Unbiased Learning-to-Rank +α

東京工業大学 経営工学系 B4
中田研究室 齋藤 優太

イントロダクション

ランキング学習とは

検索クエリに対し最適なドキュメントのランキングを返したい

例）ある検索クエリに対して10個のdocumentを表示するとき

Position (k)	Ranking 1	Ranking 2
1	5	1
2	4	2
---	---	---
9	1	5
10	2	4

Ranking 1のように
関連度の高いdocumentを
上位に表示したい
(5段階の関連度)

ランキング学習の定式化

次のように一般的な損失関数を考えてみる

$$M = \sum_{(q,d,r)\in\mathcal{L}_U} r \cdot \underbrace{f\left(q, d, \Omega_q\right)}_{\text{Rankingに関連する関数}}$$

$\mathcal{L}_U = \{(q, d, r)\}$ ：(query, document, relevance) のデータ

$\Omega_q = \{d \,|\, (q, d) \in \mathcal{L}_U\}$ ：あるqueryに対し提示されたdoc集合

ランキング学習の損失関数

損失関数として例えば、

$$ARP = \sum_{(q,d,r)\in\mathcal{L}_U} r \cdot \mathrm{rank}\left(q, d, \Omega_q\right)$$

Relevantなdocumentの順位の総和を小さくしたい

$$DCG = \sum_{(q,d,r)\in\mathcal{L}_U} r \cdot \frac{1}{\mathrm{rank}\left(q, d, \Omega_q\right)}$$

実際はlogかけたり

Relevantなdocumentの順位の逆数の総和を小さくしたい

実際に使えるのはClickデータだけ

最適化したい損失の計算には**Relevance**が必要
しかし、コストと時間がかかるので**human annotationはしたくない**

▼

安価に手に入るClickデータを使ってRanking Systemを構築したい

$$M = \sum_{(q,d,c)\in\mathcal{L}_U} c \cdot f\left(q, d, \Omega_q\right)$$

実際に使えるのはClickデータだけ

最適化したい損失の計算には**Relevance**が必要
しかし、コストと時間がかかるので**human annotationはしたくない**

安価に手に入るClickデータを使ってRanking Systemを構築したい

Relevanceだった部分をClickにそのまま入れ替えても大丈夫？

$$M = \sum_{(q,d,c) \in \mathcal{L}_U} c \cdot f\left(q, d, \Omega_q\right)$$

ClickはRelevanceの代わりになる？

ある検索クエリに対して10個のdocumentを表示したとき

Position (k)	**R**elevance	???	**C**lick
1	◎		◎
2	×		×
---	---	---	---
9	◎		×
10	×		×

必ずしも Relevance = Click とは言えなさそう…

Position-Based Model (PBM)

ClickとRelevanceを関係付けるため次の**Position-Based Model**を導入

$$C = E \cdot R$$

かつ

$$\underbrace{P(C=1|q,d,k)}_{\textbf{Click}} = \underbrace{P(E=1|k)}_{\textbf{Examination}} \cdot \underbrace{P(R=1|q,d)}_{\textbf{Relevance}}$$

$$= \theta_k \cdot \gamma_{q,d}$$

Position-Based Model (PBM)

ClickとRelevanceを関係付けるため次の**Position-Based Model**を導入

このモデル化のもとでは、

- queryとdocumentがrelevantかつexaminedのときclickが発生
- Relevanceはqueryとdocumentのみに依存
- Examinationはpositionのみに依存

$$\underbrace{P(C=1|q,d,k)}_{\text{Click}} = \underbrace{P(E=1|k)}_{\text{Examination}} \cdot \underbrace{P(R=1|q,d)}_{\text{Relevance}}$$

Position-Based Model (PBM)

ある検索クエリに対して10個のdocumentを表示したとき

Position (k)	**R**elevance	**E**xamine	**C**lick
1	◎	◎	◎
2	×	◎	×
9	◎	×	×
10	×	×	×

RelevanceとExaminationの両方が発生して初めてClickが発生

Unbiased Learning-to-Rank

Inverse Propensity Approach

Relevanceだった部分をClickにそのまま入れ替えたらダメ！

$$\cancel{M} = \sum_{(q,d,c)\in\mathcal{L}_U} c \cdot f\left(q, d, \Omega_q\right)$$

