Multi-Objective Pareto-Efficient Approaches for Recommender Systems

MARCO TULIO RIBEIRO, ANISIO LACERDA, Computer Science Department, Universidade Federal de Minas Gerais, and Zunnit Technologies, Brazil
EDLENO SILVA DE MOURA, Computer Science Department, Universidade Federal do Amazonas, Brazil
ITAMAR HATA, ADRIANO VELOSO and NIVIO ZIVIANI, Computer Science Department, Universidade Federal de Minas Gerais, Brazil

Recommender systems are quickly becoming ubiquitous in applications such as e-commerce, social media channels, content providers, among others, acting as an enabling mechanism designed to overcome the information overload problem by improving browsing and consumption experience. A typical task in many recommender systems is to output a ranked list of items, so that items placed higher in the rank are more likely to be interesting to the users. Interestingness measures include how accurate, novel and diverse are the suggested items, and the objective is usually to produce ranked lists optimizing one of these measures. Suggesting items that are simultaneously accurate, novel and diverse is much more challenging, since this may lead to a conflicting-objective problem, in which the attempt to improve a measure further may result in worsening other measures. In this paper we propose new approaches for multi-objective recommender systems based on the concept of Pareto-efficiency – a state achieved when the system is devised in the most efficient manner in the sense that there is no way to improve one of the objectives without making any other objective worse off. Given that existing multi-objective recommendation algorithms differ in their level of accuracy, diversity and novelty, we exploit the Pareto-efficiency concept in two distinct manners: (i) the aggregation of ranked lists produced by existing algorithms into a single one, which we call Pareto-efficient ranking, and (ii) the weighted combination of existing algorithms resulting in a hybrid one, which we call Pareto-efficient hybridization. Our evaluation involves two real application scenarios: music recommendation with implicit feedback (i.e., Last.fm) and movie recommendation with explicit feedback (i.e., MovieLens). We show that the proposed Pareto-efficient approaches are effective in suggesting items that are likely to be simultaneously accurate, diverse and novel. We discuss scenarios where the system achieves high levels of diversity and novelty without compromising its accuracy. Further, comparison against multi-objective baselines reveals improvements in terms of accuracy (from 10.4% to 10.9%), novelty (from 5.7% to 7.5%), and diversity(from 1.6% to 4.2%).

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Information Filtering

Additional Key Words and Phrases: Multi-Objective Recommender Systems, Pareto-Efficiency

ACM Reference Format:
Marco Tulio Ribeiro, Anisio Lacerda, Edleno Silva de Moura, Itamar Hata, Adriano Veloso, and Nivio Ziviani, 2013. Multi-Objective Pareto-Efficient Approaches for Recommender Systems. *ACM Trans. Intell. Syst. Technol.* 9, 1, Article 1 (February 2013), 20 pages.
DOI:http://dx.doi.org/10.1145/0000000.0000000

This work is partially supported by the National Institute of Science and Technology for the Web (InWeb), MCT/CNPq grant 57.3871/2008-6, and by the authors' individual grants and scholarships from CNPq.

DOI:http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

Recommender systems identify interesting items in situations where the number and complexity of possibilities outstrip the user's capability to survey them in order to reach a proper decision. Such complex situations are commonly observed in applications provided by many social media [Guy et al. 2010] and e-commerce sites [Wang and Zhang 2011]. Ideally, by suggesting interesting items, a recommender system essentially brings properties that are desired by the users, quickly becoming essential tools that change the way people interact with the Web.

There are several tasks and challenges associated with modern recommender systems. A typical task is stated as follows: given a set of items that are known to be relevant to some users (i.e., past preference data), the system must return a ranked list of suggested items, so that items that are more interesting to the user appear in the top of the rank [Cremonesi et al. 2010]. The notion of interestingness, however, may be subjective, encompassing a broad repertoire of measures, including: (i) accuracy (how well the suggested items meet the user's information need), (ii) novelty (how well the system promotes unknown items to the users), and (iii) diversity (how different the suggested items are with respect to each other).

Historically, the typical goal of a recommender system is to maximize accuracy as much as possible in predicting and matching user information needs, often by considering individual delivered items in isolation. However, it has become a consensus that accurate suggestions are not necessarily useful [McNee et al. 2006]: real value is found in the ability to suggest relevant items that are not easily discovered by the users, that is, in the novelty and diversity of the suggestions [Zhou et al. 2010]. Therefore, a pressing challenge resides on devising recommender systems able to perform suggestions that are simultaneously accurate, novel and diversified, what may lead to a conflicting-objective problem, where the attempt to improve a measure further (i.e., an objective) may result in worsening other competing measures. Thus, the need of the hour is to devise solutions that find a proper balance between accuracy, novelty and diversity, so that the returned ranked list reflects all three interestingness measures simultaneously as well as when they are taken in isolation (i.e., items in the top of the ranked list are likely to be as accurate, diverse and novel as possible).

In this paper we tackle this problem by proposing approaches based on the concept of *Pareto Efficiency*. This is a central concept in Economics, which informally states that "when some action could be done to make at least one person better off without hurting anyone else, then it should be done." This action is called Pareto improvement, and a system is said to be Pareto-Efficient if no such improvement is possible. The same concept may be exploited for the sake of devising multi-objective recommender systems that are built by combining existing recommendation algorithms. In this case, the most efficient recommender system is the one which cannot improve an objective further (i.e., accuracy, diversity or novelty) without hurting the other objectives. Given that existing recommendation algorithms are complementary in the sense that they greatly differ in their level of accuracy, novelty and diversity, we may exploit the Pareto-Efficiency concept in two distinct manners:

(1) Pareto-Efficient Ranking: Each possible item is associated with a point in an n-dimensional scattergram (which we call the user-interest space). In this case, a point is represented as $[c_1, c_2, \ldots, c_n]$, where each coordinate c_i corresponds to the relevance score estimated by a different recommendation algorithm. Points that are not dominated by any other point in the scattergram compose the Pareto frontier [Goldberg 1989; Palda 2011]. Points lying in the frontier correspond to cases for which no Pareto improvement is possible, being therefore items more likely to be simultaneously accurate, novel and diversified.

(2) Pareto-Efficient Hybridization: The final relevance score of an arbitrary item is estimated using a linear combination of the relevance scores estimated by n different existing recommendation algorithms (i.e., $\alpha \times c_1 + \beta \times c_2 + \ldots + \theta \times c_n$). In this case, we have a 3-dimensional scattergram (which we call the objective space) and each point in this scattergram corresponds to the level of accuracy, novelty and diversity achieved by a possible hybrid recommendation algorithm. We may search for weights (i.e., $\alpha, \beta, \ldots, \theta$) for which the corresponding points lie in the Pareto frontier, and then choose the hybrid that best fits the system priority.

Both approaches run in an offline setting, that is, the ranked lists are produced before the user interacts with the system. We conducted a systematic evaluation involving different recommendation scenarios, with explicit user feedback (i.e., movies from the MovieLens dataset), as well as implicit user feedback (i.e., artists from the LastFM dataset). The results show that our main goal was successfully achieved: in most cases the proposed approaches produce systems that improve diversity and novelty without compromising accuracy (when compared against the results obtained with the best algorithms in isolation). Further, the comparison against multi-objective baselines indicate the superiority of our proposed approaches, which provide significant gains in terms of all three criteria considered in our analysis.

A preliminary partial version of the hybrid recommendation approach appeared in [Ribeiro et al. 2012]. The main difference between the paper in [Ribeiro et al. 2012] and this work is the inclusion of a totally different multi-objective recommendation approach, which, in some cases, is able to overcome the approach proposed in [Ribeiro et al. 2012]. In summary, the Pareto-efficient hybridization approaches proposed in [Ribeiro et al. 2012] should be the choice if it is desirable to give emphasis to a particular objective, without significantly hurting the other objectives. The Pareto-efficient ranking approaches introduced in this work should be the choice if it is desirable to maximize all objectives simultaneously.

2. RELATED WORK

Recommender systems have been recognized as an important topic by many research communities [Rendle et al. 2011; Leung et al. 2011; Li et al. 2011; Wang and Zhang 2011]. Several tasks have being extensively studied, including problems such as cold-start [Gunawardana and Meek 2008; Zhou et al. 2011], rating prediction [Koren and Sill 2011] and top-k recommendation [Cremonesi et al. 2010]. Items that are typical target for recommendation include tags [Garg and Weber 2008; Surbjörnsson and van Zwol 2008; Guan et al. 2009; Menezes et al. 2010] and movies [Gunawardana and Meek 2009]. In this work we perform experiments involving top-k recommendation of musics and movies, with implicit and explicit user feedback.

