Expert Systems With Applications 183 (2021) 115341

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Exploring the effects of different Clustering Methods on a News Recommender System

Douglas Zanatta Ulian [a,1,4,*], João Luiz Becker [b,1,2,4], Carla Bonato Marcolin [c,1,3,4], Eusebio Scornavacca [d,4]

[a] *Escola de Administração, Universidade Federal do Rio Grande do Sul, Rua Washington Luiz, 855, 90010-460, Porto Alegre, RS, Brasil*
[b] *Departamento de Tecnologia e Ciência de Dados, EAESP/FGV, Av. 9 de Julho, 2029, 01313-902, São Paulo, SP, Brasil*
[c] *Faculdade de Gestão e Negócios, Departamento de Operações e Sistemas, Universidade Federal de Uberlândia, Av. João Naves de Ávila, 2121, Bloco 5M, Sala 100, 38400-902, Uberlândia, MG, Brazil*
[d] *Merrick School of Business, University of Baltimore, William H. Thumel Sr. Business Center 473, 11 W Mt Royal Ave Baltimore, MD 21201, USA*

ARTICLE INFO

Keywords:
Recommender systems
News recommender systems
Clustering
Collaborative filtering
Content filtering
Data mining

ABSTRACT

News recommendations distinguishes from general content recommendations as it takes in consideration news freshness, sparsity, monotony and time. Recent works approach these features using hybrid Collaborative-Content-based Filtering methods, adapting clustering techniques to handle sparsity and monotony without considering the effects that different clustering methods may have over recommendation results. Such studies often evaluate the results of varying different parameters individually, ignoring possible interaction effects between them. They also base their results on metrics such as accuracy and recall that are sensitive to bias. To investigate the importance of clustering method selection to News Recommender System results we evaluated the effects of different traditional techniques in recommending news articles. We implemented an algorithm that used a hybrid Collaborative-Content-based Filtering method to incorporate user behavior, user interest, article popularity and time effect. The system uses an article selection method that built the recommendation set based on content features. With this algorithm, we examined the existence of interaction effects between the input parameters. We used a Gaussian regression process to explore the response surface while sequentially optimizing parameters. To avoid being misled by underlying biases we used Informedness, an accuracy metric that captures both positive and negative information from prediction results. Our results demonstrated that different clustering methods had a significant influence on the recommendation results. It was also found that a traditional hierarchical method outperformed optimization methods with important performance improvement. In addition, we demonstrated that parameters may interact with each other and that analyzing them separately may mislead interpretation.

1. Introduction

In the new data-driven digital ecosystem, Recommender Systems (RS) play a vital role in individual consumption. These systems are an essential part of on-line retailers such as Amazon and E-bay as well as digital entertainment companies such as Netflix (Ji, Hong, Shangguan, Wang, & Ma, 2016). News services such as Google News and Yahoo! are also leveraging from statistical techniques to understand their audience and recommend news articles according to individual needs at different point of the consumption cycle (Bai, Cambazoglu, Gullo, Mantrach, & Silvestri, 2017; Das, Datar, Garg, & Rajaram, 2007; Schlesinger & Doyle, 2015). While item recommendation is already a common practice across most online businesses, it still represents a challenge to online news companies. In the literature there is a current discussion

[*] Corresponding author.
E-mail addresses: douglas.ulian@gruporbs.com.br (D.Z. Ulian), joao.becker@fgv.br (J.L. Becker), carla@ufu.br (C.B. Marcolin), escornavacca@ubalt.edu (E. Scornavacca).

[1] This work was produced while Ulian, Becker and Marcolin were at the Graduate Program in Management (PPGA) in the School of Management (EA) at the Federal University of Rio Grande do Sul (UFRGS).

[2] Becker gratefully acknowledges the support from CNPq, Federal Brazilian funding agency.

[3] Marcolin gratefully acknowledges the support from CAPES, Federal Brazilian funding agency.

[4] This work was developed while the first author of the study was working at Grupo RBS. The authors thank the partner organization for kindly providing the resources necessary to conduct the research.

https://doi.org/10.1016/j.eswa.2021.115341
Received 5 July 2020; Received in revised form 21 April 2021; Accepted 2 June 2021
Available online 7 June 2021

with opposing views on how articles should be selected — either by the professional judgment of an editor or by capturing audience interest (Welbers, Van Atteveldt, Kleinnijenhuis, Ruigrok, & Schaper, 2016). While using quantitative methods can assist to precisely reveal the relationship between audience preferences and editorial decisions, it still poses substantial challenges to researchers (Lee, Lewis, & Powers, 2014). In this context, research aimed at improving the quality of News Recommender Systems has the potential to make a positive contribution to the online media industry.

To produce recommendations, Recommender Systems usually consider the users' reading history, individual profile, content characteristics and user-generated feedback (Li, Zheng, Yang, & Li, 2014). These systems can generate direct connections among users and/or items by dynamically matching users' preferences and/or feedback about items (Shi, Larson, & Hanjalic, 2014).

Several challenges arise when recommending news due to the recommendation process itself. For example, dodging monotony can be challenging, as the RS seeks for some diversification of items while avoiding unwanted ones (Gu, Yang, & Dong, 2017; Li et al., 2014; Wang, Gao, Feng, & Pan, 2017). Scalability is another obstacle since the growth of both the number of users and items increases the computation time exponentially (Lee, Hong, Jung, Shin, & Kim, 2016; Lu, Qin, Cao, Liu, & Wang, 2014; Zahra et al., 2015). Another traditional challenge for RS is the cold start problem, as the lack of data for new users or new items reduces the capability of the system to identify the users' preferences (Jonnalagedda, Gauch, Labille, & Alfarhood, 2016; Lin, Xie, Guan, Li, & Li, 2014; Wei, He, Chen, Zhou, & Tang, 2017; Zheng, Li, Hong, & Li, 2013). Time dynamics represent another constraint, and the RS must address the effect of time over the users' interest (Beutel et al., 2018; Larrain, Trattner, Parra, Graells-Garrido, & Nørvåg, 2015; Song, Elkahky, & He, 2016; Wei et al., 2017).

Other challenges relate to the proper nature of the news. News freshness may be the most important challenge as the users' interest over an article decays as time passes (Sottocornola, Symeonidis, & Zanker, 2018; Zheng et al., 2013). On the other hand, news items that have been available for some time and, consequently, may have received the appraisal of many users, are likely to be of little interest. This puts forward the relevance of the item cold-start problem in this context.

Another difficulty relates to data sparsity, a traditional RS problem intensified by the large difference between the numbers of articles produced and articles exposed to the user while browsing the news (Kille et al., 2015; Lin et al., 2014; Sottocornola et al., 2018).

These challenges have been the focus of recent work around News Recommender Systems (NRS) using clustering methods as a way to overcome them (Bouras & Tsogkas, 2014; Cleger-Tamayo, Fernández-Luna, Huete, Pérez-Vázquez, & Cano, 2010; Das et al., 2007; Lee et al., 2016; Lee & Park, 2007; Li, Wang, Li, Knox, & Padmanabhan, 2011; Lu et al., 2014; Sadhasivam, Saranya, & Praveen, 2014; Wei, Xun, & Wang, 2009; Wu, Ding, Wang, & Xu, 2010; Zahra et al., 2015; Zheng et al., 2013). Clustering techniques are primarily used to reduce the user/item space and, therefore, the computational cost of making recommendations. This quality makes them popular in NRS. However, their successful application depends heavily on the selection of the method itself — a factor that is rarely taken into consideration. For example, Karimi, Jannach, and Jugovac (2018) reviewed over 140 NRS articles published between 2005 and 2016. They identified 13 articles that applied clustering methods as part of their algorithm. Only one single paper evaluated the impact of different clustering methods.

As a result, the goal of this paper is to evaluate the effects of different clustering methods on the results of an NRS. The experiment developed demonstrates that the choice for a particular method has significant influence on overall NRS results. It also shows the existence of interaction effects among different input parameters — something that should not be neglected when optimizing multiple parameters. The findings should help future NRS research to evaluate different clustering methods as well as input parameters influence on overall results.

The remaining of the paper is structured as follows. Section two reviews recent studies regarding the application of clustering techniques in NRS. This is followed by a description of the model developed. Section four presents the methodology used in our experimental design. This is followed by a presentation of the results and implications. The paper concludes with a discussion of research contributions and implications for practice and future research.

2. Recent work

The use of clustering techniques for news recommendation is very popular among news applications and platforms (Karimi et al., 2018). There are many clustering methods deployed by NRS and the impact of its selection is often overlooked in the literature. One of the most popular clustering methods in NRS is K-means. For example, Zahra et al. (2015) proposed improvement over random centroid creation for K-means. They modeled user interest as the centroid profile of the users' cluster. Sadhasivam et al. (2014) proposed to build recommendations by integrating user profiles, composed by user location and user sentiments, with cluster profiles obtained with K-means. Lee and Park (2007) used K-means to model users' interest based on (a) demographic data; (b) the similarity between each user's access patterns over articles' segments; and (c) article's importance and recency. Wei et al. (2009) proposed using one-class classification to identify important financial news articles. They trained a classifier based on a recall threshold and used the profile generated by it to evaluate the importance of new articles. The model was tested against K-means showing better results. Bouras and Tsogkas (2014) proposed a method that modeled users' interest by clustering reading sessions using K-means. The clusters were used to create interest profiles with word hypernyms connected to articles keywords. It is important to notice that among all these studies the reasons for choosing K-means were not specified. K-means potential disadvantages regarding outliers (Arora, Varshney, & Deepali, 2016) and scalability (Arora et al., 2016; Velmurugan & Santhanam, 2010) were not considered nor other methods were tested.

In their work, Lee et al. (2016) proposed a method called Adaptive Collaborative Filtering Based on Scalable Clustering. They used Expectation Maximization to cluster users and to cluster items, modeling user preferences and content characteristics, reducing data sparsity and improving scalability. They also did not mention the reasons for choosing an Expectation Maximization method.

In Li et al. (2011) and Zheng et al. (2013) soft and hard clustering were used to create multi-level representations of user interest. The impacts of selecting different clustering methods were discussed and tested. To model interest, Li et al. (2011) tested Latent Dirichlet Analysis (LDA) plus Locality Sensitive Hashing against K-means direct clustering. As the similarity concept was not shared by both methods the results cannot be evaluated regarding solely the clustering method. Zheng et al. (2013) tested Probabilistic Latent Semantic Indexing (PLSI), Fuzzy K-means and LDA to model interest with the later presenting better results. Both Li et al. (2011) and Zheng et al. (2013) also tested Average Linkage hierarchical clustering against K-means for clustering articles with the former presenting significant advantage over the later. The tests were run only once with metrics subject to bias (Powers, 2011) and without evaluating possible interaction effect over input variables. The reasons for selecting these methods were also not mentioned.

