Calibrated Recommendations

Harald Steck
Netflix
Los Gatos, California
hsteck@netflix.com

ABSTRACT

When a user has watched, say, 70 romance movies and 30 action movies, then it is reasonable to expect the personalized list of recommended movies to be comprised of about 70% romance and 30% action movies as well. This important property is known as *calibration*, and recently received renewed attention in the context of fairness in machine learning. In the recommended list of items, calibration ensures that the various (past) areas of interest of a user are reflected with their corresponding proportions. Calibration is especially important in light of the fact that recommender systems optimized toward accuracy (e.g., ranking metrics) in the usual offline-setting can easily lead to recommendations where the lesser interests of a user get crowded out by the user's main interests–which we show empirically as well as in thought-experiments. This can be prevented by *calibrated recommendations*. To this end, we outline metrics for quantifying the degree of calibration, as well as a simple yet effective re-ranking algorithm for post-processing the output of recommender systems.

CCS CONCEPTS

• **Information systems → Collaborative filtering**;

KEYWORDS

Recommender Systems; Calibration; Fairness; Diversity

ACM Reference Format:
Harald Steck. 2018. Calibrated Recommendations. In *Twelfth ACM Conference on Recommender Systems (RecSys '18), October 2–7, 2018, Vancouver, BC, Canada.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3240323.3240372

1 INTRODUCTION

Recommender systems provide a personalized user experience in many different application domains, including online-shopping, social-networks and music/video streaming.

In this paper, we show that recommender systems trained toward *accuracy* (e.g., ranking metrics) can easily generate lists of recommended items that focus on the main areas of interest of a user–while the user's lesser areas of interest tend to be under-represented or even absent. Over time, such unbalanced recommendations carry the risk of gradually narrowing down the user's areas of interest–which is similar to the effect of echo chambers or filter bubbles. This problem also applies to the case of several users sharing the same account, where the interests of the less active users within the same account may get crowded out in the recommendations. We demonstrate this effect in several thought experiments in Section 2 as well as in experiments on real-world data in Section 6.

Calibration is a general concept in machine learning, and recently experienced a resurgence in the context of fairness of machine learning algorithms. A classification algorithm is called calibrated if the predicted proportions of the various classes agree with the actual proportions of data points in the available data. Analogously, in this paper the goal of *calibrated recommendations* is to reflect the various interests of a user in the recommended list, and with their appropriate proportions. To this end, we outline metrics for quantifying the degree of calibration in Section 3. In Section 4, we propose an algorithm for post-processing a given ranked list of recommendations with the objective of making it (close to) calibrated. In Section 5, which discusses related concepts and literature, we also outline that diversity in its typical sense of minimal similarity or redundancy among the recommended items is different from calibration. In our experiments on real-world data in Section 6, we demonstrate the effect that the lesser interests of users can get crowded out easily. We then show the effectiveness of our proposed approach in achieving (close to) calibrated recommendations.

For the ease of exposition in this paper, we will paraphrase 'users who interact with items' as well as 'categories of items', using *'users who play movies'* and *'genres'*. This paper naturally caries over to the general case, see also the last paragraph in Section 4 for further generalizations.

2 MOTIVATING EXAMPLE

In this section, we outline a thought experiment that illustrates a core mechanism that can cause the list of recommended items to be unbalanced. We develop it in three steps, starting from the most extreme scenario.

We consider the typical *offline setting* throughout this paper, where the data set is comprised of historical user-item-interactions, and it is split into a training and test set (e.g., based on time, or randomly); the evaluation objective is to achieve the best accuracy in predicting which items the user interacted with in the test set, which is typically quantified in terms of ranking metrics. This setting has the advantage that it is easy to implement, and applicable to publicly available data sets for collaborative filtering.

