Exploiting Regression Trees as User Models for Intent-Aware Multi-attribute Diversity

Paolo Tomeo[1], Tommaso Di Noia[1], Marco de Gemmis[2], Pasquale Lops[2],
Giovanni Semeraro[2], Eugenio Di Sciascio[1]
[1] Polytechnic University of Bari – Via Orabona, 4 – 70125 Bari, Italy
[2] University of Bari Aldo Moro – Via Orabona, 4 – 70125 Bari, Italy
[1]{firstname.lastname}@poliba.it [2]{firstname.lastname}@uniba.it

ABSTRACT

Diversity in a recommendation list has been recognized as one of the key factors to increase user's satisfaction when interacting with a recommender system. Analogously to the modelling and exploitation of query intent in Information Retrieval adopted to improve diversity in search results, in this paper we focus on eliciting and using the profile of a user which is in turn exploited to represent her intents. The model is based on regression trees and is used to improve personalized diversification of the recommendation list in a multi-attribute setting. We tested the proposed approach and showed its effectiveness in two different domains, i.e. books and movies.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Search and Retrieval

Keywords

Personalized diversity; Intent-aware diversification; Regression Trees

1. INTRODUCTION

In the recent years, diversification has gained more and more importance in the field of recommender systems. Engines able to get excellent results in terms of accuracy of results have been proved to be not effective when we consider other factors related to the quality of user experience [10]. As a matter of fact, when interacting with a system exposing a recommendation service, the user perceives as good suggestions those showing also an appropriate degree of diversity, novelty or serendipity, just to cite a few. The attitude of populating the recommendation list with similar items could exacerbate the over-specialization problem that content-based recommender systems tend to suffer from [9], even though it appears also in collaborative-filtering approaches. Improving diversity is generally a good choice to foster the user satisfaction as it increases the odds of finding relevant recommendations [1].

Here our focus is on both the *individual* (or *intra-list*) diversity, namely the degree of dissimilarity among all items in the list provided to a user, and the *aggregate diversity* [3], namely the number and distribution of distinct items recommended across all users. The item-to-item dissimilarity can be evaluated by using content-based attributes (e.g. genre in movie and music domains, product category in e-commerce) [18] or statistical information (e.g. number of co-ratings) [23]. Usually, approaches to the diversification take into account only one single attribute while, in the approach we present here, multiple attributes are selected to describe the items. The rationale behind this choice is that we believe there are numerous and heterogeneous item dimensions conditioning user's interests and choices. Moreover, depending on the user these dimensions may interact with each other thus contributing to the creation of her intents. The question is how to tackle multiple attributes to address the diversification problem.

In this paper we use regression trees as user modeling technique to infer the individual interests, useful to provide an intent-aware diversification. Compared to approaches where item attributes are treated independently one to each other, regression trees make possible to represent user tastes as a combination of interrelated characteristics. For instance, a user could have a preference for horror movies of the 80s irrespective of the director, or for horror movies of the 90s directed by a a specific director. In a regression tree, conditional probability lets to build such inference rules about user's preferences. We conducted experiments on the movie and on the book domains to empirically evaluate our approach. The performance was measured in terms of accuracy and both individual and aggregate diversity.

The main contributions of this paper are:

- a novel intent-aware diversification approach able to combine multiple attributes. It bases on the use of regression trees (and rules) to infer and encode the model of users' interests;
- a novel method to combine different diversification approaches;
- an experimental evaluation which shows the performance of the proposed approaches with respect to both accuracy and diversity measures.

The paper is organized as follows. Section 2 describes the greedy approach to diversification problem, the xQuAD algorithm and some evaluation metrics. We then continue in

CBRecSys 2015, September 20, 2015, Vienna, Austria.

Section 3 by showing how to face the multi-attribute diversification and how to leverage regression trees in the diversification process with xQuAD to provide more personalized recommendations. Section 4 describes the experimental configuration and the datasets used for the experiments while Section 5 presents and describes the experimental results, showing the competitive performance of the proposed approach. In Section 6 we review the related work at the best of our knowledge. Conclusions close the paper.

2. DIVERSITY IN RECOMMENDATIONS

The recommendation step can be followed by a re-ranking phase finalized to improve other qualities besides accuracy [3]. Some of re-ranking approaches proposed so far are based on greedy algorithms designed to handle the balance between accuracy and diversity in a recommendations list [26]. Their scheme of work is explained through Algorithm 1, where $\mathbf{P} = \langle 1, ..., n \rangle$ is the recommendation list for user u generated using the predicted ratings and the output is the re-ranked list $\mathbf{S}$ of recommendations, such that $\mathbf{S} \subset \mathbf{P}$ and whose length is $N \leq n$.

Data: The original recommendation list $\mathbf{P}$, $N \leq n$
Result: The re-ranked recommendation list $\mathbf{S}$

1 $\mathbf{S} = \langle\rangle$;
2 **while** $|\,\mathbf{S}\,| \leq N$ **do**
3 $\quad i^* = \underset{i \in \mathbf{P} \setminus \mathbf{S}}{\operatorname{argmax}}\ f_{obj}(i, \mathbf{S})$;
4 $\quad \mathbf{S} = \mathbf{S} \circ i^*$;
5 $\quad \mathbf{P} = \mathbf{P} \setminus \{i^*\}$
6 **end**
7 **return S**.

