Improving Sales Diversity by Recommending Users to Items

Saúl Vargas and Pablo Castells
Universidad Autónoma de Madrid, Spain
saul.vargas@uam.es, pablo.castells@uam.es

ABSTRACT

Sales diversity is considered a key feature of Recommender Systems from a business perspective. Sales diversity is also linked with the long-tail novelty of recommendations, a quality dimension from the user perspective. We explore the inversion of the recommendation task as a means to enhance sales diversity – and indirectly novelty – by selecting which users an item should be recommended to instead of the other way around. We address the inverted task by two approaches: a) inverting the rating matrix, and b) defining a probabilistic reformulation which isolates the popularity component of arbitrary recommendation algorithms. We find that the first approach gives rise to interesting reformulations of nearest-neighbor algorithms, which essentially introduce a new neighbor selection policy. The second approach, as well as the first, ultimately result in substantial sales diversity enhancements, and improved trade-offs with recommendation precision and novelty. Two experiments on movie and music recommendation datasets show the effectiveness of the resulting approach, even when compared to direct optimization approaches of the target metrics proposed in prior work.

Categories and subject descriptors: H3.3 [Information Search & Retrieval]: Information Filtering

Keywords: Recommender Systems; Sales Diversity; Novelty

1. INTRODUCTION

Sales diversity has been pointed out as a relevant quality of recommendation from the business point of view [7]. Sales diversity means that all or most products in the business catalog get purchased to some extent, rather than having sales concentrating around a few items. Sales diversity gets meaning in the context of recommendation in the sense that recommending a product exposes it to being sold. By linking recommendation to purchase in the analysis of diversity, "sales diversity" becomes a shorthand for "promoting sales diversity".

Sales diversity can be measured, for instance, by the total number or the ratio of items that are recommended in the top-N to at least one user [1]. Better yet, we can use the Gini coefficient as a finer measure of the concentration of top N recommendations around a few items [7]. Prior research has found there is an indirect connection between sales diversity and selling (recommending) in the long tail [1]: promoting long-tail (novel) items has a positive effect on sales diversity, even though sales diversity and long-tail novelty are not in themselves the same thing. Several approaches have been proposed in the literature to enhance recommendations over such metrics, most of which consist of re-sorting the top-K items ($K > N$) of an initial ranking produced by a baseline recommendation algorithm [1, 17]. Other authors have also researched the effect of different basic recommendation methods on sales concentration [7].

In the research presented here, we consider a different outlook on the problem of sales diversity. If we aim to procure a fair opportunity for most items to be recommended, one may consider selecting which users each item should be recommended to, instead of the other way around. This view entails a symmetric swap of the recommendation task, whereby users are recommended to items rather than the opposite. From this perspective, we address three main research questions: a) How can we define suitable and effective algorithms that recommend users to items? b) Does the inverted formulation actually enable any improvements in sales diversity? c) If so, what trade-offs if any are involved with respect to other qualities such as precision or recommendation novelty from the user point of view?

To address these questions we propose, firstly, to explore the application of state of the art collaborative filtering algorithms to the inverted recommendation task, that is, simply swapping the role of items and users in the algorithms. We find interesting derivations, equivalences, and new insights on the behavior of neighborhood-based algorithms in particular, where the inversion results in the emergence of new neighbor selection policies, with an impact on the potential connections to item popularity. We furthermore find that the inversion approach results in a significant increase of sales diversity while retaining a good trade-off on top-N item recommendation precision. In addition to this, we develop a probabilistic scheme which formulates user recommendation to items as a Bayesian layer which can be applied on top of any recommendation algorithm. The probabilistic scheme provides a principled means to isolate the item popularity component of the baseline algorithm; by means of simple smoothing techniques, the presence of this popularity component can be calibrated (i.e. kept unchanged or neutralized) to any desired degree. This parametrization is shown to enable an enhanced precision-diversity trade-off,

RecSys'14, October 6–10, 2014, Foster City, Silicon Valley, CA, USA.

ACM 978-1-4503-2668-1/14/10 ...$15.00.
http://dx.doi.org/10.1145/2645710.2645744 .

even above, somewhat surprisingly, direct optimization approaches targeting the precision vs. long-tail novelty tradeoff. Furthermore, the resulting algorithmic scheme is competitive with respect to direct optimization even in terms of long-tail novelty.

The rest of the paper is structured as follows. In Section 2 we review the related work on assessment and enhancement of sales diversity and novelty. Section 3 discusses the popularity bias in collaborative filtering algorithms detected by several authors, and how this bias also influences the evaluation of Recommender Systems in ranking tasks. Section 4 introduces the inverted recommendation task of recommending users to items. Then, Section 5 describes our first proposal of using inverted neighborhoods to improve sales diversity. An analysis of the properties of standard and inverted neighborhoods and their differences is shown in Section 6. Section 7 presents our second proposal, a probabilistic reformulation layer that allows the calibration of the popularity bias in state-of-the-art collaborative filtering algorithms. Experiments with two different recommendation scenarios – movies and music – are described in Section 8. Finally, Section 9 offers the conclusions and future work.

2. SALES DIVERSITY AND NOVELTY

As Anderson [4] stated, some businesses or economic models present a *Long tail* effect, in which a few of the most popular items are extremely popular, while the rest – the long tail – is much less known. Promoting the recommendation of items in this long tail may offer benefits for both users and the business behind the recommender system. From the user side, offering not so popular items may help receive less obvious, unexpected recommendations, which correspond with the natural notion of a recommender system as a tool to help users discover new content. From the system side, avoiding the recommendation of short-head items may also contribute to make the most of the catalog.

