3A2-5

相対比較に基づく効率的なランキング推定アルゴリズム

Efficient Ranking Estimation Based on Pairwise Comparison

本多淳也 *1*2 Junya Honda
小宮山純平 *1 Jumpei Komiyama
前原貴憲 *2 Takanori Maehara
横山大作 *1 Daisaku Yokoyama

*1東京大学 The University of Tokyo
*2理化学研究所 RIKEN

In the estimation problem of ranking over items, we often face a situation such that direct evaluation of items is infeasible and only the results of pairwise comparisons are accessible. This paper constructs a new algorithm that outputs the correct ranking by $O(K \log K/\delta)$ comparisons with error probability at most δ.

1. はじめに

推薦システムや検索エンジン，オンライン広告など，個々の選好からアイテム集合のランキングを推定する問題は様々な場面で現れる．特に，アイテム間の相対比較の結果は観測可能である場面が多く多様な場面で用いられる．また，相対比較に基づくランキング推定はビデオゲームやチェス，囲碁などのプレイヤーのランキング推定といったものにも用いることができる．このような枠組みでは，適切な比較ペアを選択することでなるべく少ない比較回数で正しいランキングを復元することが主な目的となる．

本論文ではこの問題を相対比較に基づく PAC ランキング推定問題として定式化する．この問題では K 個のアイテムがあり，未知の選好行列 $M = \{\mu_{ij}\}_{i,j\in[K]}$ が定まっている．ここで $n \in \mathbb{N}$ に対して $[n] = \{1, 2, \cdots, n\}$ とした．アイテム x_i と x_j が比較されるとき，前者は後者に対して確率 μ_{ij} で好まれる．ここで各アイテムは勝ち越し関係の意味で全順序関係があると仮定する．すなわち，ある全順序 $([K], \succ)$ が存在して $x_i \succ x_j$ ならば $\mu_{ij} > 1/2$ が成り立つとする．予測者はそれまでの比較結果に基づいて比較するペアを各時刻に選択し，推定したランキングが誤り確率 δ 以下で正しいと判定した時点で比較を終了し，そのランキングを出力する．

この問題は計算量理論における雑音付きソートと深い関連がある．これは各ペアの比較結果が確率 $p < 1/2$ で誤っている場合のソートとして定式化される．ここで比較結果が受動的に与えられたときに尤度が最大の順序を計算する問題は NP 困難であることが知られている [Ailon et al., 2008]．また，予測者が比較するペアを選択できる設定において，Feige et al. [1994] は平均 $O(K \log K/\delta)$ 回の比較で確率 $1-\delta$ 以上で正しい順序を出力するアルゴリズム（以下 FPRU アルゴリズムとよぶ）を構成した．ただし雑音付きソートの文脈では p が既知であることを仮定し，これは実際には現実的でない．

そこで，相対比較に基づく PAC ランキング推定の文脈ではそのような事前知識を用いないランキング推定の方法が考えられてきた．相対比較モデルにおいては比較ペアとして $\Theta(K^2)$ 個の候補があり，単純に選好行列 M の全要素を推定するような素朴なアルゴリズムは $\Omega(K^2)$ 回の比較が必要となる．このような $\Omega(K^2)$ の依存性を回避するために選好行列についておく最も一般的な仮定が全順序関係の存在である．特に，最近の多くの研究では全順序よりさらに強い仮定をおき，それらの仮定のもとで $O(K(\log K) \log(K/\delta))$ の比較回数を達成するアルゴリズムが構成されている [Busa-Fekete et al., 2014, Szörényi et al., 2015]．

本論文では，PAC ランキング推定の問題において選好行列について全順序以外の仮定をおかない場合を考える．

まず準備として，FPRU アルゴリズムを $\Delta = \min_{i,j} |\mu_{ij} - 1/2| > 0$ が未知である PAC ランキング推定の設定へと拡張する．このアルゴリズムは全順序関係以外の仮定をおかずに $O(K \log K/\delta)$ の比較回数を達成する．このオーダーは最適であることが計算量理論の文脈で知られているが [Feige et al., 1994]，実際の FPRU アルゴリズムの性能は一般的な K, δ においては非常に悪い．本論文では，その原因が FPRU アルゴリズムの比較回数のうち $O(\log 1/\delta)$ の項が情報論的下限を達成していないことにあることを議論する．