In our running example, we assume in this section that a user has played 70 romance and 30 action movies in the offline training data: our objective is to generate a list of, say, 10 recommended movies such that we maximize the probability of predicting the test-movies of this user (i.e., the held-out movies played by the user in the offline test data). This maximizes the recommendation accuracy,

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

RecSys '18, October 2–7, 2018, Vancouver, BC, Canada
© 2018 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-5901-6/18/10.
https://doi.org/10.1145/3240323.3240372

e.g., ranking metrics. For simplicity of argument, let us also assume in this section that the two genres are mutually exclusive (i.e., a movie is either in the action or romance genre, but not in both).

2.1 Class Imbalance

In the first and most extreme scenario, let us assume in this subsection that we only know the user's preference for genres, but we have no additional information on the individual movies within each genre. This problem then becomes analogous to the imbalanced classification problem in supervised machine learning in the absence of any additional information: it is well known that the best prediction accuracy is obtained by *always* predicting the label of the *majority* class. In a binary classification problem where we only know that 70% of the data points have the label +1, and the remaining 30% points are labeled -1, in the absence of any additional information, it is best to predict the label +1 for *all* data points–and we can expect to be correct for 70% of the data points. In contrast, if we predicted the labels +1 and -1 randomly with probabilities 70% and 30% (with which they occur in the data), we can expect the predicted labels to be correct for only $0.7 \cdot 70\% + 0.3 \cdot 30\% = 58\%$ of the data points.

Translated to our recommendation example, in the absence of any additional information, we can expect to obtain the best accuracy on our test data if we recommend 100% romance movies to the user, and not a single action movie.

Our assumption in this subsection, namely that we have no additional information available, is obviously very extreme. In the real world, there will be more data available–however, data will always be limited or noisy, and hence this effect may still be present to some degree. Note that this problem is independent of any particular machine learning model trained for accuracy. In our experiments on real-world data in Section 6, we illustrate that indeed there is the risk of unbalanced recommendations: the genres where the user has only a slight interest can easily get crowded out when optimizing the recommender system for accuracy, while the main areas of the user's interests can get amplified.

Another perspective of this problem is in terms of biased recommendations: even in the ideal case that the available data are free of any biases, the training toward accuracy on limited data can introduce a bias in the recommended list, i.e., it is biased toward the main interests of the user.

Conversely, this suggests–not surprisingly–that the objective of making more balanced or calibrated recommendations is expected to reduce recommendation accuracy.

2.2 Varying Movie Probabilities

This section develops a slightly more involved thought experiment: we now assume that each movie i has a different probability $p(i|g)$ of being played if user u decided to play from genre g. From above, we already know the probabilities $p(g_r|u) = 0.7$ and $p(g_a|u) = 0.3$ that user u plays a movie from genre g_r (romance) and g_a (action), respectively. Assuming here for simplicity that the two sets of movies regarding the two genres are mutually exclusive, the probability that user u plays movie i in genre g is given by $p(i|u) = p(i|g) \cdot p(g|u)$. For best prediction accuracy, we hence have to find the 10 movies i with the largest probabilities $p(i|u)$ of being played by the user. Let us consider the most probable action-movie $i_{g_a,1}$(i.e., ranked first among the action movies), and the 10th most probable romance movie $i_{g_r,10}$, and we obtain

$$\frac{p(i_{g_r,10}|u)}{p(i_{g_a,1}|u)} = \underbrace{\frac{p(i_{g_r,10}|g_r)}{p(i_{g_a,1}|g_a)}}_{\approx 1/2.1} \cdot \underbrace{\frac{p(g_r|u)}{p(g_a|u)}}_{=\frac{0.7}{0.3}\approx 2.33} \approx \frac{2.33}{2.1} > 1, \qquad (1)$$

where we determined the value of 2.1 from the MovieLens 20 Million data set [13].[1] As we can see, also in this variant of the example, the 10th romance title has a higher probability of being played by the user than the best action title. Hence, in terms of accuracy, the optimal 10 titles to recommend in this example are again all romance titles, and not a single action title.

