WIKIPEDIA

Discounted cumulative gain

Discounted cumulative gain (**DCG**) is a measure of ranking quality. In information retrieval, it is often used to measure effectiveness of web search engine algorithms or related applications. Using a graded relevance scale of documents in a search-engine result set, DCG measures the usefulness, or *gain*, of a document based on its position in the result list. The gain is accumulated from the top of the result list to the bottom, with the gain of each result discounted at lower ranks.[1]

Contents

Overview

Two assumptions are made in using DCG and its related measures.

1. Highly relevant documents are more useful when appearing earlier in a search engine result list (have higher ranks)
2. Highly relevant documents are more useful than marginally relevant documents, which are in turn more useful than non-relevant documents.

DCG originates from an earlier, more primitive, measure called Cumulative Gain.

Cumulative Gain

Cumulative Gain (CG) is the predecessor of DCG and does not include the position of a result in the consideration of the usefulness of a result set. In this way, it is the sum of the graded relevance values of all results in a search result list. The CG at a particular rank position p is defined as:

$$CG_p = \sum_{i=1}^{p} rel_i$$

Where rel_i is the graded relevance of the result at position i.

The value computed with the CG function is unaffected by changes in the ordering of search results. That is, moving a highly relevant document d_i above a higher ranked, less relevant, document d_j does not change the computed value for CG. Based on the two assumptions made above about the usefulness of search results, DCG is used in place of CG for a more accurate measure.

Discounted Cumulative Gain

The premise of DCG is that highly relevant documents appearing lower in a search result list should be penalized as the graded relevance value is reduced logarithmically proportional to the position of the result. The discounted CG accumulated at a particular rank position p is defined as:[2]

$$\mathrm{DCG_p} = \sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)} = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{\log_2(i+1)}$$

Previously there has not been any theoretically sound justification for using a logarithmic reduction factor[3] other than the fact that it produces a smooth reduction. But Wang et al. (2013)[4] give theoretical guarantee for using the logarithmic reduction factor in NDCG. The authors show that for every pair of substantially different ranking functions, the NDCG can decide which one is better in a consistent manner.

An alternative formulation of DCG[5] places stronger emphasis on retrieving relevant documents:

$$\mathrm{DCG_p} = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

The latter formula is commonly used in industry including major web search companies[2] and data science competition platform such as Kaggle.[6]

These two formulations of DCG are the same when the relevance values of documents are binary;[3]:320 $rel_i \in \{0, 1\}$.

Note that Croft et al. (2010) and Burges et al. (2005) present the second DCG with a log of base e, while both versions of DCG above use a log of base 2. When computing NDCG with the second formulation of DCG, the base of the log does not matter, but the base of the log does affect the value of NDCG for the first formulation. Clearly, the base of the log affects the value of DCG in both formulations.

Normalized DCG

Search result lists vary in length depending on the query. Comparing a search engine's performance from one query to the next cannot be consistently achieved using DCG alone, so the cumulative gain at each position for a chosen value of p should be normalized across queries. This is done by sorting all **relevant** documents in the corpus by their relative relevance, producing the maximum possible DCG through position p, also called Ideal DCG (IDCG) through that position. For a query, the *normalized discounted cumulative gain,* or nDCG, is computed as:

$$\mathrm{nDCG_p} = \frac{DCG_p}{IDCG_p},$$

where IDCG is ideal discounted cumulative gain,

$$\mathrm{IDCG_p} = \sum_{i=1}^{|REL|} \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

and |REL| represents the list of relevant documents (ordered by their relevance) in the corpus up to position p.

The nDCG values for all queries can be averaged to obtain a measure of the average performance of a search engine's ranking algorithm. Note that in a perfect ranking algorithm, the DCG_p will be the same as the $IDCG_p$ producing an nDCG of 1.0. All nDCG calculations are then relative values on the interval 0.0 to 1.0 and so are cross-query comparable.

The main difficulty encountered in using nDCG is the unavailability of an ideal ordering of results when only partial relevance feedback is available.