In their work, Cleger-Tamayo et al. (2010) proposed a method that clustered users to create interest profiles. They tested soft-clustering using PLSI and hard-clustering using K-means, the first outperforming the second. Like in the works of Li et al. (2011) and Zheng et al. (2013), the difference of similarity concepts between the methods prevents from evaluating the clustering method impact on the results. The reasons for selecting these methods were also not mentioned.

Table 1
Recent studies using clustering methods in NRS.

Author	Title	Clustering methods	Clustered items	Method justification	Evaluation metric	Parameter optimization
Das et al. (2007)	Google news personalization: scalable online collaborative filtering	MinHash clustering	Users	Not specified	Precision, recall	Single
Lee and Park (2007)	MONERS: A news recommender for the mobile web	K-means	Users	Not specified	Number of hits (true positives)	Single
Wei et al. (2009)	One-class classification based finance news story recommendation	Rocchio, k-means, one-class SVM	Articles	Not specified	Recall, Precision, BEP, F-score	Single
Chu and Park (2009)	Personalized recommendation on dynamic content using predictive bilinear models	K-means	Users	Not specified	Number of clicks per rank	K-fold cross validation
Qiu, Liao, and Li (2009)	News recommender system based on topic detection and tracking	Topic Detection and Tracking	Articles	Not specified	Miss Probability, False Alarm Probability	Not done
Wu et al. (2010)	Topic based automatic news recommendation using topic model and affinity propagation	Affinity propagation, LDA	Topics	Not specified	Recall, Precision, F-score	Single
Cleger-Tamayo et al. (2010)	A proposal for news recommendation based on clustering techniques	K-means and affinity propagation	Users	Popularity/ Not Specified	Spearman's rank correlation, Kullback–Leibler divergence	Not clear
Li et al. (2011)	SCENE: A scalable two-stage personalized news recommendation system	LSH, Average linkage, K-means	Articles	Popularity	F-score, accuracy, recall	Single
Zheng et al. (2013)	PENETRATE: Personalized news recommendation using ensemble hierarchical clustering	EM, Consensus hierarchy, LDA, Ensemble hierarchy, Kmeans	Articles and Users	Not specified	F1-Score, recall	Single
Lu et al. (2014)	Scalable news recommendation using multi-dimensional similarity and Jaccard–Kmeans clustering	K-means	Users	Not specified	Accuracy, recall	Single
Sadhasivam et al. (2014)	Personalization of news recommendation using genetic algorithm	Not specified	Users	Not specified	Accuracy, F-score	Evolutionary optimization
Bouras and Tsogkas (2014)	Improving news articles recommendations via user clustering	K-means, Wk-means	Users	Not specified	CI, MAE, F-score	Not clear
Zahra et al. (2015)	Novel centroid selection approaches for KMeans-clustering based recommender systems	K-means	Users	Not specified	MAE, ROC-sensitivity, precision, recall, F-score	Single
Lee et al. (2016)	Adaptive collaborative filtering based on scalable clustering for big recommender systems	EM	Articles and Users	Not specified	MAE	Not done
Iglesias, Tiemblo, Ledezma, and Sanchis (2016)	Web news mining in an evolving framework	Evolving fuzzy systems	Articles	Not specified	Accuracy	Single

Table 1 summarizes our analysis of recent studies using clustering methods in NRS. The analysis demonstrates that while the clustering methods used are highly diverse, the clustered items, on the other hand, are mostly limited to either users or articles. It suggests that researchers often overlook the impacts of the clustering method and tend to not consider evaluation metric biases (see ”Method Justification” and ”Evaluation Metric” columns). In addition, interaction effects among input parameters are also often neglected (see ”Parameter Optimization” column).

In this sense, the work of Lu et al. (2014) is particularly interesting to our study. They proposed to adapt K-means to work with a multi-dimensional similarity. The modification aimed at scaling computational power but ignored its impacts on K-means' clustering mechanism. Their model included multiple input parameters assembled into two similarity measures, the first accounting for user content interest and the second for user behavior. Although they explored the effects of varying input parameters they did not account for interaction effect between them. Finally, they used accuracy and recall as evaluation metrics, both susceptible to bias as they do not account for the proportion between positive and negative prediction results. These characteristics made it a perfect candidate to serve as the base of the RS we would implement on our research.

3. Algorithm implementation

The algorithm implemented in this paper has its foundation on the algorithm structure developed by Lu et al. (2014). We note, though, that our work is not an extension of Lu et al. (2014). While we use a similar structure, we have modified some of the inner parts of the algorithm in order to achieve our goals of testing different clustering methods and investigating the effect of input parameters choices. The system is a memory-based algorithm that applies Collaborative Filtering and Content Filtering to measure the distance between users. The choice considers two important aspects of their work. First, the decisions made by the authors regarding clustering method selection, evaluation metric selection and interaction effect consideration, which are the focus of our study. Second, the data used by the RS, user behavior records and content features, which are usually available in the news business, making it easily reproducible. From this data the algorithm models user interest, user behavior, article popularity and time effect, giving multiple dimensions and input parameters to test. It combines them on a single similarity measure, enabling the test of different methods with the same input. In the evaluation process, we use an unbiased metric to measure prediction efficiency (Schröder, Thiele, & Lehner, 2011). We form the recommendation set using an article selection method based on content features. We also use a Gaussian regression process to sequentially optimize parameters (Bartz-Beielstein, 2010).

We perform one modification to the RS proposed by Lu et al. (2014) as we use tags already available instead of extracting them from the article's text using LDA. We do so on the belief that tags produced by human experts, in our case the articles' editors, should outperform LDA. To reduce data sparsity, we applied plural stemming to these tags using the rslp R package (Falbel, 2016). To test our RS, we used a dataset provided by a digital news platform of a major media company.

Throughout the paper we use the following parameters and datasets:

A**:** Set of articles ordered from 0 to $|A| - 1$;

J: Set of tags associated with articles, ordered from 0 to $|J|-1$;

A_j: Set of articles associated with tag $j \in J$; $\bigcup_{j\in J} A_j = A$; A_j and A_k might not be disjoint for some $j,k \in J$; $\left|A_j\right| \geq 1$ for all $j \in J$;

U: Set of users, ordered from 0 to $|U|-1$;

${}_uA$ Set of articles accessed by user u, where $u \in U$; $\bigcup_{u\in U} {}_uA = A$; ${}_uA$ and ${}_vA$ might not be disjoint for some u, $v \in U$; $|{}_uA| \geq 1$ for all $u \in U$.

E: Set of access event times of articles in A by users in U, ordered by the instant in time in which the events have occurred;

E_a: Set of access event times of article $a \in A$ by users in U; $\bigcup_{a\in A} E_a = E$; given the accuracy of timestamps, E_a and E_b might not be disjoint for some $a, b \in A$; $|E_a| \geq 1$ for all $a \in A$

${}_uE$: Set of access event times of articles in A by user $u \in U$; $\bigcup_{u\in U} {}_uE = E$; given the accuracy of timestamps, E_u and E_v might not be disjoint for some $u, v \in U$; $|{}_uE| \geq 1$ for all $u \in U$

${}_uE_a$: Set of access event times by user $u \in U$ to article $a \in A$; $\bigcup_{u\in U} {}_uE_a = E_a$; $\bigcup_{a\in A} {}_uE_a = {}_uE$; ${}_uE_a$ might be empty for some $a \in A$ and some $u \in U$; given the accuracy of timestamps, ${}_uE_a$ and ${}_vE_a$ might not be disjoint for some $a \in A$ and some $u, v \in U$

m: The most recent time of access by any user in U to any article in A.

3.1. User to user behavior distance

We define the user to user behavior distance as the weighted Jaccard distance between the articles access' vectors of two users. This distinguishes old and fresh news as well as popular and unpopular articles. Differently from the traditional Jaccard distance that sums the matches between two vectors, the weighted Jaccard distance summed, for each match, their weights (Lu et al., 2014). The weight of a match is calculated using a different numerator, depending on (a) the article's popularity and (b) the time difference between the accesses of each user. We use the article's popularity to adjust the matches in the numerator multiplying it by the logarithm of the article's total number of accesses. The popularity factor (Lu et al., 2014) for article a, $p(a)$, is defined as:

$$p(a) = \log(1 + |E_a|) \tag{1}$$

We also adjust the Jaccard distance numerator considering how further apart in time were the accesses of each user over the matching articles. The time adjustment factor is given by the inverse exponential of the time difference between the accesses, measured in days, multiplied by a weighting coefficient (α) (Lu et al., 2014). The value of α will be defined in the training and testing processes, as explained later in the paper. To improve computational performance, we calculate the exponential for each user before the distance calculation process leaving only the subtraction between exponentials inside it. The time adjustment factor $f(\alpha, u, v, a)$ for users u and v regarding article a is defined as:

$$f(\alpha,u,v,a) = \begin{cases} e^{-\alpha\left|\left(\frac{\sum_{s\in{}_uE_a} s}{|{}_uE_a|}\right)-\left(\frac{\sum_{s\in{}_vE_a} s}{|{}_vE_a|}\right)\right|} & \text{if } {}_uA \cap {}_vA \neq \Phi \\ 0 & \text{if } {}_uA \cap {}_vA = \Phi \end{cases} \tag{2}$$

A behavior distance is then defined, for users u and v, as:

$$b(\alpha,u,v) = \begin{cases} 1 - \dfrac{\sum_{a\in{}_uA\cap{}_vA} \frac{f(\alpha,u,v,a)}{\log(1+|E_a|)}}{|{}_uA \cup {}_vA|} & \text{if } u \neq v \\ 0 & \text{if } u = v \end{cases} \tag{3}$$

3.2. User to user content distance

We define the content distance of two users as the cosine dissimilarity between their tags' profile vectors, that is,

$$c(\lambda,u,v) = 1 - \cos(\vec{u}_\lambda, \vec{v}_\lambda) = 1 - \frac{\vec{u}'_\lambda \vec{v}_\lambda}{|\vec{u}_\lambda||\vec{v}_\lambda|} \tag{4}$$

where $\vec{u}_\lambda$ and $\vec{v}_\lambda$ are the tags profile vectors, as defined in what follows, of users u and v respectively, adjusted by a coefficient labeled λ.