2.3 Latent Dirichlet Allocation

This running example got inspired by the Latent Dirichlet Allocation model (LDA) [5], which describes a user's process of selecting a movie in a two-step procedure: the user first selects a genre (or topic) and then a movie (or word) within the selected genre. We mention LDA for three reasons.

First, if we assume in this section that the real-world user truly follows this two-step procedure of selecting a movie, then the LDA model is the correct model. When the LDA model is trained, it hence is able to capture the correct balance of each user's interests, and with their correct proportions. Hence, balanced recommendations can be expected when following its generative process, where the list of recommended titles is generated iteratively by appending one title at a time: first, a genre g is sampled from the learned genre-distribution $p(g|u)$ for user u, followed by sampling a movie i from the learned distribution $p(i|g)$ regarding genre g. *Sampling* movies results in lower accuracy compared to *ranking* movies according to $p(i|u)$, where $p(i|u) = \sum_g p(i|g) \cdot p(g|u)$. The reason is that also movies i with small $p(i|u)$ may be sampled to be near the top of the recommended list for user u. In contrast, ranking is deterministic and guarantees that the movies i with the largest probabilities $p(i|u)$ that user u likes them, will be at the top of the recommended list, which obviously can be expected to achieve the best accuracy on test data if the learned probabilities $p(i|u)$ are correctly estimated. Unlike sampling, however, ranking unfortunately does not maintain the balance in the recommended list–we illustrated this in our example in the previous section, where the movies were also ranked by their probabilities $p(i|u)$.

Second, note that the problem of unbalanced recommendations is not restricted to the case when explicit categories (e.g., genres) are used, but also applies to the case when latent topics or embeddings are used–LDA is such a model.

Third, the problem of unbalanced recommendations may arise irrespective of the fact whether a movie belongs to a single genre (hard assignment), or whether it partially belongs to several genres, like in the LDA model.

[1] We further assumed for simplicity of the argument in Eq. 1 that the most probable movie in each genre has the same conditional probability, i.e., $p(i_{g_r,1}|g_r) = p(i_{g_a,1}|g_a)$. If this is not the case, one can simply include an additional factor in the equation.

3 CALIBRATION METRICS

In this section, we outline metrics that quantify the degree of calibration of a list of recommended movies, with respect to the user's history of played movies. To this end, we consider two distributions, both of which are based on the distribution of genres g for each movie i, denoted by $p(g|i)$, which are assumed to be given:

- $p(g|u)$: the distribution over genres g of the set of movies $\mathcal{H}$ played by user u in the past:
$$p(g|u) = \frac{\sum_{i \in \mathcal{H}} w_{u,i} \cdot p(g|i)}{\sum_{i \in \mathcal{H}} w_{u,i}}, \tag{2}$$
where $w_{u,i}$ is the weight of movie i, e.g., how recently it was played by user u. See also Eq. 7 for a regularized version.
- $q(g|u)$: the distribution over genres g of the list of movies recommended to user u:
$$q(g|u) = \frac{\sum_{i \in \mathcal{I}} w_{r(i)} \cdot p(g|i)}{\sum_{i \in \mathcal{I}} w_{r(i)}}, \tag{3}$$
where $\mathcal{I}$ is the set of recommended movies. The weight of movie i due to its rank $r(i)$ in the recommendations is denoted by $w_{r(i)}$. Possible choices include the weighting schemes used in ranking metrics, like in Mean Reciprocal Rank (MRR) or normalized Discounted Cumulative Gain (nDCG).

There are various established methods for determining if these two distributions, $q(g|u)$ and $p(g|u)$, are similar. As to account for the fact that these distributions are estimated from finite data, comprised of N recommended movies and M movies played by the user, respectively, one may carry out a statistical hypothesis test, with the Null hypothesis that the two distributions are the same. This is typically cast as an independence test regarding the multinomial distribution over two random variables: the genres g, and a variable reflecting the two sets of movies, $\mathcal{I}$ and $\mathcal{H}$. Given that N or M might actually be very small numbers, this may call for exact tests, like the multinomial test or Fisher's exact test. These tests may, however, be computationally prohibitive in practice. A computationally efficient alternative are asymptotic tests, if applicable, like the G-test or χ^2-test.

