Technical Report manuscript No.
(will be inserted by the editor)

Optimizing Novelty and Diversity in Recommendations

Jorge Díez · David Martínez-Rego · Amparo Alonso-Betanzos · Oscar Luaces · Antonio Bahamonde

Received: date / Accepted: date

Abstract The articles in the long tail are those that are not popular in some sense, but all together often represent a large proportion of the products covered by a Recommender System. For companies, it is important to recommend these items that otherwise could be unknown to their customers. It is also interesting for users because knowing about these items might constitute a pleasant surprise. But long tail items are not the only we might wish to recommend. Thus, some companies promote products on seasonal offers. It is a challenge to manage the preferences on items whose interaction with users is scarce. There is a trade-off between recommending items that users like and those belonging to a certain kind. We present a framework to address recommendations where the items will have a weight that quantifies our interest in recommending them in a broad sense. Then we derive a factorization method that optimizes the award of the recommendations. To test the method, we present an exhaustive experimentation with a real-world dataset on digital news. We show that it is possible to improve dramatically the novelty (those items of special interest) and diversity of items with a tiny penalization in the accuracy.

Keywords Recommender Systems · Novelty · Diversity · Matrix Factorization · Probabilistic approach to preferences · Trade-off optimization.

This work was funded by grants TIN2015-65069-C2-1-R and TIN2015-65069-C2-2-R from Ministerio de Economía y Competitividad. We also would like to thank the newspaper El País for providing us with the dataset used in this paper.

J. Díez, O. Luaces and A. Bahamonde
Universidad de Oviedo, Artificial Intelligence Center, Gijón, Asturias, Spain
E-mail: {jdiez,oluaces,abahamonde}@uniovi.es

D. Martínez-Rego and A. Alonso-Betanzos
Laboratory for Research and Development in Artificial Intelligence (LIDIA) , University of A Coruña, 15071 A Coruña, Spain
E-mail: {dmartinez,ciamparo}@udc.es

1 Introduction

Long tails in marketing were highlighted by Anderson [2] in reference to those items that are not very popular but represent an important portion of the set of goods or services provided by a company.

In Recommender Systems (RS), long tail items are a challenge since there is a large number of such items but only a few data about their compatibility with users. The consequence is that RS that suggest long tail items have a real risk of decreasing their accuracy. Therefore, RS need to assume a trade-off between accuracy and the possibility of suggesting not-so-popular items that, however, may be useful for users that probably do not even know about their existence; see [4, 11, 27, 29, 22, 1].

The phenomenon of the long tail is quite stunning. In [7] the authors report that the 33% of all ratings are typically concentrated in a very small proportion of items. In Netflix dataset, this select group is only 1.7% of movies (302 items). In Movilens dataset only 213 movies (5.5%) include a third of all rates. Therefore, long tail items are really a lot of items.

The literature contains a number of proposals on how to include long tail items in the recommendations, [29, 31, 26, 25, 24]. Sometimes, we find different names to mean roughly the same idea; thus, *diversity*, *serendipity* or *novelty* are terms used when RS attempt to include the not so common items in the recommendation lists. The overall motivation is that recommenders should

try to improve the *non-obviousness* of their suggestions, and then enhance user satisfaction [30].

However, the phenomenon of the long tail is only a case of a set of items that we might want to appear in the recommendations. For instance, the motivation may be to promote products that are on seasonal offer. In this article, we present a framework that is sufficiently general to be able to address recommendations where the items will have a weight that quantifies our interest in recommending them in a broad sense. So, we define an award for recommendations that take into account items' weights. The idea is that a recommendation of an item that a user likes is weighted by a measure of its interest. Then we derive a factorization method that optimizes the award of the recommendations.

To illustrate the proposal we use a dataset of digital news. In this context, the goal is to improve the reading experience of users in order to increase their engagement. By showing useful and surprising recommendations, readers remain connected for longer times and thus increase chances for cross-selling of advertising and other related products. Capturing traffic is an explicit aim of the recommenders devised by Google [16] or Yahoo! [14] news aggregators, and has been less exploited by more traditional online papers, relying these on general recommendations for all readers.

The next section reviews some related work. The formal framework to deal with long tail items and the learning algorithm is then described. Next, we introduce the representation of readers and news used in the report of results obtained in a number of experiments detailed in the last section. There, we use a real-world data from *El País*[1], Spain's most popular newspaper and probably the most influential in the global Spanish-speaking community.

2 Related Work

Long tail recommendations have been tackled in a number of approaches. In [23], the authors propose a *clustering tail* method. Long tail items are clustered in order to obtain groups with more ratings to be handled by a standard classifier instead of plain items. Then, in [22] the method is refined to become an *adaptive* release where clusters are algorithmically devised.

On the other hand, in [1], a method to improve the *diversity* in a ranking of items is proposed. The standard ranking is combined (reranked) with alternative item ranking functions, such as item popularity. The aim is to find a heuristic way to improve the aggregate diversity of recommendations while maintaining adequate accuracy.

There are other kinds of approaches that formulate an optimization objective that considers the compatibility of users and items and additionally the diversity (or novelty or serendipity) of the suggestions. This is the case of [10] that analyzes weighted objective functions that allow the trade-off between the diversity of the affinity of items and users. The paper presents a control parameter allowing explicit tuning of this trade-off. The proposal needs to solve a binary quadratic programming problem with linear constraints that may use heuristic methods.

Another way to face the optimization trade-off is using a multi-objective approach, as in [27]. To optimize accuracy and the presence of long tail items, the authors propose an evolutionary algorithm that aims to find a set of solutions by optimizing two objective functions simultaneously. The final selection of items to be recommended should be taken from a set of Pareto dominance solutions.

A different approach can be found in [28]. The interaction between users and items is represented by an undirected edge-weighted graph. The recommendations are obtained by algorithms based on the Hitting and Absorbing Time to enhance the long tail items in the list of items suggested to users.

