Learning to Rank

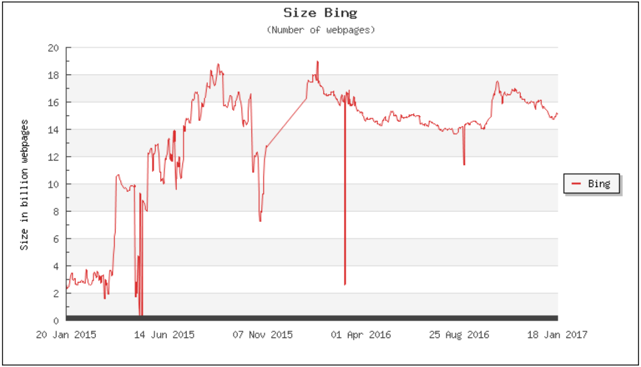
Dr. Josif Grabocka

ISMLL, University of Hildesheim

Business Analytics

Information Retrieval - Motivation

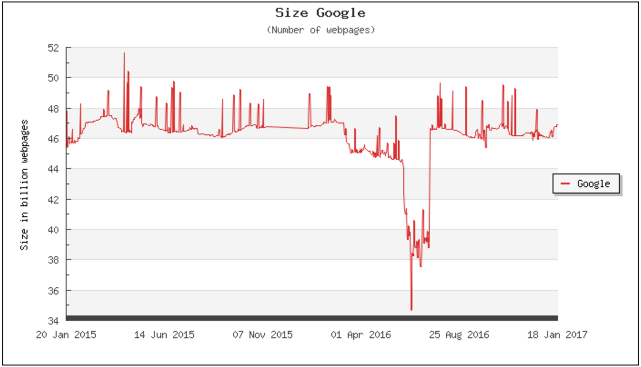
Bing indexes ca. 16 billion websites:



Source: worldwidewebsize.com (18.01.2017)

Information Retrieval - Motivation (II)

Google indexes ca. 46 billion websites:



Source: worldwidewebsize.com (18.01.2017)

Information Retrieval - Motivation (III)

Amazon offers:

- Totally 353,710,754 products, among which:

- Cell Phones & Accessories: 82,039,731 products
- Home & Kitchen: 64,274,875 products
- Clothing, Shoes & Jewelry: 33,422,437 products (including categories for Men, Women, Girls, Boys and Baby)
- Electronics: 31,604,887 products
- Sports & Outdoors: 23,997,293 products

Source: 360pi.com (18.01.2017)

Information Retrieval - Motivation (IV)

YouTube has:

- 1,3 billion users
- 4,9 billion videos viewed daily
- 300 hours of new content uploaded every minute
- 3.2 billion hours of videos watched each month
- 10,113 videos with more than 1 billion views

Source: statisticbrain.com (18.01.2017)

Indexing and Retrieval

- Information is worthless without retrieval.
- Two stage process:
  (i) Indexing: preprocessing and storing information, **crawling and indexing**
  (ii) Retreival: issuing a **query**, accessing the index, and finding **documents** relevant to the query

Retrieval Terms

- Document: A piece of information, such as a web page, article, book, video, song
  - Usually text information.
  - But, what about feature-rich data (audio, image, video)?
- Query: Text containing the user's information need
- Relevance:
  - Indicates how relevant is a particular document for the query
  - Relevance is defined within the scope of a query, it is a binary relation between documents and queries
  - A document can result on multiple queries with different relevances
  - How is relevance determined?

Illustrative Example

Query: "Brexit":

Relevance	Document	Features
1	Wikipedia, United Kingdom"s withdrawal ...	x_1
1	BBC, Brexit: All you need to know ...	x_2
1	Independent, Theresa May challenged ...	x_3
0	Fidessa, Brexit hangover ...	x_4
0	Vanguard, Brexit: What does Vanguard think ...	x_5

Problem Definition

- For each query $q = 1, \ldots, Q$,
- Given a list of n query-matching documents' features $x_i \in \mathbb{R}^M, i = 1, \ldots, n$,
- Given the relevances of the documents within the query $l_1, l_2, \ldots, l_n$
- Learn a function $f : \mathbb{R}^M \rightarrow \mathbb{R}$ that predicts relevance scores $s_i = f(x_i), i = 1, \ldots, n$
- Such that:
  - The ranking of estimated relevances s matches the ranking of the true relevances l
  - According to a ranking loss $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ that measures the correctness of the estimated relevances for query q

Problem Definition (II)

- f is a parametric function with parameters θ, e.g. a linear function:
  - $f(x_i) = \sum_{m=1}^{M} x_{i,m}\theta_m$
- Or a neural network, a decision tree, an ensemble of trees, etc ...