Several recommendation approaches have being proposed with the specific objective of providing to the user the most accurate suggestions as possible [Bellogín et al. 2011; Pan et al. 2008; Hu et al. 2008; Zhang et al. 2008]. More recent studies have shown that accuracy alone does not guarantee high user satisfaction in recommender systems [McNee et al. 2006; Ge et al. 2010]. As a result, attention has also being devoted to other properties associated with suggested items, such as diversity and novelty [Castells et al. 2011]. In [Zhou et al. 2010], the authors showed that the choice between accuracy and diversity is not necessarily a dilemma, and that it is possible to simultaneously achieve gains in accuracy and diversity by proposing hybrid approaches. In this paper we reached similar conclusions, but using different approaches.

Traditionally, hybrid recommendation algorithms are obtained by the combination of two different families of base algorithms – namely, content-based and collaborative filtering [Adomavicius and Tuzhilin 2005]. In this paper, we combine a broad repertoire of recommendation algorithms, including different content-based and collaborative filtering algorithms, algorithms that deal with explicit and implicit feedback etc. We treat each recommendation algorithm as a black-box, so adding or removing recommendation algorithms is easy. Different hybridization approaches have being proposed to combine recommendation algorithms, such as weighted approaches [Claypool et al. 1999], voting approaches [Pazzani 1999], switching between different algorithms [Lekakos and Caravelas 2008; Billsus and Pazzani 2000], and re-ranking the results of one algorithm with another [Burke 2002].

In [Ziegler et al. 2005], the authors introduced the notion of "topic diversification", which ensures diversity by balancing suggestions across different topics. Further, the authors proposed the intra-list similarity measure for assessing the diversity of suggested items. In [Lathia et al. 2010], the authors proposed a different measure of item diversity, which assess the extent to which the same items are being recommended to users over and over again. In this paper we employed diversity measures proposed in [Vargas and Castells 2011].

In [Kawamae 2010], the authors introduced a new metric for assessing item novelty by hypothesizing that the degree of user's surprise is proportional to the estimated time spent searching the item. In [Hurley and Zhang 2011], the authors proposed a statistical model for assessing novelty in recommender systems using a concentration index, which measures the ability to suggest novel items. They analyzed various recommendation approaches using the concentration index to determine which approaches are more suited towards diversity. In [Fouss and Saerens 2008], the authors proposed a novelty measure which assumes that less popular items are less likely to be widely known by users, and thus a penalty is imposed depending on the frequency or popularity of the suggested items. Again, in this paper we adopt novelty measures proposed in [Vargas and Castells 2011].

An excellent survey involving several measures used for evaluating recommender systems is found in [Shani and Gunawardana 2011]. Approaches for multi-objective, or multi-criteria recommender systems, were proposed in [Adomavicius et al. 2011; Adomavicius and Kwon 2007; Lee and Teng 2007].

To the best of our knowledge, the approaches we introduce in this paper differ from all existing multi-objective recommendation approaches. We exploit the notion of Pareto-Efficiency in order to sort items that balance accuracy, novelty and diversity. The Pareto-Efficiency concept was already employed in recommender systems that must cope with additional dimensions such as user privacy [Dokoohaki et al. 2010] and friendship [Naruchitparames et al. 2011], but our scenario is much more challenging, involving competing objectives. Our first approach employs the Pareto frontier to find a partial ordering between items, and by avoiding items located at the extreme positions of the frontier it finds items that are likely to be simultaneously interesting in terms of accuracy, diversity and novelty. Our second approach employs the Pareto frontier to find hybrids that are more likely to perform suggestions that are simultaneously interesting in terms of accuracy, diversity and novelty. Our proposed approaches are highly practical and effective for multi-objective recommender systems, as shown in our experiments.

3. PARETO-EFFICIENT RANKING

In this section we introduce our approach for Pareto-efficient ranking. We start by discussing how possible items are disposed in a user-interest space by exploiting different recommendation biases within existing recommendation algorithms. Then, we discuss how the user-interest space is used in order to aggregate multiple ranked lists into a final, Pareto-efficient, ranked list.

3.1. Recommendation Bias and User-Interest Space

Typically, a recommender system arranges items into a ranked list, so that the top-k items are those most interesting to the user. Although being naturally subjective, the potential interest a user will have in the top-k items may be approximated by the following interestingness measures:

— Accuracy: returns how well the top-k items meet the user's information need.
— Novelty: is inherently linked to the notion of discovery and returns how novel to the user are the top-k items. Further, top-k items are assumed to be accurate (i.e., relevant).
— Diversity: returns how different with respect to each other are the top-k items. Further, top-k items are assumed to be accurate.

Existing recommendation algorithms differ by large in their level of accuracy, novelty and diversity. The difference is due to a distinct recommendation bias which is followed by each algorithm, that is, existing algorithms may favor different interestingness measures. However, it is already a consensus that all three interestingness measures are essential to effective recommendation, since together these measures have a complementary effect which is highly desirable for recommender systems. That is, accurate suggestions are of little value if they are obvious to the user. Besides, suggesting items that are too similar to each other leads to monotonous and ineffective recommendations. Therefore, in order to ensure effective results, the top-k items within a ranked list must be as accurate, novel and diverse as possible.

Consider the set of constituent recommendation algorithms $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$ and assume that these algorithms assign to each possible item a score $\hat{p}_{a_j}(u_i|t)$ corresponding to the potential interest user t has on item u_i, we may represent each item u_i as a point in a n-dimensional user-interest space: $\mathcal{S}_t = [\hat{p}_{a_1}(u_i|t), \hat{p}_{a_2}(u_i|t), \ldots, \hat{p}_{a_n}(u_i|t)]_{i=1}^{m}$, where m is the number of possible items and each $\hat{p}_{a_j}(u_i|t)$ is calculated using one out of n different constituent recommendation algorithms (i.e., following different recommendation biases). Figure 1 (Left) depicts a 2-dimensional user-interest space. The dominance operator relates two items in such space, so that the result of the dominance operation has two possibilities: (i) one item dominates another or (ii) the two items do not dominate each other. We need now the following definition.

Definition 1: *A Pareto-Efficient ranked list for user t is an ordered list of m items $\mathcal{L}_t = \{u_1, u_2, \ldots, u_m\}$ such that there is no pair $(u_i, u_j) \in \mathcal{L}_t$ for which u_i dominates u_j, given that $i > j$.*

3.2. Building Pareto-Efficient Ranked Lists

Algorithm 1 builds a Pareto-Efficient ranked list for user t. Items that are not dominated by any other item in $\mathcal{S}_t$ lie on the Pareto frontier, as shown in Figure 1 (Right). Stripping off an item from the Pareto frontier, and building another frontier[1] from the remaining items in $\mathcal{S}_t$ reveals a partial ordering between the items, which we call a Pareto-Efficient ranking.

Next, we discuss different strategies for building Pareto-Efficient ranked lists for each user t. These strategies are based on Algorithm 1, and the only difference between them resides on the item that is selected at each iteration (i.e., step b). Still, our strategies try to avoid selecting items located at extreme positions of the frontier, since such items may privilege a specific measure. Instead, highly dominant items, or

[1] There are efficient algorithms for building and maintaining the Pareto frontier, such as the ones based on skyline queries [Lin et al. 2007; Papadias et al. 2003]. In particular, we employed the skyline operator algorithm proposed in [Börzsönyi et al. 2001], ensuring $O(n \times m \times k)$ complexity.