In this paper, we illustrate the general purpose framework, presented in the next section, using a digital news recommender. In this field, there are many related works that should be mentioned. In [16] the authors described a personalized news recommendation system based on profiles learned from registered users' activity in Google News. Their proposed approach is a hybrid between collaborative filters and content recommenders. The recommender described in [16] uses the recommender developed by [8]. This hybrid method has been reported to improve the quality of news recommendations and increase traffic.

Combinations of content-based and collaborative filters have also been used to make personalized digital news suggestions, for instance in [5, 15, 20].

We present a content-free recommendation. However, the inclusion of any information available about news items or readers it would be straightforward. Thus, this is not essential in the framework proposed here.

As mentioned above, we used matrix factorization, which has been successfully applied in other recommender systems - the paradigmatic case being the winner of the Netflix Prize [13, 12]. The overall idea is to embed reading trajectories and news in a common Euclidean space and then use metric properties to represent affinities. Embedding has also been used to suggest

[1] http://www.elpais.com

music playlists for the recommendation in [21,6]. In this case, a probabilistic perspective was adopted.

A quite preliminary work with the dataset used here was presented in our paper [9], where we focused on analyzing the feasibility of our matrix factorization approach to improve the recommendations in terms of precision.

3 General Framework

Let us consider a set $\mathcal{U}$ of users and a set $\mathcal{I}$ of items of some kind. We consider a recommendation framework whose items are classified in two groups for each user: those that the user likes (represented by +1) and those that she/he does not like (respectively, -1). Therefore, we have a dataset

$$\mathcal{D} = \{(\boldsymbol{u}, \boldsymbol{i}, z) : \boldsymbol{u} \in \mathcal{U}, \boldsymbol{i} \in \mathcal{I}, z \in \{+1, -1\}\}. \tag{1}$$

We will try to learn a model from $\mathcal{D}$ in order to predict items that the user will like, if she/he knows about them in the future. To describe an algorithm to learn recommendations for each user $\boldsymbol{u}$, we aggregate the *positive* (respectively, *negative*) items in

$$\boldsymbol{u}^+ = \{\boldsymbol{i} : (\boldsymbol{u}, \boldsymbol{i}, +1) \in \mathcal{D}\},$$

$$\boldsymbol{u}^- = \{\boldsymbol{i} : (\boldsymbol{u}, \boldsymbol{i}, -1) \in \mathcal{D}\}.$$

The union of those items are the *items rated* by the user,

$$\text{items}(\boldsymbol{u}) = \boldsymbol{u}^+ \cup \boldsymbol{u}^- = \{\boldsymbol{i} : (\boldsymbol{u}, \boldsymbol{i}, z) \in \mathcal{D}\}. \tag{2}$$

The dual concept for items is the set

$$\text{users}(\boldsymbol{i}) = \{\boldsymbol{u} : (\boldsymbol{u}, \boldsymbol{i}, z) \in \mathcal{D}\}. \tag{3}$$

In this context a Recommender Systems (RS) is a function R that depends on a parameter θ that maps users into nonempty subsets of items,

$$R(\boldsymbol{u}, \boldsymbol{\theta}) \subset \mathcal{I}.$$

Additionally, let us assume that for each item $\boldsymbol{i}$ or each pair user-item $(\boldsymbol{u}, \boldsymbol{i}) \in \mathcal{D}$ we have a *weight* that establishes the *interest* in recommending it

$$\text{weight}(\boldsymbol{i}, \boldsymbol{u}) \in \mathbb{R}. \tag{4}$$

4 Evaluation of an RS Pretending to Suggest Interesting and Relevant Items

To measure the performance of $\mathrm{R}(\cdot, \boldsymbol{\theta})$, we consider a timeline point of view. Thus, the training dataset $\mathcal{D}$ will include all the items assessed by the user up to a given moment t, those that the user liked and those that did not.

On the other hand, we will use as test set $\mathcal{T}$. Their elements are pairs $(\boldsymbol{u}, \widehat{\boldsymbol{i}})$ where $\widehat{\boldsymbol{i}}$ is the set of those positive (liked) items assessed after t, pretending that the user is still unaware of them. In other words, $\mathcal{T}$ will have all pairs $(\boldsymbol{u}, \widehat{\boldsymbol{i}})$, where $\widehat{\boldsymbol{i}}$ is the set of items that the user $\boldsymbol{u}$ is going to like.

In the general framework introduced in the previous section, we must consider both the relevancy of items suggested (accuracy) and their *interest* (given by the weight (4)). Thus, we define a measure that awards these recommendations,

$$\text{reward}(\mathrm{R}(\boldsymbol{u}, \boldsymbol{\theta}), \text{weight}) = \max_{\boldsymbol{i} \in \mathrm{R}(\boldsymbol{u}, \boldsymbol{\theta})} \{\text{weight}(\boldsymbol{i}, \boldsymbol{u}) : (\boldsymbol{u}, \widehat{\boldsymbol{i}}) \in \mathcal{T}, \boldsymbol{i} \in \widehat{\boldsymbol{i}}\}. \tag{5}$$

Let us remark that we use the maximum instead of the average value. The reason is that the size of the set $\widehat{\boldsymbol{i}}$ of items that $\boldsymbol{u}$ likes in $\mathcal{T}$ is not constant, while $\mathrm{R}(\boldsymbol{u}, \boldsymbol{\theta})$ has a constant size. In the experiments reported at the end of paper, we will always have 5 recommendations for each user. The intended idea is to measure the maximum weight of the recommended items that she or he likes.

