Decision Support Systems 117 (2019) 14–27

Contents lists available at ScienceDirect

Decision Support Systems

journal homepage: www.elsevier.com/locate/dss

A utility-based news recommendation system

Morteza Zihayat[a,*], Anteneh Ayanso[b], Xing Zhao[c], Heidar Davoudi[c], Aijun An[c]

[a] Ted Rogers School of Information Technology Management, Ryerson University, Canada
[b] Goodman School of Business, Brock University, Canada
[c] School of Computer Science and Engineering, York University, Canada

ARTICLE INFO

Keywords:
News recommendation systems
News utility model
Big clickstream data
MapReduce
Rule mining

ABSTRACT

News platforms exhibit both the challenges as well as opportunities for enhancing the functionalities of recommendation systems in today's big data environment. Novel use of big data storage and programming models can improve news recommendation systems through efficient handling and analysis of clickstream data and a better understanding of users' interests. Most existing approaches to news recommendation consider users' clicks as the implicit feedback to understand user behaviors. However, "clicks" may not be an effective indicator of real user interests. We address this problem by developing a novel news recommendation system based on a *news utility model*. Given the new utility model, we propose a two stage news recommendation framework. The framework first generates article-level recommendation rules based on the utility model, then integrates the notion of utility and probabilistic topic models and generates topic-level recommendation rules. We argue that the proposed utility-based news recommendation system also addresses the news cold start problem which is one of the most challenging obstacles for news agencies. We evaluate the framework on a massive real dataset (*two billion records*) obtained from a major newspaper (i.e., The Globe and Mail) in Canada and show that it outperforms the existing methods.

1. Introduction

Recent advancements in Internet-based technologies and the production of digital contents have shifted news consumption models from reading physical newspapers to visiting online news websites. Understanding and modeling users' interests is a critical task for the value proposition and prosperity of any online service provider. Moreover, in the digital age, value creation has become value cocreation between companies and customers [1]. Thus, many companies focus on their big historical data to model users' behaviors and boost profits. Similarly, online newspaper agencies aim to provide personalized contents to improve visit experiences. This does not only increase the frequency of visits, which boosts the revenue through the advertisements, but also improves user engagement, leading to more subscriptions [2]. In fact, from an online news provider's business perspective, increasing the revenue through advertisements and subscriptions is a major objective. Therefore, for most online newspapers, developing effective recommendation systems to help users find interesting articles and keep them engaged is of paramount importance.

Building an effective and efficient recommendation system for the news domain is much more challenging than other domains:

1. ***Business Objectives Trade-off.*** As the business model evolves, many online news publishers are struggling to balance the availability of high value advertising inventory with the need for content that maximizes opportunities for subscriber acquisition and retention. Thus, in many cases, a recommendation system is needed in which recommended articles satisfy more than one business objective, even if those objectives are in contention with one another.
2. ***Beyond Click-through Rates.*** Conventional news recommendation systems only consider a click as an implicit feedback. This is quite problematic because a user might click on an article but may not be interested in reading it (e.g., the title may be appealing to her while the content may not meet her expectations). Therefore, the number of clicks may not be helpful to address an intended domain specific objective. For example, it is very likely that user engagement increases the number of subscriptions, therefore a news agency might be interested in a type of recommendation which increases the engagement (e.g., dwell time of visited articles) thus the subscription rate. However, click analysis does not necessarily maximize this objective.
3. ***Big Data of Unknown Visitors.*** Newspapers are rich repositories of visitors' historical data. In 2016, the number of average monthly

[*] Corresponding author.
E-mail address: mzihayat@ryerson.ca (M. Zihayat).

https://doi.org/10.1016/j.dss.2018.12.001
Received 20 July 2018; Received in revised form 5 December 2018; Accepted 5 December 2018
Available online 15 December 2018

unique visitors for top 50 U.S. newspapers was more than 11 billion[1]. The user interaction data of *The Globe and Mail*[2], Canada's foremost news media company, is about 30 million sessions in one month. Building a recommendation system that can handle big data requires applying scalable techniques and platforms [3]. Moreover, despite the availability of such huge volume of historical data, most interactions belong to un-subscribed users (i.e., unknown visitors). Therefore, user profiling and collaborative filtering techniques such as *Matrix factorization* [4] are not effective recommendation approaches in the news domain. This is due to the fact that such approaches need to identify users and collect their reading histories to discover their interests and the similarity among them.

4. ***News/Users Cold Start Problems.*** Articles are generated continuously and unboundedly at a high speed. This makes recommending new articles much harder than recommending new items (e.g., new products, new travel packages [5]) in other e-commerce domains. Thus, the ability to handle the cold start problem is essential for newspapers. There are two types of cold start problems: *user cold start* (when a new or unknown user visits the portal) and *item cold start* (when a new article is published). Despite different solutions [6, 7] to this major issue, this problem is still challenging, particularly in the news domain.

To address the aforementioned challenges, this paper presents a *Utility-based News Recommendation SYStem*, called *URecSYS*, which works based on a news utility model. The idea of the news utility model is inspired by the concept of utility in intelligent agents. In intelligent agents and machine learning, *goal-based*agents and *utility-based* agents are two common classes of agents [8]. A goal-based agent can only differentiate between goal and non-goal states in the environment. However, it is important to measure how desirable a particular state is. To do so, a utility function is defined such that it measures how satisfied the agent (e.g., a visitor) will be if it moves to a particular state (e.g., article). This helps agents to move toward a goal state (e.g., increase user engagement) with the highest satisfaction. It is important to note that the concept of utility in our model is very similar to its common usage in economics. In economics, the term utility is used to describe the measurement of *usefulness* and *satisfaction* that a consumer obtains from any good [9]. The utility is not a characteristic of a particular good (e.g., article), but rather of each consumer's reactions (e.g., user's engagement) to that good. In this paper, we argue that the utility can be defined in the context of recommendation. We model the utility to represent a desirable domain specific objective and recommend article with respect to this specific objective. In other words, we design a utility model to recommend those articles that persuade the user to move toward a higher value of the objective. For example, if the objective is to increase user engagement, a utility model can be designed such that a recommender suggests articles that lead to a highly engaged visit.

The news utility model is designed based on two broad types of attributes: *article* and *user-article interaction* attributes. Importantly, the model can be engineered to address one or more objectives even if those objectives are in conflict with one another. *URecSYS* recommends news articles by discovering article-level rules based on the news utility model. Moreover, by leveraging topic modeling and a probabilistic framework, it generalizes rules from the article-level to the topic-level. This addresses the news cold-start problem properly as newly-published articles can be recommended to a user by matching them to the topic-level rules. Moreover, as the news utility model is built based on reading sessions of all users, the recommendation rules (at either the article-level or the topic-level) can be used for new users, thus also resolves the *user cold start problem.*

To the best of our knowledge, this study is the first step toward exploring the impact of both article and user-article interaction attributes in a unified framework. Our contributions are summarized as follows:

1. We define a novel model, called *News Utility Model*, to simultaneously consider both article attributes (e.g., the recency of the article) and user-article interaction attributes (e.g., DwellTime). We argue that it is more beneficial to recommend news articles based on the utility of articles rather than the browsing frequency of articles.
2. We propose a novel and scalable rule engine to discover article-level recommendation rules based on the news utility model. At its heart, the rule engine uses multiple MapReduce-like steps to discover article-level recommendation rules in parallel.
3. We propose a novel probabilistic approach on top of topic-based models to generalize article-level news recommendation rules. The output is a topic-level recommendation rule engine that links topics of articles based on the news utility model, thus the domain specific objective. Such rules recommend newly-published articles based on topics of interests to users.
4. We apply the proposed framework to a dataset of *two billion records* collected from a major Canadian news agency (*The Globe and Mail*[3]) and demonstrate the effectiveness of the recommended articles by comparing the proposed framework to other state-of-the-art recommendation systems in practice.

The rest of the paper is organized as follows. In Section 2, we provide a review of prior work relevant to news recommenders. In Section 3, we introduce terms and concepts used in this paper. In Section 4, we present the proposed framework. In Section 5, we outline the experimental settings and discuss the results. Finally, in Section 6 we provide conclusions and point out limitations and future research directions.

2. Literature review

The underlying techniques used in recommender systems can be categorized into three broad classes: *content-based* [10, 11], *collaborative filtering* [4, 12-15] and *hybrid* [7, 16, 17] approaches. Several studies have been conducted on content-based news recommendation systems [10]. For example, Liang et al. [18] propose a time-aware content recommendation system. In another work, Agrawal et al [19] take activity freshness into account and significantly outperform *click-through rate (CTR)*. Even though content-based recommendation systems are easy to implement, they only look at a user's profile as a bag of words which is not good enough to keep track of the exact reading interest of the user.

On the other hand, collaborative filtering approaches are built based on the users' past rating behaviors to predict news ratings, or modeling users' behaviors in a probabilistic way [12]. *Matrix factorization* [4] is a popular collaborative filtering approach that has been applied to news recommendation. This approach needs users' reading histories to find users' interests and the similarity between users. However, in the news domain, most of the readers are not subscribed users. Moreover, it is very hard to identify users to collect their browsing histories. Some approaches consider news article co-visitations, which does not need to identify users, but only consider articles read by users in a session. For example, one main component of Google news recommender [13] is built on a network which represents co-visited articles. Alternatively, since the nature of news reading behavior is sequential, the intuitive approach to model news reading behavior is to use k-order Markov model. In this model, the next article is predicted based on the last k visited articles [14]. Nonetheless, it is not clear how to select the order

[1] http://www.journalism.org/chart/newspaper-website-unique-visitors/.

[2] https://www.theglobeandmail.com.

[3] http://www.theglobeandmail.com/.

k. Approaches like context-tree [15] try to alleviate the problem by introducing a variable-order Markov model. However, they treat all articles equally important which may not be suitable in practice. Collaborative filtering systems are efficient in cases where historical consumption across users and the content universe is almost static. However, in the news domain, the contents (i.e., items) change over time drastically. Moreover, we usually have news and user cold start problems. These issues make collaborative filtering ineffective in news recommendation environments. In Ref. [16], the authors investigate how dwell time is computed from a large scale web log. Later, for collaborative filtering, they use dwell time as a form of implicit feedback from users and demonstrate how it can be incorporated into a state-of-the-art matrix factorization model. Moreover, the authors in Ref. [7] propose to apply the WWW as an external database to calculate the correlations between items. Then, such correlations are converted to estimate the similarity between the items. The authors show that the estimated similarities improve the performance of conventional item-based collaborative filtering systems. In Ref. [17], the authors address the cold start problem by proposing a unique method of building models derived from explicit ratings. The method first predicts actual ratings and subsequently identifies prediction errors for each user.

