情報検索システム特論
Advanced Information Retrieval Systems
第5回 Lecture #5

2023-05-08

湯川 高志 Takashi Yukawa

Reference Collections

Refer the material for lecture #4, pp.33-47

More Variation in Performance Evaluation

User-Oriented Measures (1)

- Recall and precision assumes that the set of relevant documents for a query is independent of the users

- Different users might have different relevance interpretations

- User-oriented measures have been proposed

User-Oriented Measures (2)

- As before,
  - consider a reference collection, an information request I, and a retrieval algorithm to be evaluated
  - with regard to I, let R be the set of relevant documents and A be the set of answers retrieved
- Also, let K be the set of documents of the collection known to the user
- The set $K \cap R \cap A$ is composed of the relevant documents known to the user that have been retrieved
- The set $(R \cap A) - K$ is composed of relevant documents that have been retrieved by are not known to the user

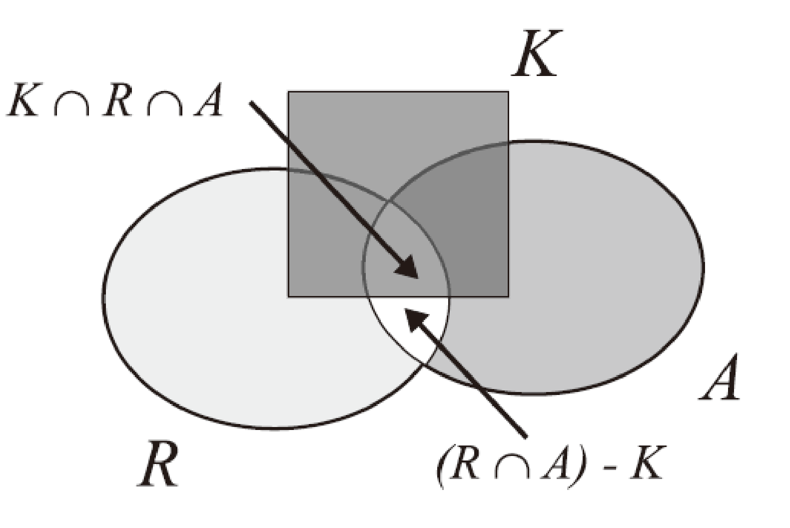
User-Oriented Measures (3)

- The coverage ratio: the fraction of the documents known and relevant that are in the answer set

$$coverage = \frac{|K \cap R \cap A|}{|K \cap R|}$$

- The novelty ratio: the fraction of the relevant documents in the answer set that are not known to the user

$$novelty = \frac{|(R \cap A) - K|}{|R \cap A|}$$



User-Oriented Measures (4)

- A high coverage indicates that the system is finding most of the relevant documents the user expected to see
- A high novelty indicates that the system is revealing many new relevant documents which were unknown
- Additionally, two other measures can be defined
  - The **relative recall** is the ratio between the number of relevant documents found and the number of relevant documents the user expected to find
  - The **recall effort** is the ratio between the number of relevant documents the user expected to find and the number of documents examined in an attempt to find the expected relevant documents

Discounted Cumulated Gain (1)

- Precision and recall allow only binary relevance assessments
- As a result, there is no distinction highly relevant documents and mildly relevant documents
- These limitations can be overcome by adopting graded relevance assessments and metrics that combine them
- The **discounted cumulated gain (DCG)** is metric that combine graded relevance assessments effectively

Discounted Cumulated Gain (2)

- When examining the results of a query, two key observations can be made
  - highly relevant documents are preferable at the top of the ranking than mildly relevant ones
  - relevant documents that appear at the end of the ranking are less valuable
- Consider that the results of the queries are graded on a scale 0–3 (0 for non-relevant, 3 for strong relevant documents)

Discounted Cumulated Gain (3)

- An example
  - for query q_1 and q_2
  - graded relevance
    $R_{q_1} = \{[d_3, 3],\ [d_5, 3],\ [d_9, 3],\ [d_{25}, 2],\ [d_{39}, 2],\ [d_{44}, 2],$
    $[d_{56}, 1],\ [d_{71}, 1],\ [d_{89}, 1], [d_{123}, 1]\ \}$
  - $R_{q_2} = \{[d_3, 3],\ [d_{56}, 2],\ [d_{129}, 1]\}$
- while document d_3 is highly relevant to query q_1, document d_{56} is just mildly relevant

Discounted Cumulated Gain (4)

- Specialists associate a graded relevance score to the top 10-20 results of the IR algorithm for a given query q
  - This list of relevance scores is referred to as the gain vector G
- Considering the top 15 documents in the ranking produced for queries q_1 and q_2, the gain vectors for these queries

$$G_1 = (1, 0, 1, 0, 0, 3, 0, 0, 0, 2, 0, 0, 0, 0, 3)$$
$$G_2 = (0, 0, 2, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 3)$$