Inverse Propensity Approach

Relevanceだった部分をClickにそのまま入れ替えたらダメ！

$$\cancel{M} = \sum_{(q,d,c)\in\mathcal{L}_U} c \cdot f\left(q, d, \Omega_q\right)$$

IPSのアイデア：**両辺をExamination parameterで割る**

$$\frac{P(C=1|q,d,k)}{P(E=1|k)} = \underline{P(R=1|q,d)}$$

Relevanceを分離！

Inverse Propensity Approach

Relevanceだった部分をClickにそのまま入れ替えたらダメ！

$$\cancel{M} = \sum_{(q,d,c)\in\mathcal{L}_U} c \cdot f\left(q, d, \Omega_q\right)$$

Examination確率の逆数でlossを重み付け [Joachims et al. (2017)]

$$M_{IPS} = \sum_{(q,d,k,c)\in\mathcal{L}} \frac{c}{\theta_k} f\left(q, d, \Omega_q\right) = \sum_{(q,d,k,\underline{c=1})\in\mathcal{L}} \frac{1}{\theta_k} f\left(q, d, \Omega_q\right)$$

データに残ってる！

Inverse Propensity Approach

$$\underline{\mathbb{E}\left[M_{IPS}\right]} = \sum_{(q,d,k,c)\in\mathcal{L}} \frac{\mathbb{E}[C]}{\theta_k} f\left(q,d,\Omega_q\right)$$

重み付けした
損失の期待値

$$= \sum_{(q,d,k,c)\in\mathcal{L}} \frac{\theta_k \cdot \gamma_{q,d}}{\theta_k} f\left(q,d,\Omega_q\right)$$

Unbiased!

$$= \sum_{(q,d,k,c)\in\mathcal{L}} \gamma_{q,d} \cdot f\left(q,d,\Omega_q\right)$$

$$= \sum_{(q,d,k,c)\in\mathcal{L}} \Pr(R=1\,|\,q,d) f\left(q,d,\Omega_q\right) = \underline{M}$$

真の損失

Naiveな損失を使ってしまった場合

仮に**de-biasしなかった場合**に何が起こるかというと...

過去の検索方策に似かよるような重み

$$
\begin{aligned}
\mathbb{E}\left[M_{naive}\right] &= \sum_{(q,d,k,c)\in\mathcal{L}} \mathbb{E}[C]\cdot f\left(q,d,\Omega_q\right) \\
&= \sum_{(q,d,k,c)\in\mathcal{L}} \theta_k \cdot \gamma_{q,d} \cdot f\left(q,d,\Omega_q\right) \\
&= \sum_{(q,d,k,c)\in\mathcal{L}} \underbrace{\mathbb{P}(E=1|k)}_{past\,policy} \cdot \mathbb{P}(R=1|q,d)\cdot f\left(q,d,\Omega_q\right) \\
&\neq M
\end{aligned}
$$

Biased!

過去の検索方策を真似てしまう？？

Naiveな損失を最適化した末の提示ランキングの例

Pos	Exam (固定)	Relevance	Exam (過去)	Click
1	1	0.9	0.6	**0.90**
2	0.8	0.3	1	**0.24**
3	0.6	0.6	0.4	**0.36**
4	0.4	0.2	0.8	**0.08**
5	0.2	0.7	0.2	**0.14**
---	---	---	---	**0.344**

過去の検索方策を真似てしまう？？

各positionのexamination確率

Pos	Exam (固定)	Relevance	Exam (過去)	Click
1	1	0.9	0.6	**0.90**
2	0.8	0.3	1	**0.24**
3	0.6	0.6	0.4	**0.36**
4	0.4	0.2	0.8	**0.08**
5	0.2	0.7	0.2	**0.14**
---	---	---	---	**0.344**

過去の検索方策を真似てしまう？？

各documentのrelevanceの期待値

Pos	Exam (固定)	Relevance	Exam (過去)	Click
1	1	0.9	0.6	**0.90**
2	0.8	0.3	1	**0.24**
3	0.6	0.6	0.4	**0.36**
4	0.4	0.2	0.8	**0.08**
5	0.2	0.7	0.2	**0.14**
---	---	---	---	**0.344**

