Contents lists available at ScienceDirect

Software Impacts

journal homepage: www.journals.elsevier.com/software-impacts

Original software publication

RIROSE: Rational Information Retrieval in Online Search Environments

Debora Di Caprio [a,*], Francisco J. Santos-Arteaga [b]

[a] *Department of Economics and Management, University of Trento, Trento, Italy*
[b] *Departamento de Análisis Económico y Economía Cuantitativa, Universidad Complutense de Madrid, Madrid, Spain*

ARTICLE INFO

Keywords:
Information retrieval
Online search
Decision trees
Benchmarking
Click-through rate
Pattern recognition

ABSTRACT

We design a benchmark algorithm that mimics the sequential behavior of users when retrieving information from the set of alternatives provided by an engine within the first page of online search results. The benchmark defined by the algorithm is designed to evaluate deviations from the rational retrieval strategies determined by the subjective preferences and beliefs of users. The algorithm accounts for the 2047 nodes composing the binary decision tree defined by the ten alternatives ranked within the first page of results. The flexibility of the algorithm allows to incorporate modifications accounting for search frictions and different degrees of impatience on the side of users, as well as testing the categorization capacities of machine learning techniques.

Code metadata

Current code version	*v1*
Permanent link to code/repository used for this code version	https://github.com/SoftwareImpacts/SIMPAC-2021-202
Permanent link to Reproducible Capsule	https://codeocean.com/capsule/1348726/tree/v1
Legal Code License	*MIT License*
Code versioning system used	*Code Ocean*
Software code languages, tools, and services used	*MATLAB*
Compilation requirements, operating environments & dependencies	
If available Link to developer documentation/manual	https://help.codeocean.com/en/collections/500063-user-manual
Support email for questions	

The main intuition on which the algorithms described in this paper are built is the design of an information retrieval process that is as close as possible to the behavior of a standard rational decision maker (DM) formalized in the decision theoretical literature while accounting for the sequential evaluation structures common to the main empirical studies.

In a nutshell, the benchmark algorithm defines an information retrieval framework where each DM sets out to evaluate the ten alternatives composing the initial page of results provided by a search engine. The DM follows a sequential process determined by the observations retrieved from the alternatives displayed, which condition his subsequent retrieval behavior.

Fig. 1 illustrates the retrieval process corresponding to the first four alternatives from a binary decision tree, where the behavior of the DM is determined by the value of the realization observed, x_i, $i = 1, 2, \dots, 10$, relative to a satisfying cutoff value, c_i, $i = 1, 2, \dots, 10$, conditioned by his preferences. This latter value is subjectively defined by the DM based on his beliefs regarding the potential realizations that may be observed from the characteristics composing the different alternatives. Trivially, whenever $x_i > c_i$, the DM evaluates the corresponding alternative before proceeding to observe the next one. Note how the DM must account for the whole history of evaluations performed – or omitted – as he proceeds through the ten alternatives composing the initial set of results provided by the engine.

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.
* Corresponding author.
E-mail addresses: debora.dicaprio@unitn.it (D. Di Caprio), fransant@ucm.es (F.J. Santos-Arteaga).

https://doi.org/10.1016/j.simpa.2022.100248
Received 3 January 2022; Received in revised form 28 January 2022; Accepted 31 January 2022

