Evaluation: Normalised Discounted Cumulated Gain (NDCG)

COMP3009J: Information Retrieval

Dr. David Lillis (david.lillis@ucd.ie)

UCD School of Computer Science
Beijing Dublin International College

Järvelin, K., & Kekäläinen, J. (2002). Cumulated gain-based evaluation of IR techniques. https://doi.org/10.1145/582415.582418

Normalised Discounted Cumulated Gain (NDCG)

- The metrics so far make use of **binary** relevance judgments:
  - Documents are judged to be **relevant** or **non-relevant**.
  - Any document that helps to satisfy an information need in any way is considered to be **relevant**.
- **BUT:** In reality, some documents are more relevant than others.
  - **Normalised Discounted Cumulated Gain (NDCG)** supports **graded relevance judgments**.

NDCG: Motivation

- NDCG rewards systems that:
  - Rank highly-relevant documents ahead of mildly relevant ones.
  - Position relevant documents in early positions in the ranking.

NDCG: Graded Relevance

- The first thing that is required is a set of **graded relevance judgments**.
  - Let us assume that a 0-3 scale where 3 is a highly relevant document and 0 is a non-relevant document.
  - R = {[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]}
    - i.e. d_3, d_5 and d_9 are highly relevant documents.
    - d_{25}, d_{39}, and d_{44} are relevant documents.
    - d_{56}, d_{71}, d_{89} and d_{123} are somewhat relevant documents.
    - Everything else is non-relevant.

NDCG: Example

- Let's revisit the example from before.
- This time, relevant documents are also shown with the degree of relevance attached.
- We create a **gain vector** that records the relevance level at each rank:
- G = (1, 0, 1, 0, 0, 3, 0, 0, 0, 2, 0, 0, 0, 0, 3)
- This is the starting point for our calculations.

Rank	**Document**	**G**
1	d_{123} (r: 1)	1
2	d_{84}	0
3	d_{56} (r: 1)	1
4	d_6	0
5	d_8	0
6	d_9 (r: 3)	3
7	d_{511}	0
8	d_{129}	0
9	d_{187}	0
10	d_{25} (r: 2)	2
11	d_{38}	0
12	d_{48}	0
13	d_{250}	0
14	d_{113}	0
15	d_3 (r: 3)	3

Cumulated Gain

- Next, we calculated a **cumulated gain vector** (also sometimes called a “cumulative gain vector”).
- Each rank *i* has its own CG value:
  - $CG[i] = \begin{cases} G[1], & i = 1; \\ G[i] + CG[i-1] & i > 1 \end{cases}$
- e.g. at rank 10:
  - CG[10] = G[10] + CG[9]
    = 2 + 5 = 7

Rank	Document	G	CG
1	d_{123} (r: 1)	1	1
2	d_{84}	0	1
3	d_{56} (r: 1)	1	2
4	d_{6}	0	2
5	d_{8}	0	2
6	d_{9} (r: 3)	3	5
7	d_{511}	0	5
8	d_{129}	0	5
9	d_{187}	0	5
10	d_{25} (r: 2)	2	7
11	d_{38}	0	7
12	d_{48}	0	7
13	d_{250}	0	7
14	d_{113}	0	7
15	d_{3} (r: 3)	3	10

Cumulated Gain

- The idea is that each relevant document we find should add to the gain (i.e. the overall usefulness of the list).
- More highly relevant documents add more to the gain than less relevant documents.
- **BUT:** documents late in the list can add as much to the gain as documents early in the list (e.g. at rank 15, a highly-relevant document adds 3 to the cumulated gain: the same as at rank 6).