In practice, both content-based and collaborative filtering systems have their own advantages. To build a news recommendation system with more effective recommendations, extensive studies have been conducted on combining these methods [6, 20-23]. In Ref. [23], the authors argue that a truly successful recommendation system not only considers what the customer needs but also maximizes the customers' after-sale satisfaction. To do so, they differentiate between the customer purchase and the customer endorsement and proposed a rating classification model based on the customer's profile and feedback. In Ref. [6] the authors focus on sequential information provided by a user's click to predict a user's next page visit. They show that the sequential information can improve the prediction performance. Moreover, they argue that the next page visit of a user can give useful information about their likes and tastes. They apply similarity upper approximation and singular valued decomposition for developing the recommendation system. In Ref. [20], the authors design a recommendation system that generates personalized recommendations based on preference similarity, recommendation trust, and social relations. The advantage of the proposed system, compared to traditional collaborative filtering systems, is its comprehensive consideration of recommendation sources.

Recently, some approaches consider the concept of utility in their proposed recommendation systems. For example, the authors in Ref. [24] propose a utility-based link recommendation system. They argue that existing approaches overlook the benefit a recommended link could bring to an operator. Therefore, they formulate the problem of utility-based link recommendation problem. The proposed method recommends links based on the value, cost, and linkage likelihood of a link, while existing methods focus solely on linkage likelihood. However, the concept of utility in this paper is limited to the cost and profit analysis. In Ref. [25], the authors leverage recommendation systems to predict consumers' willingness to pay and estimate non-linear utility functions to provide a better approximation of consumers' preference structures. They show that exponential utility functions are better suited for predicting optimal recommendation ranks for products than linear functions. The utility function proposed by Scholz et al. [25] is more of a cost function for the proposed optimization problem than a function that represents users' interests. Moreover, none of these approaches are applicable to the news domain directly.

Our work is essentially a hybrid recommendation approach. What differentiates our proposed system from prior methods is that it takes a utility model into account and can handle huge volume of data as the underlying method is designed based on the MapReduce framework. Moreover, our utility model is broadly defined to be flexible and fine-tuned as desired depending on the specific business objective. That is, it can consider different factors all together for the recommendation.

Table 1
Summary of notations.

Notation	Description
nw	News article
S_i	Session i
D	Clickstream dataset
$\Psi(nw)$	Article-driven measure of news article nw calculated based on attributes of nw
$\Upsilon(nw,S_r)$	User-driven measure of article nw in session S_r
$\varphi(nw, S_r)$	Utility of news article nw in session S_r
$\sigma(nw,P)$	Local utility of news article nw in news reading pattern P
$\varphi(P, D)$	Utility of news reading pattern P in clickstream dataset D
$R: X \rightarrow Y$	An article-level rule R (news reading pattern X implies news reading pattern Y)
$uconf(R)$	The utility confidence of article-level rule R
$min\_uconf(R)$	The minimum utility confidence of article-level rule R
$R: t_i \rightarrow t_j$	The topic-level rule R (topic t_i implies topic t_j)
$tconf(R)$	The utility confidence of topic-level rule R
$min\_tconf(R)$	The minimum topic confidence of topic-level rule R

3. Notations and definitions

For convenience, Table 1 summarizes the concepts and notations we define in this paper.

Let $\mathcal{N} = \{nw_1, nw_2, ..., nw_n\}\mathcal{N} = \{nw_1, nw_2, ..., nw_n\}$ be a set of distinct articles. A *clickstream dataset* consists of several user sessions. A *user session S* (or session in short) is defined as an ordered list of viewed articles $\langle nw_1, nw_2, \ldots, nw_z\rangle$ within a visit. Each article is represented by different attributes such as *popularity*, *topics*, *published date*. These attributes are called *article attributes*. Once an article is visited by a user, different attributes (e.g., *dwell time*, *timestamp*) are initialized based on interactions of the user with the article. These attributes are called *user-article interaction attributes* and present the quality of a visit. The article and user-article interaction attributes are selected with respect to an objective. There are two common approaches to select objective-oriented attributes: 1) Attributes/metrics presented by prior literature related to the objective, 2) Feature selection approaches [26] that can be used to discover data-driven attributes with respect to the objective. Later, we present an example of the first approach when the objective is to increase user engagement.

We consider these categories of attributes to design the *news utility model*. Our goal is to propose a general mapping from the attributes in a clickstream dataset to a news utility model with respect to a domain specific objective. In the next section, we present a framework to utilize this model for article recommendation.

Definition 1 (*Article-driven measure***).** Given news article *nw*, Article-driven measure is denoted and defined as:

$$\Psi(nw) = F_{\Psi}(f_1, f_2, ..., f_k) \tag{1}$$

where $\{f_1, f_2, \ldots, f_k\}$ are article attributes and F_{Ψ} is an aggregation function.

Definition 2 (*User-driven measure***).** Given a session S_r and a set of user-article interaction attributes $f_1', f_2', \ldots, f_k'$, the user-driven measure is defined as:

$$\Upsilon(nw, S_r) = F_{\Upsilon}(f_1', f_2', ..., f_k') \tag{2}$$

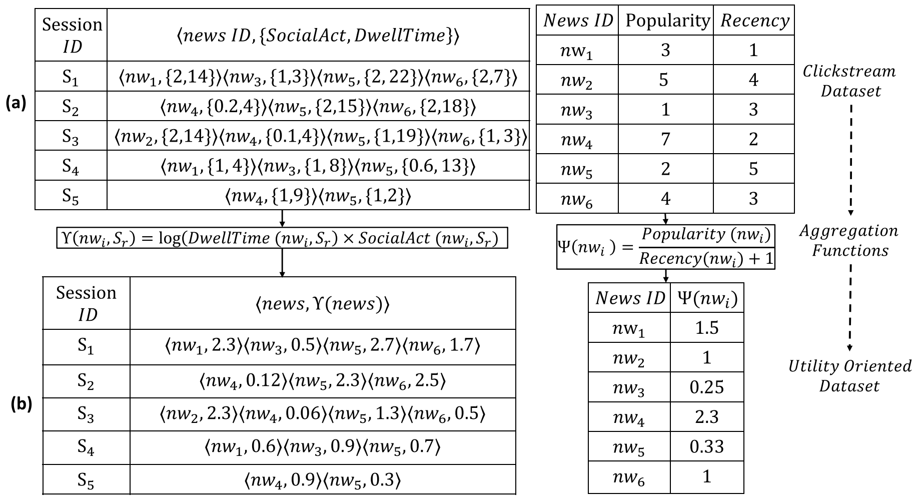
where F_{Υ} is an aggregation function.

Definition 3 (*News Utility Model***).** Given an article *nw* and a session S_r, the *news utility model* is defined and denoted as follows:

$$\varphi(nw, S_r) = F_{\varphi}(\Psi(nw), \Upsilon(nw, S_r)) \tag{3}$$

where F_{φ} is the function for aggregating the article-driven and the user-driven measures.

In the above definitions, we do not limit the aggregation functions



Session ID	$\langle news\ ID, \{SocialAct, DwellTime\}\rangle$
S_1	$\langle nw_1, \{2,14\}\rangle\langle nw_3, \{1,3\}\rangle\langle nw_5, \{2,22\}\rangle\langle nw_6, \{2,7\}\rangle$
S_2	$\langle nw_4, \{0.2,4\}\rangle\langle nw_5, \{2,15\}\rangle\langle nw_6, \{2,18\}\rangle$
S_3	$\langle nw_2, \{2,14\}\rangle\langle nw_4, \{0.1,4\}\rangle\langle nw_5, \{1,19\}\rangle\langle nw_6, \{1,3\}\rangle$
S_4	$\langle nw_1, \{1,4\}\rangle\langle nw_3, \{1,8\}\rangle\langle nw_5, \{0.6,13\}\rangle$
S_5	$\langle nw_4, \{1,9\}\rangle\langle nw_5, \{1,2\}\rangle$

News ID	Popularity	Recency
nw_1	3	1
nw_2	5	4
nw_3	1	3
nw_4	7	2
nw_5	2	5
nw_6	4	3

$\Upsilon(nw_i, S_r) = \log(DwellTime\ (nw_i, S_r) \times SocialAct\ (nw_i, S_r)$

$\Psi(nw_i) = \dfrac{Popularity\ (nw_i)}{Recency(nw_i) + 1}$

Session ID	$\langle news, \Upsilon(news)\rangle$
S_1	$\langle nw_1, 2.3\rangle\langle nw_3, 0.5\rangle\langle nw_5, 2.7\rangle\langle nw_6, 1.7\rangle$
S_2	$\langle nw_4, 0.12\rangle\langle nw_5, 2.3\rangle\langle nw_6, 2.5\rangle$
S_3	$\langle nw_2, 2.3\rangle\langle nw_4, 0.06\rangle\langle nw_5, 1.3\rangle\langle nw_6, 0.5\rangle$
S_4	$\langle nw_1, 0.6\rangle\langle nw_3, 0.9\rangle\langle nw_5, 0.7\rangle$
S_5	$\langle nw_4, 0.9\rangle\langle nw_5, 0.3\rangle$

News ID	$\Psi(nw_i)$
nw_1	1.5
nw_2	1
nw_3	0.25
nw_4	2.3
nw_5	0.33
nw_6	1

Fig. 1. (a) Clickstream dataset with article and user-article interaction attributes, (b) Utility-oriented dataset with article-driven and user-driven measures.

F_Ψ, F_Υ and F_φ to any specific form. In general, the aggregation functions can be formulated with respect to the given business objective(s). Without loss of generality, we formulate some aggregation functions as examples and illustrate them in our experiments. However, we reiterate that any other aggregation function can be formulated and utilized as desired depending on the specific business objective.

Fig. 1 (a) shows a clickstream dataset consisting of five sessions $\{S_1, S_2, S_3, S_4, S_5\}$.

Assume that our domain specific objective is to recommend articles that increase *user engagement*. It has been shown that the *recency* of an article, its *popularity* and also *social media activity*(i.e., social media activity score of the user regarding an article) are indicators/triggers of the user engagement [27, 28]. Moreover, according to Yi et al. [16], *DwellTime* (i.e., the time that the user spends on the clicked article) is one of the effective attributes to measure engagement. Therefore, two attributes are captured per article: *popularity* (how popular an article is) and *recency* (how recent an article is). For example nw_1 has the popularity score of 3 based on the clicks and the recency of 1 (lower value means more recent the article is). The aggregation function is defined as follows:$\Upsilon(nw, S_r) = \log(DwellTime(usr, nw)) \times SocialAct(usr, nw)$. We use $\log(DwellTime(usr, nw))$ since after a certain amount of time, DwellTime does not necessarily represent the user engagement. In addition, the *article-driven* measure of *nw* is calculated as follows: $\Psi(nw) = \frac{popularity(nw)}{recency(nw) + 1}$. In this example, two user-article interaction attributes *SocialAct*, and *DwellTime* are also captured per visited article. Higher value of *SocialAct* means more social activities (e.g., share the article, like the article, etc.), thus higher engagement. The item $\langle nw_1, \{2,14\}\rangle$ in S_1 means that a user visited the news article nw_1, spent 14 minutes on nw_1 and had a relatively high social activity score on the article (e.g., clicked on the *like* button and shared it on social media). The tables in Fig. 1 (b) show the results of applying the aggregation functions. The result dataset, Fig. 1 (b), is called *utility oriented dataset.* Later, we define F_φ as a multiply function. That is, $\varphi(nw, S_r) = \Psi(nw) \times \Upsilon(nw, S_r)$.