When there is not any special interest in items, the weight is constant and the reward is just recording when there are items in the recommendation that the user likes. We will refer to this measure as the accuracy. In symbols, if $\mathbf{1}$ is a constant weight (4), for instance that returns always one,

$$\text{accuracy}(\mathrm{R}(\boldsymbol{u}, \boldsymbol{\theta})) = \text{reward}(\mathrm{R}(\boldsymbol{u}, \boldsymbol{\theta}), \mathbf{1}). \tag{6}$$

To evaluate the interest of recommendations throughout the test set $\mathcal{T}$, we consider the average rewards

$$\text{avg. reward}(\mathrm{R}(\cdot, \boldsymbol{\theta}), \text{weight}) = \frac{\sum_{\mathcal{T}} \text{reward}(\mathrm{R}(\boldsymbol{u}, \boldsymbol{\theta}), \text{weight})}{|\mathcal{T}|}. \tag{7}$$

On the other hand, to measure the performance of R with respect to the *diversity* of the items recommended for all users, we may use the *aggregate* diversity of [1]

$$\text{AggDiv}(R(\cdot, \boldsymbol{\theta})) = \left| \cup \left(R(\boldsymbol{u}, \boldsymbol{\theta}) : \boldsymbol{u} \in \mathcal{T} \right) \right|. \tag{8}$$

The idea is that the greater the diversity, the more likely it is that the RS is recommending items from the *long tail*. In any case, the recommendations are more varied and that is a positive quality of the RS.

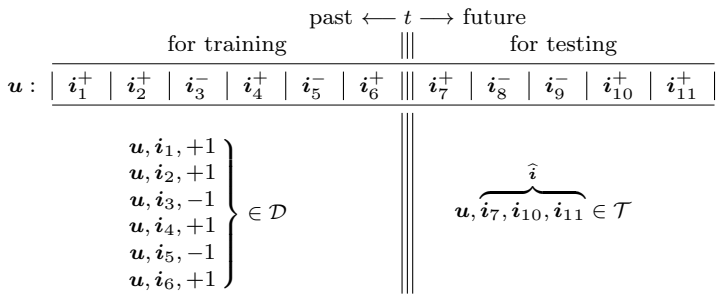

Figure 1 Time line representation of items that a user $\boldsymbol{u}$ likes ($\boldsymbol{i}^+$) and does not like ($\boldsymbol{i}^-$). All items evaluated in the past are recorded in the training set $\mathcal{D}$, and the collection of items that the user is going to like in the future are gathered in the test set $\mathcal{T}$)

5 When the Interest of Items is their Novelty

The approach presented above is quite general, as we only have a vague definition of *interest* by means of the weight of items.

In this section, we follow [3], where Castells et al. present a number of ways to define the *novelty*, a precise and standard method to formalize the interest of items. However, we may also use different ways of dealing with the interest of items, for example, according to seasonal offers of a company.

First, from a global point of view, the novelty of items may be defined as the opposite of the popularity. An item will be novel if a few users have interacted with it. To formalize these concepts, the usual definition of popularity is

$$\text{popularity}(\boldsymbol{i}) = \frac{|\,\text{users}(\boldsymbol{i})|}{|\mathcal{U}|}. \tag{9}$$

Then the negative log is the novelty

$$\text{novelty}(\boldsymbol{i}) = -\log \text{popularity}(\boldsymbol{i}). \tag{10}$$

However, the novelty of an item may be defined for each user. In fact, personalized recommendations should take this point of view. In this case, the novelty can be understood as *unexpectedness*, and the definition of the novelty of an item $\boldsymbol{i}$ must consider the set of items that a user $\boldsymbol{u}$ has rated and the *distance* from them to $\boldsymbol{i}$. In symbols,

$$\text{expect}(\boldsymbol{i}, \boldsymbol{u}) = \frac{1}{|\,\text{items}(\boldsymbol{u})|} \sum_{\boldsymbol{j} \in \text{items}(\boldsymbol{u})} \frac{|\,\text{users}(\boldsymbol{i}) \cap \text{users}(\boldsymbol{j})|}{|\,\text{users}(\boldsymbol{j})|}. \tag{11}$$

Then, the novelty of item $\boldsymbol{i}$ for a user $\boldsymbol{u}$ is defined by

$$\text{novelty}(\boldsymbol{i}, \boldsymbol{u}) = -\log \text{expect}(\boldsymbol{i}, \boldsymbol{u}). \tag{12}$$

6 How to Recommend Interesting Items using Multitask Logistic Regression

To derive a recommender aiming to optimize the reward (5) we propose a probabilistic approach to estimate from $\mathcal{D}$ a distribution for modeling the links of users and items. We chose the logistic function:

$$\Pr(z|\boldsymbol{u}, \boldsymbol{i}, \boldsymbol{\theta}) = \sigma(z \cdot g(\boldsymbol{u}, \boldsymbol{i}, \boldsymbol{\theta})),$$
$$\sigma(x) = \frac{1}{1 + \exp(-x)}. \tag{13}$$

Here $\boldsymbol{\theta}$ is a parameter to be found using a *Maximum a Posteriori Probability (MAP)*.

The function g is a *compatibility* or *scoring* relation between users and items. It is defined by the following inner product:

$$\text{g}(\boldsymbol{u}, \boldsymbol{i}, \boldsymbol{W}, \boldsymbol{A}) = \langle \boldsymbol{W}\boldsymbol{u}, \boldsymbol{A}\boldsymbol{i} \rangle. \tag{14}$$

Thus, the parameter $\boldsymbol{\theta}$, introduced in Section 3, is now the pair of matrices $\boldsymbol{W}$ and $\boldsymbol{A}$, which can be seen from a geometrical point of view as linear embedding projections of users and items in a common Euclidean space $\mathbb{R}^k$.

$$\mathbb{R}^{|\mathcal{U}|} \to \mathbb{R}^k, \quad \boldsymbol{u} \mapsto \boldsymbol{W}\boldsymbol{u}$$
$$\mathbb{R}^{|\mathcal{I}|} \to \mathbb{R}^k, \quad \boldsymbol{i} \mapsto \boldsymbol{A}\boldsymbol{i}. \tag{15}$$

Therefore, g is the dot product of the projections of users and items. In other words, we use a *matrix factorization* approach. It might be argued that this kind of factorization limits the capacity of the model since we restrict ourselves to a subspace of rank k. However, in practice, this is not a limitation for learning an accurate model. In fact, this restriction helps to filter noise and may even be considered to express users and items in terms of a set of *latent variables*, just as latent topic models do for text analysis.