The users' tags profile is a $|J|$-dimensional vector accounting for the number of accesses of each tag by each user, adjusted by the age of each access. The age of a given access is defined as the time difference between the most recent access time in the whole dataset and the time of that access. As a user might access the same tag several times (through several articles), we take the average age of all accesses to that tag to build the profile. We define each user tags profile component x as the average time of all accesses to tag j by user u adjusted by λ, as:

$$x(\lambda,u,j) = \begin{cases} \displaystyle\sum_{a\in{}_uA\cap A_j} e^{-\lambda\left(m-\frac{\sum_{s\in{}_uE_a} s}{|{}_uE_a|}\right)} & \text{if } {}_uA \cap A_j \neq \Phi \\ 0 & \text{if } {}_uA \cap A_j = \Phi \end{cases} \tag{5}$$

Coefficient λ represents a weight given to time in the content distance, adjusting relevance of recent versus old accesses in content similarity between users (Lu et al., 2014).

3.3. User to user hybrid distance

We define the mixed distance m as a weighted average of the behavior and the content distances with weight β. We express the mixed distance calculation as:

$$m(\beta,\alpha,\lambda,u,v) = \beta b(\alpha,u,v) + (1-\beta)c(\lambda,u,v) \tag{6}$$

All user-to-user distances are calculated in advance. Fig. 1 illustrates the calculation steps.

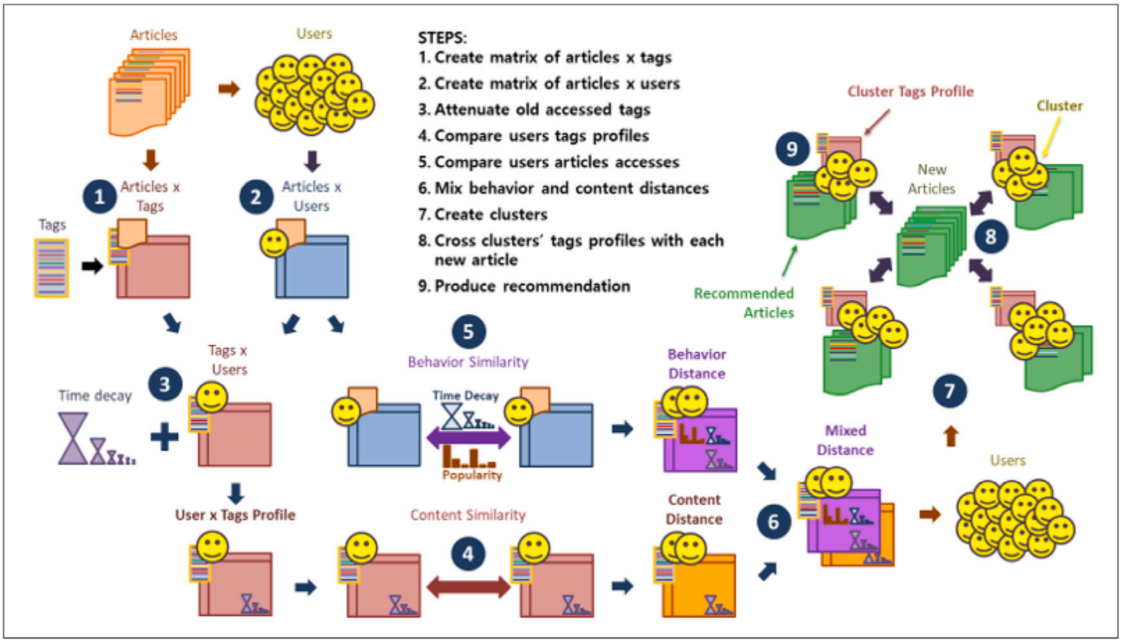
3.4. Clustering method

In order to carry out the analysis over clustering method selection we considered traditional optimization and hierarchical methods (Kaufman & Rousseeuw, 2009). As for optimization methods, we selected k-medoids and discarded K-means, the method proposed by Lu et al. (2014), for three reasons. First, K-means' inability to take the dissimilarity matrix as input, like k-medoids, making its testing against other methods subject to distance calculation. Second, K-means has performance disadvantages when compared do k-medoids (Arora et al., 2016; Velmurugan, 2012; Velmurugan & Santhanam, 2010). Finally, K-means can be heavily affected by outliers and produces more intense overlapping problems than k-medoids (Arora et al., 2016).

Two implementations of k-medoids were available in the Comprehensive R Archive Network, PAM (Maechler, Rousseeuw, Struyf, Hubert, & Hornik, 2014) and K-Medoids (Mouselimis, 2018). Both implementations are based on the Partitioning Around Medoids (PAM) (Kaufman & Rousseeuw, 2009) Fortran algorithm. Since there were no reason for choosing one over another, we proceeded with both.

As for hierarchical methods, we selected six of the eight most traditional methods: Ward.D, Ward.D2, Single Linkage, Average Linkage, Complete Linkage, Mcquitty, Centroid and Median (Everitt, Landau, Morven, & Stahl, 2011; Kaufman & Rousseeuw, 2009; Müllner et al., 2013). We discarded the Centroid and Median methods as they did not take the dissimilarity matrix as input (similarly to the K-means optimization method) (Everitt et al., 2011).

Several methods exist to evaluate cluster quality, influence and robustness (Everitt et al., 2011). As there is no single method that ensures validity and practical significance of a cluster solution (Hair,



STEPS:
1. Create matrix of articles x tags
2. Create matrix of articles x users
3. Attenuate old accessed tags
4. Compare users tags profiles
5. Compare users articles accesses
6. Mix behavior and content distances
7. Create clusters
8. Cross clusters' tags profiles with each new article
9. Produce recommendation

Fig. 1. Distance calculation process.

Table 2
Dataset split.

Dataset	Events	Articles	Days
Training	329.142	23.418	152
Testing	98.867	6.681	45
Total	428.009	30.099	197

Hair, Black, Babin, & Anderson, 2013), we decided not to apply any quality, influence or robustness test. Instead, we left these evaluations, together with the decision of the number of clusters, k, to the parameter optimization process. As a result, the cluster solution quality would be defined by real-world testing.

3.5. Recommendation evaluation

We tested the effect of the algorithm on an off-line dataset, splitting it into two portions: training and testing sub-datasets. We used the training sub-dataset to build the user profiles based on user behavior and content similarity. We then used user profiles to build the recommendation set on the testing sub-dataset and also assess how well informed it is in relation to users' interest. Although users participate on both the training and testing sub-datasets, no behavior nor content data is shared between them. In the process of splitting the dataset we aimed at creating an approximate 80/20 split for the window of time as well as for the volume of data. A larger proportion of data in the training dataset is an important factor to capture users' characteristics to predict their behavior. On the other hand, the size of the testing dataset cannot be too small, to avoid overfitting. The test was done a single time for each experiment run, assessing the NRS's capability to predict all the user readings in the testing sub-dataset at once. Table 2 shows the profile of the dataset split for events and articles.

The method for selecting articles to recommend could vary in the way it selects tags for the cluster's profile as well as in the way it selects articles to recommend. Lu et al. (2014) did not mention the methods used to select either tags or articles, so we decided to test two different approaches, index, and top$n$. In both methods, we ordered the items based on their weight values from the largest to the smallest. The index method accumulates the weight values sum until reaching a given threshold, selecting these items. The top$n$ method selects the first n items. When using the top$n$ method we always select the same number of items for all clusters, while the index method selects several items that account for the same amount of relevance across all clusters.

Having defined the recommendation set, we selected a metric to assess its effect. We used Schröder et al. (2011) method for selecting a metric, defining our goal as "to select the best set of articles ranked so they would reflect the users' preferences". Proceeding with Schröder et al. (2011) method, we selected classification accuracy metrics from the ranking recommendation category, since our data are all implicit. From that category, we chose Informedness, as it combines probabilities of recommending an accessed article and of not recommending an article not accessed. Informedness also takes into consideration all the information from the confusion matrix, avoiding possible bias (Powers, 2011).

4. Methodology

The resulting model has three exogenous settings and six exogenous parameters that need to be defined *a priori*. They are:

- Clustering method, either "PAM", "K-Medoids", "Ward.D", "Ward.D2", "Single", "Average", "Complete" or "Mcquitty";
- Tags selection method, either "index" or "top$n$";
- Articles selection method, either "index" or "top$n$";
- α coefficient, defining the attenuation weight of access time difference in the behavior distance, ranging between zero and one, zero meaning no attenuation at all and one a full exponential attenuation.
- β coefficient, defining the proportion between behavior and content distance on the mixed distance, ranging from zero to one, zero meaning the use of only content distance and one the use of only behavior distance;
- λ coefficient, defining the weight of time on content distance, ranging between zero and one, zero meaning that only the most recent events would be considered, and one considering the same weight to all events, regardless of time;
- γ coefficient, defining the amount of tags to select, according to the tags selection method; for the index method, it denotes the selection percentile, ranging from zero to one, zero meaning the

Table 3
Experiment's settings and parameters bounds table.

Parameter		From	To
α		0	1
β		0	1
γ	Index	0	1
	top$n$	1	4941
ζ	Index	0	1
	top$n$	1	6681
λ		0	1
k		2	77
Clustering method		PAM, K-Medoids, Ward.D, Ward.D2, Single, Average, Complete, Mcquitty	

selection of the first tag and one meaning all tags; for the top$n$ method, it denotes the number of tags to select, ranging from 1 to $|J|$;

- ζ coefficient, defining de amount of articles to select, according to the articles selection method; for the index method, it denotes the selection percentile, ranging from zero to one, zero meaning the selection of the first article and one meaning all articles; for the top$n$ method, it denotes the number of articles to select, ranging from 1 to $|A|$ in the testing dataset;
- k, defining the number of clusters, ranging from 2 to the squared root of the total items, a traditional empiric metric (Han, Pei, & Kamber, 2011);

The chosen parameters are intimately related to the NRS domain. Time, a variable that plays a central role to news freshness and user interest, is calibrated by parameters α and λ. Tags and articles selection, which are key for building the recommendation set and dealing with the item cold-start problem, are optimized by parameters γ and ζ. The weight associated to behavior and content similarities, adjusted by parameter β, aims to balance reading habits and text resemblance in the process of building user profiles. Finally, parameter k tries to identify the best set of user interest profiles, represented by the cluster collection.

The model performance depends on the choice of all settings and parameters, and one would like to know which settings and parameter values would maximize Informedness. This is something that cannot be determined analytically so we used a parameter optimization method to search for global optima. We evaluated traditional Response Surface and Computer Experiments methodologies for parameter optimization, choosing the latter (Montgomery, 2017). We used a sequential, surrogate model-based approach to global optimization (Bartz-Beielstein & Zaefferer, 2017) and carried out the experiments using a Latin Hypercube Design and a Gaussian regression process, also known as a Kriging model (Bartz-Beielstein & Zaefferer, 2017).