Note that, the above example presents only one potential option for the news utility model. In practice, any other utility model can be plugged in as desired. For further illustration, Fig. 2 shows examples of news utility models with respect to different objectives.

To take this model into account, we first define *news reading pattern* and then, we show how we calculate the utility of a pattern.

Definition 4 (*News Reading Pattern*). a news reading pattern P is a set of news articles $\{nw_1, nw_2, \ldots, nw_L\}$ where L is the length of the pattern.

Definition 5 (*Local Utility of a News Article*). Given a news reading pattern P, the local utility of a news article nw_i in P, is defined as the sum of the utility values of nw_i in all the sessions where P resides: $\sigma(nw_i, P) = \sum_{P \in S_j, S_j \in D} \varphi(nw_i, S_j)$, where D is the clickstream dataset.

Definition 6 (*High Utility News Reading Pattern*). Given clickstream dataset D, a news reading pattern P is a *high utility news reading pattern* iff $\varphi(P, D)$ is not less than a given user specified threshold, where $\varphi(P, D) = \sum_{S_r \in D} \sum_{nw \in P, P \subseteq S_r} \varphi(nw, S_r)$.

4. URecSYS: a utility-based news recommendation system

Given the proposed news utility model to present a domain specific objective, the most important challenge is how to incorporate it into the recommendation process. To address this challenge, we design a *Utility-based news Recommendation SYStem (URecSYS)*. *URecSYS* is a rule-based recommendation system which is designed and developed using *Apache Spark* and *MapReduce* framework. URecSYS first finds recommendation rules from the clickstream dataset and then applies the discovered rules to recommend articles with respect to the current session of a user. Fig. 3 shows our proposed framework. There are two main stages to discover recommendation rules: *1) article-level recommendation rule discovery*, and *2) topic-level recommendation rule discovery*. In Section 4.1, we describe *Stage 1* and in Section 4.2, we describe *Stage 2*. In Section 4.3, we discuss how URecSYS recommends articles based on the discovered rules.

4.1. Stage 1: article-level recommendation rule discovery

Most rule-based recommendation systems generate association rules from data. That is, they first discover all the news reading patterns whose frequency (e.g., occurrences of the pattern in the data) is no less than a minimum support threshold [29]. Then, given a frequent news reading pattern P, all the rules in the form of R: $A \rightarrow B$ are generated from P if the *confidence* of R is no less than a *confidence threshold.* The confidence is the percentage of sessions containing the pattern B among the set of

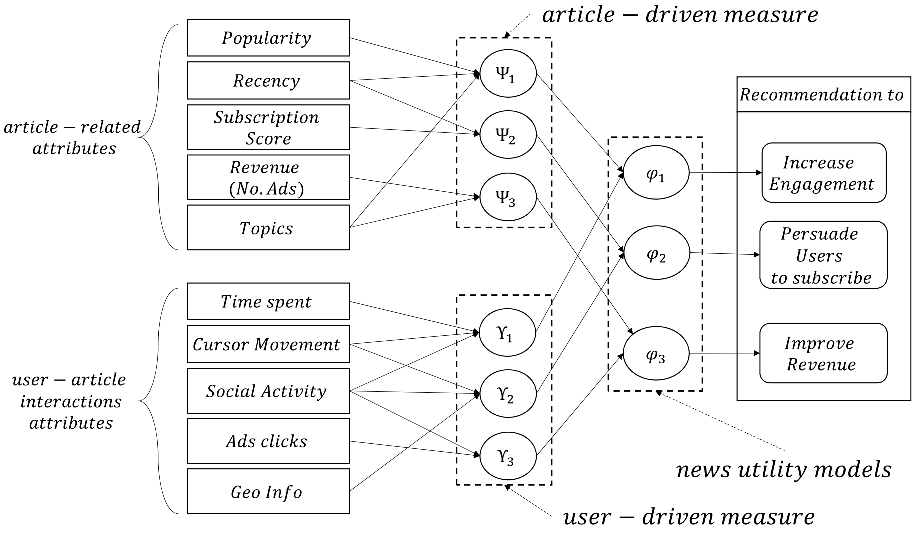


Fig. 2. Examples of the news utility model for different objectives in recommendation.

sessions containing the pattern A. However, this approach is not able to tell how satisfactory articles in B are by knowing that the articles in A were satisfactory with respect to a domain specific objective.

We propose a novel rule discovery framework to find all *article-level rulesR*: $A \rightarrow B$ based on the proposed news utility model. We argue that recommendations are *satisfactory* if they meet two conditions: 1) A reading session, including visited articles and the recommended ones, should form a *high utility news reading pattern*, 2) The recommended articles should increase the utility to some extent by providing more *satisfactory* information. To meet these conditions, we define an *article-level news recommendation rule* as follows. Given two news reading patterns X and Y, a rule R: $X \rightarrow Y$ (X implies Y) is an **article-level news recommendation rule** iff:

1. $X \neq \emptyset$, $Y \neq \emptyset$ and $X \cap Y = \emptyset$.
2. Y and $X \cup Y$ are high utility news reading patterns.
3. Its **utility confidence (*uconf(R)*)** is no less than a user-defined minimum threshold (*min_uconf*), where

$$uconf\,(R) = \frac{\sigma(Y, X \cup Y)}{\sigma(X \cup Y, X \cup Y)} \tag{4}$$

Rationale: The utility-confidence of the rule R: $X \rightarrow Y$ reflects the utility contribution of Y to the utility of news reading pattern $X \cup Y$. That is, it recommends Y, associated to X, where $X \cup Y$ forms a high utility news reading pattern. A higher confidence indicates a higher total utility value of the recommended news articles.

Our proposed framework to discover such rules consists of two parts: (1) high utility new reading pattern discovery (*to meet condition 1*), and (2) identifying article-level news recommendation rules (*to meet condition 2*). Mining high utility news reading patterns from a big clickstream dataset is not an easy task. To mine such patterns, the method should deal with the critical combinatorial explosion of search space caused by sequencing among articles in sessions. Moreover, pruning the search space is more difficult than that in traditional frequent pattern mining as the news utility model is not necessarily monotonic, and thus *downward closure property*does not hold for mining high utility news reading patterns. That is, the utility of a pattern may be higher than, equal to, or lower than its super-patterns and sub-patterns [30]. In addition, with a considerably large number of sessions whose information may not be entirely loaded into memory, developing a distributed method is the only solution to identify article-level recommendation rules.

Fig. 4 shows an overview of the proposed framework to discover article-level recommendation rules using several MapReduce jobs. In this work, we use an extended version of a recently proposed approach called *BigHUSP* [30] to discover article-level news recommendation rules. In the first MapReduce, we apply Definitions 1 and 2 to build the utility-oriented dataset (see Fig. 1 (b)). Given the utility-oriented dataset, using one Mapper and one MapReduce, the framework discovers high utility news reading patterns. Lastly, article-level recommendation rules are discovered using the last MapReduce.

Algorithm 1 shows the proposed method to discover high utility news reading patterns. In lines 1–8, we first distribute the data among nodes in a cluster of computers, and then convert the input dataset

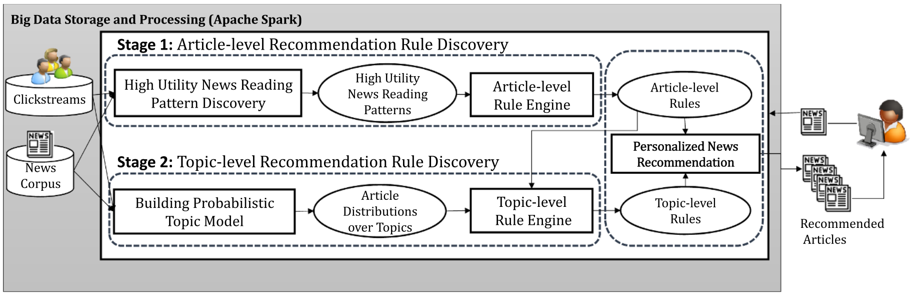

Big Data Storage and Processing (Apache Spark)
Stage 1: Article-level Recommendation Rule Discovery
Clickstreams
News Corpus
High Utility News Reading Pattern Discovery
High Utility News Reading Patterns
Article-level Rule Engine
Article-level Rules
Personalized News Recommendation
Stage 2: Topic-level Recommendation Rule Discovery
Building Probabilistic Topic Model
Article Distributions over Topics
Topic-level Rule Engine
Topic-level Rules
Recommended Articles

Fig. 3. URecSYS, the proposed framework.

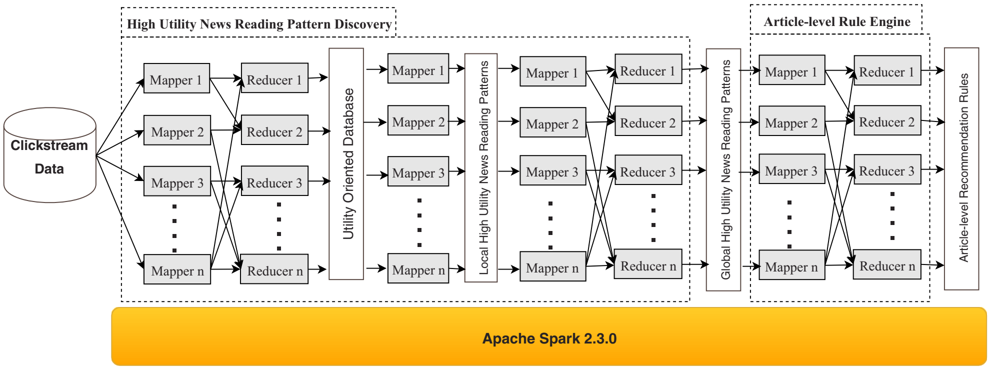

High Utility News Reading Pattern Discovery
Article-level Rule Engine
Clickstream Data
Mapper 1
Mapper 2
Mapper 3
Mapper n
Reducer 1
Reducer 2
Reducer 3
Reducer n
Utility Oriented Database
Local High Utility News Reading Patterns
Global High Utility News Reading Patterns
Article-level Recommendation Rules
Apache Spark 2.3.0

Fig. 4. The conceptual big data framework for article-level rule discovery.

based on the given news utility model. In line 9, the algorithm runs a procedure [30] to discover high utility news reading patterns in each mapper. A high utility news reading pattern in a mapper is called *local high utility news reading pattern* and the set of local high utility news reading patterns in the mapper *i* is denoted as χ_i. Then, using a MapReduce job, we discover high utility news reading patterns by aggregating patterns in all χ_is. For more details regarding the aggregation process please read Ref. [30].