Rank	Document	G	CG
1	d_{123} (r: 1)	1	1
2	d_{84}	0	1
3	d_{56} (r: 1)	1	2
4	d_{6}	0	2
5	d_{8}	0	2
6	d_{9} (r: 3)	3	5
7	d_{511}	0	5
8	d_{129}	0	5
9	d_{187}	0	5
10	d_{25} (r: 2)	2	7
11	d_{38}	0	7
12	d_{48}	0	7
13	d_{250}	0	7
14	d_{113}	0	7
15	d_{3} (r: 3)	3	10

Cumulated Gain

- This motivates us to create a vector for **Discounted Cumulated Gain (DCG)**.
- Here, later documents add less to the gain than earlier ones.
- It is calculated in the same way as CG, except that the gain at each rank is **discounted** by dividing it by the log of the rank (except for the first rank, which is unaffected).
- Each rank *i* has its own DCG value:
  - $$DCG[i] = \begin{cases} G[1], & i = 1; \\ \frac{G[i]}{log_2 i} + DCG[i-1] & i > 1 \end{cases}$$

Rank	Document	G	CG
1	d_{123} (r: 1)	1	1
2	d_{84}	0	1
3	d_{56} (r: 1)	1	2
4	d_{6}	0	2
5	d_{8}	0	2
6	d_{9} (r: 3)	3	5
7	d_{511}	0	5
8	d_{129}	0	5
9	d_{187}	0	5
10	d_{25} (r: 2)	2	7
11	d_{38}	0	7
12	d_{48}	0	7
13	d_{250}	0	7
14	d_{113}	0	7
15	d_{3} (r: 3)	3	10

DCG

- For this type of data, we could graph the DCG vector to compare two systems.
- However, it is not very suitable in this form.
    - Too many data points
    - Difficult to compare

Rank	Document	G	CG	Calculation	DCG
1	d_{123} (r: 1)	1	1	1	1
2	d_{84}	0	1		1
3	d_{56} (r: 1)	1	2	$\frac{1}{log_2 3} + 1$	1.6
4	d_6	0	2		1.6
5	d_8	0	2		1.6
6	d_9 (r: 3)	3	5	$\frac{3}{log_2 6} + 1.6$	2.8
7	d_{511}	0	5		2.8
8	d_{129}	0	5		2.8
9	d_{187}	0	5		2.8
10	d_{25} (r: 2)	2	7	$\frac{2}{log_2 10} + 2.8$	3.4
11	d_{38}	0	7		3.4
12	d_{48}	0	7		3.4
13	d_{250}	0	7		3.4
14	d_{113}	0	7		3.4
15	d_3 (r: 3)	3	10	$\frac{3}{log_2 15} + 3.4$	4.2

DCG: Analysis

- We have now calculated a Discounted Cumulated Gain vector for a query:

DCG=(1.0, 1.0, 1.6, 1.6, 1.6, 2.8, 2.8, 2.8, 2.8, 3.4, 3.4, 3.4, 3.4, 3.4, 4.2)

- This shows how finding relevant documents increases the quality of the results to that point.
  - More relevant documents contribute more to gain.
  - Relevant documents found earlier in the ranked list also contribute more to gain.

DCG: Analysis

- We have now calculated a Discounted Cumulated Gain vector for a query:

DCG=(1.0, 1.0, 1.6, 1.6, 1.6, 2.8, 2.8, 2.8, 2.8, 3.4, 3.4, 3.4, 3.4, 3.4, 4.2)

- **BUT**: By itself, this is not easy to compare with others.
  - Is 4.2 a good score for this query?
  - Which figure(s) do we choose to compare?
- Most evaluation metrics give a **single value** that is in the range **between 0 and 1**.
- **Normalised** DCG allows us to achieve this.

Normalised DCG (NDCG)

- **Normalised Discounted Cumulated Gain** is calculated by comparing the **DCG** vector against an **Ideal DCG vector**.
- The Ideal DCG vector is the DCG vector that we would see if the IR system had perfect retrieval.
  - i.e. it begins with all the documents of relevance level 3.
    - then it includes all the documents of relevance level 2.
    - then it includes all the documents of relevance level 1.
- We calculate an Ideal DCG vector to be the same length as the DCG vector (with 0 relevance values inserted at the end if there are not enough relevant documents).