As two of the input settings are binary (Tags and Articles selection methods) we need four different experiments to test their combination. The cluster method set has eight different possibilities raising the number of experiments to thirty-two. Table 3 displays the settings and parameters' bounds we applied. For each combination we executed 200 runs of the Kriging optimization model.

It is important to point out that the Kriging model is intent to linear optimization problems, taking continuous parameters as input. The k parameter, as well as the γ and ζ coefficients, when defined by top$n$ criteria, are integers, so we rounded them before passing on to the test function. While acknowledging this might lead to non-optimum results, we made no further efforts to address this issue, since the optimization process was not the main and final purpose of our research.

5. Results

The main output from the Kriging model are the test function maxima and the parameter values that generate it. The optimization function recorded these values while it ran and returned them for analysis, enabling us to evaluate its evolution. The thirty-two experiments ran for two hundred times each totalizing sixty-four hundred runs. We present the results in Fig. 2.

Although none of the experiments converged to its best solution after 200 runs, the results lead to some interesting conclusions. Perfect parameter optimization through adaptive surrogate models is very difficult due to parameter interaction and parameter sensitivity, so approximately optimal solutions were expected (Wang et al., 2014). If the parameters were perfectly optimized by one of the experiments, without any level of insensitivity or interaction, that would by easily identified. When looking to the chart depicted in Fig. 2 we would see that at a certain run the experiment would find its highest result and that the following runs would always reproduce it. This would draw a straight horizontal line starting at that run until the end of the experiment. That is almost the case in experiment g3, where after 20 runs results stop varying much, sitting a little above 0.1 degrees of Informedness.

Another way to identify stability would be to analyze the differences between median, third quartile and maximum values for the accuracy metric chosen (in our case, Informedness). Smaller differences indicate closeness to stability. Based on this criterion we can observe in Table 4 that experiment (e1) and (e3), both using Single Linkage hierarchical method, although presenting poor results, are closer to stability than others. We present Precision and Recall results among other commonly used RS metrics in appendix Table A.1. A strong correlation of Informedness and Recall was expected since Informedness captures all the information from the confusion matrix using Recall and Inverse Recall (Powers, 2011). The correlation for the 32 experiments was of 0.9999819. Relevant correlation between Informedness and Precision was also expected as they are both obtained from the confusion matrix. The resulting correlation between Informedness and Precision for the 32 experiments was of 0.9542111.

We performed a Three-way Fixed Effects ANOVA model over the 32 experiments. We analyzed Informedness means to assess what factors influence the overall results and whether there was an interaction effect among them. The analysis demonstrated that all factors significantly influence the result. It also detected a significant interaction effect between tags and articles selection methods. In addition, a significant interaction effect was also found between cluster method and articles selection method. The results are presented on Table 5.

With regards to the tags selection settings the index method outperformed the top$n$ method. On all sixteen sets of experiments it always presented highest maxima and highest medians (see Table 4). This also emerges from the top left part of Fig. 3 which shows a box-plot of the differences between Informedness medians and maxima across all sixteen sets of corresponding experiments (differences taken between tags selection methods index versus top$n$ criterion). The average median difference is 0.0399, the difference peaking at 0.0932 for experiments (d2) and (d4). The average maxima difference is 0.0545, the difference peaking at 0.1624 for experiments (g2) and (g4). The bottom left part of Fig. 3 shows a box-plot of the ratio between Informedness medians and maxima across all sixteen sets of corresponding experiments. The average median ratio is 1.2434, the ratio peaking at 1.5848 for experiments (h2) and (h4). The average maxima ratio is 1.2417, the ratio peaking at 1.7237 for experiments (g2) and (g4). Both differences and ratios were all positive, a clear indication of prevalence from index over top$n$ method for tags selection.

As for the articles selection settings, the values in Table 4 also indicate an advantage of the index method over the top$n$ criterion, although not as clear as for the tags selection settings.

The top right part of Fig. 3 shows a box-plot of the differences between Informedness medians and maxima across all sixteen sets of corresponding experiments (differences taken between articles selection methods index versus top$n$ criterion). The average median difference is 0.0574, the difference peaking at 0.1363 for experiments

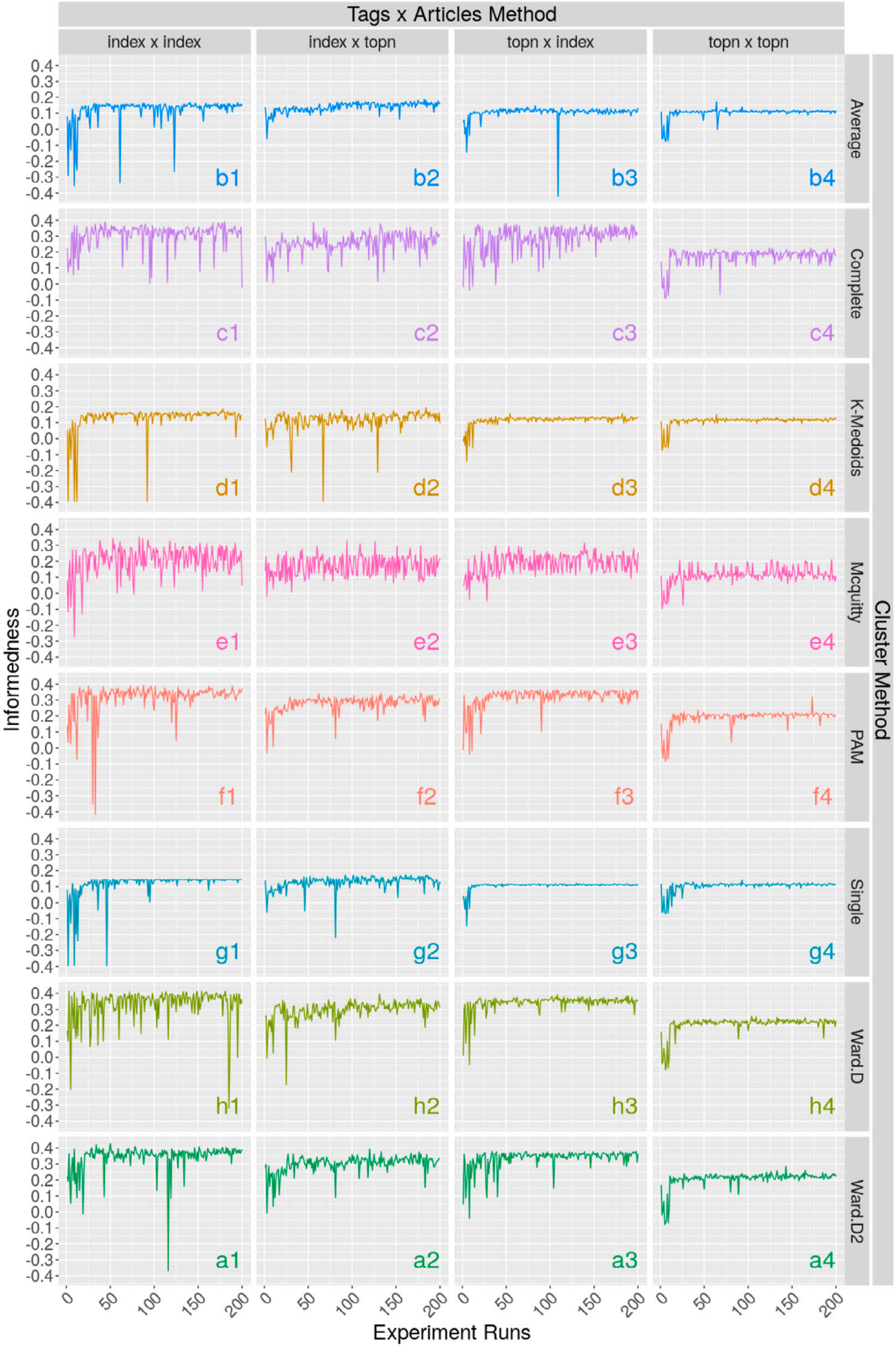


Fig. 2. Results evolution through 200 runs of each experiment.

(d3) and (d4). One of the differences is slightly negative, though, for experiments (e3) and (e4). The average maxima difference is 0.0352, the difference peaking at 0.1539 for experiments (g3) and (g4). Five differences are negative, the lowest of which is equal to −0.0272 for the experiments (e1) and (e2). The bottom right part of Fig. 3 shows a box-plot of the ratio between Informedness medians and maxima across all sixteen sets of corresponding experiments. The average median ratio is 1.2886, the ratio peaking at 1.7902 for experiments (h3) and (h4). The average maxima ratio is 1.1232, the ratio peaking at 1.6858 for experiments (g3) and (g4).

The advantage of the index method versus the top*n* method in both tags selection and articles selection settings also emerges when one compares box plots of the Informedness of all experimental runs segmented by the tags selection method and by the articles selection method, as is depicted in Figs. 4(a) and 4(b).

We also note that the best experiment using the combination of top*n* criterion for both tags and articles selection settings, experiment (d4), only came in 17th position, signaling that such setup did not perform well in our experimental sampling.

When analyzing the combination of top*n* and index criteria across tags and articles selection settings, the disadvantage of the top*n* criteria

Table 4
Experiments results summary after 200 runs each // Accuracy metric: Informedness.