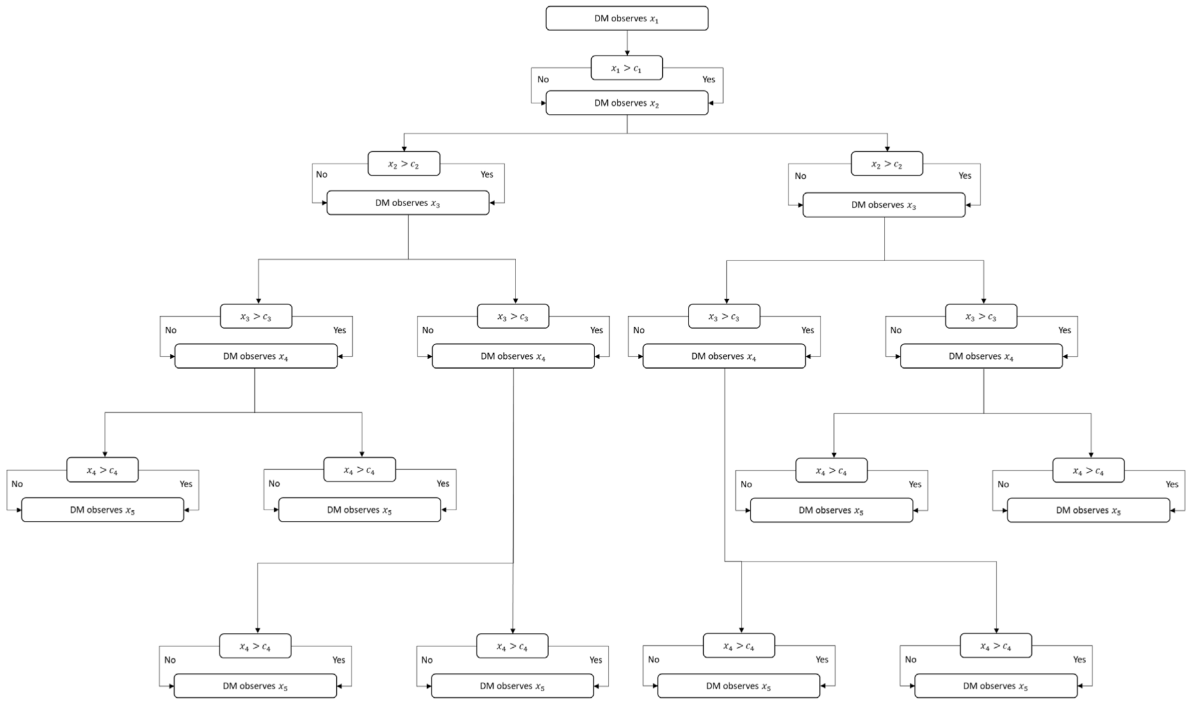

Fig. 1. Benchmark information retrieval process: Four initial alternatives composing a binary decision tree.

That is, the benchmark algorithm is a binary decision tree defined over a total of ten alternatives, accounting for 1023 binary decision nodes and 1024 final nodes, namely, a total of 2047 nodes. The algorithm allows to consider scenarios where the DM sets out to evaluate any given number of satisfying alternatives out of the initial page of results delivered by a search engine. Note that the complexity of the retrieval process increases substantially with each additional alternative considered. In this regard, the bound imposed on the benchmark algorithm is based on the behavior of DMs observed empirically, who focus on the first ten alternatives provided by search engines [1–3].

From an intuitive viewpoint, a basic algorithm consisting of ten individual observations, each with its corresponding cutoff value, can also be defined to describe the retrieval behavior of DMs observed empirically. Indeed, the output obtained from this basic algorithm cannot be differentiated from that of the benchmark algorithm accounting for the whole set of potential retrieval paths. However, the basic algorithm must be extended to formalize a retrieval process where DMs are asked to identify less than ten alternatives satisfying their subjective requirements among those provided by the engine. Even in this case, the basic algorithm does not identify the whole set of retrieval paths that may be generated by the DM, preventing the analysis of multiple interactions across the sets of potential observations and evaluations.

We must highlight the fact that the binary decision tree defining the retrieval process of rational DMs – within an online search environment – common to the economic, decision theoretical, and operational research literature had not been previously coded and tested. This is the case despite the fact that these research branches acknowledge the formal sequential process on which the algorithm is based and have validated empirically the retrieval behavior of DMs and their aim to observe and evaluate a predetermined number of satisfying alternatives [4].

The benchmark algorithm described in the following papers formalizes the sequential retrieval structure and subsequent paths summarized in Fig. 1 for a total of ten alternatives. The definition of each and every step that may be taken by the DM allows for the inclusion of a variety of modifications at specific points through the retrieval path determined by the observations retrieved. Moreover, alternative algorithms designed to test specific behavioral strategies relative to the benchmark scenario were also designed and evaluated. The main contributions of these algorithms can be summarized as follows:

• Di Caprio et al. [5] illustrated the effect that frictions – triggered by observations that do not satisfy the subjective requirements of DMs throughout the different nodes of the decision tree – have on their retrieval behavior.

• Di Caprio et al. [6,7, Source Code 2021] introduced a set of complementary algorithms allowing to compare the behavior of DMs as their impatience grew. Two variants of the main benchmark algorithm were presented, with impatience determining the stopping behavior assumed on the DMs, which ranged

- from strict scenarios where DMs concluded the retrieval process as soon as an alternative underperformed relative to their subjective preferences, that is, as soon as an observation did not satisfy their cutoff expectations
- to more complex environments where DMs proceeded through underperforming alternatives until a satisfying one was found and then stopped as soon as a new underperforming alternative was observed.