Adomavicius et al. [1] proposed to measure sales diversity in terms of the aggregate diversity, i.e., how many different items of the catalog were recommended to the users. Given a recommender system S, they proposed a metric, denoted here as aggr-div, which return the total number of different items that have been recommended to at least one user:

$$\text{aggr-div}(S) = \left| \bigcup_{u \in \mathcal{U}} L_u^S \right| \tag{1}$$

where L_u^S is the recommendation list generated by the system S for the user u. This metric, while being simple and intuitive, may not be very robust, since the contribution to the metric of an item that has been recommended just once is equal to that of other item recommended a thousand times. Therefore, we think that using a metric to measure the imbalance in the number of times each item is recommended would be more adequate. Fleder and Hosanagar [7] proposed a better alternative by using the *Gini coefficient* to measure sales concentration:

$$\text{gini}(S) = 1 - \frac{1}{N-1} \sum_{k=1}^{N} (2k - N - 1) p(i_k \,|\, S) \tag{2}$$

where $p(i_k \,|\, S)$ is the probability of the k-th least recommended item being drawn from the recommendation lists generated by a system S:

$$p(i \,|\, S) = \frac{\sum_u \mathbf{1}_{i \in L_u^S}}{\sum_{u,j} \mathbf{1}_{j \in L_u^S}} \tag{3}$$

Note that in our definition we use the complement of the standard definition for the Gini coefficient so that higher values of our metric correspond to more balanced recommendations.

In previous work [17], we proposed the *expected popularity complement* metric to measure the long-tail novelty of the items in a recommendation list, which is defined as the average novelty of the recommended items:

$$\text{EPC}(L_u) = \frac{1}{|L_u|} \sum_{i \in L_u} nov(i) \tag{4}$$

where $nov(i)$ measures the *novelty* of an item as the probability of not to being known by a user:

$$nov(i) = 1 - \frac{\sum_u \mathbf{1}_{r_{u,i}>0}}{\sum_{u,j} \mathbf{1}_{r_{u,j}>0}} \tag{5}$$

Several proposals to promote sales diversity and novelty in recommender systems have been proposed [1, 11, 15, 17]. A simple approach, which we compare against our proposals, consist in a re-scoring of a previously generated recommendation baseline by using a normalized linear combination between the scores provided by the baseline and the novelty component described in 5:

$$s_{NR}(u,i) = (1-\lambda)\ norm_s\, s(u,i) + \lambda\, norm_{nov}\, nov(i) \tag{6}$$

where $norm_s$ and $norm_{nov}$ are normalizing functions that help making a balanced combination of both relevance and novelty components, such as the *standard score* $norm_X(x) = \frac{x - \mu_X}{\sigma_X}$.

3. THE POPULARITY BIAS IN RECOMMENDATIONS

In a ranking prediction task, the popularity-based recommendation is the obvious baseline to beat [3]. In terms of sales diversity and recommendation novelty, personalized algorithms easily improve over the popularity-based recommendation which, by definition, has the lowest scores in terms of the metrics described in the previous section. Improving the precision of popularity recommendation is less obvious than one might think but is also a feasible goal [6]. While it is clear that collaborative filtering algorithms outperform popularity-based recommendations in terms of accuracy and sales diversity, some authors have pointed out that they still suffer from a bias [18].

Collaborative filtering algorithms, in particular, are known to generally have a bias towards recommending popular items, commonly known as the "Harry Potter Effect"[1]. There is a natural reason for this trend to begin with: collaborative filtering thrives on the populated regions of the user-item interaction history matrix (rating matrix for short), and falls short in the sparser regions. Popular items live by definition in the more populated areas, since they carry more rating data that populates matrix cells, and collaborative algorithms are therefore more prone to end up recommending these items. The popularity bias of collaborative filtering algorithms has been pointed out by several authors and studied by some. For instance, Zhao et al. [18] show empirical evidence that popular items tend to be more recommended than not so popular ones, and proposes methods to alleviate his effect. Steck [15] examined this issue in further depth and justified this popularity bias by the selection bias towards popular items in the available data.

[1] `http://recsyswiki.com/wiki/Harry_Potter_effect`

Furthermore, common precision-based evaluation methodologies reward this behavior, since popular items have more test ratings and are more likely to be counted as hits for more users. This has motivated several evaluation protocols [5, 6, 15] that try to enable a less biased assessment of the personalized relevance of the recommendations, by removing the tip of the bias in the test data.

The popularity bias has a negative effect on the discovery-related added value and practical usefulness of collaborative filtering recommendations, as well as their effect on sales concentration [7]. The research and findings we report here provide means to better cope, directly and indirectly, with this bias, as we discuss in the next sections.

4. RECOMMENDING USERS TO ITEMS: PROBLEM STATEMENT

The recommendation task can be formulated as defining a scoring function $s : \mathcal{U} \times \mathcal{I} \to \mathbb{R}$ for pairs of users $u \in \mathcal{U}$ and items $i \in \mathcal{I}$ so that, for each user, a ranked list of items $L_u \subset \mathcal{I} \times \mathcal{I} \times \ldots$ is defined by sorting items by decreasing score order. The scoring function of a recommender algorithm is based on previous interactions between users and items recorded in a matrix $R = (r_{u,i})_{u,i} \in \mathbb{R}^{|\mathcal{U}|,|\mathcal{I}|}$ and, possibly, additional sources, as in the case of content or social-based recommenders. We shall focus here on collaborative filtering approaches, which use only the interaction matrix R.

Analogously to the original task of recommending items to users, the task of recommending users to items can be formulated as defining a scoring function

$$\tilde{s} : \mathcal{I} \times \mathcal{U} \to \mathbb{R} \tag{7}$$

which induces a ranking of users by their decreasing predicted relevance to item i. In a pure collaborative filtering setting, the input data for this task consists of the transposed rating matrix $\tilde{R} = R^t \in \mathbb{R}^{|\mathcal{I}|,|\mathcal{U}|}$.