Algorithm 1. High utility news reading pattern discovery.

```
Input: A utility-oriented dataset $D$, News utility model $F_\varphi$, Minimum utility threshold $\delta$
Output: High Utility News Reading Patterns $\Re$
 1: Distribute $D$ among $m$ nodes in the cluster $\{D_1, D_2, \ldots, D_m\}$
 2: for $\forall S_i \in$ each $D_j$ do
 3:   for $\forall nw \in S_i$ do
 4:     $\varphi(nw, S_i) \leftarrow F_\varphi(\Psi(nw), \Upsilon(nw, S_i))$
 5:     Add $\langle nw, \varphi(nw, S_i)\rangle$ to session $S_i'$
 6:   end for
 7:   Add $S_i'$ to $D_j'$
 8: end for
 9: $\chi_i \leftarrow$ Find high utility news reading patterns in node $i$ using $\delta$ and $D_i'$
10: $X \leftarrow$ Aggregate $\chi_i$s to find high utility news reading patterns over $D'$
11: Return $X$
```

Given a set of high utility news reading patterns, a naive solution to discover article-level rules is to generate all the rules using an exhaustive search. However, this approach produces all the possible subsets of a high utility news reading pattern *P* with *k* articles, that is 2^k subsets which grows exponentially. As we are dealing with a huge volume of data, we are more interested in finding possible antecedents for a consequent efficiently (e.g., in a linear time) without missing rules with highest confidence among all rules mined from the same pattern. Our goal is to design a greedy algorithm to discover article-level news recommendation rules in a linear time.

Theorem 1. *Given a news reading pattern P = {nw_1,nw_2,…,nw_n} where σ(nw_1,P) < σ(nw_2,P) < … < σ(nw_n,P), X = {nw_1, nw_2, …,nw_k} ⊆ P where k ≤ n and any article nw_i from P where σ(nw_i,P), the following statement holds:*

$$uconf\,(\langle\{X - nw_k\} \rightarrow nw_i\rangle) > uconf\,(\langle\{X - nw_{k-1}\} \rightarrow nw_i\rangle) \tag{5}$$

where term X − nw_k excludes the article nw_k from its residing set X.

Given Theorem 1, Algorithm 2 shows the overview of our proposed greedy algorithm to discover article-level recommendation rules. The algorithm takes a high utility news reading pattern and *min_uconf* as inputs and returns the article-level news recommendation rules. In line 2, we first sort articles in pattern *P* in an ascending order based on the local utility of articles in the session. Given each article *nw*, Algorithm 2 generates all article-level recommendation rules whose antecedent is *nw*.

Algorithm 2. Article-level rule engine.Algorithm 2 calls Algorithm 3 to

```
Input: A set of high utility news reading patterns $PSet$, $min\_uconf$
Output: A set of article-level recommendation rules $\Re$
 1: $\Re \leftarrow \emptyset$
 2: $PSet' \leftarrow$ sort articles $nw \in P$ in ascending order based on $\sigma(nw, P)$ where $P \in PSet$
 3: for each $P = \{nw_1, nw_2, ...., nw_n\} \in PSet'$ do
 4:   for $i \leftarrow n : 1$ do
 5:     if $nw_i \in PSet'$ then
 6:       $X \leftarrow \{(nw_1, nw_2, ..., nw_{i-1})\}$
 7:       $Y \leftarrow nw_i$
 8:       $\Re \leftarrow$ Algorithm 3 with $X$, $Y$, $\Re$, $min\_uconf$, $PSet$
 9:     else
10:       Return
11:     end if
12:   end for
13: end for
14: Return $\Re$
```

check the validity of each generated rule using the conditions presented in Section 4.1. Theorem 1 is applied to Algorithm 3 in the step of the article-level rule evaluation and it ensures the algorithm has a linear running time complexity. Algorithm 3 is a recursive algorithm where lines 1 and 2 set the base case to guarantee that it terminates. In line 10, the algorithm recursively calls itself to look for a possible antecedent but with size decremented by one. The time complexity of this algorithm is $O(k)$ compared to the exhaustive search which is $O(2^k)$ (check all the subsets), where k is the length of the input high utility news reading pattern.

Algorithm 3. Backwards analysis.

```
Input: Antecedent ($antec$), Consequent ($cons$), current $\Re$, $min\_uconf$, $Pset$
Output: updated $\Re$
 1: if ($X = \emptyset$ or $antec \cup cons \notin PSet$) then
 2:     return
 3: end if
 4: $uconf \leftarrow \frac{\sigma(Y, X\cup Y)}{\sigma(X\cup Y, X\cup Y)}$
 5: if $uconf \geq min\_uconf$ then
 6:     $R \leftarrow \langle X \rightarrow Y, uconf, \sigma(Y, X \cup Y)\rangle$
 7:     Add $R$ to $\Re$
 8: end if
 9: for $\forall nw \in X$, $X' \leftarrow X - nw$ do
10:     Call Algorithm 3 ($X'$, Y,$\Re$, $min\_uconf$, $Pset$)
11: end for
```

Example 1. Given a high utility news reading pattern $\langle nw_6, nw_2, nw_4, nw_5\rangle$, all potential rules of recommending the article nw_5 are: r_1: $\{nw_6, nw_2, nw_4\} \rightarrow nw_5$; r_2: $\{nw_6, nw_2\} \rightarrow nw_5$; r_3: $\{nw_6\} \rightarrow nw_5$. Then, each rule is evaluated. Suppose the utility confidences are 0.45, 0.5 and 0.65 and $min\_uconf = 0.6$. Therefore, only rule r_3 is chosen as an article-level rule that recommends nw_5.

4.2. Stage 2: topic-based news recommendation rule discovery

Article-level news recommendation rules represent the utility-based connection among articles effectively. However, they may not recommend newly published articles. A straightforward solution is to apply a content-based approach (e.g., based on topics' similarity) to recommend newly-published articles with similar content to visited articles by a user. However, such approaches have two main drawbacks: (1) *Repellent recommendations.* Since recommended articles are only based on content similarity and they do not consider news reading patterns, recommended articles may not be interesting to the user, and (2) *Lack of diversity*. It is very common that a user reads articles with a wide variety of topics ranging from *politics* to *sports*. For example, a user may visit a news website to read some articles about *presidential election* and to get some updates on a *basketball league*. Recommending articles without considering the user-based relationships among topics might not satisfy the diversity of users' information needs. Therefore, we identify topic-level recommendation rules based on the news utility model such that the rules represent the relationships among topics.

A probabilistic topic model infers latent topics from a text corpus. For example, LDA represents a topic as a distribution over words. In particular, given a news article corpus $\mathcal{N}$ with a vocabulary $\mathcal{V}$, a topic t is a multinomial distribution over words $w_i \in \mathcal{V}$, where $p(w_i|t)$ is the probability of word w_i in topic t. Moreover, each document is a mixture of topics. As such, given a news article nw (i.e., document) and a set of topics $T = \{t_1, t_2, \cdots, t_i, \cdots, t_m\}$, $p(t_i|nw)$ is a probability of topic t_i in the article nw. In the second stage of URecSYS, we apply LDA to detect latent topics and represent each article as a distribution over topics. Then, we generate topic-level rules using our news utility model. Note that, our proposed approach works based on a probabilistic topic model, therefore any other topic modeling technique that infers a distribution for a document can be applied here as an alternative.

Given news articles represented by topics, the goal is to define relationships among topics based on article-level news recommendation rules. A naive approach is to map each article of a rule to the top-1 topic of the article (i.e., the topic with highest probability). However, this approach assumes that each article has only one topic; moreover, it generates a large number of rules which are not necessarily attractive.

Intuitively, a topic-level rule $t_j \rightarrow t_i$ is satisfactory if $p(t_i|t_j)$ is high. We estimate this probability using the news utility model. Given an article, assuming that topics are conditionally independent, we can estimate $p(t_i|t_j)$ as follows:

$$\begin{aligned} p(t_i|t_j) &= \frac{\sum_{nw\in\mathcal{N}} p(t_i, nw)p(t_j|t_i, nw)}{p(t_j)} \\ &= \frac{\sum_{nw\in\mathcal{N}} p(nw)p(t_i|nw)p(t_j|nw)}{p(t_j)} \\ &= \frac{\sum_{nw\in\mathcal{N}} p(nw)p(t_i|nw)p(t_j|nw)}{\sum_{nw\in\mathcal{N}} p(nw)p(t_j|nw)} \end{aligned} \tag{6}$$

We can estimate the $p(nw)$ using our news utility model as:

$$p(nw) = \frac{\sum_{S_r\in D} \varphi(nw, S_r)}{\sum_{nw_i\in\mathcal{N}} \sum_{S_r\in D} \varphi(nw_i, S_r)} \tag{7}$$

where D is the given clickstream dataset and S_r is a session in D. Note that in Eq. (6) we obtain the probability of $p(t_i|nw)$ from the probabilistic topic model. We define *topic utility confidence* of a topic-level rule R: $t_j \rightarrow t_i$ (i.e., $tconf(R)$) as $p(t_i|t_j)$. Formally, given a topic-level utility confidence and a minimum topic confidence threshold min_tconf, Algorithm 4 finds topic-level recommendation rules. It first draws two topics, then computes the conditional probabilities $p(t_j|t_i)$ and $p(t_i|t_j)$. If $p(t_j|t_i)$ or $p(t_i|t_j)$ is no less than a specified threshold, a new topic rule R: $\{t_i \rightarrow t_j\}$ or R: $\{t_j \rightarrow t_i\}$) is discovered respectively.

Algorithm 4. Topic-level rule engine.

```
Input: A clickstream dataset $D$, topic distributions on news articles corpus $T$, minimum topic confidence $min\_tconf$
Output: A set of topic-level news recommendation rules $\Re_T$
for each $t_i, t_j \in T$ do
    Calculate $p(t_i|t_j)$ and $p(t_j|t_i)$ using Equation 1
    if $p(t_i|t_j) \geq min\_tconf$ then
        Add $R : \langle t_j \rightarrow t_i, p(t_i|t_j)\rangle$ to $\Re_T$
    end if
    if $p(t_j|t_i) \geq min\_tconf$ then
        Add $R : \langle t_i \rightarrow t_j, p(t_j|t_i)\rangle$ to $\Re_T$
    end if
end for
return $\Re_T$
```

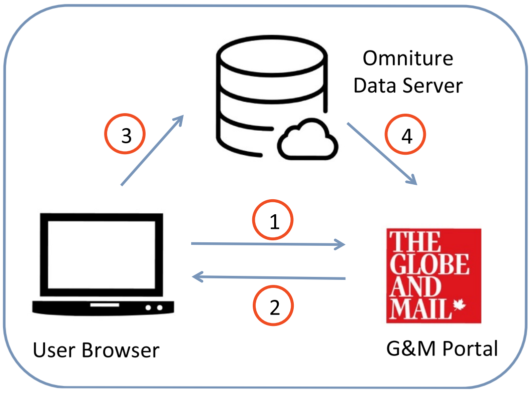


Fig. 5. Data collection platform.