Discounted Cumulated Gain (5)

- By summing up the graded scores up to any point in the ranking, we obtain the cumulated gain (CG)
- For query q_1, for instance, the cumulated gain at the first position is 1, at the second position is 1+0, and so on
- Thus, the cumulated gain vectors for queries q_1 and q_2 are given by

$$CG_1 = (1, 1, 2, 2, 2, 5, 5, 5, 5, 7, 7, 7, 7, 7, 10)$$
$$CG_2 = (0, 0, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 6)$$

- For instance, the cumulated gain at position 8 of CG_1 is equal to 5 → $CG_1[8] = 5$

Discounted Cumulated Gain (6)

- We also introduce a discount factor that reduces the impact of the gain as we move upper in the ranking
- A simple discount factor is the logarithms of the ranking position
- If we consider logs in base 2, this discount factor will be $\log_2 2$ at position 2, $\log_2 3$ at position 3, and so on
- By dividing a gain by the corresponding discount factor, we obtain the **discounted cumulated gain (DCG)**

$$DCG_j[i] = \begin{cases} G_j[1] & \text{if } i = 1 \\ \dfrac{G_j[i]}{\log_2 i} + DCG_j[i-1] & \text{otherwise} \end{cases}$$

Discounted Cumulated Gain (7)

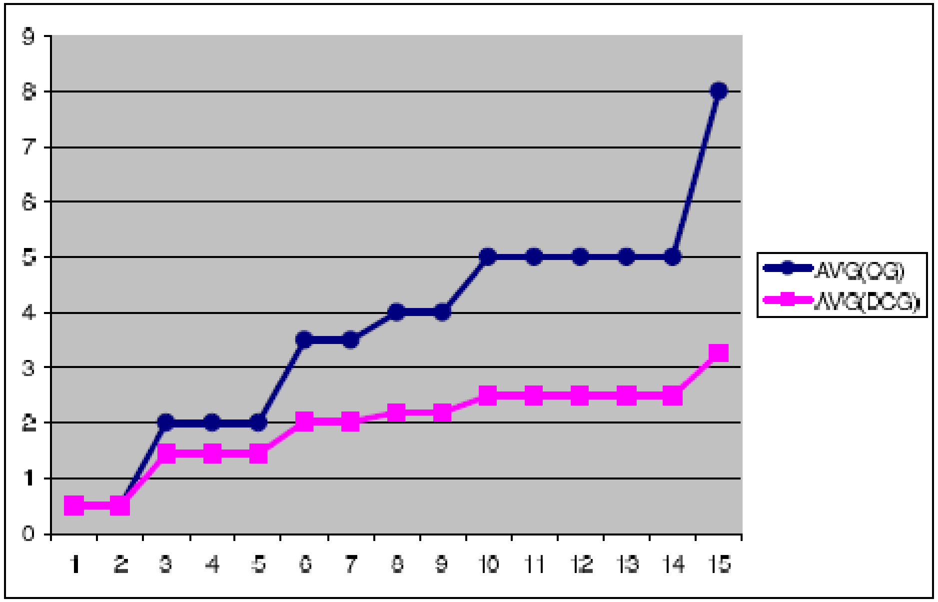
- For the example queries q1 and q2, the DCG vectors are given by
DCG_1
$= (1, 1, 1.6, 1.6, 1.6, 2.7, 2.7, 2.7, 2.7, 3.3, 3.3, 3.3, 3.3, 3.3, 4.1)$
DCG_2
$= (0, 0, 1.2, 1.2, 1.2, 1.2, 1.2, 1.6, 1.6, 1.6, 1.6, 1.6, 1.6, 1.6, 2.3)$
- Discounted cumulated gains are much less affected by relevant documents at the end of the ranking
- By adopting logs in higher bases the discount factor can be accentuated

DCG Curves

- To produce CG and DCG curves over a set of test queries, we need to average them over all queries
- Given a set queries, $\overline{CG}[i]$ and $\overline{DCG}[i]$ are averages over all queries
- For instance, for the example queries q_1 and q_2, these averages are given by
  $\overline{CG}$
  $= (0.5, 0.5, 2.0, 2.0, 2.0, 3.5, 3.5, 4.0, 4.0, 5.0, 5.0, 5.0, 5.0, 5.0, 8.0)$
  $\overline{DCG}$
  $= (0.5, 0.5, 1.4, 1.4, 1.4, 2.0, 2.0, 2.1, 2.1, 2.4, 2.4, 2.4, 2.4, 2.4, 3.2)$

DCG Curves (cont'd)