Since collaborative filtering algorithms do not depend on the content or internal characteristics of both users and items, they can be adapted for this task without any modification apart from the change of roles between users and items. An initial observation is that, for many popular, state-of-the-art collaborative filtering algorithms, the scoring function $\tilde{s}$ is actually identical to that of the original problem s. That is the case, for instance, of many matrix factorization approaches, such as the implicit matrix factorization of Hu et al. [9]:

$$\tilde{s}_{iMF}(i,u) = q_i\, p_u^t = p_u\, q_i^t = s_{iMF}(u,i) \tag{8}$$

There are other collaborative filtering approaches that break this symmetry. That is the case of other matrix factorization approaches [13, 14, 16] which, even having the same scoring function as in Equation 8, have non-interchangeable roles for users and items in their model training, and thus provide new scoring functions between users and items. However, we focus on the case of nearest neighbors approaches, whose asymmetry offers an interesting, new alternative for generating diverse recommendations.

5. INVERTED NEAREST NEIGHBORS

A first application of the inverted recommendation task lies in the asymmetry of the nearest neighbor approaches when applied to the inverted recommendation task. Throughout this section we will focus on the user-based k-nearest neighbors (kNN) approach, since most of the observations, unless explicitly discussed, are straightforwardly translatable to the item-based alternative.

sim	u_1	u_2	u_3	u_4
u_1	-	0.1	0.3	0.4
u_2	0.1	-	0.5	0.0
u_3	0.3	0.5	-	0.6
u_4	0.4	0.0	0.6	-

u	$N_2(u)$	$N_2^{-1}(u)$
u_1	$\{u_3, u_4\}$	$\{u_2, u_4\}$
u_2	$\{u_1, u_3\}$	$\{u_3\}$
u_3	$\{u_2, u_4\}$	$\{u_1, u_2, u_4\}$
u_4	$\{u_1, u_3\}$	$\{u_1, u_3\}$

Table 1: Example of user neighborhoods of size 2.

The scoring functions of the user-based and item-based kNN recommenders[2, 6] can be formulated as follows :

$$s_{UB}(u,i) = \sum_{v \in \mathcal{U}} \mathbf{1}_{v \in N(u)}\; sim(u,v)\; r_{v,i} \tag{9}$$

$$s_{IB}(u,i) = \sum_{j \in \mathcal{I}_u} \mathbf{1}_{i \in N(j)}\; sim(i,j)\; r_{u,j} \tag{10}$$

where $sim(u,v)$ is a similarity function between two users and $N(u)$ is the neighborhood of user u, containing the top-most similar users to item u.

Reformulating these algorithms in the inverted recommendation task, the scoring functions become:

$$\tilde{s}_{UB}(i,u) = \sum_{j \in \mathcal{I}} \mathbf{1}_{j \in N(i)}\; sim(i,j)\; \tilde{r}_{j,u} \tag{11}$$

$$\tilde{s}_{IB}(i,u) = \sum_{v \in \mathcal{U}_i} \mathbf{1}_{u \in N(v)}\; sim(u,v)\; \tilde{r}_{i,v} \tag{12}$$

As previously commented, the symmetry of the nearest neighbors scoring functions with respect to the original problem is broken. In particular, the user-based approach s_{UB} in Equation 9 is notably different from the scoring function $\tilde{s}_{UB}$ in Equation 11. Interestingly, it is almost equivalent to the item-based approach $\tilde{s}_{IB}$ for the inverted recommendation in Equation 12, the difference lying on the neighborhood selection criterion, i.e., $\mathbf{1}_{u \in N(v)}$ against $\mathbf{1}_{v \in N(u)}$. Actually, one can re-formulate the scoring function $\tilde{s}_{IB}$ as a variant of the standard user-based approach s_{UB} in which the policy for neighbor selection is inverted, that is, by considering user *inverted neighborhoods* $N^{-1}(u)$ defined as

$$N^{-1}(u) = \{v \in \mathcal{U} \;:\; u \in N(v)\} \tag{13}$$

where $N(v)$ is the original neighborhood for a user v, so that $\mathbf{1}_{u \in N(v)} = \mathbf{1}_{v \in N^{-1}(u)}$. The concept of inverted neighborhoods originally appeared in [12], where it was proposed as an ad-hoc method to efficiently predict ratings for item-based approaches, without any relationship with the inverted recommendation tasks or the improvement of sales diversity.

Note what the resulting inverted neighborhood formation policy means: instead of selecting $N(u)$ as the top-K most similar users to the target user u, all the users v for which the target user is among the K most similar to v are selected as the neighbors $N^{-1}(u)$ of u. Table 1 shows an example of a community of users and their corresponding standard and inverted neighborhoods for $K = 2$. This has several consequences. In the first place, the resulting, inverted neighborhoods no longer have all the same size. The size of the inverted neighborhood of a user u is the number of users to whose neighborhood u belongs – in particular this means that some users might have an empty neighborhood at the cost of user coverage of the recommendation, but we have observed in our experiments that this situation does not happen in practice if the original neighborhoods are large enough. Having different neighborhood sizes is not necessarily a drawback, on the contrary, it may be favorable that users have as large a neighborhood as the reliability of the available data for each user enables.

	user	item
Netflix	209.25	5,654.50
MSD	43.03	127.88

Table 2: Average profile size

	user-based	item-based
S	$\underset{u}{\text{avg}} \underset{v \in N(u)}{\text{avg}} \lvert\mathcal{I}_v\rvert$	$\underset{i}{\text{avg}} \underset{j \in N(i)}{\text{avg}} \lvert\mathcal{U}_j\rvert$
G	$Gini(\lvert N^{-1}(u)\rvert)$	$Gini(\lvert N^{-1}(i)\rvert)$
C	$\rho(\lvert\mathcal{I}_u\rvert, \lvert N^{-1}(u)\rvert)$	$\rho(\lvert\mathcal{U}_i\rvert, \lvert N^{-1}(i)\rvert)$

Table 3: Definition of the neighborhood properties. N denotes in this case a generic user or item neighborhood, either standard or inverted.