- The ultimate objective to be optimized is:

$$\underset{\theta}{\operatorname{argmin}} \quad \sum_q \mathcal{L}(l^{(q)}, s^{(q)})$$

$$s_i^{(q)} = f(x_i; \theta), \quad i = 1, \ldots, n$$

Approaches for Ranking Loss

- Point-wise:
  - Treat the relevance prediction as a regression
- Pair-wise:
  - Decompose ranking accuracy through pair-wise ranking
- List-wise:
  - Measure ranking over the full set

Why is the pairwise approach not optimal in information retrieval?

Normalized discounted cumulative gain (NDCG)

The discounted cummulative gain (**DCG**):

- Sort the documents according to the estimated relevances $s \in \mathbb{R}^n$
- Compute:

$$DCG@K = \sum_{i=1}^{K} \frac{2^{l_i} - 1}{\log_2(i+1)}$$

The normalized cumulative gain (**NDCG**):

- Sort the documents according to the ground-truth relevances $l \in \mathbb{R}^n$ to get the ideal $DCG@K$, denoted $IDCG@K$
- Compute:

$$NDCG@K = \frac{DCG@K}{IDCG@K}$$

NDCG Example (I)

- Let our query have 5 documents $x_1, \ldots, x_5$ with relevances $l = [3, 2, 1, 0, 0]$
- We learned a function f that predicts relevances $s = [3, 0, 2, 1, 0]$
- Compute terms:

rank	x_i	l_i	$\log_2(i+1)$	$\frac{2^{l_i}-1}{\log_2(i+1)}$
1	x_1	3	0.30	23.25
2	x_3	1	0.47	2.09
3	x_4	0	0.60	0
4	x_2	2	0.69	4.29
5	x_5	0	0.77	0

- $DCG@5 = 29.64$

NDCG Example (II)

- Optimal is sorted by l:

rank	x_i	l_i	$\log_2(i+1)$	$\frac{2^{l_i}-1}{\log_2(i+1)}$
1	x_1	3	0.30	23.25
2	x_2	2	0.47	6.28
3	x_3	1	0.60	1.66
4	x_4	0	0.69	0
5	x_5	0	0.77	0

- $IDCG@5 = 31.02$
- $NDCG@5 = \frac{DCG@K}{IDCG@K} = \frac{29.64}{31.20} = 0.94$

NDCG Example (III)

- Another algorithm outputs $s = [3, 2, 0, 1, 0]$
- Sorted by s:

rank	x_i	l_i	$\log_2(i+1)$	$\frac{2^{l_i}-1}{\log_2(i+1)}$
1	x_1	3	0.30	23.25
2	x_2	2	0.47	6.28
3	x_4	0	0.60	0
4	x_3	1	0.69	1.43
5	x_5	0	0.77	0

- $DCG@5 = 30.97$
- $IDCG@5 = 31.02$
- $NDCG@5 = \frac{DCG@K}{IDCG@K} = \frac{30.97}{31.20} = 0.99$

Pairwise Rank Approach

Given a ranking order among all documents of query q:

$$i <_q j \text{ iff } l_i > l_j$$

We **estimate** the probability that a pair is correctly ranked as:

$$\hat{P}_{i,j} = \hat{P}(i <_q j) = \frac{1}{1 + \exp^{-(s_i - s_j)}}$$

Pairwise Rank Loss

The loss of correctly ranking a pair i, j is

$$\mathcal{L}_{i,j} = -P_{i,j} \log(\hat{P}_{i,j}) - (1 - P_{i,j}) \log(1 - \hat{P}_{i,j})$$

where the ground-truth probability follows the given relevances:

$$P_{i,j} = \begin{cases} 1 & l_i > l_j \\ 0.5 & l_i = l_j \\ 0 & l_i < l_j \end{cases}$$

Pairwise Rank Loss (II)

Introduce $S_{i,j}$ for $P_{i,j} = \frac{1}{2}(1 + S_{i,j})$:

$$S_{i,j} = \begin{cases} 1 & l_i > l_j \\ 0 & l_i = l_j \\ -1 & l_i < l_j \end{cases}$$

yielding the loss:

$$\mathcal{L}_{i,j} = \frac{1}{s}(1 - S_{i,j})(s_i - s_j) + \log(1 + e^{-(s_i - s_j)})$$

Pairwise Rank Loss (III)

Given the loss:

$$\mathcal{L}_{i,j} = \frac{1}{2}(1 - S_{i,j})(s_i - s_j) + \log(1 + e^{-(s_i - s_j)})$$

The gradients are:

$$\frac{\partial \mathcal{L}_{i,j}}{\partial s_i} = \left(\frac{1}{2}(1 - S_{i,j}) - \frac{1}{1 + e^{(s_i - s_j)}}\right) = -\frac{\partial \mathcal{L}_{i,j}}{\partial s_j}$$

How to update model parameters?