Table 2
Categories of attributes in The Globe and Mail Dataset.

Attributes	Description
Traffic variables	• Count the instances of specific events on a web page (e.g., Number of clicks on advertisements). • Group the pages (logically) based on variable set in hits (e.g., User logged in/not logged in).
Conversion variables	• Conversion variables are persistent and hold the values for a longer period time (e.g., Number of viewed articles).
Events	• A point on the portal in which a successful event occurs (e.g., share an article in social media).

Table 3
Preprocessing steps on The Globe and Mail dataset.

Step	Description
Filtering out irrelevant hits	We only keep the hits related to articles and important events such as social activities.
Extracting events of interest	We extract events such as like, share, comments.
Computing Dwell time	Dwell time is calculated using two consecutive viewed article timestamps.
Roll-up from hits to a visit and a user	The aggregation is done using the user unique id, visit and page id.
Converting to expressive format	Each user activity record is stored as a json format to facilitate the analysis in later stages.

4.3. *Personalized news recommendation*

URecSYS calculates three scores to rank articles for recommendations:

- **Article-level News Recommendation Score:** Given an active user session S_r, the recommendation score of an article *nw* regarding the current reading pattern *P* in the session S_r is denoted and calculated as:$AL(nw) = \varphi(P, S_r) \times uconf(P \rightarrow nw)$, where $\varphi(P, S_r)$ is the utility value of the current reading pattern *P* in the user session S_r. As a subset of *P*, it may have higher recommendation score with respect to *nw*, we choose the highest recommendation score from all the subsets of *P* as the recommendation score of *nw*. For a reading pattern *P* with *k* articles, we use all the 2^k subsets of *P* to generate the candidate recommendations. The same size candidate(subset of P) with the highest score is considered for recommendation.
- **Topic-level News Recommendation Score:** Given the current user session $S_r = \{nw_i, nw_{i+1}, ..., nw_k\}$, we first discover top-k important topics from the visited articles. They are obtained by calculating the average probability of topics in the visited articles and selecting *k* topics with the highest probability. Then, all the topic-level rules whose antecedents are one of the top-k topics are discovered (i.e., *TRule*). Given a set of rules *TRule*, the recommendation score for a newly-published article (e.g., *nw*) is calculated as: $TL(nw) = \sum_{(t_i \rightarrow t_j) \in TRule} p(t_j|nw) \times tconf(t_i \rightarrow t_j)$. In our experiments, for simplicity and computing efficiency, the number of topics considered for each article is limited to 5.
- **Combined News Recommendation Score:** The two scores are combined in ranking the candidates for news recommendation: $Com(nw) = AL(nw) \times TL(nw)$. Combining the article-level score and the topic-level score offers the advantages of both approaches and shows an improved news recommendation performance over the one using either the article-level news recommendation rules or the topic-level news recommendation rules alone.

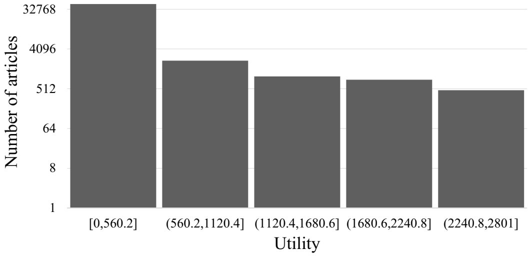


Fig. 6. The utility of articles in the dataset.

5. Experimental results and discussions

The experimental environment consists of one master node and six worker nodes. Each node is equipped with Intel Xeon 2.6 GHz(each 12 core) and 128 GB main memory. The framework is implemented on *Spark 2.3.0*.

5.1. *Data collection and data preparation*

The data used in this paper are collected from a major Canadian news agency (The Globe and Mail[4]). Every time a user reads an article, watches a video or generally takes an action, it is tracked on the website, and then is recorded as a *hit*. In data collection frameworks (e.g., Omniture by Adobe), a hit simply shows a row (record) in the data warehouse. It contains rich information about the visitors and their actions. Typically, a hit contains information like *date*, *time*, *user id*, *user environment variables* (e.g., browser type), *id of article*and special events of interest such as cursor movement, social activities, sign in and etc. Fig. 5 illustrates the collection procedure for *The Globe and Mail*dataset. The sequence of interactions is as follows: (1) the user browser requests an article from The Globe and Mail news portal (i.e., G&M portal), (2) The Globe and Mail Web server responds by sending the requested article including a small Javascript code, (3) Each user-article interaction will be sent by the Javascript code to a third party server (i.e., Omniture data server), (4) The data collection server sends the

[4] https://www.theglobeandmail.com/.

Table 4
Top-5 recommendation rules generated by *URecSY S_{AL}* and *AR-SYS*.

Approach	ID.	Rule (Title of the news in the rule)	Dwell time (mins.)	Conf.
URecSY S_{AL}	UR_1	*Police charge Garland with murder…→ Meet the sobriety coach hired …*	12	0.888
	UR_2	*In pictures: Scenes from crash …→ MH17:Disaster ratchets up…*	11.6	0.880
	UR_3	*La Prairie, Quebec mayor dies …→ MH17:Separatists hand over black…*	9.9	0.839
	UR_4	*MH17 Disaster ratchets up Russia-Ukraine …→ I refused to play the Conservatives…*	8.2	0.812
	UR_5	*Liberals take two Toronto ridings …→ Full by election results from Trinity-Spadina…*	9.4	0.812
AR-SYS	AR_1	*Harper alleges Supreme Court Chief Justice …→ Chief Justice hits back at Prime Minister…*	$<$ 1	0.953
	AR_2	*Body of missing Canadian Dave Walker …→ Rob Ford takes leave as recent drug video emerge…*	2	0.924
	AR_3	*Our Rob Ford problem: We forgive him …→ Assembly of First Nations national chief Shawn…*	$<$ 1	0.897
	AR_4	*Toronto's scandal mayor: World media …→ The end of the line for Toronto's rogue mayor…*	3	0.891
	AR_5	*Ontario election expected to be close race between Grits …→ Ontario's deficit includes the politicians…*	$<$ 1	0.882

collected data periodically to The Globe and Mail to be used in the data analytics pipeline.

The Globe and Mail clickstream dataset is composed of *2 billion* hits. It contains *246 attributes* which capture different aspects of interactions with users. Table 2 describes the various types of attributes in The globe and Mail clickstream dataset.

Although clickstream data collection can provide a lot of insights into users' behaviors, it is usually noisy, and contains a lot of unrelated information. As such, we need careful data preprocessing and cleaning steps before exploring, extracting and summarizing any useful pattern. For example, since dwell time is one of the factors which indicates user's engagement, we are supposed to focus on articles whose time spent is high. However, a long time does not necessarily imply that a user spends more time on reading clicked articles. It is possible that a user is using a multi-tab browser and (s)he is engaging in other activities. Therefore, we use both data aggregation and data cleaning to solve the problem in the data preprocessing phase. We first aggregate the data to user session levels. Each session represents a visit which is a set of viewed articles in the ascending order by time. Then, we filter out sessions whose time spent is longer than 30 min. Table 3 shows the general preprocessing steps which are done on The Globe and Mail clickstream dataset.

To calculate the dwell time per article, we use timestamps captured by the data collection platform. That is, the difference between any two consecutive page click timestamps is used to calculate the time that a user spent on an article. Similar to other studies, the last visited article in a session is ignored as we cannot estimate its dwell time. Moreover, attributes which are not required in the calculation of the proposed news utility model are filtered out. Lastly, we remove outlier visits whose time spent value deviates more than three times of the standard deviation from the average time spent for a specific article. This removes unreasonable values from the measures.

5.2. News utility model

We use a utility model to increase *user engagement*. However, other utility models can be plugged in as desired (e.g., see Fig. 2). Given a news article *nw* and a user *usr*, since the engagement of *nw* is dynamic and varying from time to time, the *article-driven* measure of *nw* is calculated using *recency* and *popularity* as follows:

$$\Psi(nw) = \frac{popularity(nw)}{accessDate(nw) - releasedDate(nw) + 1} \tag{8}$$

where *accessDate* is the date that *usr* clicked on *nw* and *releasedDate* is the published date of *nw*. Note that 1 in the denominator is added to avoid zero division.

In addition, the *user-driven* measure (e.g., $\Upsilon(nw,S_r)$) of *nw* to *usr* is calculated using two user-interaction attributes: 1) *DwellTime (in seconds)*: the time that *usr* spent on *nw*, and 2) *social media activity (SocialAct)*: this measure represents if *usr* shares the article in social media and it was engaging (e.g., number of likes). The aggregation function is defined as follows:

$$\Upsilon(nw, S_r) = \log(DwellTime(usr, nw)) \times SocialAct(usr, nw) \tag{9}$$

In this formula, we use $\log(DwellTime(usr, nw))$ since after a certain amount of time, dwell time does not necessarily represent user engagement. Lastly, the utility model is defined and calculated as follows:

$$\varphi(nw, usr) = \frac{popularity(nw) \times \log(DwellTime(usr, nw)) \times SocialAct(usr, nw)}{accessDate(nw) - releasedDate(nw) + 1} \tag{10}$$

Note that, the only criteria on the model is that it should return positive value as the utility values. The distribution of utility values is not tied to a specific type of distribution and it depends on the design of the news utility model. Fig. 6 shows the distribution of utility value of articles for the dataset used in our experiment. As the figure shows, all the values are positive and the majority are within 0 and 560. As the system is looking for maximizing the utility, the higher utility value implies that the article is more satisfactory with respect to the objective defined.