The inverted neighborhoods approach implies, on the other hand, that all users will appear in exactly the same number $|N(u)|$ of inverted neighborhoods (except perhaps a few low activity users for which it was not possible to form a direct neighborhood of size K in the first place). This flattens the influence power of all users, so that all users' opinions "count" to the same extent overall in the produced recommendations. This may be expected to avoid a concentration of recommendations over the tastes of a reduced set of users, thereby indirectly enhancing a more even distribution of items across recommendations to the user population.

In the case of the item-based variant this effect is even more direct: if all items appear in the inverted neighborhood of the same number of items (neighbor items being the candidates for recommendation in the item-based kNN method), they will have more even chances of making it to the top-N of recommendations, whereby one may expect better distributed recommendations over the set of items (i.e. more diverse "sales"). Moreover, long-tail items, by getting a more equal opportunity to be recommended with respect to popular items, might make for a long-tail novelty enhancement of recommendations.

In order to have a preliminary understanding of these potential effects, we will analyze more closely in the next section the relation between user and item characteristics, namely profile size, and their distribution across neighborhoods for the direct and inverted selection policies. Our discussion of the potential effects on final recommendation diversity is so far speculative and needs to be tested empirically, as we report in Section 8.2.

6. NEIGHBORHOOD BIAS ANALYSIS

We test and illustrate the biases suggested in the previous section by taking some measurements on data from the Netflix Prize and the Million Song Dataset, for which we study the characteristics of user and item neighborhoods with different neighborhood sizes K.

We show in Table 4 the following measurements:

- Average profile size of the neighbors (**S**), in order to detect any possible bias towards neighbors with profile sizes significantly different from the average profile size (displayed in Table 2).
- Gini coefficient of the items distribution across neighborhoods (**G**), as an indicator to detect any imbalance in the distribution of the number $|N_K^{-1}(u)|$ of neighborhoods a user belongs to.
- Correlation between the profile size of a user and the number of neighborhoods she belongs to (**C**), to see if an existing imbalance is caused for a bias to users with big profiles in the neighbor selection process.

		S		G		C	
	K	N_K	N_K^{-1}	N_K	N_K^{-1}	N_K	N_K^{-1}
Netflix	10	286.39	137.91	0.18	1.00	0.07	-
	20	291.07	131.91	0.20	1.00	0.09	-
	50	298.20	124.68	0.22	1.00	0.11	0.00
	100	304.38	120.08	0.25	1.00	0.12	0.00
	200	311.24	116.46	0.27	1.00	0.15	0.00
	500	321.53	113.25	0.30	1.00	0.18	0.01
	1000	330.29	111.58	0.33	1.00	0.22	0.01
	2000	339.96	110.41	0.35	1.00	0.26	0.02
	5000	353.90	110.12	0.21	1.00	0.32	0.02
MSD	10	23.12	35.41	0.18	1.00	-0.11	0.00
	20	24.29	35.58	0.21	1.00	-0.11	0.00
	50	26.27	35.96	0.26	1.00	-0.12	0.00
	100	28.07	36.41	0.30	1.00	-0.12	0.00
	200	30.06	37.07	0.34	1.00	-0.12	0.01
	500	32.75	38.63	0.40	1.00	-0.11	0.01

Table 4: Properties of user neighborhoods with cosine similarity for the Netflix and Million Song datasets. Dashes mark undefined correlations since $|N^{-1}(u)|$ was constant for all the users. See Table 3 for the meaning of S, G and C.

		S		G		C	
	K	N_K	N_K^{-1}	N_K	N_K^{-1}	N_K	N_K^{-1}
Netflix	10	24,018.26	8,622.86	0.48	1.00	0.22	-
	20	25,484.00	7,888.98	0.49	1.00	0.28	-
	50	27,674.89	7,208.90	0.50	1.00	0.37	-
	100	29,497.64	6,824.87	0.50	1.00	0.44	-
	200	31,272.44	6,513.84	0.51	1.00	0.50	-
	500	32,565.92	6,525.21	0.51	1.00	0.56	-
	1000	30,964.19	7,301.09	0.54	1.00	0.57	-
	2000	26,208.78	9,150.45	0.62	1.00	0.53	0.01
	5000	17,001.67	12,839.88	0.80	1.00	0.26	0.01
MSD	10	146.76	120.09	0.60	1.00	0.02	0.00
	20	176.99	114.08	0.64	1.00	0.05	0.01
	50	248.41	108.56	0.67	0.99	0.12	0.02
	100	344.79	106.58	0.70	0.97	0.23	0.04
	200	497.18	116.77	0.70	0.94	0.42	0.05
	500	822.14	158.30	0.62	0.86	0.64	0.08
	1000	1,147.84	216.59	0.53	0.78	0.71	0.11
	2000	1,475.06	302.41	0.45	0.67	0.75	0.15
	5000	1,793.47	462.71	0.37	0.52	0.76	0.21

Table 5: Properties of item neighborhoods for the Netflix and Million Song datasets. Dashes mark undefined correlations since $|N^{-1}(i)|$ was constant for all the items. See Table 3 for the meaning of S, G and C.

A more formal definition of the above measurements is given in Table 3, where we denote by $N_K(u)$ the direct neighborhood formed by the K most similar users to a user u, and by $N_K^{-1}(u)$ the inverted neighborhood for u.

The results in Table 4 reveal, as hypothesized, biases and concentrations in the selection of user for standard neighborhoods. In the case of Netflix data, the standard neighborhood method is slightly biased towards selecting users with big profiles and shows a clear concentration on a small subset of users. In the case of the Million Song Dataset, there is an opposite bias towards small profiles, which also causes a concentration of neighbors. A possible explanation of why these methods differ in the direction of the bias may lie in the incomparable number of items between them and the different levels of sparsity in each dataset. In any case, the inverted selections strategy corrects this biases, that is, elim-

inates the correlation between profile size and the number of neighborhoods a user belongs to and, at the same time, creates a perfectly balanced distribution of this number.