過去の検索方策を真似てしまう？？

各documentが過去に提示された際の
examination確率 (past policy)

Pos	Exam (固定)	Relevance	Exam (過去)	Click
1	1	0.9	0.6	**0.90**
2	0.8	0.3	1	**0.24**
3	0.6	0.6	0.4	**0.36**
4	0.4	0.2	0.8	**0.08**
5	0.2	0.7	0.2	**0.14**
---	---	---	---	**0.344**

過去の検索方策を真似てしまう？？

Naive損失を使うとここの積に対して最適化される

Pos	Exam (固定)	Relevance	Exam (過去)	Click
1	1	0.9	0.6	**0.90**
2	0.8	0.3	1	**0.24**
3	0.6	0.6	0.4	**0.36**
4	0.4	0.2	0.8	**0.08**
5	0.2	0.7	0.2	**0.14**
---	---	---	---	**0.344**

過去の検索方策を真似てしまう？？

Relevanceに対して最適化した結果clickのPrecision@5が向上

Pos	Exam (固定)	Relevance	Exam (過去)	Click
1	1	0.9	0.6	**0.90**
2	0.8	0.7	0.2	**0.56**
3	0.6	0.6	0.4	**0.36**
4	0.4	0.3	1	**0.12**
5	0.2	0.2	0.8	**0.04**
---	---	---	---	**0.396**

Result Randomization

IPSによりUnbiasedなlossを計算できるが, θ_k が必要

$$M_{IPS} = \sum_{(q,d,k,c=1)\in\mathcal{L}} \frac{1}{\theta_k} f\left(q, d, \Omega_q\right)$$

Result Randomization

IPSによりUnbiasedなlossを計算できるが, θ_k が必要

$$M_{IPS} = \sum_{(q,d,k,c=1)\in\mathcal{L}} \frac{1}{\theta_k} f\left(q, d, \Omega_q\right)$$

Result Randomization [Joachims et al. (2017)]

上位N番目までの検索結果を**Randomize**することで (きつい...)

$$\mathbb{E}_{(q,d)\sim P(q,d)}\left[\gamma_{q,d}\right] = \mathbb{E}_{(q,d)\sim P(q,d)}\left[\gamma_{q,d} \,\middle|\, k\right]$$

Result Randomization

Randomizeによりposition kでのCTRが θ_k に比例するので、
Position間でのexamination確率の比がわかる

$$\underbrace{\mathbb{E}[C|k]}_{\text{kでのCTR}} = \int_{q,d\in\mathcal{R}_k} \mathbb{E}[C|q,d,k]P(q,d)$$

$$= \int_{q,d\in\mathcal{R}_k} \theta_k \cdot \gamma_{q,d} P(q,d)$$

$$= \theta_k \cdot \underbrace{\int_{q,d\in\mathcal{R}_k} \gamma_{q,d} P(q,d)}_{\text{Randomizeにより定数}}$$

$$\propto \theta_k$$

関連研究

関連研究

WSDM2017で枠組みが提案されて以降、

Examination確率をいかに推定するかが研究の主な焦点

- Result Randomization [Joachims et al. WSDM2017]
  - ランダムなランキング表示データを用いて推定
- Regression-EM [Wang et al. WSDM2018]
  - ランダム配信なしでパラメータをEM-basedな手法で推定
- Intervention-Harvesting [Wang et al. WSDM2019]
  - Relevanceモデルをexamination parameterの推定時に必要としない

関連研究

そのほかの研究も、**Examination確率の推定 -> 学習の手順を前提**

- Trust Position-Based Model [Agarwal et al. WWW2019]
  - Position-Based ModelにTrust Biasを導入した新たなモデル化
- Unbiased LambdaMART [Hu et al. WWW2019]
  - Inverse Propensity ScoreをPairwise lossに拡張
- Context Dependent Examination Bias [Fang et al. SIGIR2019]
  - Examination確率がcontextにも依存するというモデル化

Dual Learning Algorithm

問題意識

- 近年の手法の複雑化にわかるようにExaminationの推定は困難
- さらに、モデルの学習毎にExaminationの推定も行わなければならない

- Examination parameterの推定を容易にしたい
- **end-to-end**でexamination parameterの推定とrelevanceの予測を達成するモデルが欲しい