Algorithm 1: The greedy strategy

At each iteration, the algorithm selects the item maximizing the objective function f_{obj} (line 3) – which in turn can be defined to deal with the trade-off between accuracy and diversity – and then adds it to the re-ranked list (line 4).

For our purpose, we focus on the intent-aware approach xQuAD (eXplicit Query Aspect Diversification), with the aim to diversify the user intents. It was proposed for search diversification in information retrieval by Santos et al. [15], as a probabilistic framework to explicitly model an ambiguous query as a set of sub-queries that will cover the potential aspects of the initial query. Then it was adapted for recommendation diversification by Vargas and Castells [20], replacing query and relative aspects with user and items categories, respectively. Hereafter we refer to generic *item features* - such as categories - as *features*, considering the features as possible instances of a generic attribute.

More formally, xQuAD greedily selects diverse recommendations maximizing the following objective function:

$$f_{obj}(i, \mathbf{S}, u) = \lambda\ r^*(u, i)\ +\ (1 - \lambda) div(i, \mathbf{S}, u) \qquad (1)$$

with $r^*(u, i)$ being the score predicted by the baseline recommender; the λ parameter allowing to manage the accuracy-diversity balance, where higher values give more weight to accuracy, while lower values give more weight to diversity. The last component in Equation 1 promotes the diversity, providing a measure of *novelty* with respect to the items already selected in S. As for the function $div(i, \mathbf{S}, u)$, the original formulation in [20] is:

$$div^{orig}(i, \mathbf{S}, u) = \sum_f p(i|f) p(f|u) \prod_{s \in \mathbf{S}} (1 - p(s|f)) \qquad (2)$$

where *p(i|f)* represents the likelihood of item i being chosen given the feature f while *p(f|u)* represents the user interest in the feature.

A number of measures have been proposed to evaluate the diversity in a recommendation list. Smyth and McClave [17] proposed the ILD (Intra-List Diversity), that computes the average distance between each couple of items in the list L:

$$ILD(L) = \frac{1}{|L|\,(|L| - 1)} \sum_{i,j \in L, i \neq j} (1 - sim(i, j)) \qquad (3)$$

The *sim* function is a configurable and application-dependent component which can use content-based item features or statistical information (e.g. number of co-ratings) to compute the similarity between items. We used also the metric α-nDCG, that is the redundancy-aware variant of Normalized Discounted Cumulative Gain proposed in [5]. We adopt the adapted version for recommendation proposed in [16]:

$$\alpha\text{-nDCG}(L, u) = \frac{1}{\alpha\text{-iDCG}} \sum_{r=1}^{|L|} \frac{\sum_{f \in F(L_r)} (1 - \alpha)^{cov(L, f, r-1)}}{log_2(1 + r)} \qquad (4)$$

where $cov(L, f, r - 1)$ is the number of items ranked up to position $r - 1$ containing the feature f. $F(L_r)$ represents the set of features of the r-th item. The α parameter is used to balance the emphasis between relevance and diversity. α-iDCG denotes the value of α-nDCG for the best "ideally" diversified list. Considering that the computation of the ideal value is NP-complete [5], we adopt a greedy approach: at each step we select solely the item with the highest value, regardless of the next steps.

3. INTENT-AWARE MULTI-ATTRIBUTE DIVERSITY

In this section we show how we address the intent-aware diversity problem when dealing with multi-attribute item descriptions. The presentation relies on content-based attributes (e.g. genres, years, etc. in the movies domain), but the proposed approach can be used independently of the attributes types. Therefore, one could also use statistical information as item attributes, e.g., popularity or rating variance. As explained in the previous section, we refer to features as possible instances of a generic attribute. We tried different reformulations of the *div* function in xQuAD (Equation 2) to deal with multi-attribute values. After an empirical evaluation, we chose the best div^{ma} (for **m**ulti-**a**ttribute) in terms of accuracy-diversity balance:

$$div^{ma}(i, \mathbf{S}, u) = \sum_{A \in \mathcal{A}} \frac{\sum_{f \in dom(A)} p(i|f) p(f|u) (1 - \text{avg}_{j \in \mathbf{S}}\, p(j|f))}{\sum_{f \in dom(A)} p(f|u)} \qquad (5)$$

where:

- $\mathcal{A}$ is the set of attributes;
- for each attribute $A \in \mathcal{A}$ and each feature in the attribute domain $f \in dom(A)$, $p(i|f)$ represents the importance of f for the item i. It is computed as a binary function that returns 1 if the item contains f, 0 otherwise;
- $p(f|u)$ represents the importance of the feature f for the user u and is computed as the relative frequency of the feature f on the rated items from the user u.

Here after we will refer to xQuAD using Equation 5 as *basic xQuAD*.

Besides dealing with multi-attribute descriptions, the idea behind our approach is to infer and model the user profile by means of a regression tree, a predictive model where the user interest represents the target variable, which can take continuous values. Once a regression tree is produced for a user u, then it is converted into a set of rules $RT(u)$. Each rule maps the presence/absence of a categorical feature or a constraint on a numerical one to a value v in a continuous interval. This latter indicates the predicted interest of the user on the items satisfying the rule. In our implementation we used the interval $[1, 5]$ since the value of the target variable has been calculated as the rating mean of the training instances classified by the inferred rule. Please note that the choice of a specific value interval for the target variable does not affect the overall approach. Each rule m has then the form

$$body(m) \mapsto interest = v$$

with $body(m) = \{c_1, \ldots, c_n\}$. An example of a set of rules produced for a user is shown in Figure 1.