Table 5 shows the equivalent measurements for item neighborhoods. Again, we can observe biases and concentration in the direct selection method that are partly solved by the inverted neighborhoods. The Netflix data shows a bias towards popular items that, ultimately, compose the majority of the neighborhoods. These issues are solved by the inverted item neighborhoods, which achieve a perfectly equitative distribution of the items in the neighborhoods, doing away with the bias towards popular items. In the Million Song dataset, a bias towards popular items is also observed for large neighborhood sizes, and an uneven distribution of the items in the distribution is observed for all neighborhood sizes. Again, inverted neighborhoods help solving these effects by significantly reducing the bias towards popular items and achieving more uniform distributions in the number of neighborhoods each item belongs to.

7. PROBABILISTIC REFORMULATION LAYER

The inverted recommendation task can also be addressed in probabilistic terms. Probabilistic formulations have been used extensively in the conventional item recommendation task as a means to develop collaborative filtering methods. For instance, Hofmann [8] proposed ranking items by the decreasing probability $p(i|u)$ that the target user would prefer each item over the others. This principle is developed by means of an adaptation of the probabilistic Latent semantic Indexing (pLSA) framework into an effective scoring procedure for producing ranked recommendations.

Turning the task around, recommending users for items would consist of estimating $p(u|i)$ for each user u given an item i. A straightforward way of linking any recommendation algorithm to a probabilistic formulation can be established by assuming that the recommender scoring function $s(u,i)$ is roughly proportional to $p(u,i)$. This assumption, coarse as it may be, provides a very direct means to bring the recommendation algorithm to a probabilistic interpretation as per:

$$p(u \mid i; s) \sim \frac{s(u,i)}{\sum_v s(v,i)} \tag{14}$$

We can therefore use this approach to obtain an inverted recommendation method out of any direct item recommendation algorithm. Note that the resulting formulation produces a totally equivalent output as its scoring function preserves the original ranking, i.e., $p(u \mid i; s) \propto s(u,i)$. The formulation is however useful as it enables a probability-based manipulation of the popularity bias in recommendation algorithms, as we see next.

First, the resulting output of the inverted recommendation should be reverted to a list of ranked items to be delivered to each user. A principled way to do this is by applying Bayesian inversion on $p(u|i)$, thereby obtaining an estimate for $p(i|u)$ as a suitable scoring function for ranking items for each user:

$$p(i \mid u; s) = \frac{p(u \mid i; s)\, p(i; s)}{\sum_j p(u \mid j; s)\, p(j; s)} \tag{15}$$

where the prior $p(i; s)$ represents how likely the item is to be the favorite of a random user.

	UB	IB	iMF
Netflix	0.99	0.73	0.95
MSD	0.92	0.78	

Table 6: Pearson correlation between prior $p(i; s, 0)$ and item popularity.

Note that we could instead have derived an estimate of $p(i \mid u; s)$ by an equivalent symmetric version of equation 14. However, the advantage of equation 15 is that it explicitly reflects the popularity component carried by the item prior $p(i; s)$. Using the same assumption as before between the scoring function and probabilities, we have:

$$p(i; s) \sim \frac{\sum_u s(u,i)}{\sum_j \sum_u s(u,j)} \tag{16}$$

Now that the popularity component is isolated, we propose to smooth the prior estimate in a way that has it range from the literal estimate based on the recommender's scores, to a flat uniform background prior where all items are considered equally popular. We do so by an entropic regularization of the estimate – similar to the tempered expectation maximization algorithm in [8], which simply introduces an exponent in the expression:

$$p(i; s, \alpha) \sim \frac{\left(\sum_u s(u,i)\right)^{1-\alpha}}{\sum_j \left(\sum_u s(u,j)\right)^{1-\alpha}} \tag{17}$$

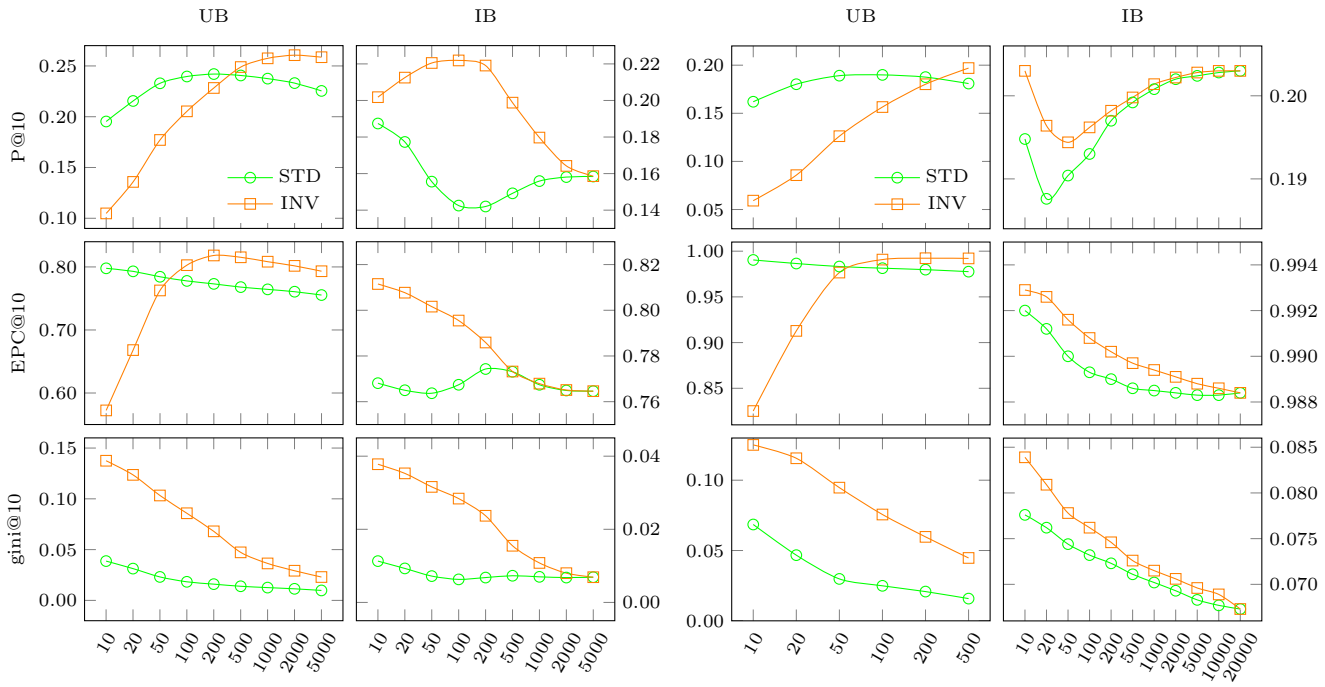
In the above expression, the $\alpha \in [0, 1]$ smoothing parameter allows controlling how much of the algorithm popularity bias we wish to leave as is or remove.