次に本論文の主要な貢献として，$O(K \log K/\delta)$ の比較回数を達成するとともに $O(\log 1/\delta)$ の項が情報論的下限を達成するアルゴリズムを構成する．提案アルゴリズムはさまざまな設定において既存手法を大きく上回る性能を達成することをシミュレーションにより確認する．

2. 定式化

K 個のアイテム集合 $\{x_i\}_{i\in[K]}$ に対するランキングを考える．x_i が x_j に対して好まれる確率 μ_{ij} に対して $\mu_{ji} = 1 - \mu_{ij}$ が成り立つ．ここで他の文献と同様に $\mu_{ij} \neq 1/2$ が全ての $i \neq j$ に対して成り立つと仮定する．また，ある全順序 $([K], \succ)$ が存在して $i \neq j$ に対して $i \succ j$ ならば $\mu_{ij} > 1/2$ が成り立つと仮定する．

選好行列を $M = \{\mu_{ij}\}_{(i,j)\in[K]^2}$ で表し，全順序関係が存在する選好行列全体の集合を $\mathcal{M}$ で表す．ペア (i, j) の比較結果を $Y_{ij} \in \{-1, +1\}$ で表し，これは x_i が x_j より好まれた場合に $+1$，そうでない場合 -1 となる．

真の順序は $[K]$ の置換 $\sigma^* = \{\sigma_i^*\}$ で表す，すなわち，$x_{\sigma_1^*} \succ x_{\sigma_2^*} \succ \cdots \succ x_{\sigma_K^*}$ が成り立つ．ここで一般性を失うことなしに $\sigma_i^* = i$，すなわち $x_1 \succ x_2 \succ \cdots \succ x_K$ と仮定する．各時刻にアルゴリズムはペア (x_i, x_j) を選択し，そのいずれが選ばれたかを確率 μ_{ij} に従って観測する．また，各時刻にアルゴリズムは比較を続けるか，あるいは比較を停止して推定順序 $(\hat\sigma_1, \hat\sigma_2, \ldots, \hat\sigma_K)$ を出力する．あるアルゴリズムが確率 $1-\delta$ 以上で有限時間で停止し正しい順序を出力するとき，そのアルゴリズムは δ-PAC であるという．アルゴリズムが停止するまでの比較回数の期待値をサンプル複雑度とよぶ．

連絡先: 本多 淳也，email: honda@stat.t.u-tokyo.ac.jp

Algorithm 1: ペア (i,j) の比較.

入力: 信頼度 δ.

n_{ij}, $s_{ij} := 0$.
repeat
　ペア (i,j) を比較して $Y_{ij} \in \{-1,+1\}$ を観測する.
　$n_{ij} := n_{ij}+1$, $s_{ij} := s_{ij}+Y_{ij}$.
until $|s_{ij}| > s'_{n_{ij}}(\delta)$.
$s_{ij} > 0$ ならば "$i \succ j$" を，そうでないなら "$i \prec j$" を出力する.

3. サンプル複雑度の下界

本節では δ-PAC 問題においてアルゴリズムが必要とする比較回数の下界について考える．ベルヌーイ分布の KL ダイバージェンス $d(p,q) = p\log\frac{p}{q} + (1-p)\log\frac{1-p}{1-q}$ に対して

$$H_M = \sum_{j=2}^{K} \frac{1}{d(\mu_{j-1,j}, 1/2)}$$

と定義する．このとき δ-PAC アルゴリズムのサンプル複雑度の下界は以下で与えられる．

定理 1 (計算論的・情報論的サンプル複雑度下界)**.** あるアルゴリズムが δ-PAC であるとき，その比較回数 N について以下が成り立つ．(i) ある選好行列 $M \in \mathcal{M}$ に対して

$$\mathbb{E}[N] = \Omega(K\log(K/\delta))\,. \tag{1}$$

(ii) 任意の $M \in \mathcal{M}$ に対して

$$\mathbb{E}[N] \geq H_M \log(2.4\delta)^{-1}\,. \tag{2}$$

これらの結果はそれぞれ Feige et al. [1994] および Kaufmann et al. [2016] から容易に導かれる．

4. アルゴリズムの構成

本節ではまず $\Delta = \min_{i\neq j}|\mu_{ij} - 1/2|$ が未知の場合に対しての FPRU アルゴリズムの拡張を考え，これが情報論的下限 (2) を達成できないことを議論する．次に，情報論的下限，計算論的下限を共に達成するアルゴリズムを提案する．