1. $\{horror \in dom(genres),\ western \notin dom(genres),$ $DarioArgento \in dom(directors)\} \mapsto interest = 4.2$
2. $\{horror \notin dom(genres),\ thriller \in dom(genres)\}$ $\mapsto interest = 2.1$
3. $\{year > 1990,\ horror \notin dom(genres),$ $drama \in dom(genres),\ Aronofsky \in dom(directors)\}$ $\mapsto interest = 4.0$
4. $\{year < 1990,\ drama \in dom(genres),$ $AlPacino \in dom(actors)\} \mapsto interest = 3.9$
5. $\{horror \notin dom(genres)\} \mapsto interest = 3.2$

Figure 1: Example of a set of rules generated via the regression tree

Eventually, under the assumption that they represent specific user interests, the computed rules are used in the re-ranking phase as item features to improve the intent-aware recommendation diversity.

We propose also a *div* function for xQuAD so that each item is evaluated according to the rules it satisfies.

$$div^{rules}(i, \mathbf{S}, u) = \sum_{m \in M(u,i)} p(m|u)(1 - \text{avg}_{j \in \mathbf{S}}\, p(j|m)) \quad (6)$$

Here $M(u, i)$ represents the set of rules for the user u matched by the item i while $p(m|u)$ represents the importance of the rule m for u and is computed as:

$$p(m|u) = \frac{interest_m}{|M(u, i)|} \quad (7)$$

In Equation 7, $interest_m$ is the normalized predicted outcome of the regression tree for the rule m. Finally, the last component in Equation 6 indicates the complement of the coverage of the rule among the already selected recommendations. We propose two different versions of this adapted xQuAD.

- **RT.** $p(j|m)$ is a binary function that returns 1 if the item j matches the rule, 0 otherwise.
- **DivRT.** $p(j|m)$ is the average similarity between m and each rule covered by item j. More formally:

$$p(j|m) = \text{avg}_{m' \in M(u,j)}\, sim(m, m') \quad (8)$$

The rationale behind this formulation is that some rules may be similar with each other thus not bringing any actual diversification if considered separately. The computation of $sim(m, m')$ takes into account the overlapping between the rules m and m' as follows:

$$sim(m, m') = \frac{\sum_{c_i \in body(m)} overlap(m, m', c_i)}{max(|body(m)|, |body(m')|)}$$

For instance, considering the attributes represented in Figure 1, we have for $actor, genre$ and $director$:

$$overlap(m, m', c_i) = \begin{cases} 1, & c_i \in body(m) \wedge c_i \in body(m') \\ 0, & \text{otherwise} \end{cases}$$

For the numerical attribute $year$ we may adopt a different formulation for the function $overlap(m, m', c_i)$. Here we compute, if any, the overlap between the interval in $body(m)$ and the one in $body(m')$ normalized with respect to maximum interval's length. As an example, if $year > 1990$ is in $body(m)$ and $year < 2010$ is in $body(m')$ we may define the overlapping function as $overlap(m, m', c_i) = \frac{|1990-2010|}{max(dom(year))-min(dom(year))}$.

The functions introduced above have been used in the experimental setting in order to compute the function $overlap(m, m', c_i)$ (see Section 4).

RT and DivRT can be used instead of the basic xQuAD as diversification algorithms in the re-ranking phase. Alternatively, basic xQuAD and RT or DivRT can be pipelined to benefit from the strengths of them both. For instance, one could use xQuAD to select 50 diversified recommendations and then RT to select 20 recommendations from those 50, or vice versa. Hereafter, we use the syntax X-after-Y, e.g. xQuAD-after-RT, to indicate that algorithm X is executed on the results of Y.

4. EXPERIMENTS

We carried out a number of experiments to evaluate the performance of the methods presented in the Section 3 on two well known datasets: MovieLens1M and LibraryThing.

`MovieLens 1M`[1] dataset contains 1 million ratings from 6,040 users on 3,952 movies. The original dataset contains information about genres and year of release, and was enriched with further attribute information such as actors and directors extracted from DBpedia[2]. More details about this DBpedia enriched version of the dataset are available in [11]. Because not all movies have a corresponding resource in DBpedia, the final dataset contains 998,963 ratings from 6,040 users on 3,883 items. We built training and test sets by employing a 60%-40% temporal split for each user.

Moreover, we used the `LibraryThing`[3] dataset, which contains more than 2 million ratings from 7,279 users on 37,232 books. As in the dataset there are many duplicated ratings,

[1]Available at `http://grouplens.org/datasets/movielens`
[2]`http://dbpedia.org`
[3]Available at `http://www.macle.nl/tud/LT`

when a user has rated more than once the same item, we selected her last rating. The unique ratings are 749,401. Also in this case, we enriched the dataset by mapping the books with BaseKB[4], the RDF version of Freebase[5] and then extracting three attributes: *genre*, *author* and *subjects*. The subjects in Freebase represent the topic of the book, for instance Pilot experiment, Education, Culture of Italy, Martin Luther King and so on. The dump of the mapping is available online[6]. The final dataset contains 565,310 ratings from 7,278 users on 27,358 books. We built training and test sets by employing a 80%-20% hold-out split. The different ratio used for LibraryThing respect to Movielens (60%-40%) depends on its higher sparsity: holding 80% to build the user profile ensures a sufficient number of ratings to train the system.