Example

Presented with a list of documents in response to a search query, an experiment participant is asked to judge the relevance of each document to the query. Each document is to be judged on a scale of 0-3 with 0 meaning not relevant, 3 meaning highly relevant, and 1 and 2 meaning "somewhere in between". For the documents ordered by the ranking algorithm as

$$D_1, D_2, D_3, D_4, D_5, D_6$$

the user provides the following relevance scores:

$$3, 2, 3, 0, 1, 2$$

That is: document 1 has a relevance of 3, document 2 has a relevance of 2, etc. The Cumulative Gain of this search result listing is:

$$CG_6 = \sum_{i=1}^{6} rel_i = 3 + 2 + 3 + 0 + 1 + 2 = 11$$

Changing the order of any two documents does not affect the CG measure. If D_3 and D_4 are switched, the CG remains the same, 11. DCG is used to emphasize highly relevant documents appearing early in the result list. Using the logarithmic scale for reduction, the DCG for each result in order is:

i	rel_i	$\log_2(i+1)$	$\dfrac{rel_i}{\log_2(i+1)}$
1	3	1	3
2	2	1.585	1.262
3	3	2	1.5
4	0	2.322	0
5	1	2.585	0.387
6	2	2.807	0.712

So the DCG_6 of this ranking is:

$$DCG_6 = \sum_{i=1}^{6} \frac{rel_i}{\log_2(i+1)} = 3 + 1.262 + 1.5 + 0 + 0.387 + 0.712 = 6.861$$

Now a switch of D_3 and D_4 results in a reduced DCG because a less relevant document is placed higher in the ranking; that is, a more relevant document is discounted more by being placed in a lower rank.

The performance of this query to another is incomparable in this form since the other query may have more results, resulting in a larger overall DCG which may not necessarily be better. In order to compare, the DCG values must be normalized.

To normalize DCG values, an ideal ordering for the given query is needed. For this example, that ordering would be the monotonically decreasing sort of all known relevance judgments. In addition to the six from this experiment, suppose we also know there is a document D_7 with relevance grade 3 to the same query and a document D_8 with relevance grade 2 to that query. Then the ideal ordering is:

$$3, 3, 3, 2, 2, 2, 1, 0$$

The DCG of this ideal ordering, or *IDCG (Ideal DCG)* , is computed to rank 6:

$$\mathrm{IDCG}_6 = 8.740$$

And so the nDCG for this query is given as:

$$\mathrm{nDCG}_6 = \frac{DCG_6}{IDCG_6} = \frac{6.861}{8.740} = 0.785$$

Limitations

1. Normalized DCG metric does not penalize for bad documents in the result. For example, if a query returns two results with scores $1,1,1$ and $1,1,1,0$ respectively, both would be considered equally good even if the latter contains a bad result. For the ranking judgments *Excellent, Fair, Bad* one might use numerical scores $1,0,-1$ instead of $2,1,0$. This would cause the score to lower if bad results are returned, prioritizing the precision of the results over the recall. Note that this approach can result in an overall negative score which would shift the lower bound of the score from 0 to a negative value.
2. Normalized DCG does not penalize for missing documents in the result. For example, if a query returns two results with scores $1,1,1$ and $1,1,1,1,1$ respectively, both would be considered equally good, assuming ideal DCG is computed to rank 3 for the former and rank 5 for the latter. One way to take into account this limitation is to enforce fixed set size for the result set and use minimum scores for the missing documents. In previous example, we would use the scores $1,1,1,0,0$ and $1,1,1,1,1$ and quote nDCG as nDCG@5.
3. Normalized DCG may not be suitable to measure performance of queries that may typically often have several equally good results. This is especially true when this metric is limited to only first few results as it is done in practice. For example, for queries such as "restaurants" nDCG@1 would account for only first result and hence if one result set contains only 1 restaurant from the nearby area while the other contains 5, both would end up having same score even though latter is more comprehensive.

References

1. Kalervo Jarvelin, Jaana Kekalainen: Cumulated gain-based evaluation of IR techniques. ACM Transactions on Information Systems 20(4), 422–446 (2002)
2. "Introduction to Information Retrieval - Evaluation" (http://www.stanford.edu/class/cs276/handouts/EvaluationNew-handout-6-per.pdf) (PDF). Stanford University. 21 April 2013. Retrieved 23 March 2014.
3. B. Croft; D. Metzler; T. Strohman (2010). *Search Engines: Information Retrieval in Practice. Addison Wesley".*
4. Yining Wang, Liwei Wang, Yuanzhi Li, Di He, Wei Chen, Tie-Yan Liu. 2013. A Theoretical Analysis of Normalized Discounted Cumulative Gain (NDCG) Ranking Measures. In Proceedings of the 26th Annual Conference on Learning Theory (COLT 2013).
5. Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. In Proceedings of the 22nd international conference on Machine learning (ICML '05). ACM, New York, NY, USA, 89-96. DOI=10.1145/1102351.1102363
   http://doi.acm.org/10.1145/1102351.1102363

6. "Normalized Discounted Cumulative Gain" (https://www.kaggle.com/wiki/NormalizedDiscountedCumulativeGain).
   Retrieved 23 March 2014.

Retrieved from "https://en.wikipedia.org/w/index.php?title=Discounted_cumulative_gain&oldid=820022249"

This page was last edited on 12 January 2018, at 16:57.