Cluster/Tags/Articles methods	mean	sd	min	Q1	med	Q3	max
(a1) PAM/Index/Index	0.3150	0.0970	−0.4127	0.3204	0.3405	0.3544	0.3924
(a2) PAM/Index/Top N	0.2779	0.0500	−0.0299	0.2607	0.2914	0.3059	0.3462
(a3) PAM/Top N/Index	0.3180	0.0623	−0.0405	0.3099	0.3361	0.3519	0.3651
(a4) PAM/Top N/Top N	0.1904	0.0522	−0.0833	0.1932	0.2053	0.2101	0.3198
(b1) Kmedoid/Index/Index	0.1277	0.0867	−0.3963	0.1308	0.1506	0.1581	0.1864
(b2) Kmedoid/Index/Top N	0.1152	0.0633	−0.3941	0.1011	0.1297	0.1480	0.1937
(b3) Kmedoid/Top N/Index	0.1158	0.0337	−0.1459	0.1148	0.1241	0.1294	0.1569
(b4) Kmedoid/Top N/Top N	0.1109	0.0320	−0.0735	0.1138	0.1180	0.1212	0.1501
(c1) Ward.D/Index/Index	0.3316	0.1012	−0.3183	0.3195	0.3657	0.3864	0.4143
(c2) Ward.D/Index/Top N	0.2951	0.0626	−0.1721	0.2741	0.3094	0.3304	0.3972
(c3) Ward.D/Top N/Index	0.3410	0.0505	−0.0474	0.3403	0.3522	0.3607	0.3917
(c4) Ward.D/Top N/Top N	0.2075	0.0528	−0.0794	0.2117	0.2198	0.2278	0.2573
(d1) Ward.D2/Index/Index	0.3439	0.0832	−0.3683	0.3429	0.3648	0.3803	0.4287
(d2) Ward.D2/Index/Top N	0.2996	0.0552	−0.0090	0.2873	0.3122	0.3319	0.3943
(d3) Ward.D2/Top N/Index	0.3357	0.0601	−0.0416	0.3358	0.3553	0.3650	0.3795
(d4) Ward.D2/Top N/Top N	0.2069	0.0527	−0.0793	0.2063	0.2190	0.2266	0.2847
(e1) Single/Index/Index	0.1175	0.0800	−0.3957	0.1241	0.1432	0.1451	0.1483
(e2) Single/Index/Top N	0.1267	0.0425	−0.2178	0.1188	0.1365	0.1506	0.1755
(e3) Single/Top N/Index	0.1067	0.0265	−0.1466	0.1086	0.1116	0.1139	0.1212
(e4) Single/Top N/Top N	0.1039	0.0340	−0.0673	0.1056	0.1124	0.1171	0.1399
(f1) Average/Index/Index	0.1201	0.0808	−0.3531	0.1264	0.1464	0.1512	0.1680
(f2) Average/Index/Top N	0.1386	0.0306	−0.0600	0.1246	0.1456	0.1593	0.1878
(f3) Average/Top N/Index	0.1054	0.0470	−0.4192	0.1028	0.1129	0.1198	0.1480
(f4) Average/Top N/Top N	0.1039	0.0331	−0.0749	0.1067	0.1111	0.1141	0.1736
(g1) Complete/Index/Index	0.3041	0.0775	−0.0249	0.2997	0.3278	0.3460	0.3926
(g2) Complete/Index/Top N	0.2549	0.0642	0.0064	0.2320	0.2572	0.2957	0.3868
(g3) Complete/Top N/Index	0.2852	0.0805	−0.0377	0.2684	0.3062	0.3386	0.3783
(g4) Complete/Top N/Top N	0.1710	0.0549	−0.0907	0.1641	0.1883	0.1994	0.2244
(h1) Mcquitty/Index/Index	0.2057	0.0907	−0.2726	0.1556	0.2254	0.2721	0.3494
(h2) Mcquitty/Index/Top N	0.1708	0.0641	−0.0201	0.1167	0.1775	0.2227	0.3293
(h3) Mcquitty/Top N/Index	0.1862	0.0635	−0.0486	0.1371	0.2005	0.2385	0.3152
(h4) Mcquitty/Top N/Top N	0.1121	0.0501	−0.0976	0.0971	0.1120	0.1291	0.2125

Table 5
Three-way Fixed Effects ANOVA analysis of the 32 experiments; dependent variable: Informedness means.

Source	SS	df	MS	F	df1	df2	*p*-value
Tags selection (TS)	0.0092	1	0.0092	23.43	1	31	0.000
Cluster method (CM)	0.1946	7	0.0278	70.50	7	31	0.000
Articles selection (AS)	0.0187	1	0.0187	47.49	1	31	0.000
TS x CM	0.0017	7	0.0002	0.63	7	31	0.725
TS x AS	0.0050	1	0.0050	12.72	1	31	0.001
CM x AS	0.0126	7	0.0018	4.55	7	31	0.001
Residual	0.0028	7	0.0004				
TOTAL	0.2447	31					

in one of the selections is attenuated when the index was used in the other. As seen in Fig. 5, the lowest performance of the combination of top*n* criterion for both tags and articles selection settings is clear, but it becomes less clear when either one is combined with the index criteria. As already mentioned, the index method for tags selection presents an advantage over the top*n* criterion, which is supported by its steady presence in the top six best results list, experiments (d1), (c1), (c2), (d2), (g1) and (a1). We can assume that selecting the same proportion of information from all clusters, regardless of the number of tags needed to represent it, creates more custom profiles. Nevertheless, using the index method for both tags and articles selection settings did perform better for the tested sample. The top two results, experiments (d1) and (c1), use this combination of settings.

Considering the index method for both tags and articles selection, the two best results reach 0.4287 and 0.4208 points of Informedness, as seen in Table 6, suggesting that global maxima might sit around this range. The top 62 results were obtained by either Ward.D2 or Ward.D methods, indicating a clear advantage of Ward's minimization of total error sum of squares clustering criteria (Murtagh & Legendre, 2014) in our sample. The methods were implemented by both Ward.D and Ward.D2 algorithms, shared with the K-means method, which is the original choice of Lu et al. (2014). Our top two best results are based on Ward.D2, signaling its advantage over the other seven methods for the tested sample, as seen in Fig. 6.

Table 6
Top twenty results for any clustering method using any tags or articles selection method.

#	Cluster method	Tags method	Articles method	Result
1	Ward.D2	Index	Index	0.4287
2	Ward.D2	Index	Index	0.4208
3	Ward.D	Index	Index	0.4143
4	Ward.D	Index	Index	0.4134
5	Ward.D	Index	Index	0.4129
6	Ward.D	Index	Index	0.4125
7	Ward.D	Index	Index	0.4121
8	Ward.D	Index	Index	0.4106
9	Ward.D	Index	Index	0.4106
10	Ward.D	Index	Index	0.4091
11	Ward.D2	Index	Index	0.4090
12	Ward.D	Index	Index	0.4089
13	Ward.D	Index	Index	0.4089
14	Ward.D2	Index	Index	0.4087
15	Ward.D	Index	Index	0.4081
16	Ward.D	Index	Index	0.4068
17	Ward.D	Index	Index	0.4066
18	Ward.D	Index	Index	0.4062
19	Ward.D2	Index	Index	0.4058
20	Ward.D	Index	Index	0.4048

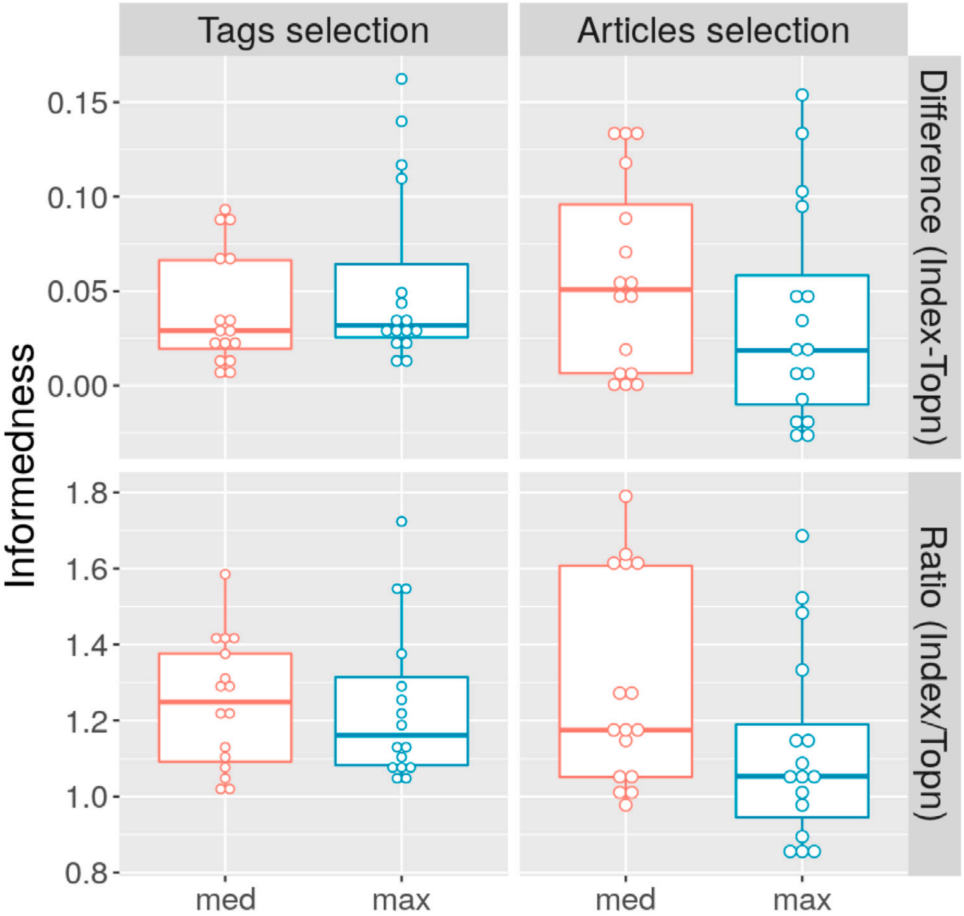
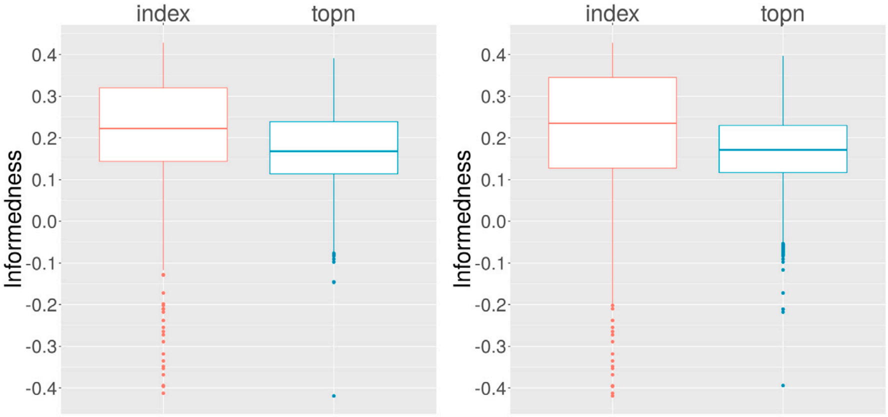


Fig. 3. Comparing the index method versus the top*n* criterion for both tags selection and articles selection settings: box plots of differences and ratios between Informedness median and maxima across sixteen sets of corresponding experiments.



(a) Results box plot for 6400 experiment runs, segmented by tags selection method.

(b) Results box plot for 6400 experiment runs, segmented by articles selection method.

Fig. 4. Results box plot for 6400 experiment runs.