	Movielens	**LibraryThing**
Number of users	6,040	7,278
Number of items	3,883	27,358
Number of ratings	998,963	565,310
Data sparsity	95.7%	99.7%
Avg users per item	275.57	20.66
Avg items per user	165.39	77.68

Table 1: Statistics about the two datasets

Since the number of distinct values was too large for year, actors and director attributes in MovieLens and for all the attributes in LibraryThing, we convert years in the corresponding decades and performed a K-means clustering for other attributes on the basis of DBpedia categories[7] for MovieLens and Freebase categories[8] for LibraryThing. Table 2 and 3 report the number of attribute values and clusters. The number of clusters was decided according to the calculation of the within-cluster sum of squares (*withinss* measure from the R Stats Package, version 2.15.3), that is picking the value of K corresponding to an evident break in the distribution of the *withinss* measure against the number of clusters extracted.

	Num. Values	**Num. Clusters**
Genres	19	-
Decades	10	-
Actors	14736	20
Directors	3194	20

Table 2: Statistics about MovieLens attributes

	Num. Values	**Num. Clusters**
Genres	270	30
Authors	12868	22
Subjects	2911	20

Table 3: Statistics about LibraryThing attributes

[4]http://basekb.com
[5]https://www.freebase.com
[6]http://sisinflab.poliba.it/semanticweb/lod/recsys/datasets/BaseKB2LibraryThing.zip
[7]http://purl.org/dc/terms/subject
[8]http://www.w3.org/1999/02/22-rdf-syntax-ns#type

4.1 Experimental Configuration

For both datasets, we used the Bayesian Personalized Ranking Matrix Factorization algorithm (BPRMF) available in MyMediaLite[9] as baseline (using the default parameters). We performed experiments using other recommendation algorithms, but we do not report results here since they are very similar to those obtained by BPRMF.

We selected the top-200 recommendations for each user to generate the initial list P used for performing the re-ranking as shown in Algorithm 1.

Accuracy is measured in terms of Precision, Recall and nDCG, but we only report nDCG values since the trend of the other two metrics is very similar. Individual diversity is measured using ILD and α-nDCG (see Section 2) with $\alpha = 0.5$ to equally balance diversity and accuracy, while aggregate diversity is measured using both the catalog coverage – computed as the percentage of items recommended at least to one user – and the entropy – computed as in [3] to analyse the distribution of recommendations among all users. These two last metrics need to be considered together, since the coverage gives a indication about the ability of a recommender to cover the items catalog and the entropy shows the ability to equally spread out the recommendations across all the items. Hence, only an improvement of both those metrics indicates a real increasing of aggregate diversity, that in turn denotes a better personalization of the recommendations [3].

As similarity measure for computing the ILD metric (Equation 3) we used the Jaccard index. Considering that there are more attributes for each item, we computed the average of the Jaccard index value for each attribute shared between two items. α-nDCG is computed as the average of the Equation 4 for each attribute.

As presented in Section 3, we propose two novel diversification approaches: RT and DivRT. We also propose a method to combine in sequence different algorithms by means of a two phase re-ranking procedure, with the aim of benefiting from the strengths of both. Therefore we evaluated other two approaches: xQuAD-after-RT and RT-after-xQuAD, applying the second re-ranking phase on the set of 50 recommendations provided from the first phase. We have also evaluated the combination with xQuAD and DivRT, but the results are very similar using RT, so they will not be shown. To evaluate the performances, we compare the top-10 recommendation list generating from all the approaches with basic xQuAD, by varying the λ parameter from 0 to 0.95 with step fixed to 0.05 in Equation 1 (higher values of λ give more weight to accuracy, lower values to diversity).

The rules are produced using M5Rules[10] algorithm available in Weka based on the M5 algorithm proposed by Quinlan [12] and improved by Wang and Witten [22]. M5Rules generates a list of rules for regression problems using a separate-and-conquer learning strategy. Iteratively it builds a model tree using M5 and converts the best leaf into a rule. We decided to use unpruned rules in order to have more rules matchable with the items.

[9]http://mymedialite.net/
[10]http://weka.sourceforge.net/doc.dev/weka/classifiers/rules/M5Rules.html

5. RESULTS DISCUSSION

Results of the experiments on MovieLens and LibraryThing are reported in Figure 2 and 3, respectively.