Interestingly, by combining equations 15 and 17, the resulting probabilistic interpretation $p(i \mid u; s)$ can be simplified to a re-scoring procedure for a standard scoring function s as follows:

$$s_{BR}(u,i) = s(u,i) \left(\sum_v s(v,i) \right)^{-\alpha} \tag{18}$$

This last reformulation allows to see more clearly the role of the parameter α. On one hand, when $\alpha = 0$ we obtain the original recommendation list created with the scoring function. On the other hand, when $\alpha = 1$ the prior is uniform and thus the recommendations to users will be completely based on $p(u \mid i; \tilde{s})$, eliminating any possible popularity bias in the items. The use of intermediate values of α is a means to provide more varied recommendation lists while retaining an appropriate level of relevance, that is α is a parameter that controls the relevance/novelty trade-off, only that novelty is not applied as the opposite to popularity (as is the case in most novelty enhancement approaches [1, 17]), but rather as neutrality with respect to popularity.

To illustrate how we can control the popularity bias by the proposed approach, we show in Table 6 the Pearson correlation values between the priors $p(i; s, 0)$ and the popularity of the items (understood as the number of users who know – i.e. have rated – the item) for some recommendations baselines – further detailed in Section 8.1 – for the Netflix and Million Song datasets. The values reveal a strong correlation between our score-based estimate of the item priors and the actual popularity of the items. This, on the other hand, empirically illustrates the popularity bias of these state of the art algorithms as discussed in Section 3, and shows how the prior component captures it, enabling its gradual adjustment by the α parameter.



(a) Netflix Dataset **(b) Million Song Dataset**

Figure 1: Experiments with Inverted Neighborhoods

	K	UB (N_K)	UB (N_K^{-1})	IB (N_K)	IB (N_K^{-1})
Netflix	**10**	0.9893	0.6602	0.9989	0.9989
	20	0.9961	0.7775	0.9989	0.9989
	50	0.9989	0.8995	0.9989	0.9989
	100	0.9989	0.9575	0.9989	0.9989
	200	0.9989	0.9871	0.9989	0.9989
	500	0.9989	0.9981	0.9989	0.9989
	≥1000	0.9989	0.9989	0.9989	0.9989
MSD	**10**	1.0000	0.8291	1.0000	1.0000
	20	1.0000	0.9178	1.0000	1.0000
	50	1.0000	0.9830	1.0000	1.0000
	100	1.0000	0.9980	1.0000	1.0000
	200	1.0000	0.9999	1.0000	1.0000
	≥500	1.0000	1.0000	1.0000	1.0000

Table 7: User coverage as the fraction of users in the test split which receive a recommendation in the Netflix and MSD data.

8. EXPERIMENTS

In order to test the effectiveness and analyze the properties of the proposed inverted nearest neighbor methods and the probabilistic popularity adjustment, we carry out two experiments on the Netflix Prize[2] and Million Song Dataset Challenge [10] datasets. The Netflix data contain 100M ratings (from one to five stars) by 480,000 users to 18,000 movies. And we used the Taste Profile Subset of the Million Song Dataset, containing 48M play count triplets by 1,100,000 users for 380,000 songs. As in the work of Aiolli [2], we take binarized play counts since, as warned by the challenge organizers, play counts are unreliable and not necessarily correlate with likings.

8.1 Setup

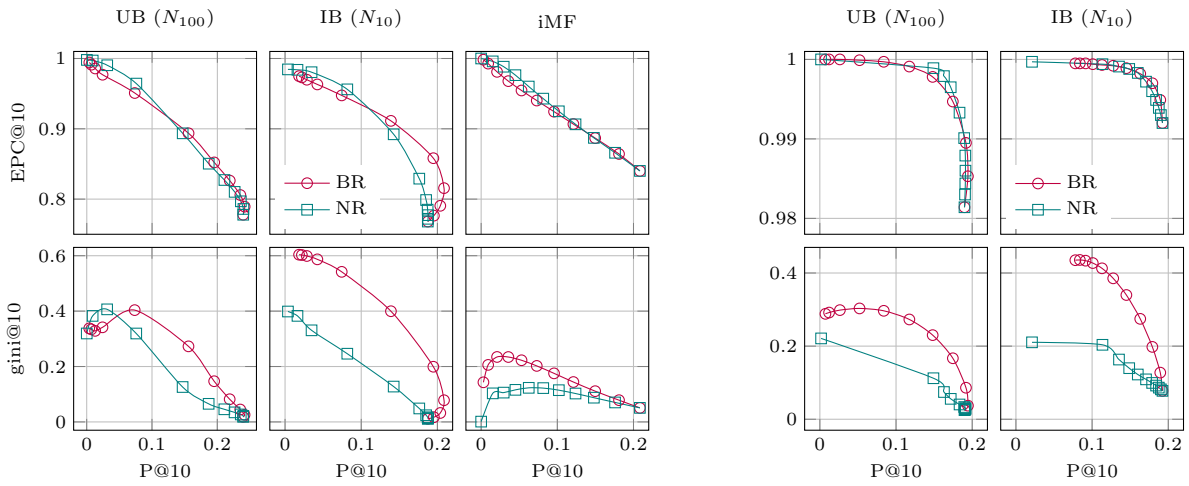
Each dataset is split for evaluation into training and test subsets. For Netflix, we do a 80-20% random split of the data, while in the Million Song dataset we take the partition provided with the data release. For every user with test data, we generate recommendation lists by ranking all the items with training data. We then measure rank-based precision, novelty and sales diversity for the top-10 items in the recommendation for each user. Precision is measured as the proportion of relevant test items of the user included in the recommendation he is delivered. Novelty is measured by EPC [17] and sales diversity by the Gini coefficient. For illustrative purposes, we also report results for aggregate diversity [1] normalized by the number of items.