• Di Caprio and Santos Arteaga [8],Di Caprio and Santos Arteaga [9, Source Code 2021] exploited the capacity of the benchmark algorithm to generate behavioral patterns that can be used to train and validate the categorization capacity of machine learning (ML) techniques. That is, the algorithm delivers both the numerical observations retrieved by DMs as well as the behavioral consequences in terms of evaluations performed. Different algorithms were designed and categorized according to the order in which the retrieval decisions were reported. The output vectors obtained from the algorithms were used as training inputs to highlight the identification and categorization capacities of ML techniques relative to standard statistical ones.

Table 1 presents the output obtained from the benchmark algorithm when the satisfying cutoff values equal $c_i = 0.5$, for all $i = 1, 2, \ldots, 10$. Each column represents a query that accounts for the ten alternatives described in the first page of results provided by the engine. The output per query corresponds to the stochastic realizations describing the characteristics of each alternative and the subsequent pages clicked by the DM.

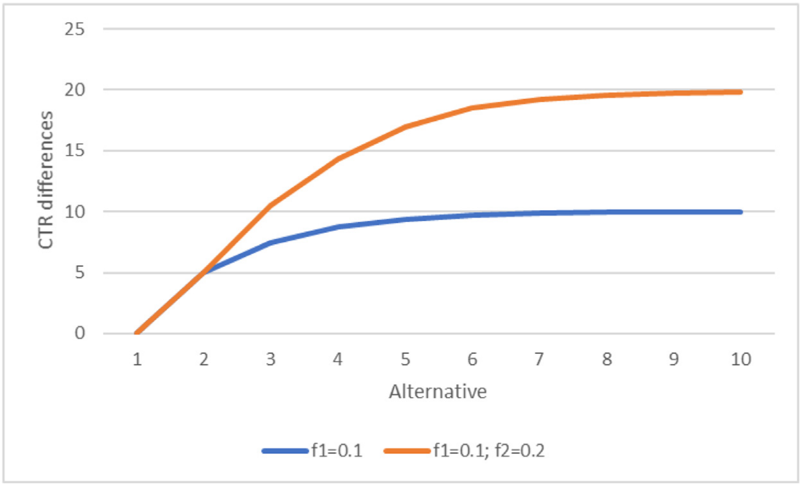

Fig. 2. CTR differences relative to the frictionless scenario for $c_i = 0.5$, $i = 1, 2, \ldots, 10$.

Table 1
Output delivered by the benchmark algorithm for $c_i = 0.5$, $i = 1, 2, \ldots, 10$.

	Search Queries				
Stochastic evaluations	0.758	0.823	0.490	0.498	0.959
	0.743	0.695	0.446	0.960	0.547
	0.392	0.317	0.646	0.340	0.139
	0.655	0.950	0.709	0.585	0.149
	0.171	0.034	0.755	0.224	0.258
	0.706	0.439	0.276	0.751	0.841
	0.032	0.382	0.680	0.255	0.254
	0.277	0.766	0.655	0.506	0.814
	0.046	0.795	0.163	0.699	0.244
	0.097	0.187	0.119	0.891	0.929
Pages clicked	1	1	3	2	1
	2	2	4	4	2
	4	4	5	6	6
	6	8	7	8	8
	0	9	8	9	10
	0	0	0	10	0
	0	0	0	0	0
	0	0	0	0	0
	0	0	0	0	0
	0	0	0	0	0

As stated above, the benchmark decision-tree algorithm allows to incorporate any potential modification to the subjective behavior of DMs defined within its main retrieval structure. Each node describes the implementation of a decision rule conditioned by the random realization of the corresponding characteristic. In this regard, the rules applied at each node can be modified to incorporate any of the strategic traits determining the different types of decision processes described in the literature.

For instance, we may consider the introduction of frictions within the search process of DMs [5,7]. That is, whenever a realization underperforms relative to the corresponding satisfying value, DMs may feel that the ranking provided by the engine does not accommodate their preferences as correctly as expected, decreasing their willingness to click on any subsequent alternative, i.e., increasing the value of the subsequent cutoffs. In this case, the strength and duration of the resulting effects can be defined according to the subjective features of DMs.

