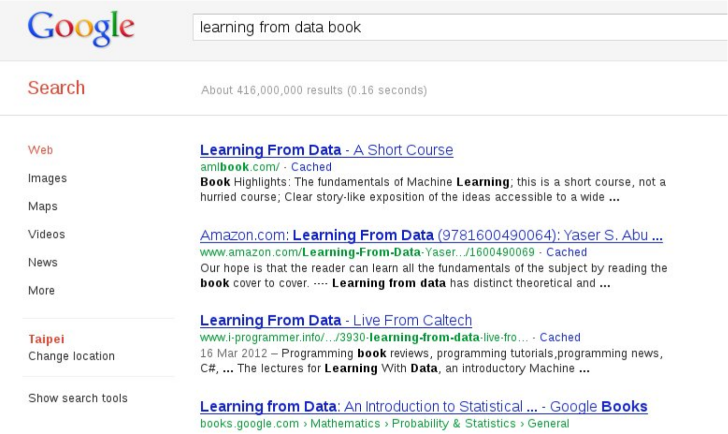
Improving Ranking Performance with Cost-sensitive Ordinal Classification via Regression

Yu-Xun Ruan[1], Hsuan-Tien Lin[1], Ming-Feng Tsai[2]

National Taiwan University[1], National Chengchi University[2]

Preference Learning @ EURO, July 10, 2012

Preference Ranking in Search Engine



not just for searching **good machine learning book**;
but also for **recommendation systems & other web service**

Three Properties of Search-Engine Ranking

- listwise with focus on **top ranks**
  - query-oriented & personalized
  - emphasis on **highly-preferred (relevant)** items
- **large scale**
  - both during **training** & testing
  - e.g. Yahoo! Learning-To-Rank Challenge 2010: 473K training URLs, 166K test URLs
- **ordinal data**
  - labeled qualitatively by human, e.g. { `highly irrelevant, irrelevant, neutral, relevant, highly relevant` }
  - **lack of quantitative info**

search-engine ranking problem:
learning a ranker from **large scale ordinal data**
with focus on **top ranks**

Search-Engine Ranking Setup

Given

for query indices $q = 1, 2, \cdots, Q$,

- a set of related documents $\{\mathbf{x}_{q,i}\}_{i=1}^{N(q)}$
- ordinal relevance $y_{q,i} \in \mathcal{Y} = \{0, 1, \ldots, K\}$ for each document $\mathbf{x}_{q,i}$

with large Q and $N(q)$

Goal

a ranker $r(\mathbf{x})$ that "accurately ranks" top $\mathbf{x}_{Q+1,i}$ from an **unseen** set of documents $\{\mathbf{x}_{Q+1,i}\}$

how to evaluate **accurate ranking around the top**?

Expected Reciprocal Rank (ERR; Chapelle et al., CIKM '09)

Assumption: Choice Probability of Single Document

for any example (document $\mathbf{x}$, rank y),

$$P(\text{user chooses document } \mathbf{x}) = (2^y - 1)/2^K$$

Assumption: Stopping Probability of **List of Documents**

$$\begin{aligned} & P(\text{user stops at position } i \text{ of list}) \\ = \; & P(\text{doesn't stop at pos. } i-1) \times P(\text{chooses document at pos. } i) \end{aligned}$$

ERR: Total **Discounted** Stopping Probability of List of Documents

$$ERR_q(r) \equiv \sum_{i=1}^{N(q)} \frac{1}{i} P(\text{user stops at position } i \text{ of the list ordered by } r)$$

large ERR $\Leftrightarrow$ small i matches large P $\Leftrightarrow$ good ranking around top

Possible Approach 1: LambdaRank (Burges et al., NIPS '06)

maximize ERR directly with non-smooth optimization on $N(q)!$ list reorderings

Pros

- respect top rank goal
- respect ordinal nature of data

Cons

- **difficult optimization problem**
- challenging to apply on large-scale data

LambdaRank: a state-of-the-art approach, but **possibly inefficient**

Possible Approach 2: SVM-Rank (Joachims, KDD ’02)

conduct listwise ranking by predicting pairwise preferences accurately

Pros

- respect ordinal nature of data (w/ comparison)
- somewhat applicable to large-scale data