MovieLens. xQuAD obtains the best results in terms of ILD (Figure 2(a)) and α-nDCG (Figure 2(b)), though the xQuAD-after-RT results are very close and, with higher λ values (namely giving more importance to the accuracy factor), the differences between them are not significant. This outcome is due to the fact that the diversity metrics are attribute-based and xQuAD operates directly diversifying the attributes values, while the proposed rule-based approaches do not take into account all the attributes values. This also explains why the pure rule-based approaches (RT and DivRT) obtain the worst diversity results, while the combined algorithms (xQuAD-after-RT and RT-after-xQuAD) obtain better results. It is noteworthy that these last two configurations have no substantial difference with ILD, but, in terms of α-nDCG, xQuAD-after-RT considerably overcomes RT-after-xQuAD. This demonstrates that the pipeline of xQuAD and the rule-based approach obtains good diversity. Considering coverage (Figure 2(c)) and entropy (Figure 2(d)) to evaluate the aggregate diversity, the results show that using the rules the recommendations are much more personalized. It is interesting to note the compromise provided by xQuAD-after-RT, that obtains equidistant results between xQuAD and the rule-based algorithms, unlike RT-after-xQuAD that slightly overcomes xQuAD. With respect to the baseline, no configuration is able to give more accurate recommendations (nDCG = 0.14); all are able to increase the individual diversity (ILD = 0.34 and α-nDCG = 0.27). With nDCG and the individual diversity, the differences are always statistically significant ($p < 0.001$), except using the pure ruled-based approaches with $\lambda > 0.65$. The situation is more complex in terms of aggregate diversity, since the coverage grows very little on the baseline (coverage = 0.29) and the entropy slightly decreases (entropy = 0.78) with higher λ values. According to a comprehensive analysis on MovieLens, the pure rule-based approaches may give personalized and diversified recommendations, also with small accuracy loss. However, when individual diversity is more important than aggregate diversity, combining xQuAD with a previous rule-based re-ranking gives a good compromise between individual and aggregate diversity.

LibraryThing. At first glance, the LibraryThing results appear similar to those on MovieLens. Although they are generally consistent, there are interesting differences. Also in this case, xQuAD obtains the best diversity values, with ILD (Figure 3(a)) and α-nDCG (Figure 3(b)). However, both the combined approaches obtain really interesting results, very close to xQuAD, except for the lower λ values (namely giving more importance to the diversification factor). Unlike what happens on MovieLens, in this case RT-after-xQuAD obtains good results also in terms of α-nDCG. The pure rule-based approaches still obtain worse results. Considering coverage (Figure 3(c)) and entropy (Figure 3(d)) to evaluate the aggregate diversity, the results show that using the rules the recommendations are much more personalized than using only xQuAD. The combined approaches are able to improve the aggregate diversity with respect to xQuAD, albeit they are still distant from the pure rule-based approaches, especially in terms of coverage. With respect to the baseline, all configurations give a little more accurate recommendations, with $\lambda > 0.65$, but the differences are not statistically significant. In terms of individual diversity, all of them are able to overcome the baseline (ILD = 0.4 and α-nDCG = 0.285) except when using the pure rule-based approaches in terms of ILD. However they are able to improve α-nDCG. For the latter two metrics, the differences are always statistically significant ($p < 0.001$). In terms of aggregate diversity, xQuAD does not improve the baseline result (coverage = 0.15 and α-nDCG = 0.77), while using the rules leads to better results. According to a comprehensive analysis on LibraryThing, the pure rule-based approaches may give more personalized recommendations with a better diversity, especially using RT, with also a small accuracy loss. Similarly to the analysis on MovieLens, the results on LibraryThing suggest that diversifying with only the rules is a good choice when aggregate diversity is more important than individual diversity, conversely xQuAD remains the best choice to improve the individual diversity and combined with the rule-based diversification improves also the aggregate diversity.

The final conclusions of this analysis are that using a regression tree to infer rules representing user interests on multi-attribute values in the diversification process with xQuAD leads to more personalized recommendations but with a less diversified list and that combining attribute-based and rule-based diversifications in two phase re-ranking is a good way for taking the advantages of both. The better degree of personalization may depend on the fact that the rules are different among the users since they represents their individual interests. The lower individual diversity values with ILD and α-nDCG are due to the nature of these metrics which are based directly on the attributes values while the pure rule-based approaches do not take into account all the attributes values.

6. RELATED WORK

There is a noteworthy effort by the research community in addressing the challenge of recommendation diversity. That interest arises from the necessity of avoiding monotony in recommendations and controlling the balance between accuracy and diversity, since increasing diversity inevitably puts accuracy at risk [25]. However, a user study in the movie domain [7] demonstrates that user satisfaction is positively dependent on diversity and there may not be the intrinsic trade-off when considering user perception instead of traditional accuracy metrics.

Typically, the proposed approaches aim to replace items in an already computed recommendation list, by minimizing the similarity among all items. Some approaches exploit a re-ranking phase with a greedy selection (see Section 2), for instance [18], or with other techniques such us the Swap algorithm [23], which starts with a list of K scoring items and swaps the item which contributes the least to the diversity of the entire set with the next highest scoring item among the remaining items, by controlling the drop of the overall relevance by a pre-defined upper bound.

Other types of approaches try to directly generate diversified recommendation lists. For instance, [2] proposes a probabilistic neighborhood selection in collaborative filtering for selecting diverse neighbors, while in [16], an adaptive diversification approach is based on Latent Factor Portfolio model for capturing the user interests range and the uncertainty of the user preferences by employing the variance of