For the inverted neighborhood approach, we compare the inverted kNN approaches described (Equations 11 and 12) to the corresponding standard user-based and item-based formulations (Equations 9 and 10). We used cosine as the similarity function between users and items, and explored a range of neighborhood sizes for both datasets.

[2]http://www.netflixprize.com/

		P@10	EPC@10	gini@10	aggr-div@10
Netflix	**Random**	0.0022	0.9866	0.9683	1.0000
	Pop	0.1026	0.6920	0.0019	0.0100
	iMF	0.2084	0.8401	0.0507	0.3571
	UB (N_{100})	0.2396	0.7778	0.0183	0.2784
	UB (N_{100}^{-1})	0.2053	0.8028	0.0858	0.9591
	IB (N_{10})	0.1874	0.7681	0.0113	0.4691
	IB (N_{10}^{-1})	0.2018	0.8115	0.0378	0.6738
MSD	**Random**	0.0000	0.9999	0.6750	0.9424
	Pop	0.0306	0.9376	0.0000	0.0001
	UB (N_{100})	0.1899	0.9814	0.0249	0.1634
	UB (N_{100}^{-1})	0.1565	0.9908	0.0756	0.3119
	IB (N_{10})	0.1924	0.9920	0.0776	0.3341
	IB (N_{10}^{-1})	0.1965	0.9929	0.0839	0.3459

Table 8: Comparison of inverted neighborhood methods to other recommendation algorithms.



(a) Netflix Dataset **(b) Million Song Dataset**

Figure 2: Experiments with Bayes Rule

For the probabilistic reformulation, a comparison between our approach and the novelty-oriented re-scoring approach defined by Equation 6 in three different recommendation baselines, namely the standard user and item-based kNN recommenders (N_{100} and N_{10} respectively) and, for the case of the Netflix dataset, the implicit matrix factorization proposed by Hu et al. [9], by considering positive ratings as implicit data. As stated in [2], matrix factorization approaches are not effective in the Million Song dataset, as our attempts at it confirmed, whereby we omit results with this baseline in this dataset. The two compared approaches have a parameter that controls the trade-off between the original scoring function ($\alpha = 0.0$ and $\lambda = 0.0$) and a pure novelty component ($\alpha = 1.0$ and $\lambda = 1.0$). We explore these trade-offs by a grid search on the full interval by steps of 0.1.

8.2 Results for Inverted Nearest Neighbors

Figure 1 shows the comparison of the direct (i.e. standard) and inverted nearest-neighbor approaches in terms of the metrics of interest for different neighborhood sizes.

The results confirm a systematic increase in sales diversity (measured by the Gini coefficient), as hypothesized in Section 5. The improvement is consistent in the user-based and item-based versions for all neighborhood sizes on both datasets. Notably moreover, for the item-based approach, the inverted method offers better accuracy and novelty for every value of K. For inverted user-based kNN, accuracy and novelty are better than in direct kNN only for large enough neighborhoods ($K \geq 100$). This is caused by user coverage degradation that occurs with smaller K, that is, many users do not receive recommendations since their inverted neighborhoods are empty, resulting in a penalization in metrics such as precision and EPC (we sum zero precision and EPC when a user cannot be delivered a recommendation). The details about the user coverage degradation are shown in Table 7. In a real recommender system this would not be acceptable, and a fallback solution, such as using the standard neighborhood, should be resorted to in those cases. However, in this analysis we are interested in the properties of the pure algorithm, and we therefore report the results for a plain version of the approach.

In order to provide a wider perspective in the context of alternative recommendation algorithms, Table 8 shows the comparison of inverted kNN for $K = 100$ in user-based kNN and $K = 10$ in item-based kNN with random, popularity-based and matrix factorization recommendations. It can be seen that the inverted kNN approaches obtain the best results – after random recommendation of course – in sales diversity and novelty. Random recommendation, as expected, produces inaccurate but highly novel and diverse results in both datasets. Popularity-based recommendation also yields predictable outcomes, producing moderately accurate results – depending on the sparsity of each dataset – and the lowest possible novelty and sales diversity – which should be so by definition. Matrix factorization (only tested on Netflix for the aforementioned reasons) has good novelty and sales diversity, outperforming the standard nearest neighborhoods methods, but not the inverted variant.

8.3 Results for the Probabilistic Reformulation

The results of our experiments with the Probabilistic Layer approach are shown in Figure 2. For each dataset-baseline pair we display two scatter plots showing the trade-offs between precision vs. novelty and sales diversity for the novelty-oriented re-scoring technique (NR) and our probabilistic approach (BR). Curves for each approach start from $\alpha = \lambda = 0.0$ as the points with the lowest novelty and diversity and, as α and λ tend to 1.0, improve in terms of EPC and Gini while – generally – having lower precision values. Assuming that the interpolated lines are a good approximation to the continuous range of the trade-off parameters, we determine that a method is better that the other if its curve is generally above the other in each plot. Under this criterion, the results in Figure 2 show the validity of our probabilistic approach.

In the Netflix data, we can see how the compared approaches show practically identical trade-offs in terms on EPC and, in terms of Gini, our probabilistic method clearly outperforms the novelty-oriented re-ranking. Surprisingly, the probabilistic approach outperforms the original scoring function even in terms of precision and, among baselines, it is the one that achieves the highest sales diversity scores. On the other side, the improvements on the matrix factorization approach, although being perceptible, are more limited that those in the nearest neighbors recommenders.