5.3. Baseline methods

We compare the performance of URecSYS against the following baseline systems:

- **News Popularity (Pop):** in this method, news articles are ranked by the DwellTime as a popularity measure. In fact, the results based on the popularity are used as a baseline in many studies [31].
- **News-to-news Collaborative Filtering (NewsKNN):** this method is similar to the popular collaborative filtering method which has been commercially used by Amazon [32]. In this approach, each article is represented by a vector of users on which they have spent time. Cosine measure is utilized to measure the similarity between articles. After several test, we found 50 as the best number of number of neighbors.
- **Non-negative Matrix factorization (NMF):** one of the common methods in news recommendation is based on Non-negative Matrix factorization [4] where the user-article matrix is factorized into two matrices with the property that all matrices have no negative value. Inspired by Yi et al. [16], we considered DwellTime to build the matrix than only *clicks*. Compared to traditional matrix factorization, the results of this method is interpretable and more proper for the ranking tasks in recommendation. We set the number of factors to 30 as higher values had no significant effects on the results.
- **Association Rule Mining (AR-SYS):** this method recommends articles using the proposed method in Ref. [33]. It considers association rules whose support and confidence are no less than some thresholds (e.g., 0.6). It recommends articles based on current visited articles by the user.
- **LDA:** this method is a content-based recommendation approach

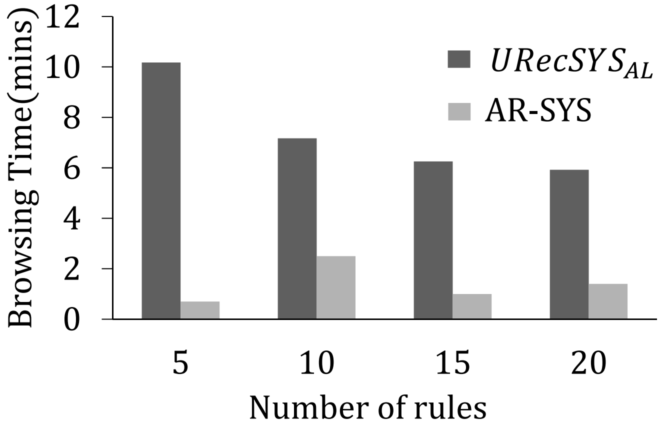


Fig. 7. Average browsing time using different sets of recommendation rules.

which uses topic modeling (e.g., LDA) for recommendation. In the method, users are considered as documents and news articles are treated as words. We use the default setting proposed by Guo et al. [34] to run the recommendation system.

- **URecSYS$_B$:** this method is our proposed recommendation system with *combined score*. The main difference is that we consider only *Dwelltime* as the news utility model. Our purpose is to show how effective the recommendations are even when the news utility model is as simple as a single measure (e.g., DwellTime).

Lastly, we evaluate three versions of URecSYS: a) Article-level URecSYS, denoted as **URecSYS$_{AL}$**, recommends articles according to the article-level news recommendation score. b) Topic-level URecSYS, denoted as **URecSYS$_{TL}$**, recommends articles based on the proposed topic-level news recommendation score, and c) Combined URecSYS, denoted **URecSYS$_C$**, recommends articles using the proposed combined news recommendation score.

5.4. Performance metrics

In our experiments, we perform 5-fold cross validation. That is, the dataset is split randomly into five folds. In each iteration, we use four folds as the training set and the remaining fold as the test set. Eventually, all folds are tested and the average results are reported as final performance results. To measure the effectiveness of recommended articles, we use *Mean Average Precision at k (i.e., MAP@K)*, where precision is defined as the number of relevant articles divided by the total number of articles in the dataset. We set K to 1, 5, and 10 respectively in our experiments. We use a validation set taken from the training set to find the optimal values for the parameters of our methods. Moreover, to evaluate how diverse the recommendation results are, URecSYS is compared to the baselines in terms of *news set diversity* described in Ref. [35]. The diversity is defined as the average dissimilarity of all pairs of news articles in the recommendation list. Given a set of news articles N, the average dissimilarity of N is defined as follows:

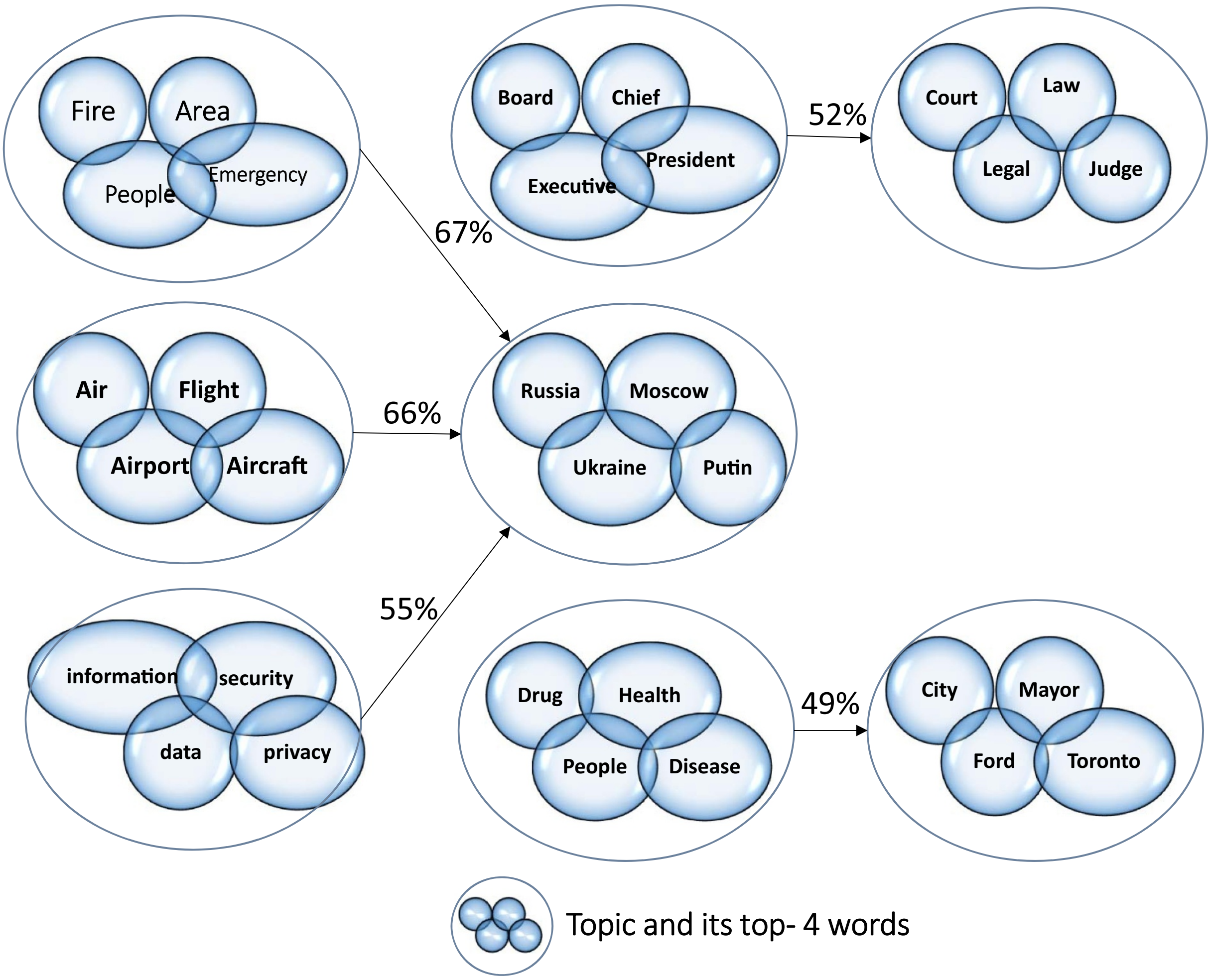


Fig. 8. Top-5 Topic-level Recommendation Rules discovered by $URecSYS_{TL}$.

Table 5
The average precision of top-5 topic-level recommendation rules.

Topic-level rules	*Possible response range*	*Actual response range*	*User judge mean (SD)*	*tconf (%)*
Fire, *People*, *Area*, *Emergency* → *Russia*, *Ukraine*, *Moscow*, *Putin*	0 –100	60 –100	78 (15)	67
Air, *Airport*, *Flight*, *Aircraft* → *Russia*, *Ukraine*, *Moscow*, *Putin*	0 –100	50 –100	64 (11)	66
Info. ,*Data*, *Security*, *Privacy* → *Russia*, *Ukraine*, *Moscow*, *Putin*	0 –100	40 –100	73 (21)	55
Board, *Executive*, *Chief*, *President* → *Court*, *Legal*, *Law*, *Judge*	0 –100	40 –100	67 (18)	52
Drug, *People*, *Health*, *Disease* → *City*, *Ford*, *Mayor*, *Toronto*	0 –100	50 –100	71 (21)	49

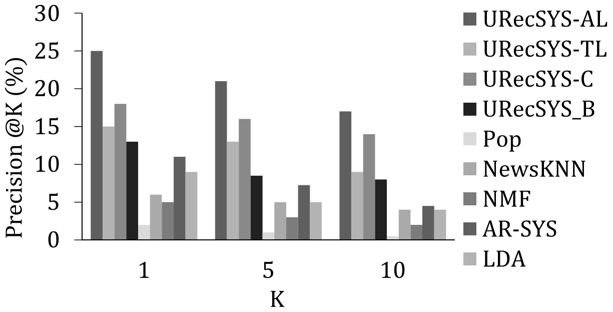


Fig. 9. Mean Average Precision @K for the different recommender systems.

$$AvgDis(N) = \frac{2}{p(p-1)} \sum_{nw_i \in N} \sum_{nw_j \in N \& nw_i \neq nw_j} (1 - Sim(nw_i, nw_j)) \tag{11}$$

In this formula, p is the number of news articles and the dissimilarity is defined as $1 - Sim(nw_i,n_j)$. $Sim(nw_i,nw_j)$ denotes the similarity of two news articles nw_i and nw_j and is calculated by cosine similarity.

We also set the utility threshold (e.g., *min_util*) such that the method discovers 100,000 high utility news reading patterns, the *min_uconf* to 0.6 and *min_tconf* to 0.45. Later, we show the sensitivity of results to the different values of these parameters.

5.5. News recommendation rules in practice

Table 4 presents top-5 article-level rules discovered by our approach and top-5 frequency-based recommendation rules (i.e., AR-SYS), which are sorted by the *utility confidence* and *support confidence* respectively.

Table 4 suggests that the rules with high support do not necessarily recommend articles that increases user engagement when we use dwell time as the measure of engagement. This is due to the fact that there are rules discovered from news reading patterns whose frequency is low (e.g., UR_1 and UR_2) but recommending such articles results in higher engagement.

Fig. 7 shows sum of the browsing time of two groups of recommendation rules. In this figure, the " x" axis refers to the top-k rules (e.g., article-level rules and frequency-based rules) and the " y" axis shows sum of the time spent based on the top-k recommendation rules. The results reveal that the rules discovered by $URecSY\ S_{AL}$ may lead users to spend more time as the recommended articles are of the user's interests.

Table 6
Diversity evaluation on the result list.