Cons

- all pairs equal, not respecting top rank goal
- **somewhat** applicable to large-scale data, because of $O(N^2)$ pairs

SVM-Rank: a baseline pairwise ranking approach, but **possibly not the best for listwise**

Possible Approach 3: Direct Regression (Cossock and Tong, COLT '06)

conduct listwise ranking by predicting real-valued scores accurately

Pros

- respect top rank goal by embedding it in regression loss
- applicable to large-scale data

Cons

- treats y as numerical score, not respecting ordinal nature of data

Direct Regression: a simple pointwise ranking approach, but **may be improved by taking ordinal property into account**

Possible Approach 4: Ordinal Classification (MCRank; Li et al., NIPS '07)

conduct listwise ranking by predicting ordinal-valued ranks accurately

Pros

- somewhat respect top rank goal
- respect ordinal nature of data
- applicable to large-scale data

Cons

- **somewhat** respect top rank goal because of a loose bound in embedding the goal

McRank: a state-of-the-art pointwise ranking approach, but **may be improved further towards top rank goal**

Our Contributions

an algorithmic development on cost-sensitive ordinal classification via regression (COCR), which ...

- **systematically respects all three properties** of search-engine ranking

algorithm	top rank	large scale	ordinal data
LambdaRank	$\star$	$\circ$	$\star$
SVM-Rank	$\times$	$\circ$	$\star$
Direct Regression	$\star$	$\star$	$\times$
McRank	$\circ$	$\star$	$\star$
COCR	$\bigstar$	$\bigstar$	$\bigstar$

- leads to **promising experimental results**

Overview of Cost-sensitive Ordinal Classification via Regression (COCR)

- reduction from listwise ranking (ERR) to cost-sensitive ordinal classification (approximately)
  —aim for **top rank** and **large scale data** (like Direct Regression)
- reduction from cost-sensitive ordinal classification to binary classification
  —aim for **respecting ordinal data** (like McRank)
- reduction from binary classification to regression
  —aim for **large scale data** and **avoiding discrete ties** (like Direct Regression)

COCR: combine the benefits of Direct Regression and McRank

Ordinal Classification via Binary Classification

(Lin & Li, Neural Computation '12)

desired pointwise ranking problem

$r(\mathbf{x})$ = *What is the rank of the document* $\mathbf{x}$*?*

reduced problems

$g_k(\mathbf{x})$ = *Is the rank of document* $\mathbf{x}$ *greater than* k*?*

- train binary classifiers with $\{(\mathbf{x}_{q,i}, [y_{q,i} > k])\}$
- predict with a simple counting ranker $r_g(\mathbf{x}) = \sum_{k=0}^{K-1} g_k(\mathbf{x})$
- **simple** and **efficient**

good theoretical guarantee:

1. absolutely good binary classifier $\Longrightarrow$ absolutely good ranker
2. relatively good binary classifier $\Longrightarrow$ relatively good ranker

Ordinal Classification via Regression

desired pointwise ranking problem

$E(y|\mathbf{x})$ = *What is the expected rank of the document* $\mathbf{x}$*?*

- exploited by both Direct Regression and McRank

reduced problems

$\tilde{g}_k(\mathbf{x}) = P(y > k|\mathbf{x})$ = *What is the probability that the rank of document* $\mathbf{x}$ *is greater than* k*?*

- train regressors with $\{(\mathbf{x}_{q,i}, [y_{q,i} > k])\}$
- predict with a simple counting estimator $E(y|\mathbf{x}) = \sum_{k=0}^{K-1} \tilde{g}_k(\mathbf{x})$

absolutely good regressor $\Longrightarrow$ absolutely good expected rank estimator