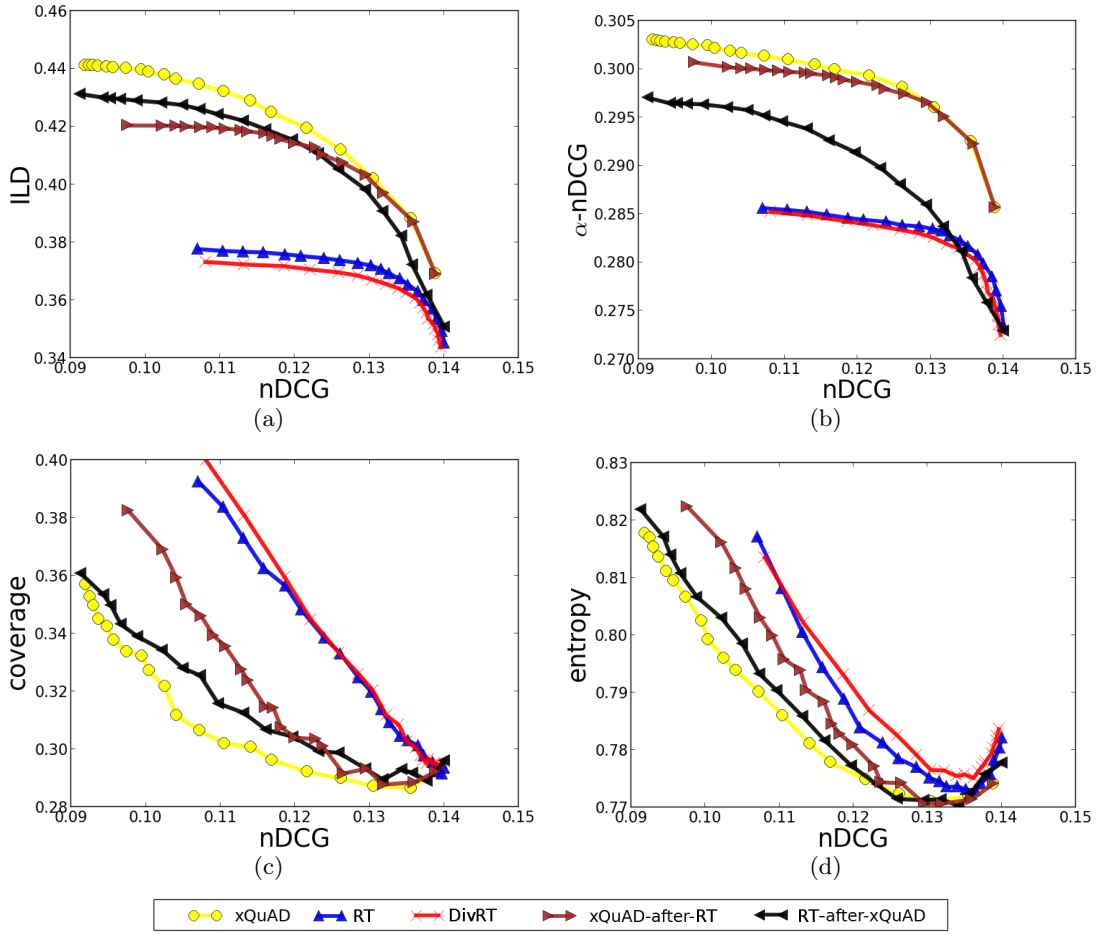

Figure 2: Accuracy-diversity curves on MovieLens at Top-10 obtained by varying the λ parameter from 0 to 0.95 (step 0.05). The statistical significance is measured based on the results from individual users, according the Wilcoxon signed-rank significance test. For nDCG and ILD 2(a), all the differences are statistically significant with ($p < 0.01$), except for those between RT and DivRT. For α-nDCG 2(b), the trend is the same, except for the differences between xQuAD and xQuAD-after-RT with $\lambda > 0.7$.

the learned user latent factors. In [13] it is proposed a hybrid method based on evolutionary search following the Strength Pareto approach for finding appropriate weights for the constituent algorithms with the final aim of improving accuracy, diversity and novelty balance. [24] considers the problem to improve diversity while maintaining adequate accuracy as a binary optimization problem and proposes an approach based on solving a trust region relaxation. The advantages of this approach is that it seeks to find the best sub-set of items over all possible sub-sets, while the greedy selections finds sub-optimal solutions.

Multi-attribute diversity has been substantially non-treated in the literature of recommender systems. A recent work [6] proposes an adaptive approach able to customize the degree of recommendation diversity of the *top-N* list taking into account the inclination to diversity of the user over different content-based item attributes. Specifically, entropy is employed as a measure of diversity degree within user preferences and used in conjunction with user profile dimension for calibrating the degree of diversification.

Furthermore, increasing attention has been paid to the intent-aware diversification, namely the process of increasing the diversity taking into account the user interests. Some approaches are based on adapted algorithms proposed for the same purpose in the Information Retrieval field, such as IA-Select [4] and xQuAD [15]. An approach for extraction of sub-profiles reflecting the user interests has been proposed in [20]. There a combination of sub-profile recommendations is generated, with the aim of maximizing the number of user tastes represented and simultaneously avoiding redundancy in the top-N recommendations. A more recent approach [19], based on a binomial greedy re-ranking algorithm, combines global item genre distribution statistics and personalized user interests to satisfy coverage and non-redundancy of genres in the final list.