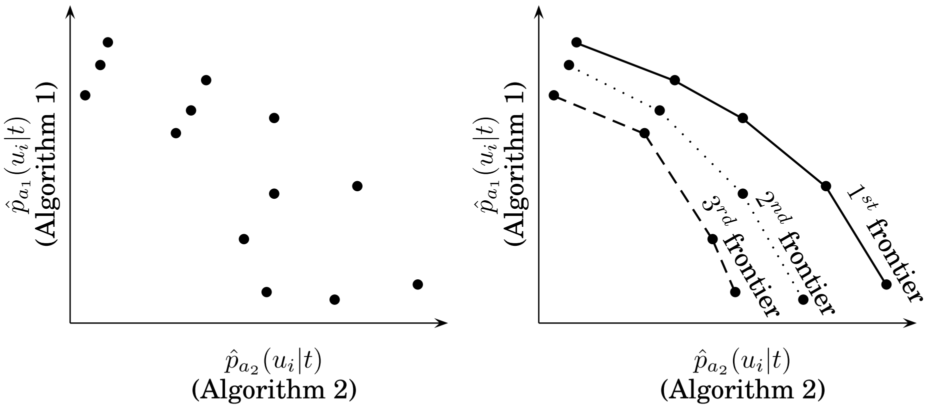

Fig. 1. Left – User-Interest space according to two different recommendation biases (i.e., different recommendation algorithms). Points are possible items and are represented by the relevance level estimated by different algorithms. Right – Non-dominated items form successive Pareto frontiers.

ALGORITHM 1: Pareto-Efficient Ranking.

```
Input: $\mathcal{S}_t$ (the $n$-dimensional interest space for user $t$), and $k$ (the number of suggested items).
Output: $\mathcal{L}_t$ (a Pareto-Efficient ranked list for user $t$).
repeat
    build the Pareto frontier in $\mathcal{S}_t$;
    include an item $x$ lying in the frontier into $\mathcal{L}_t$;
    remove $x$ from $\mathcal{S}_t$;
until $|\mathcal{L}_t| = k$;
```

items that are representative of other items in the frontier, are more likely to balance multiple objectives.

3.2.1. Most Dominant Items First. This strategy aims at selecting the item lying in the Pareto frontier which dominates more items in the user-interest space $\mathcal{S}_t$, as given by:

$$u_i \text{ such that arg max}(dom(u_i)), \forall\ u_i \in \mathcal{S}_t$$

where $dom(u_i)$ is the number of items dominated by u_i.

The number of items that are dominated by an arbitrary item u_i is easily obtained while building the Pareto frontier, and it remains unchanged as most dominant items are removed from $\mathcal{S}_t$, ensuring the efficiency of the process.

3.2.2. Learning to Rank. This strategy aims at selecting the item which is more likely to be located in the first Pareto frontier. To this end, we label items according to the frontier they are located (i.e., first, second, . . ., frontiers), so that items lying in the first frontiers are labeled as more relevant than items lying in subsequent frontiers. Then, we apply a well-known learning to rank algorithm, SVM-Rank [Joachims 2002], in order to sort items accordingly to their potential relevance level. Specifically, we model the training data as item-user pairs, and each pair is labeled with the Pareto frontier in which the corresponding item is located. SVM-Rank formalizes the ranking problem as a binary classification problem on instance pairs, and then solve the problem using SVMs [Joachims 2006].

4. PARETO-EFFICIENT HYBRIDIZATION

In this section we introduce our search approach for Pareto-efficient hybrids. We start by discussing how different recommendation algorithms are combined, so that potential hybrids are created. Then we describe the search strategy for Pareto-efficient hybrids.

4.1. Weighted Hybridization

Our hybridization approach is based on assigning weights to each constituent algorithm. We denote the set of constituent algorithms as $\mathcal{A} = \{a_1, a_2, \dots, a_n\}$, and we suppose that these algorithms assign to each possible item a score $\hat{p}_{a_j}(u_i|t)$ corresponding to the potential interest user t has on item u_i. Since the constituent algorithms may output scores in drastically different scales, a simple normalization procedure is necessary to ensure that all algorithms in $\mathcal{A}$ operate in the same scale. The aggregated score for each item i is calculated as follows:

$$\hat{p}(u_i|t) = \sum_{j=1}^{n} \hat{p}_{a_j}(u_i|t) \times w_{a_j} \quad (1)$$

where w_{a_j} is the weight assigned to algorithm $a_j \in \mathcal{A}$. The assignment of weights to each algorithm is formulated as a search problem which we discuss next.

4.2. Searching for Pareto-Efficient Hybrids

Finding a suitable hybrid, represented as a vector of weights $W = \{w_{a_1}, w_{a_2}, \dots, w_{a_n}\}$, can be viewed as a search problem in which each w_{a_i} is selected in a way that optimizes a established criterion. We consider the application of evolutionary algorithms for searching optimal solutions. These algorithms iteratively evolve a population of individuals towards optimal solutions by performing genetic-inspired operations, such as reproduction, mutation, recombination, and selection [Goldberg 1989]. Next we precisely define an individual.

Definition 2: *An individual is a candidate solution, which is encoded as a sequence of n values $[w_{a_1}, w_{a_2}, \dots, w_{a_n}]$, where each w_{a_i} indicates the weight associated with algorithm $a_i \in \mathcal{A}$.*

Each constituent algorithm a_i assigns scores to items using a cross-validation set. Finally, weights are assigned to each recommendation algorithm and their scores are aggregated according to Equation 1, producing an individual (i.e., an hybrid). A fitness function is computed for each individual in order to make them directly comparable, so that the population can evolve towards optimal solutions (i.e., individuals located closer to the Pareto frontier).

Definition 3: *An optimal solution is a sequence of weights $W = \{w_{a_1}, w_{a_2}, \dots, w_{a_n}\}$, satisfying Equation 2*:

$$\text{maximize } \phi(o_i) \, \forall o_i \in \{\text{accuracy}, \text{novelty}, \text{diversity}\} \quad (2)$$

where $\phi(o_i)$ is the value of an objective o_i, which can be either accuracy, novelty or diversity. Thus, the performance of each individual is given by a 3-dimensional objective vector, containing the average accuracy, novelty and diversity over all users in the cross validation set. Searching for optimal hybrids is a multi-objective optimization problem, in which the value of $\phi(o_i)$ must be maximized for each of the three objectives that compose an optimal solution. Therefore, multiple optimal individuals are possible.

Again, we exploit the concept of Pareto dominance for solving the multi-objective optimization problem. As a result, given the 3-dimensional objective space, the evolutionary algorithm evolves the population towards producing individuals that are located closer to the Pareto frontier, as illustrated in Figure 2.

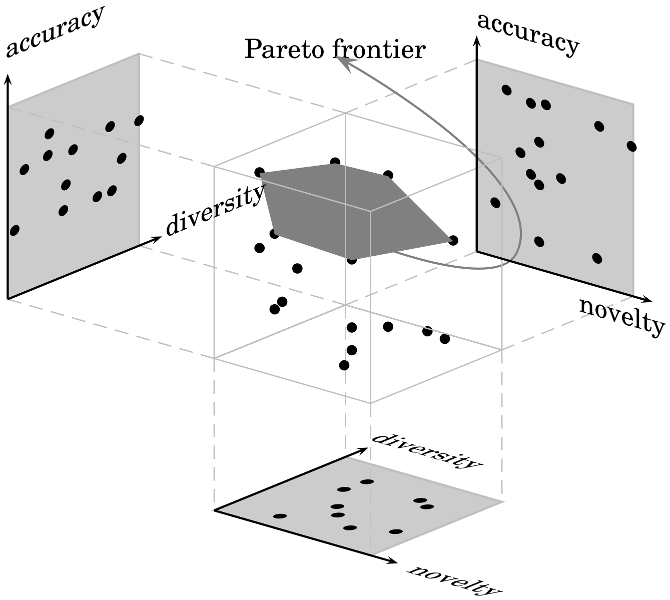

Fig. 2. A 3-dimensional objective space. Points are possible hybrids and are represented by the corresponding level of accuracy, novelty and diversity. Hybrids lying in the Pareto frontier are not dominated by any other hybrid.

The result is a set of Pareto-efficient hybrids. Under this strategy, we follow the well-known Strength Pareto Evolutionary Algorithm approach [Zitzler and Thiele 1999; Zitzler et al. 2001], which has shown to be highly effective and also because it provides more diverse individuals when compared to existing approaches [Corne et al. 2000; Deb 1999; Srinivas and Deb 1994] for many problems of interest. The Strength Pareto approach isolates individuals that achieve a compromise between maximizing the competing objectives by evolving individuals that are likely to be non-dominated by other individuals in the population. Algorithm 2 shows the basic steps of our Pareto-efficient hybridization approach.