NDCG: Example

- Let’s look again at the relevance judgments:
- R = {[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]}
- For this query, there are:
  - 3 documents at relevance level 3.
  - 3 documents at relevance level 2.
  - 4 documents at relevance level 1.
- The Ideal Gain vector (to compare with a ranked list of length 15) would be:
  - IG = (3, 3, 3, 2, 2, 2, 1, 1, 1, 1, 0, 0, 0, 0, 0)

- We can calculate the Ideal DCG vector (IDCG) in the same way as for the DCG vector

Rank	IG	Calculation	IDCG
1	3	3	3.0
2	3	$\frac{3}{log_2 2} + 3.0$	6.0
3	3	$\frac{3}{log_2 3} + 6.0$	7.9
4	2	$\frac{2}{log_2 4} + 7.9$	8.9
5	2	$\frac{2}{log_2 5} + 8.9$	9.8
6	2	$\frac{2}{log_2 6} + 9.8$	10.5
7	1	$\frac{1}{log_2 7} + 10.5$	10.9
8	1	$\frac{1}{log_2 8} + 10.9$	11.2
9	1	$\frac{1}{log_2 9} + 11.2$	11.5
10	1	$\frac{1}{log_2 10} + 11.5$	11.8
11	0		11.8
12	0		11.8
13	0		11.8
14	0		11.8
15	0		11.8

NDCG: Ideal DCG

- The IDCG vector represents the best possible DCG scores that a perfect IR system would achieve.
- IDCG=(3.0, 6.0, 7.9, 8.9, 9.8, 10.5, 10.9, 11.2, 11.5, 11.8, 11.8, 11.8, 11.8, 11.8, 11.8)
- We can **normalise** the DCG by dividing the score that was actually achieved at each rank by the ideal score, to yield a score between 0 and 1.

NDCG Calculation

Rank	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
DCG	1.0	1.0	1.6	1.6	1.6	2.8	2.8	2.8	2.8	3.4	3.4	3.4	3.4	3.4	4.2
IDCG	3.0	6.0	7.9	8.9	9.8	10.5	10.9	11.2	11.5	11.8	11.8	11.8	11.8	11.8	11.8
NDCG	0.33	0.17	0.20	0.18	0.16	0.27	0.26	0.25	0.24	0.29	0.29	0.29	0.29	0.29	0.36

- We still have the problem that we have 15 different scores for the evaluation.
- This is solved in a similar way to Precision@n: we choose a rank to measure NDCG at.
- NDCG@10 is a very commonly used metric: here it is 0.29

Features of NDCG

- Combines document ranks and graded relevance judgments.
- Single measure of quality at any rank, without needing to know recall.
- NDCG@n only considers relevant documents found to that point: not affected by many relevant documents being found very late.
- Gives less weight to relevant documents found late in the ranking.

Which metric should I use?

- We have looked at numerous different metrics for IR evaluation.
- All have their advantages and disadvantages.
- We need to use an appropriate metric in order to evaluate an IR system’s performance.
- How “performance” is defined is dependent on the final use of the system:
  - web search;
  - intranet search;
  - research environment;
  - desktop search;
  - legal search;
  - etc…..

Which metric should I use?

- A general rule is that if complete judgments are available any metric can be used.
- MAP gives a good indication of performance within a single metric as it is averaging the results over multiple queries.
- For collections that do not have complete judgments, bPref is a more suitable metric.
- For tasks such as web search (due to user behaviour), metrics like P@10 might be used.
- If graded relevance judgments are available, NDCG is preferred: this has continually gained in popularity in the last few years.
- In reality, most evaluations use multiple metrics.

Performing Evaluation

- A document collection for IR evaluation consists of:
  - Documents.
  - Standard queries.
  - Relevance judgments for the standard queries.

- A common source of documents includes the Text REtrieval Conference (TREC: https://trec.nist.gov).
  - Sets a set of IR-related challenges each year for participants to take part in (groups in both UCD an BJUT normally take part).
  - Results from each group, and the evaluation results are made available to researchers afterwards.