", The Journal of Prediction Markets, Vol. 1, No. 3, pp. 169–187, 2012.

[8] Benter, William. "Computer based horse race handicapping and wagering systems: A report", Efficiency of racetrack betting markets, pp. 183–198, 2008.

[9] Edelman, David. "Adapting support vector machine methods for horserace odds prediction", Annals of Operations Research, Vol. 151, No. 1, pp. 325, 2007.