Given the gradients:

$$\frac{\partial \mathcal{L}_{i,j}}{\partial s_i} = \left(\frac{1}{2}(1 - S_{i,j}) - \frac{1}{1 + e^{(s_i - s_j)}}\right) = -\frac{\partial \mathcal{L}_{i,j}}{\partial s_j}$$

Utilize the chain-rule of derivations as:

$$\theta_m \leftarrow \theta_m - \eta \left(\frac{\partial \mathcal{L}_{i,j}}{\partial s_i}\frac{\partial s_i}{\partial \theta_m} + \frac{\partial \mathcal{L}_{i,j}}{\partial s_j}\frac{\partial s_j}{\partial \theta_m}\right)$$

Learning Algorithm

procedure LEARNPAIRWISERANKING

input: $\mathcal{I}^{(q)} := \left\{(i,j) \mid l_i^{(q)} < l_j^{(q)}\right\}, \eta, \sigma$

$\theta_m \sim N(0, \sigma\mathbf{I}),\ m = 1, \ldots, M$

repeat

for $q = 1, \ldots, Q$ **do**

for $(i,j) \in \mathcal{I}^{(q)}$ **do** ▷ In a random order

for $m = 1, \ldots, M$ **do**

$\frac{\partial s_i}{\partial \theta_m} \leftarrow \frac{\partial f(x_i, \theta)}{\partial \theta_m}$

$\frac{\partial s_j}{\partial \theta_m} \leftarrow \frac{\partial f(x_j, \theta)}{\partial \theta_m}$

$\theta_m \leftarrow \theta_m - \eta \left(\frac{\partial \mathcal{L}_{i,j}}{\partial s_i} \frac{\partial s_i}{\partial \theta_m} + \frac{\partial \mathcal{L}_{i,j}}{\partial s_j} \frac{\partial s_j}{\partial \theta_m} \right)$

end for

end for

end for

until convergence

return θ

Improved Learning Runtime

$$\frac{\partial \mathcal{L}_{i,j}}{\partial \theta_m} = \frac{\partial \mathcal{L}_{i,j}}{\partial s_i}\frac{\partial s_i}{\partial \theta_m} + \frac{\partial \mathcal{L}_{i,j}}{\partial s_j}\frac{\partial s_j}{\partial \theta_m}$$

Remember our loss:

$$\mathcal{L}_{i,j} = \frac{1}{2}(1 - S_{i,j})(s_i - s_j) + \log(1 + e^{-(s_i - s_j)})$$

The gradients are:

$$\frac{\partial \mathcal{L}_{i,j}}{\partial s_i} = \left(\frac{1}{2}(1 - S_{i,j}) - \frac{1}{1 + e^{(s_i - s_j)}}\right) = -\frac{\partial \mathcal{L}_{i,j}}{\partial s_j}$$

Improved Learning Runtime (II)

$$
\begin{aligned}
\frac{\partial \mathcal{L}_{i,j}}{\partial \theta_m} &= \frac{\partial \mathcal{L}_{i,j}}{\partial s_i}\frac{\partial s_i}{\partial \theta_m} + \frac{\partial \mathcal{L}_{i,j}}{\partial s_j}\frac{\partial s_j}{\partial \theta_m} \\
\frac{\partial \mathcal{L}_{i,j}}{\partial \theta_m} &= \lambda_{i,j}\left(\frac{\partial s_i}{\partial \theta_m} - \frac{\partial s_j}{\partial \theta_m}\right)
\end{aligned}
$$

where

$$
\lambda_{i,j} = \left(\frac{1}{2}(1 - S_{i,j}) - \frac{1}{1 + e^{(s_i - s_j)}}\right) \tag{1}
$$

Improved Learning Runtime (III)

Notation: Denote $\mathcal{I} := \{(i,j) \mid l_i < l_j\}$, dropping index q for simplicity.

The total amount of updates on θ_m:

$$\delta\theta_m = -\eta \sum_{(i,j)\in\mathcal{I}} \left(\lambda_{i,j} \frac{\partial s_i}{\partial \theta_m} - \lambda_{i,j} \frac{\partial s_j}{\partial \theta_m} \right)$$

Define:

$$\lambda_i = \sum_{j:(i,j)\in\mathcal{I}} \lambda_{i,j} - \sum_{j:(j,i)\in\mathcal{I}} \lambda_{j,i} \tag{2}$$

Leading to:

$$\delta\theta_m = -\eta \sum_i \lambda_i \frac{\partial s_i}{\partial \theta_m}$$