It is important to realize that the projection of items given by matrix $\boldsymbol{A}$ is the same for all users. In this way,

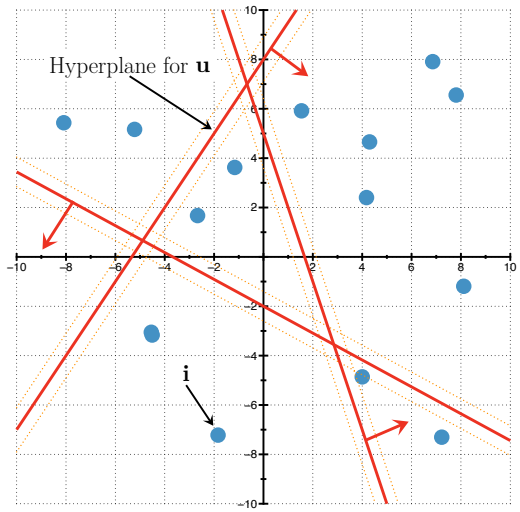

Figure 2 Geometrical representation of users and items in $\mathbb{R}^2$. Points are the projections of items, and straight lines represent users (red arrows indicate the positive region, which contains the relevant items for the user)

all classification tasks share knowledge in order to help one to another.

To enrich to expressiveness of equation (14), we add an extra 1 at the end of the vectors that represent items:

$$\boldsymbol{i}^{\mathrm{T}} \leftarrow [\boldsymbol{i}^{\mathrm{T}} 1].$$

Hence, the function g is a weighted sum of the products of the components of users and items:

$$\begin{aligned} \mathrm{g}(\boldsymbol{u}, \boldsymbol{i}, \boldsymbol{W}, \boldsymbol{A}) &= \langle \boldsymbol{W}\boldsymbol{u}, \boldsymbol{A}\boldsymbol{i} \rangle = \boldsymbol{u}^{\mathrm{T}} \boldsymbol{W}^{\mathrm{T}} \boldsymbol{A} \boldsymbol{i} \\ &= \sum_{b=1}^{|\mathcal{I}|} \sum_{a=1}^{|\mathcal{U}|} \boldsymbol{u}_a \alpha_{a,b} \boldsymbol{i}_b + \sum_{a=1}^{|\mathcal{U}|} \alpha_{a,s} \boldsymbol{u}_a, \end{aligned} \tag{16}$$

where $s = |\mathcal{I}| + 1$.

Geometrically, users can be seen as separator hyperplanes in $\mathbb{R}^k$ for items projected by $\boldsymbol{A}$, see Figure 2. According to (16), the intercept term depends only on the user.

Therefore, to have an RS, we need to minimize

$$\begin{aligned} -\log \prod_{(\boldsymbol{u},\boldsymbol{i},z)} \Pr(z|\boldsymbol{i}, \boldsymbol{u}, \boldsymbol{W}, \boldsymbol{A})) \\ + \nu\big(\mathrm{reg}(\boldsymbol{W}) + \mathrm{reg}(\boldsymbol{A})\big). \end{aligned} \tag{17}$$

Here, the *regularization* terms are computed by

$$\mathrm{reg}(\boldsymbol{X}) = \frac{\|\boldsymbol{X}\|^2}{\sqrt{\dim(X)}},$$

where dim is number of elements of a matrix, its dimension. The hyperparameter ν is the *regularization rate*.

Algorithm 1 To learn the parameters $\boldsymbol{\theta}$ of the pdf (13)

Input: $\mathcal{D}$; {see (1)}
Input: $\beta > 0$; {*stress factor*, see (18)}
assign random values to parameters $\boldsymbol{\theta}$;
repeat
  Fetch random $\boldsymbol{u}$;
  Fetch a positive item $\boldsymbol{i} \in \boldsymbol{u}^+$;
  {from a distribution proportional to weight$^\beta$(18)}
  Update $\boldsymbol{\theta}$ according to $(\boldsymbol{u}, \boldsymbol{i}, +1)$;
  Fetch a random *negative* item $\boldsymbol{neg} \in \boldsymbol{u}^-$;
  Update $\boldsymbol{\theta}$ according to $(\boldsymbol{u}, \boldsymbol{neg}, -1)$;
until *stop criterion*
return $\boldsymbol{\theta}$

To solve the optimization problem (17) we may use a Stochastic Gradient Descent (SGD) algorithm. Nevertheless, we want to favor somehow the recommendations of interesting items. And this has not been included so far. Our proposal is the use the SGD with a sampling method that explicitly considers the definition of weight. See Algorithm 1.

The core point is that the algorithm boosts the *positive* examples ($z = +1$) of users and items according to their weight when they are not constant. Moreover, since these scores are positive values, we may *stress* them using a number $\beta > 0$ to consider

$$\big(\mathrm{weight}(\boldsymbol{i}, \boldsymbol{u})\big)^{\beta}. \tag{18}$$

For greater values of β we expect to obtain recommendations with more reward (5) while suffering some decrease in accuracy (6).

Of course, when $\beta = 0$, all items have the same weight, 1, and there is no special interest in suggesting some items instead of others. Then, the reward (5) is just the *accuracy*.