ALGORITHM 2: Pareto-Efficient Hybridization.

```
Input: $\mathcal{P}$ (the current population of individuals), $p$ (the next population of individuals), and $g$
        (the maximum number of generations).
Output: Hybrids lying in the Pareto frontier.
repeat
    include the best individuals from $\mathcal{P}$ into $p$ (those closer to the frontier);
    apply genetic operators to individuals in $p$;
    update $\mathcal{P}$ with individuals in $p$;
until $g$ generations are produced;
```

4.3. Adjusting the System Priority

It is well recognized that the role that a recommender system plays may vary depending on the target user. For instance, according to [Herlocker et al. 2004], the suggestions performed by a recommender system may fail to appear trustworthy to the user because it does not recommend items the user is sure to enjoy but probably already knows about. Based on this, a recommender system might prioritize accuracy instead of novelty or diversity for new users, while prioritizing novelty for users that have already used the system for a while. This is made possible by our hybridization approach, by searching which individual in the Pareto frontier better solves the user's current needs.

The choice of which individual in the Pareto frontier is accomplished by performing a linear search on all of the individuals, in order to find which one maximizes a simple weighted mean on each of the three objectives in the objective vector, where the weights in the weighted mean represent the priority given to each objective. It is worth noting that fitness values are always calculated using the cross-validation set. Therefore, considering a 3-dimensional priority vector $Q = \{q_1, q_2, q_3\}$, that represents the importance of each objective j, the individual in the Pareto frontier $\mathcal{P}$ is chosen as follows:

$$\underset{i \in \mathcal{P}}{arg\ max} \sum_{j=1}^{3} q_j \times \phi(o_j) \tag{3}$$

5. EXPERIMENTAL EVALUATION

In this section we empirically analyze the effectiveness of our proposed Pareto-efficient approaches for the sake of multi-objective recommender systems. We assume an evaluation setting where recommendation approaches are compared without user interaction (i.e., offline setting). The experiments were performed on a Linux-based PC with a Intel I5 4.0 GHz processor and 4.0 GBytes RAM.

5.1. Evaluation Methodology

The evaluation methodology we adopted in this paper is the one proposed in [Cremonesi et al. 2010], which is appropriate for the top-N recommendation task. For each dataset, ratings are split into two subsets: the training set (denoted as $\mathcal{M}$), and the test set (denoted as $\mathcal{T}$). The training set $\mathcal{M}$ may be further split (if necessary) into two subsets: the cross-validation training set (denoted as $\mathcal{C}$), and the cross-validation test set (denoted as $\mathcal{V}$), which are used in order to tune parameters or adjust models (when applicable). The test set $\mathcal{T}$ and the cross-validation test set $\mathcal{V}$ only contain items that are considered relevant to the users in the dataset. For explicit feedback (i.e., MovieLens), this means that the sets $\mathcal{T}$ and $\mathcal{V}$ only contain 5-star ratings.

In the case of implicit feedback (i.e., Last.fm), we normalized the observed item access frequencies of each user to a common rating scale [0,5], as used in [Vargas and Castells 2011]. Namely, $r(u,i) = n * F(frec_{u,i})$, where $frec_{u,i}$ is the number of times user u has accessed item i, and $F(frec_{u,i}) = |j \in \mathbf{u}|f_{u,j} < f_{u,i}|/|\mathbf{u}|$ is the cumulative distribution function of $frec_{u,i}$ over the set of items accessed by user u, denoted as $\mathbf{u}$. In this case, the test set and the cross validation test set only contain ratings such that $r(u,i) >= 4$, since the number of 5-star ratings is very small using this mapping of implicit feedback into ratings. It is worth noting that all the sets have a corresponding implicit feedback set, used by the recommendation algorithms that can deal with implicit feedback.

The detailed procedure to create $\mathcal{M}$ and $\mathcal{T}$ is the same used in [Cremonesi et al. 2010], in order to maintain compatibility with their results. Namely, for each dataset we randomly sub-sampled 1.4% of the ratings from the dataset in order to create a probe set. The training set $\mathcal{M}$ contains the remaining ratings, while the test set $\mathcal{T}$ contains all the 5-star ratings in the probe set (in the case of explicit feedback) or 4+ star ratings (in the case of implicit feedback mapped into explicit feedback). We further divided the training set in the same fashion, in order to create the cross-validation training and test sets $\mathcal{C}$ and $\mathcal{V}$. The ratings in the probe sets were not used for training.

In order to evaluate the algorithms, we first train the models using $\mathcal{M}$. Then, for each item in $\mathcal{T}$ that is relevant to user u:

- We randomly select 1,000 additional items unrated by user u. The assumption is that most of them will not be interesting to u.
- The algorithm in question forms a ranked list by ordering all of the 1,001 items. The most accurate result corresponds to the case where the test item i is in the first position.

Since the task is top-N recommendation, we form a top-N list by picking the N items out of the 1,001 that have the highest rank. If the test item i is among the top-N items, we have a *hit*. Otherwise, we have a *miss*. Recall and precision are calculated as follows:

$$recall@N = \frac{\#hits}{|T|} \tag{4}$$

$$precision@N = \frac{\#hits}{N * |T|} = \frac{recall@N}{N} \tag{5}$$

In order to measure the amount of novelty within the suggested items, we used a popularity-based item novelty model proposed in [Vargas and Castells 2011], so that the probability of an item i being seen is estimated as:

$$P(seen|i) = \frac{|u \in U | r(u,i) \neq \emptyset|}{|U|} \tag{6}$$

where U denotes the set of users. Since the evaluation methodology supposes that most of the 1,000 additional unrated items are not relevant to user u, we used the metrics in the framework proposed in [Vargas and Castells 2011] without relevance awareness. Finally, the amount of novelty within a top-N recommendation list R presented to user u is therefore given by:

$$EPC@N = C \sum_{i_k \in R}^{i_N} disc(k)(1 - p(seen|i_k)) \tag{7}$$

where $disc(k)$ is a rank discount given by $disc(k) = .85^{k-1}$ and C is a normalizing constant given by $1/\sum_{i_k \in R}^{i_N} disc(k)$. Therefore, this metric is rank-sensitive (i.e. the novelty of the top-rated items counts more than the novelty of other items). As is the case with precision and recall, we average the EPC@N value of the top-N recommendation lists over the test set.

We used a distance based model [Vargas and Castells 2011] in order to measure the diversity of the recommendation lists without relevance-awareness. The recommendation diversity, therefore, is given by:

$$EILD@N = \sum_{i_k \in R, i_l \in R, l \neq k}^{i_N, l_N} C_k\, disc(k)\, disc(l|k)\, d(i_k, i_l) \tag{8}$$

where $disc(l|k) = disc(max(1, l - k))$ reflects a relative rank discount between l and k, and $d(i_k, i_l)$ is the cosine distance between two items, given by:

$$d(i, j) = 1 - \frac{|\mathbf{U_i} \cap \mathbf{U_j}|}{\sqrt{|\mathbf{U_i}|}\sqrt{|\mathbf{U_j}|}} \tag{9}$$

such that $\mathbf{U_i}$ denotes the users that liked item i, and $\mathbf{U_j}$ denotes the users that liked item j.

5.2. Datasets

We apply the methodology presented in the previous section to two different scenarios, in order to evaluate our Pareto-efficient approaches: movie and music recommendation. For movie recommendation, we used the MovieLens 1M dataset [Miller et al. 2003]. This dataset consists of 1,000,209 ratings from 6,040 users on 3,883 movies. For music recommendation, we used an implicit preference dataset provided by Ò.Celma [Celma and Herrera 2008], which consists of 19,150,868 user accesses to music tracks on the website Last.fm[2]. This dataset involves 176,948 artists and 992 users, and we considered the task of recommending artists to users. Mapping the implicit feedback into user-artist ratings yielded a total of 889,558 ratings, which were used by the algorithms that cannot deal with implicit feedback, and to separate the dataset into the training and test sets $\mathcal{M}$ and $\mathcal{T}$.

5.3. Recommendation Algorithms

We selected eight well-known recommendation algorithms to provide the base for our Pareto-efficient approaches. To represent latent factor models, we selected PureSVD with 50 and 150 factors (PureSVD50 and PureSVD150), described in [Cremonesi et al. 2010]. These were the only algorithms we used that are based on explicit feedback. To compute the scores for the items in the Last.fm dataset, we used the mappings of implicit feedback into ratings explained in Section 5.1.