Cost-sensitive Ordinal Classification via Regression

desired pointwise ranking problem

$E_{\mathbf{c}}(y|\mathbf{x})$ = *What is the biased expected rank of the document* $\mathbf{x}$ *if if a mis-ranking is penalized with a cost* $\mathbf{c}[r(\mathbf{x})]$*?*

- for embedding the emphasis on top rank

reduced problems

$\tilde{g}_{k,\mathbf{w}}(\mathbf{x})$ = *What is the biased probability that the rank of document* $\mathbf{x}$ *is greater than k when a wrong answer is penalized with a weight* w_k*?*

- train regressors with $\{(\mathbf{x}_{q,i}, [y_{q,i} > k], w_{q,i,k})\}$
- predict with a simple counting estimator $E_{\mathbf{c}}(y|\mathbf{x}) = \sum_{k=0}^{K-1} \tilde{g}_{k,\mathbf{w}}(\mathbf{x})$

some good theoretical guarantees follow similarly

Optimistic ERR (oERR) Cost for COCR

desired listwise criteria

How to make ERR(r) close to ERR(p), the ERR of perfect ranker?

embed criteria within cost

$$ERR(p) - ERR(r) \leq \blacksquare \cdot \left(\sum_{i=1}^{N(q)} \left(2^{y_{q,i}} - 2^{r(\mathbf{x}_{q,i})} \right)^2 + \triangle \right)$$

- $\triangle \approx 0$ if $r \approx p$ (optimistic)
- then, $\mathbf{c}[k] = \left(2^y - 2^k\right)^2$ embeds ERR

not a very tight bound, but **better than nothing**
—heuristically used in some earlier works

The Proposed Algorithm

Given

for query indices $q = 1, 2, \cdots, Q$,

- a set of related documents $\{\mathbf{x}_{q,i}\}_{i=1}^{N(q)}$
- ordinal relevance $y_{q,i} \in \mathcal{Y} = \{0, 1, \ldots, K\}$ for each document $\mathbf{x}_{q,i}$

with large Q and $N(q)$

1. construct $\{(\mathbf{x}_{q,i}, y_{q,i}, \mathbf{c}[k])\}$ with oERR cost $\mathbf{c}$
2. obtain $\{(\mathbf{x}_{q,i}, [y_{q,i} > k], w_{q,i,k})\}$ by reduction to binary classification
3. train regressors $\tilde{g}_k(\mathbf{x})$ with $\{(\mathbf{x}_{q,i}, [y_{q,i} > k], w_{q,i,k})\}$
4. predict (order) future document $\mathbf{x}$ with $\sum_{k=0}^{K-1} \tilde{g}_k(\mathbf{x})$

systematic, simple, efficient, and take all three properties into account

Empirical Comparison Using Linear Regression

data set	Direct Regression	McRank-like	oERR-COCR
LTRC1	0.4470	0.4484	0.4505
LTRC2	0.4440	0.4465	0.4461
MS10K	0.2643	0.2642	0.2792
MS30K	0.2748	0.2748	0.2942

- best ERR
- significantly better than direct regression

oERR-COCR **usually the best**, and ordinal information is important

Empirical Comparison Using M5' Decision Tree

data set	Direct Regression	McRank-like	oERR-COCR
LTRC1	0.4499	0.4526	0.4530
LTRC2	0.4489	0.4499	0.4538
MS10K	0.3014	0.3129	0.3156
MS30K	0.3298	0.3438	0.3451

- best ERR
- significantly better than direct regression

oERR-COCR **the best**

Conclusion

- Cost-sensitive Ordinal Classification via Regression
  - emphasize on top rank
  - respect ordinal data
  - regress pointwise for large-scale data
- theoretical guarantee:
  - reduction from listwise to cost-sensitive ordinal, approximately
  - reduction from cost-sensitive ordinal to binary
  - reduction from binary to regression
- obtained **good experimental results**

Thank you. Questions?