In [17], the authors use a similar equation. They present a pair-wise approach where users prefer some items over others. In order to boost preferences over the weight of items, the authors define a Serendipitous AUC (SAUC) to be optimized.

However, as it happens in our case (5), the weight has no explicit relation with the parameters of the model. Thus, it is not straightforward to optimize this kind of equations. Our proposal in Algorithm 1 is to consider that the optimization should include somehow the weight of items in the iterative algorithm to find the best model to provide recommendations.

7 Experimental Results

We carried out a set of experiments to show the performance of the approach presented above. In this section, we start introducing the datasets mentioned in the introduction and how we represented readers and news

items. Next, we report the hyper-parameter values of the experimental setting. Then we present the results obtained for different values of the stress factor, β (18), in order to test the cost in terms of accuracy when increasing the novelty in the recommendations.

7.1 Datasets

The datasets used in this paper come from the access logs of the digital version of the newspaper *El País* on September 8th, 2012. This newspaper exceeded 13 million unique users in December 2013; most of them from Spain and Latin America, since it is a newspaper written in Spanish. For this experiment we worked only with accesses from Spain between 00:00 and 23:59, discarding those accesses to pages like 'services', 'widgets', 'comments'; that is, we considered only pages containing news.

The log file was fragmented in several train/test datasets as follows: every half hour, starting at 00:00, we collected the reading data of the next 5 hours, using the first 4 to build a train set and the fifth for the corresponding test set. Table 1 shows the details of each dataset[2].

Each web page access is associated in the log file to its URL and a user identifier, allowing us to construct the trajectory of reading news for each user (Section 7.2). We used only trajectories with at least 4 news for training purposes.

The box plot of Figure 3 depicts some information about the distribution of the trajectory lengths in the 39 datasets, showing that only a small fraction of accesses have a rich trace to be used for training. Only an average of approximately 838 reading trajectories have at least 4 accesses, even though the total number of accesses to the digital newspaper is several orders of magnitude higher. The amount of trajectories dramatically decreases as their length increases, down to an average of only 1.7 trajectories with 15 news.

Figure 4 illustrates the long tail phenomenon in these datasets. The graph represents the percentage of accesses in the vertical axis for each news article in each dataset. News are represented in decreasing order of accesses, being the i-th news in the horizontal axis the i-th most accessed article in its corresponding time frame.

[2] The dataset used in the experiments can be downloaded at http://dx.doi.org/10.13140/RG.2.2.31466.21445

Table 1 Datasets built from the access log file. The column *Inst* shows the number of instances in each dataset, and the column *News* indicates the number of different article news accessed in the corresponding time frame.

	Train			Test	
Set	Time frame	Inst.	News	Time frame	Inst
D1	00:00 - 03:59	6850	208	04:00 - 04:59	429
D2	00:30 - 04:29	5705	204	04:30 - 05:29	399
D3	01:00 - 04:59	4601	199	05:00 - 05:59	460
D4	01:30 - 05:29	3723	191	05:30 - 06:29	625
D5	02:00 - 05:59	3195	184	06:00 - 06:59	933
D6	02:30 - 06:29	3012	197	06:30 - 07:29	1591
D7	03:00 - 06:59	3093	193	07:00 - 07:59	2522
D8	03:30 - 07:29	3732	205	07:30 - 08:29	4133
D9	04:00 - 07:59	5091	215	08:00 - 08:59	6488
D10	04:30 - 08:29	7683	217	08:30 - 09:29	8901
D11	05:00 - 08:59	11946	225	09:00 - 09:59	10403
D12	05:30 - 09:29	17555	233	09:30 - 10:29	11777
D13	06:00 - 09:59	23941	238	10:00 - 10:59	12233
D14	06:30 - 10:29	31047	246	10:30 - 11:29	11817
D15	07:00 - 10:59	37890	258	11:00 - 11:59	10703
D16	07:30 - 11:29	44384	275	11:30 - 12:29	9605
D17	08:00 - 11:59	48990	279	12:00 - 12:59	8851
D18	08:30 - 12:29	51972	288	12:30 - 13:29	7922
D19	09:00 - 12:59	52526	289	13:00 - 13:59	7102
D20	09:30 - 13:29	51472	292	13:30 - 14:29	5675
D21	10:00 - 13:59	48875	293	14:00 - 14:59	4176
D22	10:30 - 14:29	44272	297	14:30 - 15:29	3394
D23	11:00 - 14:59	38879	292	15:00 - 15:59	3132
D24	11:30 - 15:29	33395	291	15:30 - 16:29	3137
D25	12:00 - 15:59	29148	291	16:00 - 16:59	3201
D26	12:30 - 16:29	25268	288	16:30 - 17:29	3338
D27	13:00 - 16:59	21908	285	17:00 - 17:59	3117
D28	13:30 - 17:29	19278	287	17:30 - 18:29	3688
D29	14:00 - 17:59	17359	287	18:00 - 18:59	4385
D30	14:30 - 18:29	17318	291	18:30 - 19:29	4592
D31	15:00 - 18:59	18190	289	19:00 - 19:59	4229
D32	15:30 - 19:29	19108	294	19:30 - 20:29	3758
D33	16:00 - 19:59	19609	294	20:00 - 20:59	3476
D34	16:30 - 20:29	19937	300	20:30 - 21:29	3169
D35	17:00 - 20:59	20075	298	21:00 - 21:59	2807
D36	17:30 - 21:29	19852	302	21:30 - 22:29	2642
D37	18:00 - 21:59	19554	310	22:00 - 22:59	2660
D38	18:30 - 22:29	18070	316	22:30 - 23:29	2680
D39	19:00 - 22:59	16785	318	23:00 - 23:59	2585

7.2 Representation of Readers and News

To fit this dataset in the framework presented in this paper, users are readers, and items are news. There are several peculiarities that make news recommendation challenging and should be taken into account. Since the recommendations should be personalized, we need to capture as much information as possible from readers. However, most of the readers are frequently not identified in the website, and thus their specific profile is not known in advance and a detailed history is not available. The only certainly available information from a reader is the trajectory of the news that deserved reader's attention in the current session, and these sessions are known to be very short in practice.