In the Million Song dataset the results are analogous. Both re-ranking approaches present similar outcomes in terms of EPC, while the probabilistic approach clearly outperforms the novelty-oriented re-ranking. Again, item-based kNN is the baseline that enables a higher improvement in terms of sales diversity.

In conclusion, the proposed probabilistic approach provides a new method for improving trade-offs between accuracy, novelty and sales diversity. Compared to a simpler approach that optimizes directly the long-tail novelty of recommendations, our proposal obtains comparable results in terms of the novelty of recommendations, while it shows clearly better results in terms of sales diversity.

9. CONCLUSIONS AND FUTURE WORK

Starting from the aim of improving sales diversity by recommendation, we explore in this paper where the inversion of the recommendation task leads to. By ranking users for items, the recommendation approach focuses on the relevance of user-item pairs in a different way, and item popularity gets left aside as a result. Starting from this task inversion, we derive two approaches to improve the sales diversity of the original item recommendation task. The first one, inverted neighborhoods, results in a novel way of "democratizing" the weight of user opinions (in the user-based approach) and item opportunity (in the item-based variant), leading to substantial improvements in terms of sales diversity, competitive results in recommendation novelty and a good precision trade-off. The second approach, a probabilistic reformulation of the recommendation problem, allows isolating the popularity component of any recommendation baseline and calibrate it in order to increase the chances of less popular items to appear in recommendations lists. Experiments on two different datasets, namely Netflix for movie recommendation and the Million Song Dataset for music recommendation, confirm and illustrate the effectiveness of our proposals.

The symmetric inversion of the recommendation task entails more than a simple transposition of the rating matrix. It brings up a new view on the task where the system seeks the most appropriate users to whom an item can be recommended, even though the final action is still the delivery of a ranked list of items to each user. This problem statement can reflect real-world situations where a business is selecting targets for advertising a particular product.

As future work, we envisage deeper studies on the properties of neighborhoods we examined in Section 6 with additional metrics in order to uncover further potential biases in user and item neighborhoods. We also envision further improvements of the Bayesian reformulation. In particular, we intend to explore increasing the *exclusivity* of items, that is, recommending each item to only a limited selection of users. In our probabilistic scheme, this exclusivity could be carried out by re-defining the likelihood component $p(u \mid i; s)$ in Equation 14, for instance by creating a cut-off of the users with highest scores for item i or by means of parametrization similar to the one of the prior in Equation 17.

10. ACKNOWLEDGMENTS

This work was supported by the national Spanish project TIN2013-47090-C3-2.

11. REFERENCES

[1] G. Adomavicius and Y. Kwon. Improving aggregate recommendation diversity using ranking-based techniques. *IEEE Transactions on Knowledge and Data Engineering*, 24(5):896–911, 2012.

[2] F. Aiolli. Efficient top-n recommendation for very large scale binary rated datasets. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pages 273–280, New York, NY, USA, 2013. ACM.

[3] X. Amatriain. Mining large streams of user data for personalized recommendations. *SIGKDD Explor. Newsl.*, 14(2):37–48, Apr. 2013.

[4] C. Anderson. *The Long Tail: Why the Future of Business Is Selling Less of More*. Hyperion, 2006.

[5] A. Bellogin. *Performance prediction and evaluation in Recommender Systems: an Information Retrieval perspective*. PhD thesis, Escuela Politécnica Superior, Universidad Autónoma de Madrid, Spain, 2012.

[6] P. Cremonesi, Y. Koren, and R. Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 39–46, New York, NY, USA, 2010. ACM.

[7] D. Fleder and K. Hosanagar. Blockbuster culture's next rise or fall: The impact of recommender systems on sales diversity. *Management Science*, 55(5):697–712, May 2009.

[8] T. Hofmann. Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems*, 22(1):89–115, Jan. 2004.

[9] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, ICDM '08, pages 263–272, Washington, DC, USA, 2008. IEEE Computer Society.

[10] B. McFee, T. Bertin-Mahieux, D. P. Ellis, and G. R. Lanckriet. The Million Song Dataset Challenge. In *Proceedings of the 21st International Conference Companion on World Wide Web*, WWW '12 Companion, pages 909–916, New York, NY, USA, 2012. ACM.

[11] A. Said, B. Fields, B. J. Jain, and S. Albayrak. User-centric evaluation of a k-furthest neighbor collaborative filtering recommender algorithm. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, CSCW '13, pages 1399–1408, New York, NY, USA, 2013. ACM.

[12] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, WWW '01, pages 285–295, New York, NY, USA, 2001. ACM.

[13] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, N. Oliver, and A. Hanjalic. CLiMF: learning to maximize reciprocal rank with collaborative less-is-more filtering. In *Proceedings of the sixth ACM conference on Recommender systems*, RecSys '12, pages 139–146, New York, NY, USA, 2012. ACM.

[14] Y. Shi, M. Larson, and A. Hanjalic. Unifying rating-oriented and ranking-oriented collaborative filtering for improved recommendation. *Inf. Sci.*, 229:29–39, Apr. 2013.

[15] H. Steck. Item popularity and recommendation accuracy. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, pages 125–132, New York, NY, USA, 2011. ACM.

[16] G. Takács and D. Tikk. Alternating least squares for personalized ranking. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, RecSys '12, pages 83–90, New York, NY, USA, 2012. ACM.

[17] S. Vargas and P. Castells. Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the fifth ACM conference on Recommender systems*, RecSys '11, pages 109–116, New York, NY, USA, 2011. ACM.

[18] X. Zhao, Z. Niu, and W. Chen. Opinion-based collaborative filtering to solve popularity bias in recommender systems. In H. Decker, L. Lhotská, S. Link, J. Basl, and A. Tjoa, editors, *Database and Expert Systems Applications*, volume 8056 of *Lecture Notes in Computer Science*, pages 426–433. Springer Berlin Heidelberg, 2013.