提案手法のアイデア

$$\underbrace{P(C=1|q,d,k)}_{\text{Click}} = \underbrace{P(E=1|k)}_{\text{Examination}} \cdot \underbrace{P(R=1|q,d)}_{\text{Relevance}}$$

既存のIPSのアイデア：両辺をExamination parameterで割る

$$\frac{P(C=1|q,d,k)}{P(E=1|k)} = \underbrace{P(R=1|q,d)}_{\text{Relevanceを分離！}}$$

提案手法のアイデア

$$\underbrace{P(C=1|q,d,k)}_{\text{Click}} = \underbrace{P(E=1|k)}_{\text{Examination}} \cdot \underbrace{P(R=1|q,d)}_{\text{Relevance}}$$

本研究のアイデア：両辺を**Relevance parameter**で割る

$$\frac{P(C=1|q,d,k)}{P(R=1|q,d)} = \underbrace{P(E=1|k)}_{\text{Examinationを分離！}}$$

Inverse Relevance Weighting

次のExamination parameterの推定に対する**真の損失**を

$$M = \sum_{(q,d,e)\in\mathcal{L}_U} e \cdot f\left(q, d, \Omega_q\right)$$

わからない…

先ほどのアイデアを用いて**Relevance parameterの逆数で重み付け**

$$M_{IRW} = \sum_{(q,d,k,c)\in\mathcal{L}} \frac{c}{\gamma_{q,d}} f\left(q, d, \Omega_q\right) = \sum_{(q,d,k,\underline{c=1})\in\mathcal{L}} \frac{1}{\gamma_{q,d}} f\left(q, d, \Omega_q\right)$$

データに残ってる！

Inverse Relevance Weighting

$$\underline{\mathbb{E}\left[M_{IRW}\right]} = \sum_{(q,d,k,c)\in\mathcal{L}} \frac{\mathbb{E}[C]}{\gamma_{q,d}} f\left(q,d,\Omega_q\right)$$

重み付けした
損失の期待値

$$= \sum_{(q,d,k,c)\in\mathcal{L}} \frac{\theta_k \cdot \gamma_{q,d}}{\gamma_{q,d}} f\left(q,d,\Omega_q\right)$$

Unbiased!

$$= \sum_{(q,d,k,c)\in\mathcal{L}} \theta_k \cdot f\left(q,d,\Omega_q\right)$$

$$= \sum_{(q,d,k,c)\in\mathcal{L}} \Pr(E=1|k) f\left(q,d,\Omega_q\right) = \underline{M}$$

真の損失

Dual Learning Algorithm (DLA)

それぞれのパラメータを近似するように $f_\phi,\ g_\psi$ を最適化する

$$\underbrace{\gamma_{q,d} = P(R=1 \mid q,d)}_{\text{Relevance}} \approx \frac{e^{f_\phi(x_{q,d})}}{\sum_{d' \in \Omega_q} e^{f_\phi\left(x_{q,d'}\right)}}$$

$$\underbrace{\theta_k = P(E=1 \mid k)}_{\text{Examination}} \approx \frac{e^{g_\psi(k)}}{\sum_{k'} e^{g_\psi(k')}}$$

Dual Learning Algorithm (DLA)

分析に基づいた**重み付けによりnaiveなlossを補正**する

$$M_{IPW} = - \sum_{(q,d,k,c=1) \in \mathcal{L}_U} \frac{1}{\theta_k} \cdot \boxed{\log \frac{e^{f_\phi(x_{q,d})}}{\sum_{d' \in \Omega_q} e^{f_\phi(x_{q,d'})}}}$$

$$M_{IRW} = - \sum_{(q,d,c=1) \in \mathcal{L}_U} \frac{1}{\gamma_{q,d}} \cdot \boxed{\log \frac{e^{g_\psi(k)}}{\sum_{k'} e^{g_\psi(k')}}}$$

Softmax Loss

Dual Learning Algorithm (DLA)