4.1 FPRU アルゴリズム

$k \leq K$ に対して集合 $\mathcal{S} = \{y_1, y_2, \cdots, y_k\} \subset \{x_1, x_2, \cdots, x_K\}$ をソート済みのアイテム集合とし，順序 $y_1 \succ y_2 \succ \cdots \succ y_k$ が予測者に既知であるとする．FPRU アルゴリズム [Feige et al., 1994, Sect. 3] はこのようなソート済み系列に対して新規アイテム y を挿入する操作を繰り返すアルゴリズムであり，適切なランダムウォークに基づいた雑音付き 2 分探索を行うことにより各挿入操作が $O(\log(k/\delta))$ の比較で可能であることを用いている．ここで元のアルゴリズムは $\Delta = \min_{ij}|\mu_{ij} - 1/2|$ が既知であるとしていたが，その場合の比較操作を以下の Algorithm 1 で δ に定数 $\delta_1 < 1/2$ を代入したものに置き換えることで同オーダーの挿入およびソートが可能であることが示される．ここで $s'_n(\delta)$ は $nd((s'_n(\delta)+n)/2n, 1/2) \approx \log 1/\delta$ となるよう定められた適切な値であり，詳細は省略する．

このアルゴリズムのサンプル複雑度は定理 1 (i) の意味で最適であるが，その実際の性能は $O(K(\log K)\log(K/\delta))$ 回の比較が必要な他のアルゴリズムに比べて非常に悪い場合が多い．これは以下の議論により説明される．

まず，FPRU アルゴリズムは上述のように新規要素を $K-1$ 回挿入することによりソートを行う．さて，例えばランキング $x_1 \succ x_2 \succ x_3 \succ x_4$ を正しく推定するためには $x_1 \succ x_2$, $x_2 \succ x_3$ および $x_3 \succ x_4$ が成り立つことを確認することが必要十分であり，例えば x_1 を x_3 や x_4 と比較することは（結果的には）冗長となる．一方で，FPRU アルゴリズムでは新規に挿入するための要素を選ぶ方法がないため，冗長な比較を避けることができない．例えば，FPRU アルゴリズムが x_3 に対してまず x_4 を挿入し，次に x_1 を挿入する場合には (x_1, x_3) という冗長な比較が行われることになる．

さらに，仮に挿入の順序が偶然理想的に選ばれたとしても，FPRU アルゴリズムは定理 1(ii) の下界を達成することができない．FPRU アルゴリズムで x_K をソート済み集合 $x_1 \succ x_2 \succ \cdots \succ x_{K-1}$ に対して挿入する場合，その比較回数は FPRU アルゴリズムのパラメーター δ_0 に対して

$$\mathbb{E}[N] \gtrsim \frac{\log\delta_0^{-1}}{d(\mu_{K-1,K}, \frac{1}{2})} \times \frac{2}{\log\frac{1}{4\delta_0(1-\delta_0)}}\left(\frac{\log\frac{1-\delta_0}{\delta_0}(\log_2 K - 1)}{2} + \log\frac{1}{\delta}\right)$$

で与えられることが示される．ここで $\log\delta^{-1}$ の係数はパラメーター δ_0 に対して単調増加であり $\delta_0 \to 0$ のとき最小値 $\frac{2}{d(\mu_{K-1,K}, \frac{1}{2})}$ をとる．一方，(x_{K-1}, x_K) の比較回数の情報論的下限は $\frac{1}{d(\mu_{K-1,K}, \frac{1}{2})}$ であり，これは FPRU アルゴリズムがパラメーターを最適化したとしても情報論的下限を達成できないことを意味している．さらに，$O(\log K)$ の項の係数部（これが $K-1$ 回の挿入を行うと $O(K\log K)$ となる）は $\delta_0 \to 0$ で無限大に発散し，したがって FPRU アルゴリズムでは $O(\log\delta^{-1})$ の項の最適化と $O(K\log K)$ の項の最適化が両立できない．