As an illustrative example, we consider two scenarios determined by the cumulative frictions introduced to account for the underperformance of the alternatives

- a unique friction, $f_1 = 0.1$, is added to all remaining cutoffs after DMs face the first $x_i < c_i$ realization, $i = 1, 2, \ldots, 10$;

Table 2
CTR differences across friction scenarios.

	Frictions		
CTR	None	$f_1 = 0.1$	$[f_1 = 0.1;\ f_2 = 0.2]$
1	50.02	50.09	50.00
2	50.02	45.02	45.05
3	50.01	42.53	39.56
4	49.96	41.22	35.62
5	50.02	40.65	33.06
6	50.05	40.38	31.58
7	50.07	40.17	30.85
8	50.00	40.07	30.44
9	50.00	39.99	30.29
10	49.92	39.91	30.12

- two different frictions are added: $f_1 = 0.1$, after DMs face the first $x_i < c_i$ realization, and $f_2 = 0.2$, to all remaining cutoffs after DMs face the second $x_j < c_j$ realization, $j = 2, \ldots, 10$.

Clearly, additional friction values could be introduced based on the number of underperforming realizations. Table 2 presents the CTRs obtained through the different scenarios, while Fig. 2 describes the differences in CTRs per alternative between the frictionless benchmark and each scenario. Each simulation is composed by one million queries, within which DMs proceed through the ten alternatives ranked by the engine.

The effect that an increase in frictions has on the CTRs is determined by both the number of frictions introduced and the position of the alternatives within the ranking. At the same time, the non-linear incremental effect on the CTRs highlights its concentration within the lower half of the ranked alternatives.

All in all, different extensions and modifications of the benchmark algorithm would allow to consider multiple evaluation scenarios based on a variety of behavioral strategies implemented at each node of the decision tree and through any potential path that may be followed by a DM. Note that, besides describing the behavior of DMs, the algorithm provides a benchmark to analyze how they may deviate from a rational setting as modifications are introduced to the retrieval framework.

Among the main shortcomings, the size of the algorithm constitutes a binding limit – as is generally the case with binary decision trees –, with the coding requirements increasing considerably as additional alternatives are incorporated to the retrieval process. In this regard, algorithms based on simpler retrieval strategies can be designed to deliver similar results to those of the complex benchmark framework, highlighting the importance of bounded rationality on the assimilation capacities and subsequent behavior of DMs.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Chitika, The Value of Google Result Positioning, Chitika, 2013, Westborough (2013) Available at perma.cc/7AGC-HTDH. Chitika Insights June 7, 2013.

[2] B. Dean, We analyzed 5 million google search results, 2019, Here's what we learned about organic click through rate (2019) Available at https://backlinko.com/google-ctr-stats.

[3] Advanced Web Ranking, Google organic CTR history, 2021, https://www.advancedwebranking.com/ctrstudy/. (Accessed 26 December 2021).

[4] J. Qin, W. Zhang, X. Wu, J. Jin, Y. Fang, Y. Yu, User behavior retrieval for click-through rate prediction, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 2347–2356, http://dx.doi.org/10.1145/3397271.3401440.

[5] D. Di Caprio, F.J. Santos Arteaga, M. Tavana, A new algorithm for modeling online search behavior and studying ranking reliability variations, Appl. Intell. (2021b) http://dx.doi.org/10.1007/s10489-021-02856-8.

[6] D. Di Caprio, F.J. Santos Arteaga, M. Tavana, An information retrieval benchmarking model of satisficing and impatient users' behavior in online search environments, Expert Syst. Appl. 191 (2022) 116352, http://dx.doi.org/10.1016/j.eswa.2021.116352.

[7] D. Di Caprio, F.J. Santos Arteaga, M. Tavana, ESWA an information retrieval benchmarking model of satisficing and impatient users' behavior in online search environments [Source Code], 2021a, http://dx.doi.org/10.24433/CO.9579932.v1.

[8] D. Di Caprio, F.J. Santos Arteaga, Enhancing the pattern recognition capacity of machine learning techniques: The importance of feature positioning, Mach. Learn. Appl. 7 (2022) 100196, http://dx.doi.org/10.1016/j.mlwa.2021.100196.

[9] D. Di Caprio, F.J. Santos Arteaga, MLWA enhancing the pattern recognition capacity of machine learning techniques: The importance of feature positioning [Source Code], 2021, http://dx.doi.org/10.24433/CO.0135598.v1.