- Then, average curves can be drawn by varying the rank positions from 1 to a pre-established threshold
- In the example above, this threshold is set at 15, in the Web it is normally set at 10
- Figure below shows CG and DCG curves corresponding to the $\overline{CG}$ and $\overline{DCG}$ vectors



Ideal CG and DCG Metrics (1)

- Recall and precision: computed relatively to the set of relevant documents

⇕

- CG and DCG scores: not computed relatively to any baseline

⇓

- It might be confusing to use them directly to compare two distinct retrieval algorithms
- One solution to this problem is to define a baseline to be used for normalization
- This baseline are the ideal CG and DCG metrics

Ideal CG and DCG Metrics (2)

- For a given test query q assume that the relevance assessments made by the specialists produced:
  - n_3 documents evaluated with a relevance score of 3
  - n_2 documents evaluated with a relevance score of 2
  - n_1 documents evaluated with a score of 1
  - n_0 documents evaluated with a score of 0
- The ideal gain vector IG is created by sorting all relevance scores in decreasing order
  $IG = (3, \ldots, 3, 2, \ldots, 2, 1, \ldots, 1, 0, \ldots, 0)$
- For instance, for the example queries q_1 and q_2, we have
  $IG_1 = (3, 3, 2, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$
  $IG_2 = (3, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$

Ideal CG and DCG Metrics (3)

- Ideal CG and ideal DCG vectors can be computed analogously to the computations of CG and DCG
- For the example queries q_1 and q_2, the ideal CG vectors are
  $ICG_1 = (3, 6, 8, 9, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10)$
  $ICG_2 = (3, 5, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6)$
- The ideal DCG vectors are
  $IDCG_1 = (3, 6, 7.2, 7.7, 8.1, 8.1, 8.1, 8.1, 8.1, 8.1, 8.1, 8.1, 8.1, 8.1, 8.1)$
  $IDCG_2 = (3, 5, 5.6, 5.6, 5.6, 5.6, 5.6, 5.6, 5.6, 5.6, 5.6, 5.6, 5.6, 5.6, 5.6)$
- $\overline{ICG}$ and $\overline{IDCG}$ vectors are
  $\overline{ICG} = (3.0, 5.5, 7.0, 7.5, 8.0, 8.0, 8.0, 8.0, 8.0, 8.0, 8.0, 8.0, 8.0, 8.0, 8.0)$
  $\overline{IDCG} = (3.0, 5.5, 6.4, 6.7, 6.9, 6.9, 6.9, 6.9, 6.9, 6.9, 6.9, 6.9, 6.9, 6.9, 6.9)$
- By comparing the average CG and DCG curves for an algorithm with the average ideal curves we gain insight on how much room for improvement there is

Normalized DCG

- Precision and recall figures can be directly compared to the ideal curve of 100% precision at all recall levels

⇕

- DCG figures are not build relative to any ideal curve

⇓

- It is difficult to compare directly DCG curves for two distinct ranking algorithms

⇓

- This can be corrected by normalizing the DCG metric

Normalized DCG (cont'd)

$$NCG[i] = \frac{\overline{CG}[i]}{\overline{ICG}[i]},\ NDCG[i] = \frac{\overline{DCG}[i]}{\overline{IDCG}[i]}$$

- for the example queries q1 and q2
  $NCG =$ (0.17, 0.09, 0.29, 0.27, 0.25, 0.44, 0.44,0.50, 0.50, 0.63, 0.63, 0.63, 0.63, 0.63, 1.00)
  $NDCG =$ (0.17, 0.09, 0.22, 0.22, 0.21, 0.29, 0.29,0.32, 0.32, 0.36, 0.36, 0.36, 0.36, 0.36, 0.47)
- The area under the NCG and NDCG curves represent the quality of the ranking algorithm
- Higher the area, better the results are considered to be
- Thus, normalized figures can be used to compare two distinct ranking algorithms

Discussion on DCG Metrics

- CG and DCG metrics aim at taking into account multiple level relevance assessments
- This has the advantage of distinguish highly relevant documents from mildly relevant ones
- The inherent disadvantages are that multiple level relevance assessments are harder and more time consuming to generate

Discussion on DCG Metrics (cont’d)

- Despite these inherent difficulties, the CG and DCG metrics present benefits
  - They allow systematically combining document ranks and relevance scores
  - Cumulated gain provides a single metric of retrieval performance at any position in the ranking
  - It also stresses the gain produced by relevant documents up to a ranking position which makes the metrics more immune to outliers
  - Discounted cumulated gain allows down weighting the impact of relevant documents found late in the ranking

[TOPIC]
Performance Evaluation of Recognition/Classification Systems

[IMPORTANT] Assignment #1

That's it today

今日はここまで