4.2 提案法

本節では定理 1 の計算論的･情報論的下界の両方を達成するアルゴリズムを構成する．提案法では，入力系列 $(\tilde{x}_1, \tilde{x}_2, \cdots, \tilde{x}_K)$ に対してその正しい順序を確率 $1-\delta$ 以上で返す任意の δ_0-PAC アルゴリズム $\Pi_{\delta_0}(\tilde{x}_1, \tilde{x}_2, \cdots, \tilde{x}_K)$ をオラクルとして用いる．このようなオラクルとしては例えば FPRU アルゴリズムやマージソートにおいて各比較操作を Algorithm 1 で $\delta := \delta_0/(K\lceil\log_2 K\rceil)$ としたものに置き換えたものを用いることができる．提案法の動作は Algorithm 2 で与えられる．ここで提案法ではペア (i,j) と (j,i) を区別することに注意する．また

$$\mathcal{L}(\sigma) = \{(\sigma_1, \sigma_2), (\sigma_2, \sigma_3), \cdots, (\sigma_{K-1}, \sigma_K)\}$$

とした．

提案法は推定パートと検証パートを交互に繰り返すことによりランキング推定を行う．推定パートではオラクル Π_{δ_0} によりランキングを定数信頼度 δ_0 で粗く推定し，検証パートではその粗く推定したランキングが正しいかどうかを検証する．もし検証パートの結果が推定したランキングと定数信頼度 δ_1 で矛盾したならば，矛盾の生じたデータを破棄し推定パートを再び行う．推定したランキングが信頼度 $\delta_2 \approx \delta$ 以上で正しいことが検証パートにおいて確認された場合，そのランキングを出力してアルゴリズムを停止する．

Algorithm 2: 提案法.

入力: 信頼度 $\delta \in (0,1)$，未ソート系列 $(\tilde{x}_1, \tilde{x}_2, \cdots, \tilde{x}_K)$.
パラメーター: δ_0, $\delta_1 \in (0,1)$, δ_0-PAC アルゴリズム Π_{δ_0}.

1 $\delta_2 := (1-\delta_0)(1-\delta_1)\delta/\delta_0$.
2 $n_{ij} := 0$, $s_{ij} := 0$, $\forall i,j \in [K]$.
3 **for** $l = 1, 2, \cdots$ **do**
4 　$\Pi_{\delta_0}(\tilde{x}_1, \tilde{x}_2, \cdots, \tilde{x}_K)$ により推定ランキング $(\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_K) = (x_{\hat{\sigma}_1}, x_{\hat{\sigma}_2}, \cdots, x_{\hat{\sigma}_K})$ を求める.
5 　**while** $s_{ij} < s'_{n_{ij}}(\delta_2)$ なる $(i,j) \in \mathcal{L}(\hat{\sigma})$ が存在 **do**
6 　　$(i,j) := \mathrm{argmin}_{(i,j)\in\mathcal{L}(\hat{\sigma}): s_{ij} < s'_{n_{ij}}(\delta_2)}\, s_{ij}$.
7 　　ペア (i,j) を比較し $Y_{ij} \in \{-1,+1\}$ を観測する.
8 　　$n_{ij} := n_{ij} + 1$, $s_{ij} := s_{ij} + Y_{ij}$.
9 　　**if** $s_{ij} < -s'_{n_{ij}}(\delta_1/(K-1))$ **then**
10 　　　$n_{ij} := 0$, $s_{ij} := 0$.
11 　　　ステップ 3 のループを続ける.
12 　$(\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_L)$ を出力して停止する.