As for recommendation algorithms that use implicit feedback, we used algorithms available in the MyMediaLite package [Gantner et al. 2011]. We used WeightedItemKNN (WIKNN) and WeightedUserKNN (WUKNN) as representative of neighborhood models based on collaborative data [Desrosiers and Karypis 2011] (we only used WeightedItemKNN on the MovieLens dataset, as MyMediaLite's implementation cannot yet handle datasets where the number of items is very large, which is the case in the Last.fm dataset). Further, we also used MyMediaLite's MostPopular implementation, which is the same as TopPop in [Cremonesi et al. 2010]. We also used WRMF $-$ a weighted matrix factorization method based on [Hu et al. 2008; Pan et al. 2008], which is very effective for data with implicit feedback. Finally, we used UserAttributeKNN(UAKNN), a K-nearest neighbor user-based collaborative filtering using cosine-similarity over the user attributes, such as sex, age etc. (which both datasets provide).

5.4. Baselines

We employed three baselines for the sake of comparison. The first baseline is a voting-based approach based on Borda-Count (BC) which is similar to [Pazzani 1999], where each constituent algorithm gives n points to each item i such that $n = |R| - p_i$, where $|R|$ is the size of the recommendation list and p_i is the position of i in R. The second baseline is STREAM, a stacking-based approach with additional meta-features [Bao et al. 2009]. We used the same additional meta-features as [Bao et al. 2009], namely,

[2] www.Last.fm

the number of items that a certain user has rated and the number of users that has rated a certain item (denoted as RM_1 and RM_2). We tried the learning algorithms proposed in [Bao et al. 2009], and Linear Regression yielded the best results, so the results presented for STREAM are generated using Linear Regression as the meta-learning algorithm. Our last baseline is the weighted hybrid we proposed in Section 4.1, using equal weights for each constituent algorithm. We called this baseline Equal Weights (EW).

5.5. Pareto Efficient Hybridization Details

We apply the algorithm described in Section 4 to both datasets, combining all of the recommendation algorithms described in subsection 5.3. We used an open-source implementation of SPEA2 [Zitzler and Thiele 1999; Zitzler et al. 2001] from DEAP[3]. We used a two points crossover operator [Holland 1975], and a uniform random mutation operator with probability .05. Table I presents SPEA-2's parameters, which were sufficient for convergence.

Table I. Parameters of the SPEA2 Algorithm

Parameters	MovieLens	Last.fm
Population Size	100	100
Gene dimension	7 algorithms	6 algorithms
# of Objectives	3	3
# of Generations	300	300
Mutation Rate	.2	.2
Crossover Rate	.5	.5

In order to speed up the fitness calculations, we ran all of the constituent algorithms on the cross validation test set and stored their predictions. Then, in order to evaluate the fitness of each individual, we combine the constituent algorithms with the appropriate weights and evaluate the results on the cross validation test set $\mathcal{V}$. It is worth remembering that $\mathcal{V}$ is a list of triples (u, i, s), where u is an user, i is an item that is relevant to u and s is a set of $1{,}000$ items that are unrated by u.

Each objective in the fitness function of a certain ranking R of the items $\{i\} + s$ provided by a certain individual is given by:

$$O(R) = \sum_{(u,i,s) \in \mathcal{V}} f(u, i, s, R) \qquad (10)$$

For the accuracy objective, $f(u, i, s, R)$ is defined as follows:

$$f(u, i, s) = 21 - max(21, R_i) \qquad (11)$$

where R_i is the position of item i in the ranking. This equation provides a way to value hits up to the 20th position, with more value being given to positions closer to the top.

As for the novelty objective $f(u, i, s, R)$ is simply $EPC@20(R)$. Similarly, for the diversity objective, $f(u, i, s, R)$ is equal to $EILD@20(R)$.

5.6. Results and Discussion

The results achieved by each of the constituent recommendation algorithms can be seen in Tables II and III. There is a clear compromise between accuracy, novelty and diversity of these algorithms. For the MovieLens dataset (Table II), the constituent

[3] Freely available at http://deap.googlecode.com

Table II. Results for Recommendation Algorithms on the MovieLens dataset, with the three objectives (i.e., accuracy, novelty, and diversity). The recommender methods variants are grouped into: (i) constituent algorithms, (ii) multi-objective baselines, and (iii) our proposed Pareto-efficient approaches. We used the symbols: †, ●, ◇ to point out our method and the respective baseline. For each group, the best results for each metric are in bold. Underlined values means that the selected approach and the respective baseline are statistically different (95%).

		Accuracy								Novelty	Diversity
	Algorithm	R@1	R@5	R@10	R@20	P@1	P@5	P@10	P@20	EPC@20	EILD@20
Const. Algorithms	PSVD50 †	**.1900**	**.4155**	**.5402**	**.6643**	**.1900**	**.0831**	**.0540**	**.0332**	.8070	.8650
	PSVD150 ●◇	.1237	.3203	.4450	.5658	.1237	.0641	.0445	.0283	.8519	**.8881**
	TopPop	.0722	.2061	.2895	.3994	.0722	.0412	.0289	.0200	.7079	.7905
	WRMF	.1513	.3453	.4545	.5674	.1513	.0691	.0455	.0284	.7847	.8394
	WIKNN	.1529	.3564	.4624	.5806	.1529	.0713	.0462	.0290	.7744	.8257
	WUKNN	.1510	.3364	.4437	.5707	.1510	.0673	.0444	.0285	.7560	.8216
	UAKNN	.0614	.1762	.2504	.3387	.0614	.0352	.0250	.0169	.7386	.8173
Baselines	STREAM	**.1792**	**.3961**	**.5169**	**.6426**	**.1792**	**.0792**	**.0517**	**.0321**	.8078	.8454
	BC	.0473	.1657	.2639	.4352	.0473	.0331	.0264	.0218	**.8210**	**.8698**
	EW	.1562	.3574	.4752	.5980	.1562	.0715	.0475	.0299	.7441	.8160
Our Approaches	PEH-mean † ● ◇	.1776	.4175	.5379	.6656	.1776	.0835	.0538	.0333	.8361	.8696
	PEH-acc †	.1959	.4161	.5399	**.6689**	.1959	.0832	.0540	**.0334**	.8188	.8565
	PEH-nov ●	.1415	.3656	.4857	.5917	.1415	.0731	.0486	.0296	.8649	.8964
	PEH-div ◇	.1309	.3223	.4263	.5297	.1309	.0645	.0426	.0265	**.8828**	.9047
	PER-dom † ● ◇	**.1979**	.3722	.4368	.4910	**.1979**	.0744	.0437	.0245	.8549	**.9060**
	PER-SVM † ● ◇	.1953	**.4296**	**.5540**	.6554	.1953	**.0852**	**.0554**	.0328	.8341	.8699

algorithm that provides the most accurate recommendations is PureSVD50. The constituent algorithm that provides the most novel and diverse recommendations, with an acceptable level of accuracy, is PureSVD150, but its accuracy is much worse than the accuracy obtained by PureSVD50. TopPop provided the worst performance numbers in all criteria used.

On the Last.fm dataset (Table III), the constituent algorithm that provides the most accurate recommendations is WRMF. This is expected, as Last.fm is originally an implicit feedback dataset, to which WRMF is more suitable. Once again, PureSVD150 proved its bias to suggest novel and diverse items, being the best constituent algorithm both in terms of novelty and diversity. In this dataset the compromise between the three objectives is once again illustrated by the fact that there is no algorithm that dominates the others in every objective.

Regarding the performance of the baselines in the MovieLens dataset, STREAM performs worse then PureSVD50 on accuracy and diversity, maintaining the same level of novelty. Borda Count performed poorly on accuracy, reasonably well in terms of novelty and diversity. Equal Weights performed poorly on accuracy and novelty, and well on diversity. On the Last.fm dataset, STREAM performed slightly worse than WRMF in accuracy, and slightly better in terms of diversity and novelty. Once again, Borda Count performed poorly on accuracy. Finally, Equal Weights performed poorly on accuracy, diversity and novelty.