PAM and Complete Linkage methods follow with a slight advantage to PAM. Both achieved almost the same maximal value for informedness (0,3924 and 0,3926 respectively), PAM obtaining better average results. PAM results are more stable as well, as can be seen in Fig. 2. Single Linkage, Average Linkage, and K-Medoids methods presented the worst results. Interestingly, K-Medoids presented results significantly different from PAM, although they both claim to implement the same algorithm. This is a clear indication that the implementations differ from each other.

Table 7 shows the best parameter values obtained by the top ten runs from the experiment (d1) which used Ward.D2 method for clustering and index method for both tags and articles selection. Only two of the top twenty best runs used an α coefficient value close to 1, and the top three results were obtained with values not higher than

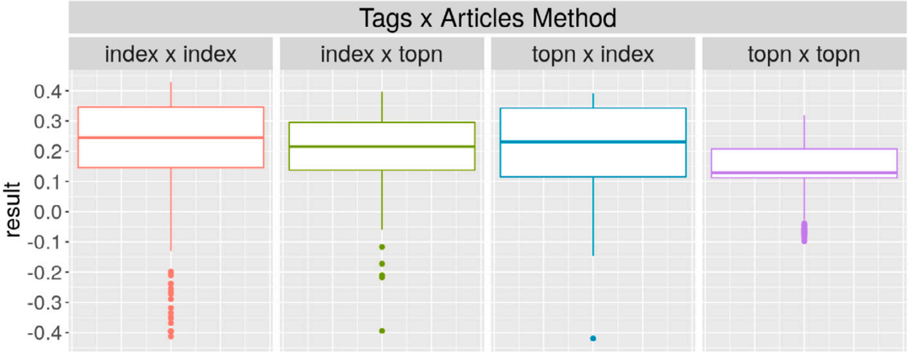


Fig. 5. Results box plot for 6400 experiment runs, segmented by tags and articles selection method.

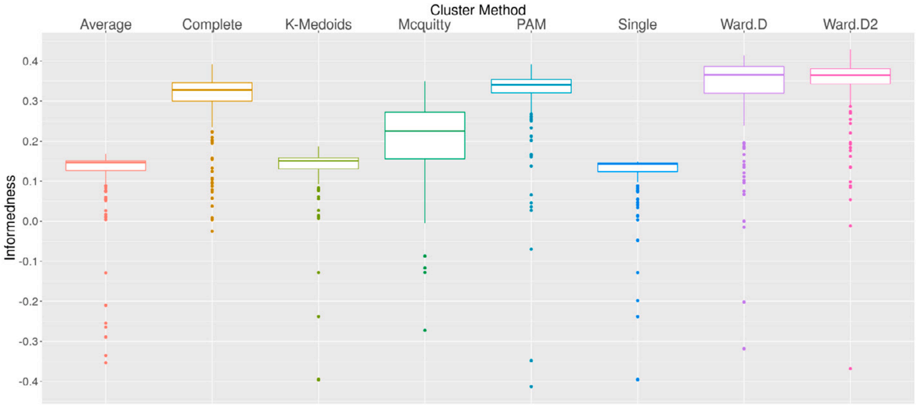


Fig. 6. Results box plot for 1600 experiment runs with index method for tags and articles selection, segmented by clustering method.

0.3851. This indicates the stronger relevance of recent similar behavior comparing with the similarity of events far apart from each other.

The top twenty best results for the experiment (d1) used β coefficient values higher than 0.5240, eight of them greater than 0.9048, indicating an advantage of behavior distance over content distance. The values vary considerably which may signal the presence of interaction effects.

The values of the λ coefficient in these top twenty best results are all scattered between 0 and 1, with a slight predominance of values below 0.5. Such behavior indicates either its irrelevance or strong interaction effects among all parameters used in the experiment.

The λ coefficient value in the best result depicted in Table 7, 0.3756, indicates that an important amount of data from the oldest records were of little contribution to the magnitude of the content distance. To enhance computational performance, this data could be taken away from the training dataset, reducing sparsity without losing much information.

The parameter k presented values ranging from 23 to 76, with the top four results sitting between 42 and 72 clusters. This suggests that a higher number of clusters may produce the best results. However, we point out the fact that the clusterization process is unsupervised and the number of individuals on each cluster may vary. The process may end up producing clusters with very few users which might be misleading. Fig. 7 presents the number of users per cluster for the top twenty best results of Ward.D2, ordered by cluster size. The most relevant clusters, with more than 50 users per cluster, are among the twenty larger ones.

Coefficient γ presented values higher than 0.6086 in 19 out of the top 20 runs, indicating that selecting a large portion of the most relevant tags of each cluster produces better results. Coefficient ζ also presented values ranging from 0.5023 to 0.5987, indicating that selecting around half a set of articles to recommend may produce better results. Although the Kriging Model was capable of estimating the model parameters' relevance, there are clear indications of interaction effects among the parameters, as already pointed out. Therefore, the assumption of parameters' relevance is questionable.

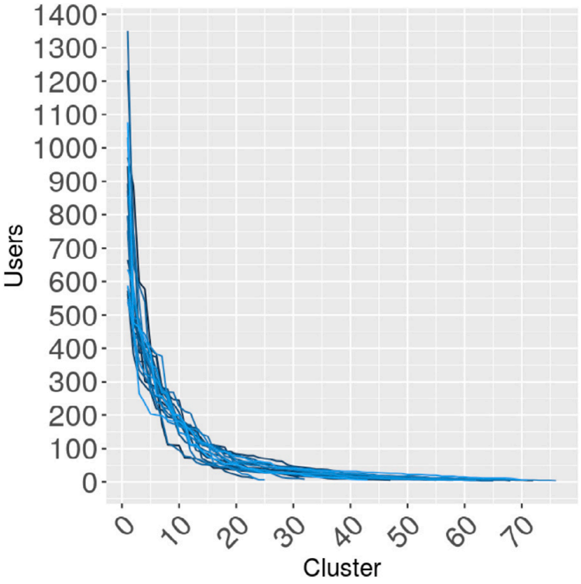


Fig. 7. Number of users per cluster for the top twenty best results of Ward.D2 clustering method using index method for tags and articles selection, ordered by cluster size.

Table 7
Top twenty results for Ward.D2 clustering method using index method for tags and articles selection.

#	α	β	γ	k	λ	ζ	Result
1	0.3821	0.9664	0.7264	54.7238	0.3756	0.5743	0.4287
2	0.2077	0.8792	0.8428	71.7226	0.4750	0.5414	0.4208
3	0.3851	0.6023	0.8481	42.1178	0.1677	0.5754	0.4090
4	0.9837	0.9274	0.6677	46.6213	0.3897	0.5520	0.4087
5	0.7011	0.5240	0.8882	31.2672	0.5566	0.5137	0.4058
6	0.4161	0.6386	0.8631	59.3369	0.2674	0.5433	0.4041
7	0.0238	0.9230	0.7144	65.2209	0.7198	0.5438	0.4032
8	0.4617	0.6752	0.9252	23.2446	0.9230	0.5450	0.4030
9	0.5446	0.9352	0.8398	68.0661	0.3700	0.5285	0.4005
10	0.2915	0.6602	0.8834	40.6738	0.1692	0.5460	0.3995
11	0.3645	0.8572	0.7718	64.7856	0.1612	0.5605	0.3995
12	0.9614	0.9717	0.6886	65.3369	0.1652	0.5449	0.3993
13	0.1142	0.7806	0.8312	31.5487	0.4192	0.5280	0.3992
14	0.5168	0.9557	0.4845	57.7599	0.0449	0.5987	0.3987
15	0.1139	0.8360	0.7123	76.4544	0.1389	0.5769	0.3987
16	0.5879	0.9048	0.7149	25.1794	0.3494	0.5606	0.3986
17	0.2465	0.7631	0.7879	55.5581	0.8338	0.5518	0.3980
18	0.3359	0.9453	0.6086	61.9349	0.4631	0.5023	0.3971
19	0.1718	0.6017	0.8980	71.1329	0.8539	0.5693	0.3965
20	0.2263	0.8868	0.7994	42.6262	0.7424	0.5228	0.3957

Table 8 shows detailed metrics for the top ten experiment runs with Ward.D2 clustering method and index method for both tags and articles selection. The precision metric was not higher than 3.21 percent meaning that we needed a very large number of recommendations until reaching each users' read. Considering users versus articles sparsity of 0.00006536, the low precision values found were within expectations (Srivastava, Bala, & Kumar, 2017).

Table 9 shows the parameter values corresponding to the best results in each experiment. The variation of values across all experiments is a clear indication of strong interaction effects among the parameters. The exception is the ζ coefficient which presented considerably high results for all experiments, either the articles that accumulated at least 44.64% of the recommendation weight (index criterion) or at least 130 articles (top*n* criterion). In both cases the number of recommendations was significantly high considering the average read rate in the dataset.

6. Conclusions

This paper aimed to demonstrate how the choice of a clustering method influences News Recommender Systems overall results. The literature review unveiled that researchers often overlook the impacts of this decision when developing NRS. We performed an experiment which consisted in the implementation of an algorithm based on Lu et al.'s work to test the impacts of selecting different clustering methods. We used a sequential parameter optimization process to explore the response surface calibrating parameters at each experiment run. In addition, we employed an unbiased accuracy metric to assess the result of each run.

Our results reveal that different clustering methods considerably influence the recommendation results. Traditional hierarchical methods outperform optimization methods with sizable performance improvements. Most significantly, model parameters interact with each other, implying that analyzing them separately in the search for best performance parameter values may mislead interpretation.

The difficulty in defining the best parameters for the RS, as well as the best clustering method, tags, and articles selection method, presents challenges faced by NRS. In such systems the recommendation is subject to many external effects related to the user and environment aspects, context, as well as to the researcher decisions, such as distance measures, clustering algorithms, recommendation set definition and evaluation calculation. When using clustering to recommend news, the selection of the clustering method should consider all these aspects. Different methods may have a significant influence on the overall recommendation result. The test of different methods, using an unbiased recommendation accuracy metric and evaluating possible interaction effects over input parameters, should produce significant improvement on recommendation quality.

7. Limitations and future research

While this study makes a positive contribution to NRS research, it also has limitations. For instance, the choice of one recommendation method (Lu et al., 2014) might have restricted the generalizability of our results. In addition, we performed some changes in Lu et al.'s method enabling the algorithm to perform the proposed tests. These adaptations, together with the specific features of the original method, may limit its generalizability.