Thus, the core issue is the representation of a single news item. For this purpose, let us consider a set N of digital news. Each $p \in N$ is going to be represented using the *one-hot* encoding, that is, by a binary

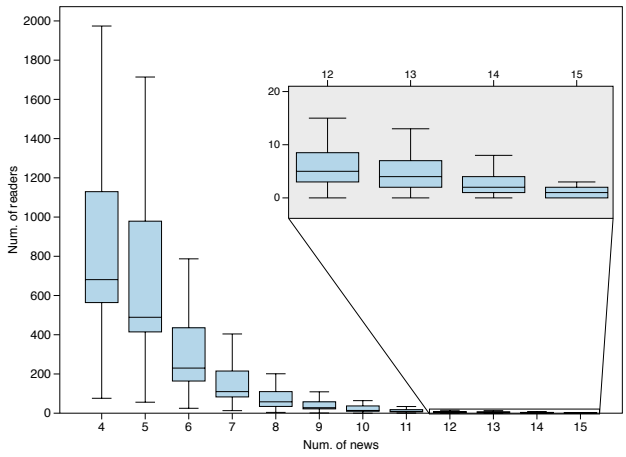

Figure 3 Characteristics of the datasets used. The vertical axis represents the amount of readers who read the number of news in the horizontal axis.

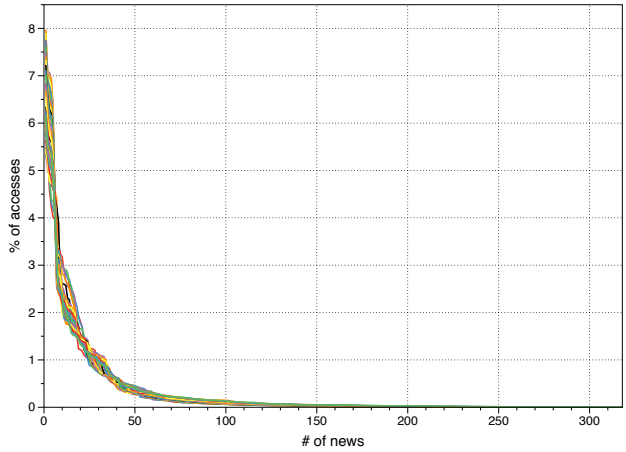

Figure 4 Long tails in all datasets. The vertical axis represents the percentage of reader who like the news represented (sorted) in the horizontal axis.

codification vector

$$\boldsymbol{p} \in \mathbb{R}^{|N|} \tag{19}$$

where all components are zero except the one with index p which is 1. We will use interchangeably in the rest of the paper, $\boldsymbol{p}$ for a news item and its vector representation.

On the other hand, to represent a reader we use only the trajectory of the news read in one session. In this paper we use the concatenation of the binary representation (one-hot) of the last ($\boldsymbol{r}^0$) and last but one ($\boldsymbol{r}^1$) articles (2-gram):

$$\widehat{\boldsymbol{r}} = [(\boldsymbol{r}^0)^T, (\boldsymbol{r}^1)^T]^T \in \mathbb{R}^{2\times|N|}. \tag{20}$$

Although other representations for readers could be used, we have experimentally checked the soundness of 2-gram for this application. Additionally, in the implementation used in the experiments, we normalized the vectors representing readers:

$$\widehat{\boldsymbol{r}} \leftarrow \frac{\widehat{\boldsymbol{r}}}{\left\|\widehat{\boldsymbol{r}}\right\|}.$$

Finally, let us remark that if we have any extra information about readers or news, we just have to concatenate the vectorial representation described above with a vectorial representation of extra knowledge. This idea has been successfully used in [19,18]. For instance, we could have some valuable information about readers, like sex, age, previous interactions with the digital newspaper, etc... On the other hand, the news could have been described by using their contents.

Table 2 Values of the SGD hyper-parameters used in the experiments

Parameter	Equat.	Value
Learning rate		in $\{10^{-2}, 10^{-1}\}$
Regularization	(17)	$\nu \in \{10^{-5}, 10^{-2}\}$
Radius	(21)	$R = 10$
$\mathbb{R}^k$ dimension	(15)	$k = 50$
Max. iterations		$100 \times$ #train. inst.

7.3 Experimental Hyper-Parameters

For the sake of reproducibility, Table 2 shows the values of the meta-parameters used in the training stage. The learning rate (γ) and the regularization factor (ν) were chosen automatically for each dataset by applying a grid search procedure in the range of values specified in the table. Also, after each iteration of the SGD algorithm, the norms of the columns of matrices $\boldsymbol{W}$ and $\boldsymbol{A}$ were constrained to be kept inside a hypersphere of radius R:

$$\left\|\boldsymbol{W}(\cdot,i)\right\| \leq R, \quad \left\|\boldsymbol{A}(\cdot,j)\right\| \leq R, \quad \forall i,j. \tag{21}$$

7.4 Results and discussion

The purpose of these experiments is to show the influence of β in the optimization problem (17). We compare the average values (7) of accuracy and novelty. While the accuracy defined in (6), we have two definitions for novelty: depending only of the item (10) and considering the user (12). The models were obtained by the