Improved Learning Algorithm

procedure LEARNPAIRWISERANKINGIMPROVED

input: $\mathcal{I}^{(q)} := \left\{(i,j) \mid l_i^{(q)} < l_j^{(q)}\right\}, \eta, \sigma$

$\theta_m \sim N(0, \sigma \mathbf{I}),\ m = 1, \ldots, M$

repeat

for $q = 1, \ldots, Q$ **do**

$s_i := f(x_i,\ \theta),\ i = 1, \ldots, n$ ▷ Compute s_i

$\lambda_{i,j} := \left(\frac{1}{2}(1 - S_{i,j}) - \frac{1}{1+e^{(s_i - s_j)}}\right),\ (i,j) \in \mathcal{I}$ ▷ Compute $\lambda_{i,j}$

$\lambda_i := \sum_{j:(i,j)\in\mathcal{I}} \lambda_{i,j} - \sum_{j:(j,i)\in\mathcal{I}} \lambda_{j,i},\quad i = 1, \ldots, n$ ▷ Compute λ_i

for $m = 1, \ldots, M$ **do**

$$\theta_m \leftarrow \theta_m - \eta \sum_{i=1}^{n} \lambda_i \frac{\partial f(x_i, \theta)}{\partial \theta_m}$$

end for

end for

until convergence

return θ

Pairwise Loss is non-optimal for NDCG

Source: Burges 2010, MSR-TR

LambdaRank Heuristic

Update the parameters by taking into account the amount of NDCG change that would result by swapping the ranking positions of the pair:

$$\lambda_{i,j} \approx \frac{-1}{1 + e^{(s_i - s_j)}} |\Delta NDCG_{i,j}|$$

In a way that maximizes the gain:

$$\begin{aligned} \theta_m &\leftarrow \theta_m + \eta \sum_{i=1}^{n} \lambda_i \frac{\partial s_i}{\partial \theta_m} \\ \lambda_i &:= \sum_{j:(i,j) \in \mathcal{I}} \lambda_{i,j} - \sum_{j:(j,i) \in \mathcal{I}} \lambda_{j,i} \end{aligned}$$

- Take into account the importance of the pair for NDCG
- A large $|\Delta NDCG|$ shows that the pair is important

How to compute the change in NDCG?

Given the old $DCG@K$:

$$DCG@K^{(\text{old})} = \sum_{i=1}^{K} \frac{2^{l_i} - 1}{\log_2(i+1)}$$

What happens if documents in positions q and r change place?

$$\begin{aligned} DCG@K^{(\text{new})} \quad &= \quad DCG@K^{(\text{old})} - \frac{2^{l_q} - 1}{\log_2(q+1)} - \frac{2^{l_r} - 1}{\log_2(r+1)} \\ &+ \quad \frac{2^{l_q} - 1}{\log_2(\mathbf{r}+1)} + \frac{2^{l_r} - 1}{\log_2(\mathbf{q}+1)} \end{aligned}$$

$IDCG@K$ remains the same, therefore $|\Delta NDCG_{q,r}|$ is an $\mathcal{O}(1)$ operation.

LambdaRank Optimization

procedure LearnLambdaRank

input: $\mathcal{I}^{(q)} := \left\{(i,j) \mid l_i^{(q)} < l_j^{(q)}\right\}, \eta, \sigma$

$\theta_m \sim N(0, \sigma \mathbf{I}), \; m = 1, \ldots, M$

repeat

for $q = 1, \ldots, Q$ **do**

$s_i := f(x_i, \; \theta), \; i = 1, \ldots, n$ ▷ Compute s_i

$NDCG@K \leftarrow \frac{DCG@K}{IDCG@K}$ ▷ Compute $NDCG@K$

$\lambda_{i,j} := \frac{-1}{1+e^{(s_i - s_j)}} |\Delta NDCG_{i,j}|, \; (i,j) \in \mathcal{I}$ ▷ Compute $\lambda_{i,j}$

$\lambda_i := \sum_{j:(i,j)\in\mathcal{I}} \lambda_{i,j} - \sum_{j:(j,i)\in\mathcal{I}} \lambda_{j,i}, \; i = 1, \ldots, n$ ▷ Compute λ_i

for $m = 1, \ldots, M$ **do**

$$\theta_m \leftarrow \theta_m + \eta \sum_{i=1}^{n} \lambda_i \frac{\partial f(x_i, \theta)}{\partial \theta_m}$$

end for

end for

until convergence

What is $f(x_i, \theta)$?

- It can be a Neural Network, known as RankNet
- It can be a Gradient Boosted Decision Tree, known as LambdaMART, (now implemented in XGBoost)

- In the next lecture, we will see how to learn Gradient Boosted Decision Trees (GBDT) for Ranking.
  - Before that, read the last GBDT slides on Predictive Analytics.