The use of a single dataset is another limitation. While the use of article tags in the recommendation process may enhance accuracy, it also restricts the number of suitable datasets available. The replication of the study with a different recommendation method and multiple datasets may help to further understand the effects of different clustering methods on NRS results.

Finally, the use of a single split test is another limitation of the study. Although a three-way split would reduce overfitting risk (Sun et al., 2020), the associated computational cost would be greatly increased. In order to prevent overfitting, training and test sub-datasets were used in different stages of the experiment. As a result, there was no shared information among them. Overfitting risk may also be reduced by using repeated random sub-sampling or multiple evaluation windows. Nevertheless, accessing the user's interest for the recommended articles would only be possible through an online real-world experiment.

There is considerable scope for future research in NRS. For instance, there are opportunities for improvements in the recommendation process regarding data quality. Concerning tags, spell check on their input by editors, or the automatic extraction of them from the article's text using text-mining techniques like in the original method proposed at Lu et al. (2014) may also enhance the capability of tags to express content.

Further studies may also extend the hybrid user-to-user distance to include other data beyond behavior and content such as user billing information, demographic data, the user search engine inputs or customer service interactions. In addition, the study of interaction effects

Table 8
Confusion metrics and informedness for the top ten result using Ward.D2 clustering method and index method for tags and articles selection.

#	True pos.	True neg.	False pos.	False neg.	Precision	Recall	Miss rate	Informedness
1	6,568	2,03,915	1,97,908	561	0.0321	0.9213	0.0787	0.4287
2	7,660	2,63,164	2,56,224	720	0.0290	0.9141	0.0859	0.4208
3	7,772	3,12,739	3,05,798	831	0.0248	0.9034	0.0966	0.4090
4	8,009	3,16,251	3,09,102	860	0.0253	0.9030	0.0970	0.4087
5	7,459	3,12,360	3,05,726	825	0.0238	0.9005	0.0995	0.4058
6	6,519	2,67,388	2,61,605	736	0.0243	0.8986	0.1014	0.4041
7	8,256	3,48,751	3,41,434	939	0.0236	0.8979	0.1021	0.4032
8	2,540	1,01,327	99,078	291	0.0250	0.8974	0.1026	0.4030
9	7,693	3,38,041	3,31,247	899	0.0227	0.8954	0.1046	0.4005
10	8,042	3,56,765	3,49,672	949	0.0225	0.8945	0.1055	0.3995

Table 9
Experiments' best results parameters summary.

Cluster/Tags/Articles	α	β	γ	k	λ	ζ	Result
(a1) PAM/Index/Index	0.4821	0.5170	0.3940	77	0.0135	0.8213	0.3924
(a2) PAM/Index/Top N	0.8547	0.8922	0.7652	10	0.6981	304	0.3462
(a3) PAM/Top N/Index	0.7816	0.0749	1888	44	0.7583	0.9993	0.3651
(a4) PAM/Top N/Top N	0.1596	0.3099	1	67	0.3408	367	0.3198
(b1) Kmedoid/Index/Index	0.6412	0.3794	0.5776	24	0.9232	0.5983	0.1864
(b2) Kmedoid/Index/Top N	0.9945	0.2915	0.6051	9	0.1968	472	0.1937
(b3) Kmedoid/Top N/Index	0.5754	0.5950	6	15	0.8400	0.5456	0.1569
(b4) Kmedoid/Top N/Top N	0.5722	0.9840	9	21	0.6080	477	0.1501
(c1) Ward.D/Index/Index	0.5179	0.7442	0.6091	37	0.8991	0.5882	0.4143
(c2) Ward.D/Index/Top N	0.1012	0.0874	0.9132	6	0.0537	261	0.3972
(c3) Ward.D/Top N/Index	0.9059	0.8634	2673	72	0.8599	0.6725	0.3917
(c4) Ward.D/Top N/Top N	0.8090	0.0726	116	42	0.1850	193	0.2573
(d1) Ward.D2/Index/Index	0.3821	0.9664	0.7264	55	0.3756	0.5743	0.4287
(d2) Ward.D2/Index/Top N	0.3441	0.8667	0.7864	7	0.8358	213	0.3943
(d3) Ward.D2/Top N/Index	0.7816	0.0749	1888	44	0.7583	0.9993	0.3795
(d4) Ward.D2/Top N/Top N	0.3080	0.9293	9	48	0.5860	126	0.2847
(e1) Single/Index/Index	0.6818	0.0256	0.6911	70	0.3794	0.5099	0.1483
(e2) Single/Index/Top N	0.0713	0.5179	0.7492	7	0.3628	475	0.1755
(e3) Single/Top N/Index	0.9799	0.8760	29	15	0.4247	0.4464	0.1212
(e4) Single/Top N/Top N	0.8675	0.6237	25	10	0.1865	746	0.1399
(f1) Average/Index/Index	0.8967	0.6591	0.7209	73	0.7289	0.5862	0.1680
(f2) Average/Index/Top N	0.9818	0.3066	0.7578	73	0.5707	462	0.1878
(f3) Average/Top N/Index	0.8290	0.2757	8	41	0.6426	0.5929	0.1480
(f4) Average/Top N/Top N	0.5722	0.9840	9	21	0.6080	477	0.1736
(g1) Complete/Index/Index	0.2385	0.9304	0.5520	29	0.9755	0.6130	0.3926
(g2) Complete/Index/Top N	0.9316	0.6324	0.7769	5	0.8892	476	0.3868
(g3) Complete/Top N/Index	0.5416	0.4392	3121	76	0.7189	0.9873	0.3783
(g4) Complete/Top N/Top N	0.2540	0.2914	3868	48	0.6499	204	0.2244
(h1) Mcquitty/Index/Index	0.7430	0.1666	0.6006	73	0.8293	0.4897	0.3494
(h2) Mcquitty/Index/Top N	0.0167	0.3523	0.8437	75	0.8788	130	0.3293
(h3) Mcquitty/Top N/Index	0.8705	0.3666	3595	63	0.2387	0.5928	0.3152
(h4) Mcquitty/Top N/Top N	0.3422	0.1820	1826	47	0.6848	203	0.2125

between parameters may help improve recommendation effectiveness. Another study could extend the popularity factor applied to articles at the behavior distance to the tags used at the content distance. This improvement may reduce the weight popular tags have on measuring users' similarities and perhaps improving the algorithm capacity to capture users' interests. Future research could also validate the effects of an NRS on the user through a real-world experiment.

Yet another possible study may consider methods for automatically selecting the number of clusters, based on cluster quality evaluation after distance calculation. Such a study could explore the definition of k based on cluster quality metrics, such as the ASW, or the effects on the recommendation. Furthermore, the evaluation process could be improved by applying multiple time windows to the dataset split or using repeated random sampling. Another possible study would improve the optimization process, considering the differences between integer and continuous variables. At last, we could further explore cluster characterization by measuring similarity between cluster solutions.

CRediT authorship contribution statement

Douglas Zanatta Ulian: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing - original draft, Visualization, Funding acquisition. **João Luiz Becker:** Conceptualization, Methodology, Validation, Formal analysis, Investigation, Resources, Writing - original draft, Visualization, Supervision, Project administration, Funding acquisition. **Carla Bonato Marcolin:** Conceptualization, Writing - original draft, Writing - review & editing, Funding acquisition. **Eusebio Scornavacca:** Writing - review & editing, Supervision, Funding acquisition.

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.

Appendix

See Table A.1.

Table A.1
Other accuracy metrics for the best Informedness result in each of the 32 experiments.

Cluster/Tags/Articles	True pos.	True neg.	False pos.	False neg.	Precision	Recall	Miss rate
(a1) PAM/Index/Index	1377	58897	57695	174	0,0233	0,8873	0,1127
(a2) PAM/Index/Top N	6523	434671	429363	1214	0,015	0,8431	0,1569
(a3) PAM/Top N/Index	65	3517	3463	10	0,0184	0,8612	0,1388
(a4) PAM/Top N/Top N	5422	424786	420576	1211	0,0127	0,8173	0,1827
(b1) Kmedoid/Index/Index	2044	345178	344072	937	0,0059	0,6856	0,3144
(b2) Kmedoid/Index/Top N	6153	1009564	1006139	2727	0,0061	0,6929	0,3071
(b3) Kmedoid/Top N/Index	4541	875434	873271	2377	0,0052	0,6563	0,3437
(b4) Kmedoid/Top N/Top N	6760	1344015	1340903	3647	0,005	0,6496	0,3504
(c1) Ward.D/Index/Index	7937	300087	292952	801	0,0264	0,9083	0,0917
(c2) Ward.D/Index/Top N	4291	184142	180371	519	0,0232	0,892	0,108
(c3) Ward.D/Top N/Index	2695	118859	116509	344	0,0226	0,8867	0,1133
(c4) Ward.D/Top N/Top N	4639	545656	542516	1498	0,0085	0,7558	0,2442
(d1) Ward.D2/Index/Index	6568	203914	197908	561	0,0321	0,9213	0,0787
(d2) Ward.D2/Index/Top N	3746	164511	161232	466	0,0227	0,8892	0,1108
(d3) Ward.D2/Top N/Index	74	3547	3484	10	0,0208	0,875	0,125
(d4) Ward.D2/Top N/Top N	2987	288278	286120	828	0,0103	0,7829	0,2171
(e1) Single/Index/Index	7815	1543565	1540001	4250	0,005	0,6477	0,3523
(e2) Single/Index/Top N	7637	1342468	1338513	3681	0,0057	0,6747	0,3253
(e3) Single/Top N/Index	11510	2562505	2558026	7030	0,0045	0,6208	0,3792
(e4) Single/Top N/Top N	10227	2108395	2103937	5768	0,0048	0,6394	0,3606
(f1) Average/Index/Index	7810	1411969	1408053	3893	0,0055	0,6673	0,3327
(f2) Average/Index/Top N	7462	1199812	1195751	3400	0,0062	0,6869	0,3131
(f3) Average/Top N/Index	7807	1516817	1513263	4252	0,0051	0,6474	0,3526
(f4) Average/Top N/Top N	7574	1346768	1342876	3681	0,0056	0,6729	0,3271
(g1) Complete/Index/Index	4043	188673	185141	510	0,0214	0,8879	0,1121
(g2) Complete/Index/Top N	6817	328749	322843	910	0,0207	0,8822	0,1178
(g3) Complete/Top N/Index	77	3724	3659	11	0,0206	0,8738	0,1262
(g4) Complete/Top N/Top N	4342	576678	573998	1661	0,0075	0,7233	0,2767
(h1) Mcquitty/Index/Index	9429	646750	639032	1710	0,0145	0,8464	0,1536
(h2) Mcquitty/Index/Top N	2713	191923	189780	569	0,0141	0,8265	0,1735
(h3) Mcquitty/Top N/Index	4373	339538	336173	1007	0,0128	0,8127	0,1873
(h4) Mcquitty/Top N/Top N	4110	572669	570227	1667	0,0072	0,7114	0,2886

References

Arora, P., Varshney, S., & Deepali, V. (2016). Analysis of K-means and K-medoids algorithm for big data. *Procedia Computer Science*, *78*, 507–512.