The aggregate diversity, also known as sales diversity, is considered another important factor in recommendation for both business and user perspective: the user may receive less obvious and more personalized recommendations, comply with the target to help users discover new content [21] and the business may increase the sales [8]. [3] proposes the concept of aggregated diversity as the ability of a system to

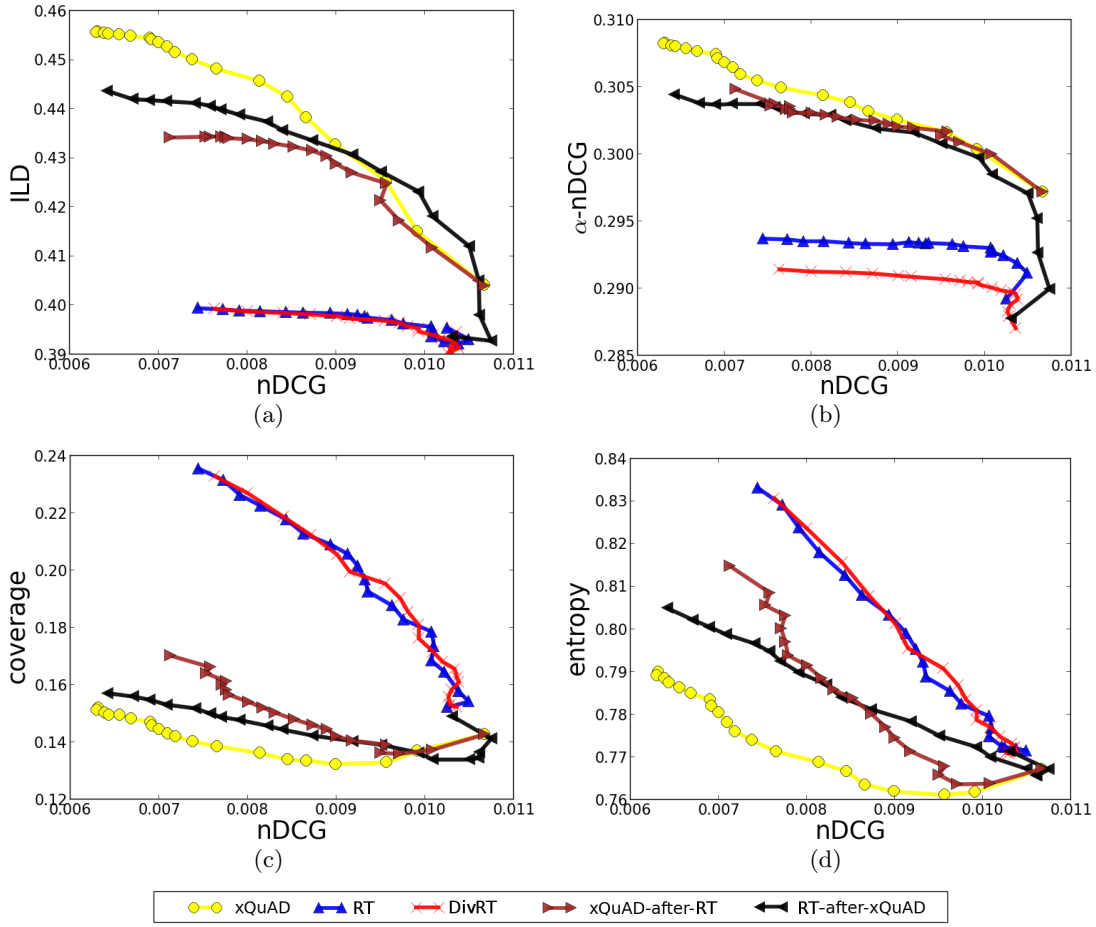

Figure 3: Accuracy-diversity curves on LibraryThing at Top-10 obtained by varying the λ parameter from 0 to 0.95 (step 0.05). The statistical significance is measured based on the results from individual users, according the Wilcoxon signed-rank significance test. For nDCG, the differences between RT and DivRT are non significant with $\lambda \in [0.2, 0.5]$. For ILD 3(a), all the differences are statistically significant with ($p<0.001$), except for those between RT and DivRT. For α-nDCG 3(b), all the differences are statistically significant ($p<0.001$).

recommend across all users as many different items as possible and proposes efficient and parametrizable re-ranking techniques for improving aggregate diversity with controlled accuracy loss. Those techniques are simply based on statistical informations such us items average ratings, average predicted rating values, and so on. [21] explores the impact on aggregate diversity and novelty inverting the recommendation task, namely ranking users for items. Specifically, two approaches have been proposed: one based on an inverted neighborhood formation and the other on a probabilistic formulation for recommending users to items. [14] proposed a k-furthest neighbors collaborative filtering algorithm to mitigate the popularity bias and increase diversity, considering also other factors in user-centric evaluation, such as novelty, serendipity, obviousness and usefulness.

7. CONCLUSIONS AND FUTURE WORK

This paper addresses the problem of intent-aware diversification in recommender systems in multi-attribute settings. The proposed approach bases on xQuAD [20], a relevant intent-aware diversification algorithm, and leverages regression trees as user modeling technique. In their rule-based equivalent representation, they are exploited to foster the diversification of recommendation results both in terms of individual diversity and in terms of aggregate one.

The experimental evaluation on two datasets in the movie and book domains demonstrates that considering the rules generated from the different attributes available in an item description provides diversified and personalized recommendations, with a small loss of accuracy. The analysis of the results suggests that a pure rule-based diversification is a good choice when the aggregate diversity is more needed than individual diversity. Conversely, basic xQuAD remains the best choice to improve the individual diversity while its combination with the rule-based diversification improves also the aggregate diversity.