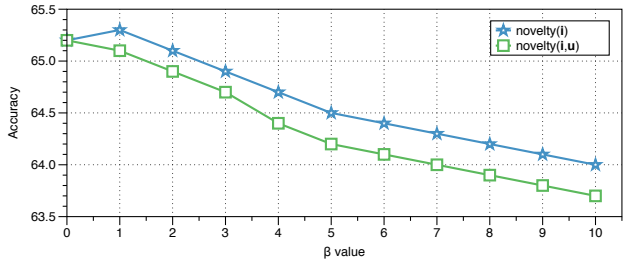

Figure 5 Weighted average (using the datasets of Table 1) of accuracy obtained for different values of β (18) represented in the horizontal axis

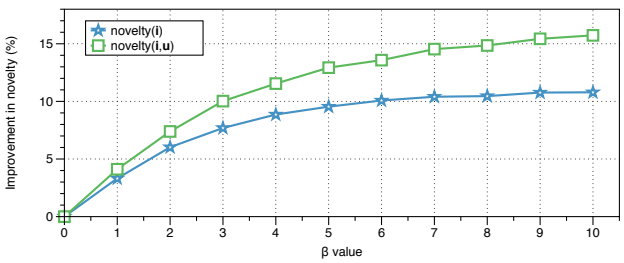

Figure 6 Weighted average (using the datasets of Table 1) of novelty (measured using (10) and (12)) obtained for different values of β (18) represented in the horizontal axis. The vertical axis represent the increase in percentage points assuming 0 for $\beta = 0$

Algorithm 1 for different values of β in the range $[0, 10]$ when weight is respectively

$$\text{weight}_i(\boldsymbol{i}, \boldsymbol{u}) = \text{novelty}(\boldsymbol{i}), \tag{22}$$

$$\text{weight}_{i,u}(\boldsymbol{i}, \boldsymbol{u}) = \text{novelty}(\boldsymbol{i}, \boldsymbol{u}). \tag{23}$$

The novelty was assessed using the definition that each model is trying to optimize.

Figure 5 depicts the effect of variations in the stress factor β with respect to the average accuracy. Figure 6 shows the average novelties, in this case, to compare the grows with β, we represent the increase in percentage points with respect to the novelty obtained when $\beta = 0$, that is when the aim is to optimize the accuracy instead of any kind of novelty. Each point in the graphs represents the weighted average scores of the results obtained in the 39 datasets.

The cost of increasing the novelty in terms of accuracy is reflected in lower accuracy scores. The graphs confirm that we have to pay a price if we want to increase the degree of novelty in our recommendations. Note that in the most extreme case, with $\beta = 10$ and using the novelty function of equation (12), we only lose 1.5 percentage points in accuracy but we win almost 16 percentage points in novelty.

The values in Figure 7 confirm what was observed in Figure 6. An increase in β brings with it a greater diversity of the recommended news. The RS's recommendations risk suggesting news of those that are surely in the long tail and therefore of which there is little evidence about the interest in readers. This risk reduces accuracy but obviously gives rise to greater diversity. In Figure 7 we see that in the extreme value ($\beta = 10$) there are increases in the number of recommended news (aggregate diversity of the equation 8) of 26.7 and 34.7 percentage points according to the definition of novelty that we are trying to optimize.

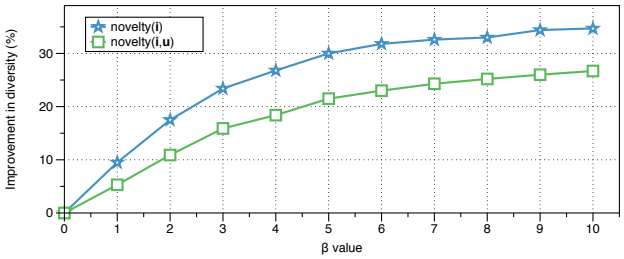

Figure 7 Weighted average (using the datasets of Table 1) of aggregate diversity (8) obtained for different values of β (18) represented in the horizontal axis. The vertical axis represents the increase in percentage points assuming 0 for $\beta = 0$

8 Concluding Remarks

We have presented a general framework to derive recommender systems that have an explicit interest in suggesting a kind of items. The proposal is quite general, but we focused on novelty and diversity. It is acknowledged that the satisfaction of users increases when RS suggest items that surprise users favorably. Frequently, this implies the exploitation of long tail items that may be very profitable for companies. These items represent an important part of the items available in typical datasets, and typically they are not recommended at all since there are just a few pieces of evidence of users who like them.

The use of these novelty items inevitably leads to a trade-off with accuracy. To formalize this issue, we defined an *award* function that takes into account a certain interest (weight) in the recommendations. The interest can be defined as the novelty (in any formulation of this concept), but it can also be the will of companies to promote a kind of items in a seasonal offer. In any case, the factorization method presented in this paper optimizes the award in this general framework.

To illustrate the performance of the proposal we used a real-world dataset from a digital newspaper, El País. We showed that the award function can be optimized to increase dramatically the novelty of the recommendations. Additionally, the recommendations include significantly more news than those suggested when considering a uniform interest. In fact, boosting the reward yielded an increase of $\simeq 16$ percentage points in the number of pieces of news used, paying just a tiny penalization in accuracy.