Pareto-Efficient Ranking

Now we turn our attention to the evaluation of our Pareto-efficient ranking approaches. First, we evaluate the simpler approach, which we call PER-dom (Pareto-

Table III. Results for Recommendation Algorithms on the Last.fm dataset, with the three objectives (i.e., accuracy, novelty, and diversity). The recommender methods variants are grouped into: (i) constituent algorithms, (ii) multi-objective baselines, and (iii) our proposed Pareto-efficient approaches. We used the symbols: †, ●, ◇ to point out our method and the respective baseline. For each group, the best results for each metric are in bold. Underlined values means that the selected approach and the respective baseline are statistically different (95%).

		Accuracy								Novelty	Diversity
	Algorithm	R@1	R@5	R@10	R@20	P@1	P@5	P@10	P@20	EPC@20	EILD@20
Const. Algorithms	PSVD50	**.3859**	.5997	.6649	.7178	**.3859**	.1199	.0665	.0359	.8878	.9561
	PSVD150 ●◇	.3265	.5241	.6055	.6667	.3265	.1048	.0605	.0333	**.8998**	**.9617**
	TopPop	.1879	.4114	.5198	.6224	.1879	.0823	.0520	.0311	.8508	.9405
	WRMF †	.3834	**.6148**	**.7073**	**.7858**	.3834	**.1230**	**.0707**	**.0393**	.8735	.9471
	WUKNN	.3272	.5662	.6562	.7340	.3272	.1132	.0656	.0367	.8481	.9352
	UAKNN	.1922	.3790	.4712	.5328	.1922	.0758	.0471	.0266	.8605	.9424
Baselines	STREAM	**.3898**	**.6022**	**.6685**	**.7185**	**.3898**	**.1204**	**.0668**	**.0359**	**.8882**	**.9563**
	BC	.2973	.5346	.6026	.6692	.2973	.1069	.0603	.0335	.8606	.9414
	EW	.3017	.5850	.6785	.7595	.3017	.1170	.0679	.0380	.8473	.9363
Our Approaches	PEH-mean † ● ◇	.4230	**.6505**	**.7250**	**.7829**	.4230	**.1301**	**.0725**	**.0391**	.8908	.9514
	PEH-acc †	**.4323**	.6476	.7232	.7819	**.4323**	.1295	.0723	.0391	.8820	.9484
	PEH-nov ●	.3751	.5911	.6659	.7246	.3751	.1182	.0666	.0362	.9219	.9643
	PEH-div ◇	.3139	.5184	.5943	.6573	.3139	.1037	.0594	.0329	**.9388**	**.9713**
	PER-dom † ● ◇	.3866	.6310	.7127	**.7829**	.3866	.1262	.0713	.0388	.9016	.9561
	PER-SVM † ● ◇	.3851	.6062	.6972	.7264	.3851	.1212	.0691	.0363	.8838	.9516

Efficient Ranking with most dominant items first). Considering the Movielens dataset, we directly compared PER-dom against two different baselines: PSVD50 and PSVD150, since these algorithms were the best performers in terms of accuracy, novelty and diversity. PER-dom is significantly superior than PSVD50 in the top of the rank, but becomes significantly worse than PSVD50 as k increases. On the other hand, PER-dom greatly outperformed PSVD50 in terms of diversity and novelty. Also, PER-dom is better than PSVD150 in terms of novelty, and it greatly outperforms PSVD150 both in terms of accuracy and diversity. In fact, Per-dom was the best performer in terms of diversity. The more sophisticate approach, which we call PER-SVM (Pareto-Efficient Ranking with SVM), was evaluated using the same procedure as to PER-dom. PER-SVM is slightly superior than PVSD50 in all three objectives considered. Also, PER-SVM is much better than PSVD150 in terms of accuracy and diversity, and slightly better in terms of novelty. In summary, PER-SVM is a good choice for cases where all objectives are simultaneously important: it was not the best performer in any of the objectives, but its performance is close to the best performers in any of the objectives.

A similar trend is observed for the Last.fm dataset. We directly compared PER-dom against two different baselines: WRMF and PSVD150, since these algorithms presented the best numbers in terms of accuracy, novelty and diversity. PER-dom is significantly superior than WRMF, in terms of all objectives considered, and particularly better in terms of novelty. Further, PER-dom is much better than PSVD150 in terms of accuracy, and slightly better in terms of novelty, but it is significantly worse than PSVD150 in terms of diversity. PER-SVM performed similarly to PER-dom, both in terms of accuracy and diversity. Also, PER-SVM greatly outperforms PVSD150 in terms of accuracy, but PSVD150 is significantly better in terms of diversity and novelty. Finally, PER-SVM is slightly better than WRMF in all objectives considered. The

Table IV. Constituent algorithms' weights for different individuals, Movielens

Individual	PSVD50	PSVD150	TopPop	WRMF	WIKNN	WUKNN	UAKNN
PEH-mean	21.60	20.19	-14.91	8.83	0.36	13.92	-3.10
PEH-acc	21.55	7.80	-11.10	10.20	5.47	10.86	-3.98
PEH-nov	25.95	22.43	-5.19	0.04	-5.07	8.18	-7.48
PEH-div	25.95	23.43	-26.94	1.20	-5.86	16.90	-1.89

same conclusion holds for Last.fm, that both PER-dom and PER-SVM are good choices if all objectives must be maximized simultaneously.

Pareto-Efficient Hybridization

Now, with our hybridization approach, we could reach any of the individuals in Figure 3, which represents the accuracy (in this case, Recall@10) and novelty (EPC@20) of the recommendations in x and y axes, and diversity (EILD@20) with a color scale. It is clear that there is a compromise between accuracy and the other two objectives: the individuals with the most accurate recommendations provide less novel and diverse lists, and so on. This compromise can be adjusted dynamically with little extra cost, since the cost of reaching these individuals is as low as a linear search (for the individual that maximizes a weighted mean, as described in Section 4.3) over the Pareto frontier individuals' scores. The Pareto frontier consists of 510 individuals in the MovieLens dataset, and of 318 individuals in the Last.fm dataset, so a linear search can be done very quickly. We chose to demonstrate a few of these individuals in Tables II and III. First, PEH-mean (Pareto-Efficient Hybrid with mean weights) represents the individual that optimizes the mean of the three normalized objectives, assuming each of them are equally important. This would be an option if personalization was not desired, or if the designers of the recommender system do not know which combination of the three objectives would result in higher user satisfaction. However, in a more realistic scenario, the system designer would most likely want to select different individuals for different users. We selected as examples the following individuals, which were found by the process explained in Section 4.3 with the represented associated weighted vectors:

— PEH-acc:[Accuracy:0.70, Novelty:0.30, Diversity:0.00]
— PEH-nov: [Accuracy:0.15, Novelty:0.50, Diversity:0.35]
— PEH-div: [Accuracy:0.10, Novelty:0.35, Diversity:0.55]

These objective weights led to the algorithm weights presented in Table IV. It is worth noticing that even though some algorithms are always highly weighted (PSVD50, for example) and others are always weighted negatively (TopPop), there are significant differences between the weights of different individuals, which lead to completely different objective values. It is interesting to notice that weaker algorithms (such as WRMF, which in this dataset is worse than PSVD50 in all three objectives) are still able to play a significant role when the algorithms are combined.