定理 2. パラメーター $\delta_0, \delta_1 \in (0,1)$ を任意に固定し，オラクル Π_{δ_0} が δ_0-*PAC* であると仮定する．このとき，提案法は δ-*PAC* であり，その比較回数 N は固定した $M \in \mathcal{M}$ に対して以下で抑えられる．

$$\mathbb{E}[N] \leq H_M \log \delta^{-1} + \frac{\mathbb{E}[N_{\Pi_{\delta_0}}]}{(1-\delta_0)(1-\delta_1)} + o(K \log K/\delta)\,.$$

ここで $N_{\Pi_{\delta_0}}$ はオラクル Π_{δ_0} の実行 1 回の比較回数を表す．

この定理から，提案法はオラクル Π_{δ_0} によらず情報論的下限 (2) を達成し，またオラクルとして FPRU アルゴリズムといった固定した信頼度に対してサンプル複雑度 $O(K \log K)$ を達成するアルゴリズムを用いることにより計算論的下限 (1) も達成できることがわかる．

5. 実験

本節では提案法の既存手法との比較を行う．既存手法としては MallowsMPR [Busa-Fekete et al., 2014]，PLPAC-AMPR [Szörényi et al., 2015]，FPRU アルゴリズム [Feige et al., 1994] を用いた．

雑音付きソートは古典的な計算量理論における設定であり，各比較において確率 p で間違った結果が観測される，すなわち，$\mu_{ij} = 1 - p$ が全ての $x_i \succ x_j$ において成り立つ．ここでは $p = 0.3$ とした．

それ以外の実験で用いた選好行列は表 2 で与えられる．**ArXiv 検索エンジン**は ArXiv.org における 6 個の検索エンジンに関する選好データを表す．**Netflix，パズル，選挙投票，**データセット PrefLib.org のものであり [*1]，それぞれ NetFlix の映画のレーティング [Bennett and Lanning, 2007]，パズルの正解までの近さ [Mao et al., 2013]，アイルランドの選挙における投票データに基づく選好行列である．**将棋**は floodgate online server[*2] におけるコンピュータ将棋の対戦データに基づく選好行列である．2016 年 7 月 19 日において 200 回以上の試合数がある 8 個のプログラムを抽出し，イロレーティングに基づき選好行列の各要素を決定した．**寿司**は Sushi Preference Dataset [Kamishima, 2003] に基づくもので，5000 人の日本人ユーザー 100 種類の寿司の選好データからなる．このデータから主な寿司を 8 個取り出し，寿司 i を寿司 j より好むユーザーの割合に基づき μ_{ij} を決定した．

表 1 は各アルゴリズムが停止するまでに行った比較の回数を表す．各結果は 100 回の試行の平均である．提案法では $\delta_0 = 1/2$, $\delta_1 = 1/3$ とした．また，オラクル Π_{δ_0} としてはマージソートを用いた．これは FPRU アルゴリズムの性能が非常に悪いためにオラクルの部分の比較回数だけで他のアルゴリズムより悪くなったことによる．

この表から分かるように提案法は全てのデータセットにおいて他のアルゴリズムを大きく上回る性能を達成した．一方，これらの比較回数は情報論的下限とは大きなギャップがあり，まだ改善の余地があることが示唆される．

6. 結論

本論文では相対比較に基づく PAC ランキング推定問題を考えた．この問題では FPRU アルゴリズムが計算論的な意味での最適性しか達成できず実際の性能が悪いのに対して，計算論的，情報論的下界の両方を達成する新たなアルゴリズムを提案した．また，提案法が様々なデータセットに対して既存手法を大きく上回る性能を達成することを確認した．

*1 http://preflib.org, ED-00004-00000087, ED-00025-00000002, ED-00001-00000001
*2 http://wdoor.c.u-tokyo.ac.jp/shogi/floodgate.html

参考文献

Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: ranking and clustering. *Journal of the ACM (JACM)*, 55(5):23, 2008.

James Bennett and Stan Lanning. The netflix prize. In *In KDD Cup and Workshop in conjunction with KDD*, 2007.

Róbert Busa-Fekete, Balázs Szörényi, Paul Weng, Weiwei Cheng, and Eyke Hüllermeier. Preference-based reinforcement learning: evolutionary direct policy search using a preference-based racing algorithm. *Machine Learning*, 97(3):327–351, 2014.