Methods	*Top@5*	*Top@10*	*Top@15*
Pop	0.541	0.438	0.360
NewsKNN	0.417	0.382	0.324
NMF	0.571	0.512	0.460
$AR-SY\ S$	0.381	0.319	0.284
LDA	0.507	0.472	0.312
$URecSY\ S_C$	**0.712**	**0.683**	**0.6552**

Bold shows that the proposed method outperforms the baselines in terms of different measures.

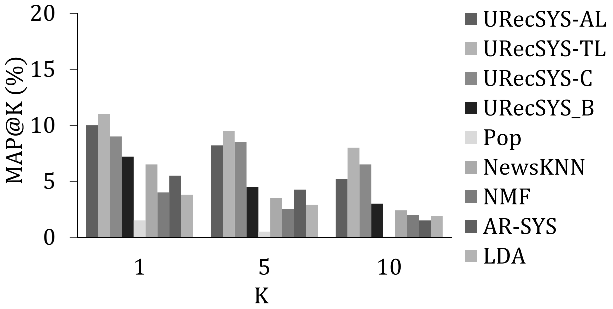


Fig. 10. MAP@K for the different systems on the dataset with the news cold-start problem.

Fig. 8 illustrates the top-5 topic-level recommendation rules obtained from the dataset. In this figure, each topic is represented by its top-4 words. The percentage on each relationship arrow shows the rule confidence. To validate the rules, we check if the top-5 rules discovered by our framework were truly associated in real life scenarios. We examine the association among topics by searching for the news articles using top-4 words from the topics as search keywords on the news portal. Since we use the data from 2014 to 2015 in the above experiments, we only consider news articles published within this time period. From the top-5 topic-level rules, we found that most of the times the rules found by our framework have very close associations in reality during the same period of time. For example, given topic $t_1 = \{Fire, People, Area, Emergency\}$, $t_2 = \{Russia, Ukraine, Putin, Moscow\}$, according to the rule in Fig. 8, 67% of users who read news articles on t_1 also read news articles on t_2. We found that in late 2014, two of the most associated topics were based on the conflicts between Russia and Ukraine in one hand, and Russian involvement in the Syrian Civil War on the other hand. The other rules revealed real world news stories in 2014 as well.

We conducted a user study to evaluate the interestingness of the top-5 article-level recommendation rules. We gave these rules to 10 undergraduate and graduate students, along with top-4 ranked keywords per topic. We asked them to judge the quality of the top-5 rules by first searching the top-4 ranked keywords in the newspaper and then giving a score between zero and one to rate each rule based on how strong they think the topics (in antecedent and consequent) in a rule are related and interesting. Table 5 shows the average results in percentage. The results show that the proposed framework for topic-level rule discovery ranks the rules very close to the ranking obtained from the survey participants.

5.6. Experimental results on recommendations

Fig. 9 shows the results of MAP@K for different recommendation systems. In this figure, " x" axis stands for different values of K and " y" axis stands for the MAP@K value obtained by each recommendation system. As the figure shows, $URecSY\ S_{AL}$ outperforms all the methods on different values of K. The results provide the following observations: 1) Although $URecSY\ S_B$ does not consider all the measures, it still works better than the baselines as it considers both topics-level and article-level rules and associations among articles based on the utility model.

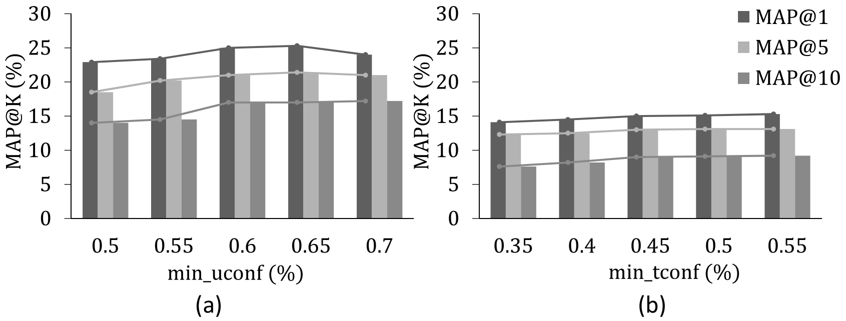


Fig. 11. (a) *URecSY S_{AL}* performance on different values of *min_uconf*, (b) *URecSY S_{TL}* performance on different values of *min_tconf*.

2) *URecSY S_{AL}* and *URecSY S_C* outperform other approaches, which demonstrates the superiority of our news utility model for news recommendation. 3) The performance gap between *URecSY S_{TL}* and the *AR-SYS* is significant, which reveals that topic-level recommendation rules are more suitable for recommendations.

Table 6 shows the average diversity results for different methods. It can be observed that, the diversity decreases as the number of recommended articles increases. This is due to the fact that when more articles are selected, the recommendation articles are getting closer to the user's interest and the recommended articles become more similar. URecSYS outperforms the other methods significantly and the diversity of its results decreases smoothly when the number of the recommended articles increases.

To simulate the news cold-start scenario, similar to Ref. [36], we divide the articles into two training and testing subsets. Since the number of articles published per day is large, we use 40% of the articles for training and 40% of articles for testing and 20% of articles in both training and testing. We compare our framework to the alternative recommendation methods on the dataset. In the scenarios of news cold-start, since there is no visits on news articles, most of the approaches do not work. Fig. 10 shows the average performance of each method across the dataset to the news cold-start scenario. In terms of MAP@K, *URecSY S_{TL}* outperforms all other approaches. The main reason is that this approach not only contains the relations among the topics based on user's interests, it can also recommend newly-published articles. In Fig. 10, *URecSY S_C* outperforms *URecSY S_{AL}* since its score is leveraged by the score obtained by *URecSY S_{TL}*.

We also evaluate the performance of *URecSY S_{AL}* and *URecSY S_{TL}* by varying *min_uconf* and *min_tconf*. Fig. 11 (a), (b) shows the sensitivity of the results with respect to different values *min_uconf* and *min_tconf*. The lower value of the *min_uconf* and *min_tconf* leads to a larger number of recommendation rules. Such added rules are not necessarily better rules as their confidence is lower. In general, lower values of the parameters decrease the performance of the recommendation systems in terms of the precision. This change is less significant for *URecSY S_{TL}* as the rules are generalized to topic level.

5.7. Discussions and implications

The literature in news recommendation systems indicates that having a model (e.g., the proposed news utility model) that can represent a domain specific objective is essential for recommendation. Given the model, the main challenge is how to effectively incorporate it during the recommendation process. Our experiments verify that our proposed utility model and its incorporation into the recommendation process improved the performance of the recommendation with respect to the objective defined.

The first implication in our experiment was the fact that applying the news utility model during article-level rule discovery, results in finding domain specific rules and therefore, recommended articles increase user experience quality with respect to the objective. The comparison against rules discovered by other methods verified that the results were promising and satisfactory. Moreover, in the topic-level, we observe that effective semantic connections among topics are built by using the utility model and the article-level rules. Therefore, we addressed the cold start problems effectively as we use the domain specific objective as well as the semantic connections during recommendations. Our experiments show that the article-level rules have better performance in the absent of cold start problem. This is due to the level of details that such rules represent. However, when there is news cold start (which is very common), the topic-level rules are of importance for having an effective recommendation system.

Analyzing two billion records in experiments verified the scalability of the proposed framework. We noticed that with the current dataset size, some baselines could not even finish the learning process due to the exponential growth of search space and we needed to adjust the parameters to get results. However, taking advantage of the Apache Spark technology and the MapReduce framework improved the performance of the system significantly.

6. Conclusions, limitations and future work

In the news recommendation context, the most challenging problem for any online news publisher is to make a balance between different business objectives (e.g., increasing user engagement through free content delivery in one hand and revenue maximization through subscription on the other hand). Such objectives are usually in contention with one another. Moreover, most existing news recommendation systems only consider the click (e.g., CTR) as an implicit feedback, which is quite problematic as a user might click on an article in which she/he is not interested in. To address these challenges, we designed a news utility model that represents business objectives by simultaneously considering both article-related and user-article interaction attributes. We showed that this model is more beneficial to recommend news articles compared to existing approaches. In addition, we argued that building a news recommendation system that can handle big data requires applying scalable techniques and platforms [3]. Moreover, despite the availability of such huge volume of historical data, most interactions belong to un-subscribed users (i.e., unknown visitors). Therefore, we proposed a novel and scalable rule engine to discover article-level recommendation rules based on the news utility model. At its heart, the rule engine uses multiple MapReduce-like steps to discover article-level recommendation rules in parallel. We also proposed a novel probabilistic approach on top of a well-known topic model (i.e., Latent Dirichlet Allocation) [37] to generalize article-level news recommendation rules. The output is a topic-level recommendation rule engine that links topics of articles based on the news utility model, thus business objectives. Such rules recommend newly-published articles based on topics of interests to a user, thus addressed the news cold start problem effectively. Lastly, we conducted extensive experiments on a dataset of two billion records collected from a major Canadian news agency (The Globe and Mail) and showed the effectiveness of the proposed recommendation system by comparing to state-of-the-art recommendation systems in practice.

In the future, we plan to study how to further improve the recommendation quality through the use of more rigorous utility models to better measure a business objective. Moreover, our proposed utility model presents an effective mapping from attributes to a model with respect to a domain specific objective, however, there are objectives that all the mapping details (e.g., aggregation functions) are not available. Particularly, when the objective is not as popular as user engagement. For such objectives, experts are required to design the model. This can be addressed by data driven modeling approaches and build an end-to-end solution to design the model automatically. In future, we investigate how data driven approaches can be applied to address this limitation.

This research sheds new light on the usage of the utility model in the user modeling, which can be consumed in more application scenarios. Note that the proposed utility model provides a general platform so that different utility models can be plugged in as desired. We plan to investigate the applicability of the proposed framework to other similar domain such as recommending research papers published in open access portals to users. The number of research paper published is exponentially growing (e.g., cold start problems) and a user (e.g., researcher) cannot keep the pace to find the most recent but essential work to his/her research interests and he/she needs to choose papers from the huge quantity of potential candidates. Our aim is to show the effectiveness of the proposed utility model and the recommendation system to effectively address challenges in this domain.

Acknowledgments

This work is funded by Natural Sciences and Engineering Research Council of Canada (NSERC), The Globe and Mail, and the Big Data Research, Analytics and Information Network (BRAIN) Alliance established by the Ontario Research Fund - Research Excellence Program (ORF-RE).