We compared PEH-acc against PureSVD50, which is the most accurate constituent algorithm. It perform equally well or better than PureSVD on accuracy, but PEH-acc performs better on novelty and worse on diversity. We compared PEH-nov against PureSVD150, which presented the most novel recommendations to the users, with reasonable accuracy. PEH-nov performs better on all three objectives, when compared to PureSVD150 - particularly accuracy and novelty. Finally, we compared PEH-div with PureSVD150, the algorithm with the most diverse recommendations. PEH-div maintains (or slightly improves)the accuracy level, while improving a lot on both novelty and diversity. PEH-mean was an individual that balanced the three objectives,

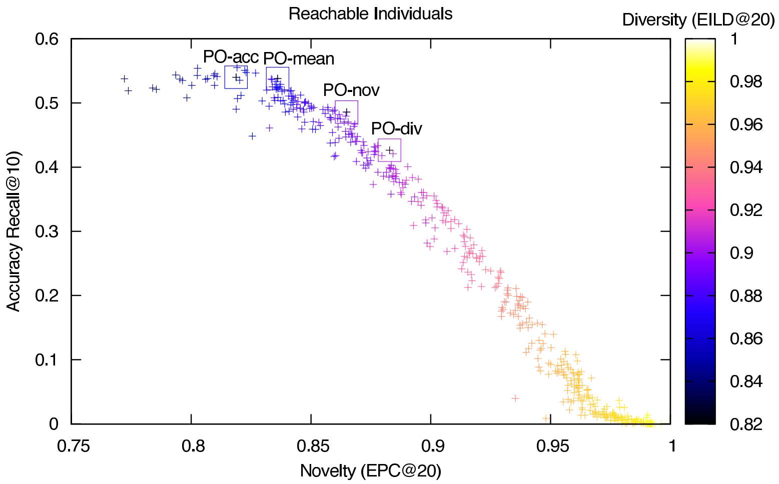

Fig. 3. (Color online) Individuals lying in the Pareto frontiers for Movielens.

performing much better than PureSVD150, but worse than PureSVD in accuracy, and better than PureSVD50 on novelty and diversity, but worse than PureSVD150. We were able to find individuals in the Pareto Frontier that performed at least as well as the best algorithms in each individual objective, but better on the other objectives. Once again, we could have chosen to compromise more accuracy if we desired even more novelty and diversity, as it is shown in Figure 3.

As for the Last.fm dataset, we selected the following individuals:

— PEH-acc: [Accuracy:0.70, Novelty:0.30, Diversity:0.00]
— PEH-nov: [Accuracy:0.15, Novelty:0.85, Diversity:0.00]
— PEH-div: [Accuracy:0.05, Novelty:0.45, Diversity:0.50]

These objective weights led to the algorithm weights presented in Table V. Once again, we notice that different priorities lead to very diverse algorithm weights, and that weaker algorithms (such as UAKNN) are able to play an important role when the algorithms are combined.

Table V. Constituent algorithms' weights for different individuals, Last.fm

Individual	PSVD50	PSVD150	TopPop	WRMF	WUKNN	UAKNN
PEH-mean	26.02	24.57	-10.53	23.21	2.77	-7.91
PEH-acc	26.02	22.43	-6.24	24.14	4.93	-7.36
PEH-nov	27.94	26.97	-9.51	13.01	-5.49	-8.60
PEH-div	26.02	21.81	-9.27	4.19	-1.90	-8.15

This time, we compared PEH-acc against WRMF, which is the most accurate constituent algorithm on this dataset. PEH-acc is much more accurate than WRMF, while also improving on novelty and performing almost as well on the diversity level. PEH-nov was compared against PureSVD150, and it performed much better on accuracy and novelty, while losing on the diversity. PEH-div was compared against PureSVD150, and it faired slightly worse on accuracy, while greatly improving on both novelty and diversity. PEH-mean was once again a balanced individual, although this time its accuracy was much better than any of the constituent algorithms. Once again, we were

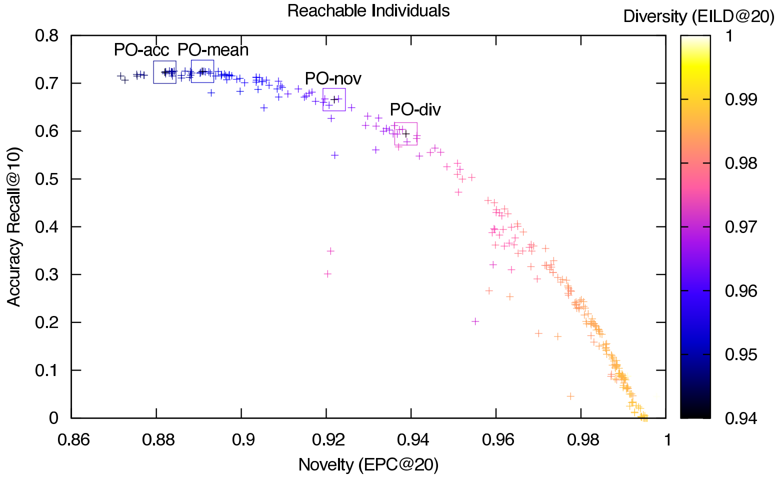

Fig. 4. (Color online) Individuals lying in the Pareto frontiers for Last.fm.

able to find effective individuals in the Pareto frontier, but we could have reached any of the individuals in Figure 4 by tweaking the weight value for each objective.

In summary, our proposed approaches are able to provided significant improvements when compared against other multi-objective approaches. Specifically, a comparison involving our best performers and the best baselines, reveals improvements in terms of accuracy (R@1), with gains ranging from 10.4% (on Last.fm) to 10.7% (on MovieLens), in terms of novelty, with gains ranging from 5.7% (Last.fm) to 7.5% (MovieLens), and also in terms of diversity, with gains ranging from 1.6% (Last.fm) to 4.2% (MovieLens).

5.7. Reproducibility

The datasets we have used in our experiments are freely available, and can be obtained following instructions in [Miller et al. 2003; Celma and Herrera 2008]. All constituent algorithms are implemented in the MyMediaLite package [Gantner et al. 2011]. The SVM-Rank implementation used in PER-SVM is freely available at http://svmlight.joachims.org/svm_rank.html. The evolutionary algorithm implementation we used to find Pareto-efficient hybrids is available at http://deap.googlecode.com.

6. CONCLUSIONS

In this paper we propose Pareto-efficient approaches for recommender systems where objectives such as accuracy, novelty and diversity must be maximized simultaneously. We show that existing recommendation algorithms do not perform uniformly well when evaluated in terms of accuracy, novelty and diversity, and thus we propose approaches that exploit the Pareto efficiency concept in order to combine such recommendation algorithms in a way that a particular objective is maximized without significantly hurting the other objectives. The Pareto-efficiency concept is exploited in two distinct manners: (i) items are placed in an n-dimensional space (i.e., n constituent algorithms) in which the coordinates are the scores assigned to the item by the algorithms. In this way, combining the constituent algorithms means maximizing all objectives simultaneously; (ii) hybrid algorithms (i.e., linear combination of the constituent algorithms) are placed in a 3-dimensional space in which the coordinates are the level of accuracy, novelty and diversity associated with each hybrid. Different hybrids may give emphasis to a particular objective, provided that this will not significantly hurt the other objectives. Our proposed Pareto-efficient approaches may be very useful in different scenarios. An obvious scenario is to provide better suggestions to the users, recommending items that are simultaneously accurate, novel and diverse. Another example is the personalization of recommendations according to particular users. For instance, new users may benefit from an algorithm which generates highly ratable items, as they need to establish trust and rapport with the recommender system before taking advantage of the suggestions it offers. The costly part of our Pareto-efficient approaches is performed entirely offline, and the online cost of choosing items or hybrids in the Pareto frontier is almost negligible, since the Pareto frontier is comprised of few items or hybrids.

We performed highly reproducible experiments on public datasets of implicit and explicit feedback, using open-source implementations. In our experiments, we demonstrated that the proposed approaches have either the ability to balance each of the objectives according to the desired compromise, or the ability to maximize all three objectives simultaneously. Finally, we show that the proposed approaches have obtained results that are competitive with the best algorithms according to each objective and almost always better on the other objectives.

REFERENCES

G. Adomavicius and Y. Kwon. 2007. New Recommendation Techniques for Multicriteria Rating Systems. *IEEE Intelligent Systems* 22, 3 (2007), 48–55.

G. Adomavicius, N. Manouselis, and Y. Kwon. 2011. Multi-Criteria Recommender Systems. In *Recommender Systems Handbook*, Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor (Eds.). Springer, 769–803.

G. Adomavicius and A. Tuzhilin. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering* 17, 6 (2005), 734–749.

X. Bao, L. Bergman, and R. Thompson. 2009. Stacking recommendation engines with additional meta-features. In *ACM Conference on Recommender Systems*. 109–116.

A. Bellogín, P. Castells, and I. Cantador. 2011. Precision-oriented evaluation of recommender systems: an algorithmic comparison. In *ACM Conference on Recommender Systems*. 333–336.

D. Billsus and M.J. Pazzani. 2000. User modeling for adaptive news access. *User modeling and user-adapted interaction* 10, 2 (2000), 147–180.

S. Börzsönyi, D. Kossmann, and K. Stocker. 2001. The Skyline Operator. In *IEEE International Conference on Data Engineering*. 421–430.