Uriel Feige, Prabhakar Raghavan, David Peleg, and Eli Upfal. Computing with noisy information. *SIAM J. Comput.*, 23(5):1001–1018, 1994.

Toshihiro Kamishima. Nantonac collaborative filtering: recommendation based on order responses. In *SIGKDD 2003*, pages 583–588, 2003.

Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On the complexity of best-arm identification in multi-armed bandit models. *Journal of Machine Learning Research*, 17(1):1–42, 2016.

Andrew Mao, Ariel D. Procaccia, and Yiling Chen. Better human computation through principled voting. In *AAAI 2013*, 2013.

Balázs Szörényi, Róbert Busa-Fekete, Adil Paul, and Eyke Hüllermeier. Online rank elicitation for Plackett-Luce: A dueling bandits approach. In *NIPS 2015*, pages 604–612, 2015.

表 1: 信頼度 $\delta = 0.05$, $\delta = 0.005$ における平均比較回数.

$\delta = 0.05$	$H_M \log(1/\delta)$	MallowsMPR	PLPAC-AMPR	FPRU	提案法
雑音付きソート ($K = 3$)	$1.85\mathrm{e}+02$	$1.25\mathrm{e}+03$	$1.12\mathrm{e}+03$	$2.39\mathrm{e}+03$	$\mathbf{7.26e+02}$
雑音付きソート ($K = 20$)	$6.93\mathrm{e}+02$	$1.53\mathrm{e}+04$	$1.05\mathrm{e}+05$	$3.07\mathrm{e}+04$	$\mathbf{7.68e+03}$
ArXiv 検索	$1.81\mathrm{e}+04$	$3.87\mathrm{e}+05$	$3.33\mathrm{e}+05$	$5.85\mathrm{e}+05$	$\mathbf{1.48e+05}$
Netflix	$1.97\mathrm{e}+03$	$1.65\mathrm{e}+04$	$1.63\mathrm{e}+04$	$2.70\mathrm{e}+04$	$\mathbf{1.05e+04}$
パズル	$1.92\mathrm{e}+02$	$1.42\mathrm{e}+03$	$1.88\mathrm{e}+03$	$2.79\mathrm{e}+03$	$\mathbf{8.82e+02}$
選挙投票	$2.68\mathrm{e}+05$	$3.21\mathrm{e}+06$	$3.38\mathrm{e}+06$	$7.05\mathrm{e}+06$	$\mathbf{1.70e+06}$
将棋	$1.70\mathrm{e}+04$	$1.72\mathrm{e}+05$	$2.00\mathrm{e}+05$	$3.75\mathrm{e}+05$	$\mathbf{9.69e+04}$
寿司	$1.05\mathrm{e}+05$	$1.27\mathrm{e}+06$	$1.23\mathrm{e}+06$	$2.43\mathrm{e}+06$	$\mathbf{6.65e+05}$

$\delta = 0.005$	$H_M \log(1/\delta)$	MallowsMPR	PLPAC-AMPR	FPRU	提案法
雑音付きソート ($K = 3$)	$3.28\mathrm{e}+02$	$1.34\mathrm{e}+03$	$1.31\mathrm{e}+03$	$2.90\mathrm{e}+03$	$\mathbf{9.71e+02}$
雑音付きソート ($K = 20$)	$1.22\mathrm{e}+03$	$1.73\mathrm{e}+04$	$1.17\mathrm{e}+05$	$3.56\mathrm{e}+04$	$\mathbf{8.13e+03}$
ArXiv 検索エンジン	$3.19\mathrm{e}+04$	$4.16\mathrm{e}+05$	$3.53\mathrm{e}+05$	$6.50\mathrm{e}+05$	$\mathbf{1.74e+05}$
Netflix	$3.48\mathrm{e}+03$	$1.83\mathrm{e}+04$	$1.83\mathrm{e}+04$	$3.42\mathrm{e}+04$	$\mathbf{1.13e+04}$
パズル	$3.40\mathrm{e}+02$	$1.60\mathrm{e}+03$	$2.18\mathrm{e}+03$	$3.61\mathrm{e}+03$	$\mathbf{1.06e+03}$
選挙投票	$4.73\mathrm{e}+05$	$3.53\mathrm{e}+06$	$3.64\mathrm{e}+06$	$8.64\mathrm{e}+06$	$\mathbf{1.88e+06}$
将棋	$3.01\mathrm{e}+04$	$1.88\mathrm{e}+05$	$2.20\mathrm{e}+05$	$4.50\mathrm{e}+05$	$\mathbf{1.16e+05}$
寿司	$1.86\mathrm{e}+05$	$1.33\mathrm{e}+06$	$1.35\mathrm{e}+06$	$2.93\mathrm{e}+06$	$\mathbf{6.97e+05}$