分析に基づいた**重み付けによりnaiveなlossを補正**する

$$M_{IPW} = - \sum_{(q,d,k,c=1)\in\mathcal{L}_U} \boxed{\frac{1}{\theta_k}} \cdot \log \frac{e^{f_\phi(x_{q,d})}}{\sum_{d'\in\Omega_q} e^{f_\phi(x_{q,d'})}}$$

$$M_{IRW} = - \sum_{(q,d,c=1)\in\mathcal{L}_U} \boxed{\frac{1}{\gamma_{q,d}}} \cdot \log \frac{e^{g_\psi(k)}}{\sum_{k'} e^{g_\psi(k')}}$$

重み付け

Dual Learning Algorithm (DLA)

重み付け部分はもう一方のパラメータに依存

$$\underbrace{\gamma_{q,d} = P(R=1 \mid q,d)}_{\text{Relevance}} \approx \frac{e^{f_\phi(x_{q,d})}}{\sum_{d' \in \Omega_q} e^{f_\phi(x_{q,d'})}}$$

$$\underbrace{\theta_k = P(E=1 \mid k)}_{\text{Examination}} \approx \frac{e^{g_\psi(k)}}{\sum_{k'} e^{g_\psi(k')}}$$

Dual Learning Algorithm (DLA)

Algorithm 1: Dual Learning Algorithm

```
  **Input:** Q = {q, π_q, c_q}, f, g, learning rate α
  **Output:** θ, φ
1 Initialize θ ⇐ 0, φ ⇐ 0
2 repeat
3   Randomly sample a batch Q′ from Q
    for (q, π_q, c_q) ∈ Q′ do
4     for x ∈ π_q do
5       Compute P(r_q^x = 1|π_q), P(o_q^x = 1|π_q) with Eq 9.
      end
    end
6   Compute L̂(S, q), L̂(E, q) with Eq 11.
7   θ = θ + α · ∂L̂(S,q)/∂θ, φ = φ + α · ∂L̂(E,q)/∂φ
  until Convergence;
8 return θ, φ
```

$$\theta = \theta + \alpha \cdot \frac{\partial \hat{L}(S,q)}{\partial \theta},\ \phi = \phi + \alpha \cdot \frac{\partial \hat{L}(E,q)}{\partial \phi}$$

3 Queryのmini-batchサンプリング

5 ExaminationとRelevanceの
Parameterを推定

6 Lossを計算. それぞれのLossは
もう一方のparameterに依存

7 Lossの勾配でparameter更新

[Ai et al. 2018]のAlgorithm1を引用

実験結果

実験設定①

Yahoo! LETOR dataset

- 29,921 queries
- 710k documents
- **5 level relevance judgement** (ただし, click dataはなし)
- 700 features

Gold-standard parametersを人工的に生成

examination $$\theta_k = \mathbb{P}(E = 1|k) = \rho_k^\eta$$

relevance $$\gamma_{q,d} = \mathbb{P}\left(R = 1 \,|\, q, d\right) = \epsilon + (1 - \epsilon)\frac{2^r - 1}{2^{r_{\max}} - 1}$$

実験設定①

Yahoo! LETOR dataset

- 29,921 queries
- 710k documents
- **5 level relevance judgement** (ただし, click dataはなし)
- 700 features

さらに、

- Click dataはexaminationとrelevanceの実現値の積によって生成
- Training dataのpositionはinitial rankerによって生成

実験結果①

Yahoo! LETOR set 1										
Ranking Model	Correction Method	MAP	nDCG@1	ERR@1	nDCG@3	ERR@3	nDCG@5	ERR@5	nDCG@10	ERR@10
DNN with DLA		0.816	0.658	0.338	0.662	0.412	0.683	0.433	0.729	0.447
Ranking SVM	NoCorrect	0.814^-	0.628^-	0.316^-	0.638^-	0.395^-	0.661^-	0.416^-	0.711^-	0.432^-
	RandList	0.812^-	0.642^-	0.330^-	0.653^-	0.407^-	0.675^-	0.428^-	0.721^-	0.442^-
DNN	NoCorrect	0.807^-	0.622^-	0.317^-	0.631^-	0.394^-	0.653^-	0.416^-	0.704^-	0.431^-
	RandList	0.814^-	0.658	0.338	0.659^-	0.412	0.679^-	0.433	0.725^-	0.447
Initial Ranker		0.804^-	0.559^-	0.271^-	0.586^-	0.357^-	0.617^-	0.381^-	0.675^-	0.397^-
Oracle DNN		0.830^+	0.667^+	0.339^+	0.675^+	0.414^+	0.695^+	0.435^+	0.740^+	0.449^+