R. Burke. 2002. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction* 12, 4 (2002), 331–370.

P. Castells, J. Wang, R. Lara, and D. Zhang. 2011. Workshop on novelty and diversity in recommender systems - DiveRS 2011. In *ACM Conference on Recommender Systems*. 393–394.

Oscar Celma and Perfecto Herrera. 2008. A new approach to evaluating novel recommendations. In *ACM Conference on Recommender Systems*. 179–186.

M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes, and M. Sartin. 1999. Combining Content-Based and Collaborative Filters in an Online Newspaper. In *ACM SIGIR Workshop on Recommender Systems*. 40–48.

D. Corne, J. Knowles, and M. Oates. 2000. The Pareto Envelope-Based Selection Algorithm for Multi-objective Optimisation. In *Parallel Problem Solving from Nature*. 839–848.

P. Cremonesi, Y. Koren, and R. Turrin. 2010. Performance of recommender algorithms on top-n recommendation tasks. In *ACM Conference on Recommender Systems*. 39–46.

K. Deb. 1999. Multi-objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems. *Evolutionary Computation* 7, 3 (1999), 205–230.

C. Desrosiers and G. Karypis. 2011. A Comprehensive Survey of Neighborhood-based Recommendation Methods. In *Recommender Systems Handbook*, Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor (Eds.). Springer, 107–144.

N. Dokoohaki, C. Kaleli, H. Polat, and M. Matskin. 2010. Achieving Optimal Privacy in Trust-Aware Social Recommender Systems. In *International Conference on Social Informatics*. 62–79.

F. Fouss and M. Saerens. 2008. Evaluating Performance of Recommender Systems: An Experimental Comparison. In *International Conference on Web Intelligence*. 735–738.

Z. Gantner, S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. 2011. MyMediaLite: a free recommender system library. In *ACM Conference on Recommender Systems*. 305–308.

N. Garg and I. Weber. 2008. Personalized, interactive tag recommendation for flickr. In *ACM Conference on Recommender Systems*. 67–74.

M. Ge, C. Delgado-Battenfeld, and D. Jannach. 2010. Beyond accuracy: evaluating recommender systems by coverage and serendipity. In *ACM Conference on Recommender Systems*. 257–260.

E. Goldberg. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc.

Z. Guan, J. Bu, Q. Mei, C. Chen, and C. Wang. 2009. Personalized tag recommendation using graph-based ranking on multi-type interrelated objects. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*. 540–547.

A. Gunawardana and C. Meek. 2008. Tied boltzmann machines for cold start recommendations. In *ACM Conference on Recommender Systems*. 19–26.

A. Gunawardana and C. Meek. 2009. A unified approach to building hybrid recommender systems. In *ACM Conference on Recommender Systems*. 117–124.

I. Guy, N. Zwerdling, I. Ronen, D. Carmel, and E. Uziel. 2010. Social media recommendation based on people and tags. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*. 194–201.

J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. 2004. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems* 22, 1 (2004), 5–53.

J.H. Holland. 1975. *Adaptation in natural and artificial systems*. Number 53. University of Michigan Press.

Y. Hu, Y. Koren, and C. Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *IEEE International Conference on Data Mining*. 263–272.

N. Hurley and M. Zhang. 2011. Novelty and Diversity in Top-N Recommendation - Analysis and Evaluation. *ACM Transactions on Internet Technology* 10, 4 (2011), 14.

T. Joachims. 2002. Optimizing search engines using clickthrough data. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 133–142.

T. Joachims. 2006. Training linear SVMs in linear time. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 217–226.

N. Kawamae. 2010. Serendipitous recommendations via innovators. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*. 218–225.

Y. Koren and J. Sill. 2011. OrdRec: an ordinal model for predicting personalized item rating distributions. In *ACM Conference on Recommender Systems*. 117–124.

N. Lathia, S. Hailes, L. Capra, and X. Amatriain. 2010. Temporal diversity in recommender systems. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*. 210–217.

H. Lee and W. Teng. 2007. Incorporating Multi-Criteria Ratings in Recommendation Systems. In *IEEE International Conference on Information Reuse and Integration*. 273–278.

G. Lekakos and P. Caravelas. 2008. A hybrid approach for movie recommendation. *Multimedia tools and applications* 36, 1 (2008), 55–70.

K. Leung, D. Lee, and W. Lee. 2011. CLR: a collaborative location recommendation framework based on co-clustering. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*. 305–314.

L. Li, D. Wang, T. Li, D. Knox, and B. Padmanabhan. 2011. SCENE: a scalable two-stage personalized news recommendation system. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*. 125–134.

X. Lin, Y. Yuan, Q. Zhang, and Y. Zhang. 2007. Selecting Stars: The k Most Representative Skyline Operator. In *IEEE International Conference on Data Engineering*. 86–95.

S. McNee, J. Riedl, and J. Konstan. 2006. Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *Conference on Human Factors in Computing Systems, Extended Abstracts*. 1097–1101.

G. Menezes, J. Almeida, F. Belém, M Gonçalves, A. Lacerda, E. de Moura, G. Pappa, A. Veloso, and N. Ziviani. 2010. Demand-Driven Tag Recommendation. In *European Conference on Machine Learning and Knowledge Discovery in Databases*. 402–417.

B. N. Miller, I. Albert, S. K. Lam, J. A. Konstan, and J. Riedl. 2003. MovieLens unplugged: experiences with an occasionally connected recommender system. In *International Conference on Intelligent User Interfaces*. 263–266.

J. Naruchitparames, M. Gunes, and S. Louis. 2011. Friend recommendations in social networks using genetic algorithms and network topology. In *IEEE Congress on Evolutionary Computation*. 2207–2214.

F. Palda. 2011. *Pareto's Republic and the new Science of Peace*. Cooper-Wolfling. 128 pages.

R. Pan, Y. Zhou, B. Cao, N. Liu, R. Lukose, M. Scholz, and Q. Yang. 2008. One-Class Collaborative Filtering. In *IEEE International Conference on Data Mining*. 502–511.

D. Papadias, Y. Tao, G. Fu, and B. Seeger. 2003. An Optimal and Progressive Algorithm for Skyline Queries. In *ACM SIGMOD International Conference on Management of Data*. 467–478.

M.J. Pazzani. 1999. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review* 13, 5 (1999), 393–408.

S. Rendle, Z. Gantner, C. Freudenthaler, and L. Schmidt-Thieme. 2011. Fast context-aware recommendations with factorization machines. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*. 635–644.

M. T. Ribeiro, A. Lacerda, A. Veloso, and N. Ziviani. 2012. Pareto-efficient hybridization for multi-objective recommender systems. In *ACM Conference on Recommender Systems*. 19–26.

G. Shani and A. Gunawardana. 2011. Evaluating Recommendation Systems. In *Recommender Systems Handbook*, Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor (Eds.). Springer, 257–297.

N. Srinivas and K. Deb. 1994. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation* 2, 3 (1994), 221–248.

B. Surbjörnsson and R. van Zwol. 2008. Flickr tag recommendation based on collective knowledge. In *International World Wide Web Conference*. 327–336.

S. Vargas and P. Castells. 2011. Rank and relevance in novelty and diversity metrics for recommender systems. In *ACM Conference on Recommender Systems*. 109–116.

J. Wang and Y. Zhang. 2011. Utilizing marginal net utility for recommendation in e-commerce. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1003–1012.

L. Zhang, K. Zhang, and C. Li. 2008. A topical PageRank based algorithm for recommender Systems. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*. 713–714.

K. Zhou, S. Yang, and H. Zha. 2011. Functional matrix factorizations for cold-start recommendation. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*. 315–324.

T. Zhou, Z. Kuscsik, J. Liu, M. Medo, J. Wakeling, and Y. Zhang. 2010. Solving the apparent diversity-accuracy dilemma of recommender systems. *National Academy of Sciences* 107, 10 (2010), 4511–4515.

C. Ziegler, S. McNee, J. Konstan, and G. Lausen. 2005. Improving recommendation lists through topic diversification. In *International World Wide Web Conference*. 22–32.

E. Zitzler, M. Laumanns, and L. Thiele. 2001. *SPEA2: Improving the Strength Pareto Evolutionary Algorithm*. Technical Report 103. 95–100 pages.

E. Zitzler and L. Thiele. 1999. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation* 3, 4 (1999), 257–271.