表 2: 実験に用いた選好行列.

0.50	0.55	0.55	0.54	0.61	0.61
0.45	0.50	0.55	0.55	0.58	0.60
0.45	0.45	0.50	0.54	0.51	0.56
0.46	0.45	0.46	0.50	0.54	0.51
0.39	0.42	0.49	0.46	0.50	0.51
0.39	0.40	0.44	0.49	0.49	0.50

(a) ArXiv 検索エンジン

0.5	0.528	0.665
0.472	0.5	0.657
0.335	0.343	0.5

(b) Netflix

0.5	0.686	0.750	0.778
0.314	0.5	0.688	0.730
0.25	0.312	0.5	0.617
0.222	0.27	0.383	0.5

(c) パズル

0.5	0.506	0.557	0.556	0.597	0.601	0.621	0.725	0.78	0.747	0.835	0.898
0.494	0.5	0.529	0.52	0.577	0.580	0.581	0.681	0.73	0.735	0.785	0.852
0.443	0.471	0.5	0.504	0.516	0.530	0.539	0.592	0.631	0.679	0.711	0.727
0.444	0.480	0.496	0.5	0.526	0.523	0.541	0.612	0.641	0.693	0.722	0.742
0.403	0.423	0.484	0.474	0.5	0.507	0.522	0.607	0.691	0.722	0.766	0.796
0.399	0.42	0.47	0.477	0.493	0.5	0.522	0.583	0.615	0.67	0.682	0.698
0.379	0.419	0.461	0.459	0.478	0.478	0.5	0.642	0.62	0.652	0.699	0.732
0.275	0.319	0.408	0.388	0.393	0.417	0.358	0.5	0.573	0.605	0.657	0.691
0.220	0.270	0.369	0.359	0.309	0.385	0.380	0.427	0.5	0.551	0.594	0.624
0.253	0.265	0.321	0.307	0.278	0.330	0.348	0.395	0.449	0.5	0.504	0.519
0.165	0.215	0.289	0.278	0.234	0.318	0.301	0.343	0.406	0.496	0.5	0.604
0.102	0.148	0.273	0.258	0.204	0.302	0.268	0.309	0.376	0.481	0.396	0.5

(d) 選挙投票

0.500	0.553	0.566	0.627	0.524	0.675	0.739	0.701
0.447	0.500	0.524	0.553	0.522	0.608	0.621	0.640
0.434	0.476	0.500	0.504	0.534	0.553	0.665	0.657
0.372	0.447	0.496	0.500	0.524	0.581	0.727	0.694
0.476	0.478	0.466	0.476	0.500	0.517	0.558	0.547
0.325	0.392	0.447	0.419	0.483	0.500	0.653	0.629
0.261	0.379	0.335	0.273	0.442	0.347	0.500	0.563
0.299	0.360	0.343	0.306	0.453	0.371	0.437	0.500

(e) 寿司

0.5	0.553	0.578	0.684	0.729	0.808	0.817	0.826
0.447	0.5	0.526	0.636	0.685	0.773	0.783	0.794
0.422	0.474	0.5	0.612	0.662	0.754	0.765	0.776
0.316	0.364	0.388	0.5	0.554	0.661	0.674	0.688
0.271	0.315	0.338	0.446	0.5	0.610	0.624	0.639
0.192	0.227	0.246	0.339	0.390	0.5	0.514	0.530
0.183	0.217	0.235	0.326	0.376	0.486	0.5	0.516
0.174	0.206	0.224	0.312	0.361	0.470	0.484	0.5

(f) 将棋