[Ai et al. 2018]のTable3を引用

ほとんどの指標でIPSで補正した場合の予測性能がNaiveな損失のものを上回った

実験結果①

Yahoo! LETOR set 1										
Ranking Model	Correction Method	MAP	nDCG@1	ERR@1	nDCG@3	ERR@3	nDCG@5	ERR@5	nDCG@10	ERR@10
DNN with DLA		0.816	0.658	0.338	0.662	0.412	0.683	0.433	0.729	0.447
Ranking SVM	NoCorrect	0.814^{-}	0.628^{-}	0.316^{-}	0.638^{-}	0.395^{-}	0.661^{-}	0.416^{-}	0.711^{-}	0.432^{-}
	RandList	0.812^{-}	0.642^{-}	0.330^{-}	0.653^{-}	0.407^{-}	0.675^{-}	0.428^{-}	0.721^{-}	0.442^{-}
DNN	NoCorrect	0.807^{-}	0.622^{-}	0.317^{-}	0.631^{-}	0.394^{-}	0.653^{-}	0.416^{-}	0.704^{-}	0.431^{-}
	RandList	0.814^{-}	0.658	0.338	0.659^{-}	0.412	0.679^{-}	0.433	0.725^{-}	0.447
Initial Ranker		0.804^{-}	0.559^{-}	0.271^{-}	0.586^{-}	0.357^{-}	0.617^{-}	0.381^{-}	0.675^{-}	0.397^{-}
Oracle DNN		0.830^{+}	0.667^{+}	0.339^{+}	0.675^{+}	0.414^{+}	0.695^{+}	0.435^{+}	0.740^{+}	0.449^{+}

[Ai et al. 2018]のTable3を引用

Result Randomizationを用いて学習した**DNN(RandList)と遜色なく**、
それ以外のbaselinesよりも良い結果を全ての指標で観測

まとめ

- Clickデータを用いたナイーブな損失設計は
  Relevanceに対する損失に対してBiasを持つ

- PBMに基づくとExamination確率の逆数でclickデータを
  重み付けする**IPSにより不偏性を満たす損失の推定量を作れる**

- Examination確率の推定にはResult Randomizationが必要だったが
  offlineでそれを推定する手法にDual Learning Algorithm等が存在

References

[Joachims et al. WSDM2017]: Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased learning-to-rank with biased feedback. In Proceedings of the 10th ACM International Conference on Web Search and Data Mining (WSDM ’17).

[Wang et al. WSDM2018]: Xuanhui Wang, Nadav Golbandi, Michael Bendersky, Donald Metzler, and Marc Najork. 2018. Position Bias Estimation for Unbiased Learning to Rank in Personal Search. In Proceedings of the 11th ACM International Conference on Web Search and Data Mining (WSDM ’18).

[Ai et al. SIGIR2018]: Qingyao Ai, Keping Bi, Cheng Luo, Jiafeng Guo, and W. Bruce Croft. Unbiased learning to rank with unbiased propensity estimation. In The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR’18).

[Agarwal et al. WSDM2019]: Aman Agarwal, Ivan Zaitsev, Xuanhui Wang, Cheng Li, Marc Najork and Thorsten Joachims. 2019. Estimating Position Bias without Intrusive Interventions. In The 12th ACM International Conference on Web Search and Data Mining (WSDM ’19)

[Hu et al. WWW2019]: Ziniu Hu and Yang Wang, Qu Peng, Hang Li. 2019. Unbiased LambdaMART: An Unbiased Pairwise Learning-to-Rank Algorithm. In Proceedings of the 2019 World Wide Web Conference (WWW ’19)

[Agarwal et al. WWW2019]: Aman Agarwal, Xuanhui Wang, Cheng Li, Mike Bendersky, and Marc Najork. 2019. Addressing Trust Bias for Unbiased Learning-to-Rank. In Proceedings of the 2019 World Wide Web Conference (WWW ’19)

[Fang et al. SIGIR2019] Fang, Z., Agarwal, A., and Joachims, T. Intervention harvesting for context-dependent examination-bias estimation. arXiv preprint arXiv:1811.01802, 2018.