Appendix A. Appendix

A.1. Theorem1 proof

Let set $X_t = X - nw_{k-t}$ that is X_t has all articles from X excludes article nw_{k-t}. For example, $X_1 = \{nw_1, nw_2, \ldots, nw_{k-2}, nw_k\}$. Since $1 \leq t < k$, we know that $\sigma(nw_{k-t},P) < \sigma(nw_k,P)$, thus: $\sum_{j=1}^{k-1} \sigma(nw_j, P) + \sigma(nw_{k-t}, P) < \sum_{j=1}^{k-1} \sigma(nw_j, P) + \sigma(nw_k, P)$. Then , $\sum_{j=1}^{k-1} \sigma(nw_j, P) < \sum_{j=1}^{k-1} \sigma(nw_j, P) - \sigma(nw_{k-t}, P) + \sigma(nw_k, P)$. In the above inequality, the left side is X_0 and the right side is X_1, so we have $\sigma(X_0,P) < \sigma(X_1,P)$. Given $\sigma(nw_i,P)$, we have $\frac{\sigma(nw_i, P)}{\sigma(X_0, P)} > \frac{\sigma(nw_i, P)}{\sigma(X_1, P)}$. Since $\frac{\sigma(nw_i, P)}{\sigma(X_0, P)}$ yields the *uconf* of a rule $\langle \{X - nw_k\} \rightarrow nw_i \rangle$, the proof is complete.

References

[1] K. Xie, Y. Wu, J. Xiao, Q. Hu, Value co-creation between firms and customers: the role of big data-based cooperative assets, Information and Management 53 (2016) 1034–1048.

[2] H. Davoudi, M. Zihayat, A. An, Time-aware Subscription Prediction Model for User Acquisition in Digital News Media, Proceedings of the SDM'2017, 2017, pp. 135–143.

[3] R. Agarwal, V. Dhar, Big data, data science, and analytics: the opportunity and challenge for IS research, 2014.

[4] D.D. Lee, H.S. Seung, Algorithms for non-negative matrix factorization, Advances in Neural Information Processing Systems, 2001, pp. 556–562.

[5] J. He, H. Liu, H. Xiong, SocoTraveler: Travel-package recommendations leveraging social influence of different relationship types, Information and Management 53 (2016) 934–950.

[6] R. Mishra, P. Kumar, B. Bhasker, A web recommendation system considering sequential information, Decision Support Systems 75 (2015) 1–10.

[7] T.C.-K. Huang, Y.-L. Chen, M.-C. Chen, A novel recommendation model with Google similarity, Decision Support Systems 89 (2016) 17–27.

[8] S.J. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, Pearson Education Limited, Malaysia, 2016.

[9] F.W. Taussig, Principles of Economics, 2 Cosimo, Inc., 2013.

[10] P. Lops, M. De Gemmis, G. Semeraro, Content-based recommender systems: state of the art and trends, Recommender Systems Handbook, Springer, 2011, pp. 73–105.

[11] J. Liu, C. Wu, W. Liu, Bayesian probabilistic matrix factorization with social relations and item contents for recommendation, Decision Support Systems 55 (2013) 838–850.

[12] X. Su, T.M. Khoshgoftaar, A survey of collaborative filtering techniques, Advances in Artificial Intelligence 2009 (2009) 4.

[13] A.S. Das, M. Datar, A. Garg, S. Rajaram, Google news personalization: scalable online collaborative filtering, Proceedings of WWW'07, ACM, 2007, pp. 271–280.

[14] G. Shani, D. Heckerman, R.I. Brafman, An MDP-based recommender system, Journal of Machine Learning Research 6 (2005) 1265–1295.

[15] F. Garcin, B. Faltings, O. Donatsch, A. Alazzawi, C. Bruttin, A. Huber, Offline and online evaluation of news recommender systems at swissinfo.ch, Proceedings of the 8th ACM RecSYS, ACM, 2014, pp. 169–176.

[16] X. Yi, L. Hong, E. Zhong, N.N. Liu, S. Rajan, Beyond Clicks: Dwell Time for Personalization, Proceedings of RecSys '14, ACM, 2014, pp. 113–120.

[17] H.-N. Kim, A. El-Saddik, G.-S. Jo, Collaborative error-reflected models for cold-start recommender systems, Decision Support Systems 51 (2011) 519–531.

[18] H. Liang, Y. Xu, D. Tjondronegoro, P. Christen, Time-aware topic recommendation based on micro-blogs, Proceedings of the 21st ACM International Conference on Information and Knowledge Management, ACM, 2012, pp. 1657–1661.

[19] D. Agarwal, B.-C. Chen, R. Gupta, J. Hartman, Q. He, A. Iyer, S. Kolar, Y. Ma, P. Shivaswamy, A. Singh, L. Zhang, Activity Ranking in LinkedIn Feed, Proceedings of the 20th ACM SIGKDD, ACM, 2014, pp. 1603–1612.

[20] Y.-M. Li, C.-T. Wu, C.-Y. Lai, A social recommender mechanism for e-commerce: combining similarity, trust, and relationship, Decision Support Systems 55 (2013) 740–752.

[21] L. Li, T. Li, News recommendation via hypergraph learning: encapsulation of user behavior and news content, Proceedings of the sixth ACM WSDM, ACM, 2013, pp. 305–314.

[22] E. Zhong, N. Liu, Y. Shi, S. Rajan, Building discriminative user profiles for large-scale content recommendation, Proceedings of the 21th ACM SIGKDD, ACM, 2015, pp. 2277–2286.

[23] Y. Jiang, J. Shang, Y. Liu, Maximizing customer satisfaction through an online recommendation system: a novel associative classification model, Decision Support Systems 48 (2010) 470–479.

[24] Z. Li, X. Fang, X. Bai, O.R.L. Sheng, Utility-based link recommendation for online social networks, Management Science 63 (2016) 1938–1952.

[25] M. Scholz, V. Dorner, M. Franz, O. Hinz, Measuring consumers' willingness to pay with utility-based recommendation systems, Decision Support Systems 72 (2015) 60–71.

[26] G. Chandrashekar, F. Sahin, A survey on feature selection methods, Computers & Electrical Engineering 40 (2014) 16–28.

[27] M. Tavakolifard, J.A. Gulla, K.C. Almeroth, J.E. Ingvaldesn, G. Nygreen, E. Berg, Tailored News in the Palm of Your Hand: A Multi-perspective Transparent Approach to News Recommendation, Proceedings of WWW '13, 2013, pp. 305–308.

[28] V. Karnowski, A.S. Kmpel, L. Leonhard, D.J. Leiner, From incidental news exposure to news engagement. How perceptions of the news post and news usage patterns influence engagement with news articles encountered on Facebook, Computers in Behavior 76 (2017) 42–50.

[29] D.-R. Liu, C.-H. Lai, W.-J. Lee, A hybrid of sequential rules and collaborative filtering for product recommendation, Information Sciences 179 (2009) 3505–3519.

[30] M. Zihayat, Z.Z. Hu, A. An, Y. Hu, Distributed and parallel high utility sequential pattern mining, IEEE International Conference on Big Data, 2016, pp. 853–862.

[31] X. He, T. Chen, M.-Y. Kan, X. Chen, Trirank: Review aware explainable recommendation by modeling aspects, Proceedings of CIKM, ACM, New York, NY, USA, 2015, pp. 1661–1670.

[32] B.N. Miller, I. Albert, S.K. Lam, J.A. Konstan, J. Riedl, MovieLens unplugged: experiences with an occasionally connected recommender system, Proceedings of ACM IUI, ACM, 2003, pp. 263–266.

[33] C. Kim, J. Kim, A recommendation algorithm using multi-level association rules, Proceeding of WI 2003, 2003, pp. 524–527.

[34] G. Guo, J. Zhang, Z. Sun, N. Yorke-Smith, LibRec: A Java Library for Recommender Systems, UMAP Workshops, 2015.

[35] M. Zhang, N. Hurley, Avoiding monotony: improving the diversity of recommendation lists, Proceedings of the 2008 ACM Conference on Recommender Systems, ACM, 2008, pp. 123–130.

[36] I. Barjasteh, R. Forsati, F. Masrour, A.-H. Esfahanian, H. Radha, Cold-start item and user recommendation with decoupled completion and transduction, Proceedings of RecSYS, ACM, 2015, pp. 91–98.

[37] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent Dirichlet allocation, Journal of Machine Learning Research 3 (2003) 993–1022.

Morteza Zihayat is an assistant professor at the School of Information Technology Management of Ryerson University from 2016 and IBM CAS Faculty Fellow from 2018. Before joining ITM, he was a Postdoctoral Fellow at University of Toronto (2015-2016). He was also a research fellow in the IBM Cloud Analytics as a member of the BRAIN ALLIANCE — Big Data Research, Analytics, and Information Network. His research concerns Big Data Analytics and machine learning. He was recently awarded multiple research grants, including NSERC Discovery Grant, NSERC Engage and MITACS. He has ongoing collaborations with industry, including IBM Canada, The Globe and Mail and AT &T Labs Research. Morteza obtained his PhD from York University where he worked on designing scalable frameworks to discover actionable knowledge from Big Data streams and social networks. His research has been published in top-tier data mining and data management venues such as Information Sciences, Machine Learning, SIGKDD, SIAM SDM, PKDD, EDBT.

Anteneh Ayanso is Professor of Information Systems and founding director of the Centre for Business Analytics at the Goodman School of Business, Brock University. He teaches Business Analytics, Database Design and Management, Data Mining Techniques & Applications and Management of IS/IT. He received PhD in Information Systems from the University of Connecticut and MBA from Syracuse University. His research interests focus primarily on data management and information retrieval, Big Data analytics, electronic commerce, and electronic government. His articles are published in leading journals such as Decision Sciences, Decision Support Systems, European Journal of Operational Research, Journal of Database Management, International Journal of Electronic Commerce, Government Information Quarterly, among others. His research has been funded by government grants, including NSERC Discovery Research Grant, NSERC Engage Grant and Voucher for Innovation and Productivity (VIP) by Ontario Centres of Excellence (OCE). He is currently serving as an Associate Editor at Decision Support Systems journal and a review board member at Journal of Database Management, and International Journal of Convergence Computing.

Aijun An is a Professor in the Department of Electrical Engineering and Computer Science at York University. She is currently leading the Big Data Research, Analytics and Information Network (BRIAN) Alliance, an Ontario-based research network that involves four universities and a dozen of private and public sector partners. Her main research area is data mining. She has worked on various research topics in data mining, including classification, clustering, data stream mining, high utility pattern mining, sentiment and emotion analysis from text, topic detection, parallel and distributed deep learning, graph mining and bioinformatics. She has published extensively in various well-respected journals and conferences in data mining, databases, optimization, and intelligent information systems. Her research has been supported by NSERC, SSHRC, and ORF-RE.

Heidar Davoudi recieved his PhD in Computer Science at Department of Electrical Engineering and Computer Science at York University, Canada. His research interest includes data mining and machine learning and, in particular, user modeling for acquisition, engagement, and recommendation.

Xing Zhao is a Master's student in Data Mining Lab at the Department of Electrical Engineering and Computer Science at York University. His research areas of interest are machine learning and Big Data. He received B.Sc., Spec. Hons. in Computer Science from York University in 2017.