Bai, X., Cambazoglu, B. B., Gullo, F., Mantrach, A., & Silvestri, F. (2017). Exploiting search history of users for news personalization. *Information Sciences*, *385*, 125–137.

Bartz-Beielstein, T. (2010). SPOT: An R package for automatic and interactive tuning of optimization algorithms by sequential parameter optimization. ArXiv preprint arXiv:1006.4645.

Bartz-Beielstein, T., & Zaefferer, M. (2017). Model-based methods for continuous and discrete global optimization. *Applied Soft Computing*, *55*, 154–167.

Beutel, A., Covington, P., Jain, S., Xu, C., Li, J., Gatto, V., et al. (2018). Latent cross: Making use of context in recurrent recommender systems. In *Proceedings of the eleventh ACM international conference on web search and data mining* (pp. 46–54). ACM.

Bouras, C., & Tsogkas, V. (2014). Improving news articles recommendations via user clustering. *International Journal of Machine Learning and Cybernetics*, 1–15.

Chu, W., & Park, S.-T. (2009). Personalized recommendation on dynamic content using predictive bilinear models In *Proceedings of the 18th international conference on world wide web* (pp. 691–700).

Cleger-Tamayo, S., Fernández-Luna, J. M., Huete, J. F., Pérez-Vázquez, R., & Cano, J. C. R. (2010). A proposal for news recommendation based on clustering techniques. In *International conference on industrial, engineering and other applications of applied intelligent systems* (pp. 478–487). Springer.

Das, A. S., Datar, M., Garg, A., & Rajaram, S. (2007). Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th international conference on world wide web* (pp. 271–280). ACM.

Everitt, B., Landau, S., Morven, L., & Stahl, D. (2011). *Wiley series in probability and statistics*, *Cluster analysis, 5th edition John Wiley & Sons* (5th ed.). Wiley.

Falbel, D. (2016). *A stemming algorithm for the Portuguese language*. The Comprehensive R Archive Network.

Gu, L., Yang, P., & Dong, Y. (2017). Diversity optimization for recommendation using improved cover tree. *Knowledge-Based Systems*, *135*, 1–8.

Hair, J., Hair, J., Black, W., Babin, B., & Anderson, R. (2013). Multivariate data analysis. In *Always learning*, Pearson Education Limited.

Han, J., Pei, J., & Kamber, M. (2011). *Data mining: Concepts and techniques*. Elsevier.

Iglesias, J. A., Tiemblo, A., Ledezma, A., & Sanchis, A. (2016). Web news mining in an evolving framework. *Information Fusion*, *28*, 90–98.

Ji, Y., Hong, W., Shangguan, Y., Wang, H., & Ma, J. (2016). Regularized singular value decomposition in news recommendation system. In *Computer science & education (ICCSE), 2016 11th international conference on* (pp. 621–626). IEEE.

Jonnalagedda, N., Gauch, S., Labille, K., & Alfarhood, S. (2016). Incorporating popularity in a personalized news recommender system. *PeerJ Computer Science*, *2*, Article e63.

Karimi, M., Jannach, D., & Jugovac, M. (2018). News recommender systems–Survey and roads ahead. *Information Processing & Management*, *54*(6), 1203–1227.

Kaufman, L., & Rousseeuw, P. J. (2009). *Finding groups in data: An introduction to cluster analysis (Vol. 344)*. John Wiley & Sons.

Kille, B., Lommatzsch, A., Turrin, R., Serény, A., Larson, M., Brodt, T., et al. (2015). Stream-based recommendations: Online and offline evaluation as a service. In *International conference of the cross-language evaluation forum for European languages* (pp. 497–517). Springer.

Larrain, S., Trattner, C., Parra, D., Graells-Garrido, E., & Nørvåg, K. (2015). Good times bad times: A study on recency effects in collaborative filtering for social tagging. In *Proceedings of the 9th ACM conference on recommender systems* (pp. 269–272). ACM.

Lee, O.-J., Hong, M.-S., Jung, J. J., Shin, J., & Kim, P. (2016). Adaptive collaborative filtering based on scalable clustering for big recommender systems. *Acta Polytechnica Hungarica*, *13*(2), 179–194.

Lee, A. M., Lewis, S. C., & Powers, M. (2014). Audience clicks and news placement: A study of time-lagged influence in online journalism. *Communication Research*, *41*(4), 505–530.

Lee, H. J., & Park, S. J. (2007). MONERS: A news recommender for the mobile web. *Expert Systems with Applications*, *32*(1), 143–150.

Li, L., Wang, D., Li, T., Knox, D., & Padmanabhan, B. (2011). SCENE: a scalable two-stage personalized news recommendation system. In *Proceedings of the 34th international ACM SIGIR conference on research and development in information retrieval* (pp. 125–134).

Li, L., Zheng, L., Yang, F., & Li, T. (2014). Modeling and broadening temporal user interest in personalized news recommendation. *Expert Systems with Applications*, *41*(7), 3168–3177.

Lin, C., Xie, R., Guan, X., Li, L., & Li, T. (2014). Personalized news recommendation via implicit social experts. *Information Sciences*, *254*, 1–18.

Lu, M., Qin, Z., Cao, Y., Liu, Z., & Wang, M. (2014). Scalable news recommendation using multi-dimensional similarity and Jaccard–Kmeans clustering. *Journal of Systems and Software*, *95*, 242–251.

Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M., & Hornik, K. (2014). Package 'cluster'. Version 1. (pp. 6–7).

Montgomery, D. C. (2017). *Design and analysis of experiments*. John Wiley & Sons.

Mouselimis, L. (2018). Package 'ClusterR'. Version 1.

Müllner, D., et al. (2013). Fastcluster: Fast hierarchical, agglomerative clustering routines for R and Python. *Journal of Statistical Software*, *53*(9), 1–18.

Murtagh, F., & Legendre, P. (2014). Ward's hierarchical agglomerative clustering method: Which algorithms implement ward's criterion? *Journal of Classification*, *31*(3), 274–295.

Powers, D. M. (2011). *Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation*. Bioinfo Publications.

Qiu, J., Liao, L., & Li, P. (2009). News recommender system based on topic detection and tracking. In *International conference on rough sets and knowledge technology* (pp. 690–697). Springer.

Sadhasivam, G. S., Saranya, K., & Praveen, E. (2014). Personalisation of news recommendation using genetic algorithm. In *2014 3rd international conference on eco-friendly computing and communication systems* (pp. 23–28). IEEE.

Schlesinger, P., & Doyle, G. (2015). From organizational crisis to multi-platform salvation? Creative destruction and the recomposition of news media. *Journalism*, *16*(3), 305–323.

Schröder, G., Thiele, M., & Lehner, W. (2011). Setting goals and choosing metrics for recommender system evaluations. In *UCERSTI2 workshop at the 5th ACM conference on recommender systems, Chicago, USA (Vol. 23)* (p. 53).

Shi, Y., Larson, M., & Hanjalic, A. (2014). Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Computing Surveys*, *47*(1), 3.

Song, Y., Elkahky, A. M., & He, X. (2016). Multi-rate deep learning for temporal recommendation. In *Proceedings of the 39th international ACM SIGIR conference on research and development in information retrieval* (pp. 909–912). ACM.

Sottocornola, G., Symeonidis, P., & Zanker, M. (2018). Session-based news recommendations. In *Companion of the the web conference 2018 on the web conference 2018* (pp. 1395–1399). International World Wide Web Conferences Steering Committee.

Srivastava, A., Bala, P. K., & Kumar, B. (2017). Transfer learning for resolving sparsity problem in recommender systems: human values approach. *JISTEM-Journal of Information Systems and Technology Management*, *14*(3), 323–337.

Sun, Z., Yu, D., Fang, H., Yang, J., Qu, X., Zhang, J., et al. (2020). Are we evaluating rigorously? Benchmarking recommendation for reproducible evaluation and fair comparison. In *Fourteenth ACM conference on recommender systems* (pp. 23–32).

Velmurugan, T. (2012). Efficiency of k-means and k-medoids algorithms for clustering arbitrary data points. *International Journal of Computer Technology & Applications*, *3*(5), 1758–1764.

Velmurugan, T., & Santhanam, T. (2010). Computational complexity between K-means and K-medoids clustering algorithms for normal and uniform distributions of data points. *Journal of Computer Science*, *6*(3), 363.

Wang, C., Duan, Q., Gong, W., Ye, A., Di, Z., & Miao, C. (2014). An evaluation of adaptive surrogate modeling based optimization with two benchmark problems. *Environmental Modelling & Software*, *60*, 167–179.

Wang, B., Gao, Q., Feng, X., & Pan, F. (2017). Recommendation strategy using expanded neighbor collaborative filtering. In *Control conference (CCC), 2017 36th Chinese* (pp. 1451–1455). IEEE.

Wei, J., He, J., Chen, K., Zhou, Y., & Tang, Z. (2017). Collaborative filtering and deep learning based recommendation system for cold start items. *Expert Systems with Applications*, *69*, 29–39.

Wei, Z., Xun, J., & Wang, X. (2009). One-class classification based finance news story recommendation. *Journal of Computational Information Systems5*, *6*, 1625–1631.

Welbers, K., Van Atteveldt, W., Kleinnijenhuis, J., Ruigrok, N., & Schaper, J. (2016). News selection criteria in the digital age: Professional norms versus online audience metrics. *Journalism*, *17*(8), 1037–1053.

Wu, Y., Ding, Y., Wang, X., & Xu, J. (2010). Topic based automatic news recommendation using topic model and affinity propagation. In *2010 international conference on machine learning and cybernetics (Vol. 3)* (pp. 1299–1304). IEEE.

Zahra, S., Ghazanfar, M. A., Khalid, A., Azam, M. A., Naeem, U., & Prugel-Bennett, A. (2015). Novel centroid selection approaches for k means-clustering based recommender systems. *Information Sciences*, *320*, 156–189.

Zheng, L., Li, L., Hong, W., & Li, T. (2013). PENETRATE: Personalized news recommendation using ensemble hierarchical clustering. *Expert Systems with Applications*, *40*(6), 2127–2136.