For future work, we would like to evaluate the impact of our approach also on the recommendation novelty. A way to improve the novelty could be the expansion of the rules by exploiting collaborative information.

Acknowledgements. The authors acknowledge partial support of PON02_00563_3470993 VINCENTE, PON04a2_E RES NOVAE, PON02_00563_3446857 KHIRA e PON01_03113 ERMES.

8. REFERENCES

[1] P. Adamopoulos and A. Tuzhilin. On unexpectedness in recommender systems: Or how to expect the unexpected. In *in Proc of RecSys '11 Intl. Workshop on Novelty and Diversity in Recommender Systems*, 2011.

[2] P. Adamopoulos and A. Tuzhilin. On over-specialization and concentration bias of recommendations: Probabilistic neighborhood selection in collaborative filtering systems. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys '14, pages 153–160. ACM, 2014.

[3] G. Adomavicius and Y. Kwon. Improving aggregate recommendation diversity using ranking-based techniques. *IEEE Trans. Knowl. Data Eng.*, 24(5):896–911, 2012.

[4] P. Castells, S. Vargas, and J. Wang. Novelty and Diversity Metrics for Recommender Systems: Choice, Discovery and Relevance. In *International Workshop on Diversity in Document Retrieval (DDR 2011) at the 33rd European Conference on Information Retrieval (ECIR 2011)*, April 2011.

[5] C. L.A. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. MacKinnon. Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 659–666. ACM, 2008.

[6] T. Di Noia, V. C. Ostuni, J. Rosati, P. Tomeo, and E. Di Sciascio. An analysis of users' propensity toward diversity in recommendations. In *ACM RecSys '14*, RecSys '14, pages 285–288. ACM, 2014.

[7] M. D. Ekstrand, F. M. Harper, M. C. Willemsen, and J. A. Konstan. User perception of differences in recommender algorithms. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys '14, pages 161–168. ACM, 2014.

[8] 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, 2009.

[9] N. Hurley and M. Zhang. Novelty and diversity in top-n recommendation – analysis and evaluation. *ACM TOIT*, 10(4):14:1–14:30, 2011.

[10] S. M. McNee, J. Riedl, and J. A. Konstan. Being accurate is not enough: How accuracy metrics have hurt recommender systems. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '06, pages 1097–1101, 2006.

[11] V. C. Ostuni, T. Di Noia, E. Di Sciascio, and R. Mirizzi. Top-n recommendations from implicit feedback leveraging linked open data. In *ACM RecSys '13*, pages 85–92, 2013.

[12] R. J. Quinlan. Learning with continuous classes. In *5th Australian Joint Conference on Artificial Intelligence*, pages 343–348, Singapore, 1992. World Scientific.

[13] M. T. Ribeiro, A. Lacerda, A. Veloso, and N. Ziviani. Pareto-efficient hybridization for multi-objective recommender systems. In RecSys '12, pages 19–26. ACM, 2012.

[14] 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. ACM, 2013.

[15] R. L.T. Santos, C. Macdonald, and I. Ounis. Exploiting query reformulations for web search result diversification. In WWW '10, pages 881–890. ACM, 2010.

[16] Y. Shi, X. Zhao, J. Wang, M. Larson, and A. Hanjalic. Adaptive diversification of recommendation results via latent factor portfolio. In *ACM SIGIR '12*, pages 175–184, 2012.

[17] B. Smyth and P. McClave. Similarity vs. diversity. In *Proceedings of the 4th International Conference on Case-Based Reasoning: Case-Based Reasoning Research and Development*, ICCBR '01, pages 347–361. Springer-Verlag, 2001.

[18] S. Vargas, L. Baltrunas, A. Karatzoglou, and P. Castells. Coverage, redundancy and size-awareness in genre diversity for recommender systems. In RecSys '14, pages 209–216, 2014.

[19] S. Vargas, L. Baltrunas, A. Karatzoglou, and P. Castells. Coverage, redundancy and size-awareness in genre diversity for recommender systems. In RecSys '14, pages 209–216. ACM, 2014.

[20] S. Vargas and P. Castells. Exploiting the diversity of user preferences for recommendation. In *OAIR '13*, pages 129–136, 2013.

[21] S. Vargas and P. Castells. Improving sales diversity by recommending users to items. In *Eighth ACM Conference on Recommender Systems, RecSys '14, Foster City, Silicon Valley, CA, USA - October 06 - 10, 2014*, pages 145–152, 2014.

[22] Y. Wang and I. H. Witten. Induction of model trees for predicting continuous classes. In *Poster papers of the 9th European Conference on Machine Learning*. Springer, 1997.

[23] C. Yu, L. Lakshmanan, and S. Amer-Yahia. It takes variety to make a world: Diversification in recommender systems. In *EDBT '09*, pages 368–378, 2009.

[24] M. Zhang and N. Hurley. Avoiding monotony: Improving the diversity of recommendation lists. In *ACM RecSys '08*, pages 123–130, 2008.

[25] T. Zhou, Z. Kuscsik, J.G. Liu, M. Medo, J.R. Wakeling, and Y.C. Zhang. Solving the apparent diversity-accuracy dilemma of recommender systems. *Proceedings of the National Academy of Sciences*, 107:4511–4515, 2010.

[26] C. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *WWW '05*, pages 22–32, 2005.