References

1. Adomavicius, G., Kwon, Y.: Improving aggregate recommendation diversity using ranking-based techniques. IEEE Transactions on Knowledge and Data Engineering **24**(5), 896–911 (2012)
2. Anderson, C.: The long tail: Why the future of business is selling less of more. Hachette Books (2006)
3. Castells, P., Hurley, N.J., Vargas, S.: Novelty and diversity in recommender systems. In: Recommender Systems Handbook, pp. 881–918. Springer (2015)
4. Castells, P., Vargas, S., Wang, J.: Novelty and diversity metrics for recommender systems: choice, discovery and relevance. In: International Workshop on Diversity in Document Retrieval (DDR 2011) at the 33rd European Conference on Information Retrieval (ECIR 2011), pp. 29–36. Citeseer (2011)
5. Chen, C.C., Chen, M.C., Sun, Y.: PVA: A Self-Adaptive Personal View Agent. Journal of Intelligent Information Systems **18**(2-3), 173–194 (2002)
6. Chen, S., Moore, J., Turnbull, D., Joachims, T.: Playlist Prediction Via Metric Embedding. In: Proc. of the 18^{th} ACM SIGKDD, pp. 714–722 (2012)
7. Cremonesi, P., Koren, Y., Turrin, R.: Performance of recommender algorithms on top-n recommendation tasks. In: Proceedings of the fourth ACM conference on Recommender systems, pp. 39–46. ACM (2010)
8. Das, A.S., Datar, M., Garg, A., Rajaram, S.: Google News Personalization: Scalable Online Collaborative Filtering. In: Proc. of the 16th WWW, pp. 271–280. ACM (2007)
9. Díez, J., Martínez-Rego, D., Alonso-Betanzos, A., Luaces, O., Bahamonde, A.: Metrical Representation of Readers and Articles in a Digital Newspaper. In: 10th ACM Conference on Recommender Systems (RecSys). Workshop on profiling user preferences for dynamic online and real-time recommendations (RecProfile) (2016)
10. Hurley, N., Zhang, M.: Novelty and diversity in top-n recommendation–analysis and evaluation. ACM Transactions on Internet Technology (TOIT) **10**(4), 14 (2011)
11. Jambor, T., Wang, J.: Optimizing multiple objectives in collaborative filtering. In: Proceedings of the fourth ACM conference on Recommender systems, pp. 55–62. ACM (2010)
12. Koren, Y.: Collaborative Filtering with Temporal Dynamics. Communications of the ACM **53**(4), 89–97 (2010)
13. Koren, Y., Bell, R., Volinsky, C.: Matrix Factorization Techniques for Recommender Systems. Computer **42**(8), 30 –37 (2009)
14. Li, L., Chu, W., Langford, J., Schapire, R.E.: A Contextual-Bandit Approach to Personalized News Article Recommendation. In: Proceedings of the 19th international conference on World Wide Web, WWW 2010, pp. 661–670. ACM (2010)
15. Lin, C., Xie, R., Guan, X., Li, L., Li, T.: Personalized news recommendation via implicit social experts. Information Sciences **254**(0), 1 – 18 (2014)
16. Liu, J., Dolan, P., Pedersen, E.R.: Personalized News Recommendation Based on Click Behavior. In: Proceedings of the 15th International Conference on Intelligent User Interfaces, IUI '10, pp. 31–40. ACM, New York City, NY, USA (2010)
17. Lu, Q., Chen, T., Zhang, W., Yang, D., Yu, Y.: Serendipitous personalized ranking for top-n recommendation. In: Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology-Volume 01, pp. 258–265. IEEE Computer Society (2012)
18. Luaces, O., Díez, J., Alonso-Betanzos, A., Troncoso, A., Bahamonde, A.: Including content-based methods in peer-assessment of open-response questions. In: IEEE International Conference on Data Mining (ICDM) Workshop on Data Mining for Educational Assessment and Feedback (ASSESS'15) (2015)
19. Luaces, O., Díez, J., Joachims, T., Bahamonde, A.: Mapping preferences into euclidean space. Expert Systems with Applications **42**(22), 8588–8596 (2015)
20. Miranda, T., Claypool, M., Gokhale, A., Mir, T., Murnikov, P., Netes, D., Sartin, M.: Combining Content-Based and Collaborative Filters in an Online Newspaper. In: In Proceedings of ACM SIGIR Workshop on Recommender Systems, 2010 (1999)
21. Moore, J., Chen, S., Joachims, T., Turnbull, D.: Learning to Embed Songs and Tags for Playlist Prediction. In: Proceedings of the International Society for Music Information Retrieval (ISMIR), 2012 (2012)
22. Park, Y.J.: The adaptive clustering method for the long tail problem of recommender systems. IEEE Transactions on Knowledge and Data Engineering **25**(8), 1904–1915 (2013)
23. Park, Y.J., Tuzhilin, A.: The long tail of recommender systems and how to leverage it. In: Proceedings of the 2008 ACM conference on Recommender systems, pp. 11–18. ACM (2008)
24. Shani, G., Gunawardana, A.: Evaluating recommendation systems. In: Recommender systems handbook, pp. 257–297. Springer (2011)
25. Sugiyama, K., Kan, M.Y.: Serendipitous recommendation for scholarly papers considering relations among researchers. In: Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries, pp. 307–310. ACM (2011)
26. Vargas, S., Baltrunas, L., Karatzoglou, A., Castells, P.: Coverage, redundancy and size-awareness in genre diversity for recommender systems. In: Proceedings of the 8th ACM Conference on Recommender Systems, pp. 209–216. ACM (2014)
27. Wang, S., Gong, M., Li, H., Yang, J.: Multi-objective optimization for long tail recommendation. Knowledge-Based Systems **104**, 145–155 (2016)
28. Yin, H., Cui, B., Li, J., Yao, J., Chen, C.: Challenging the long tail recommendation. Proceedings of the VLDB Endowment **5**(9), 896–907 (2012)
29. Zhang, M., Hurley, N.: Avoiding monotony: Improving the diversity of recommendation lists. In: Proceedings of the 2008 ACM Conference on Recommender Systems,

RecSys '08, pp. 123–130. ACM, New York, NY, USA (2008)
30. Ziegler, C.N., McNee, S.M., Konstan, J.A., Lausen, G.: Improving recommendation lists through topic diversification. In: Proceedings of the 14th international conference on World Wide Web, pp. 22–32. ACM (2005)
31. Zuo, Y., Gong, M., Zeng, J., Ma, L., Jiao, L.: Personalized recommendation based on evolutionary multi-objective optimization [research frontier]. Computational Intelligence Magazine, IEEE **10**(1), 52–62 (2015)