Instead of computing p-values, we suggest to ignore the effect of the finite data sizes N and M, and to directly compare the distributions $p(g|u)$ and $q(g|u)$. To this end, we use the Kullback-Leibler (KL) divergence as calibration metric $C_{\text{KL}}(p,q)$ in this paper:

$$C_{\text{KL}}(p,q) = \text{KL}(p||\tilde{q}) = \sum_g p(g|u) \log \frac{p(g|u)}{\tilde{q}(g|u)}, \tag{4}$$

where we use $p(g|u)$ as the target distribution. If $q(g|u)$ is similar to it, $C_{\text{KL}}(p,q)$ takes small values. Given that the KL divergence diverges if $q(g|u) = 0$ and $p(g|u) > 0$ for a genre g, we instead use

$$\tilde{q}(g|u) = (1-\alpha) \cdot q(g|u) + \alpha \cdot p(g|u) \tag{5}$$

with a small $\alpha > 0$, so that $q \approx \tilde{q}$. In our experiments, we used $\alpha = 0.01$. The KL-divergence has several properties desirable for quantifying the degree of calibration in the context of recommendations:

(1) it is zero in case of perfect calibration: $p(g|u) = \tilde{q}(g|u)$.

(2) it is very sensitive to small discrepancies between $p(g|u)$ and $\tilde{q}(g|u)$ when $p(g|u)$ is small. For instance, if a user played a genre only 2% of the time, recommending it only 1% is considered a larger discrepancy by the KL divergence, than if a genre was played 50% but recommended only 49% of times.

(3) it favours more uniform, and hence less extreme distributions: as illustrated in Table 1, if a user played a genre 30% of the time, recommendations with 31% of this genre are considered better than with 29%.

These properties ensure that the genres that the user rarely played will also be reflected in the recommended list with their corresponding proportions. Instead of the KL-divergence, one may also use other f-divergences in general, like the Hellinger distance between p and q, $C_{\text{H}}(p,q) = H(p,q) = ||\sqrt{p} - \sqrt{q}||_2/2$, where $||\cdot||_2$ denotes the 2-norm of the probability-vector (across genres). The Hellinger distance is well defined in the presence of zero values; it also is sensitive to small discrepancies between p and q when p is small, however, to a lesser degree than the KL-divergence is, as we found in our experiments.

The overall calibration metric C is obtained by averaging $C(p,q)$ across all users.

4 CALIBRATION APPROACHES

The calibration of recommendations is a list-property. As many recommender systems are trained in a pointwise or pairwise manner, one may not be able to include calibration into the training. This suggests to re-rank the predicted list of a recommender system in a post-processing step, a common approach of calibrating machine learning approaches [10, 30]. As to determine the optimal set $\mathcal{I}^*$ of N recommended movies, we use *maximum marginal relevance* [6]:

$$\mathcal{I}^* = \underset{\mathcal{I}, |\mathcal{I}|=N}{\arg\max}\ (1-\lambda) \cdot s(\mathcal{I}) - \lambda \cdot C_{\text{KL}}(p, q(\mathcal{I})) \tag{6}$$

where $\lambda \in [0,1]$ determines the trade-off between two terms: (1) the scores $s(i)$ of the movies $i \in \mathcal{I}$ predicted by the recommender system, where $s(\mathcal{I}) = \sum_{i \in \mathcal{I}} s(i)$. Note that one may also use a monotone transform of each movie's score. (2) the calibration metric (see Eq. 4), where we have explicitly denoted the dependence of q on the set of recommended movies $\mathcal{I}$, which we optimize in Eq. 6. Also note that better calibration entails a lower calibration score, so that we have to use its negative in this maximization problem.

The trade-off between accuracy-focused recommendations and calibration can be controlled by λ in Eq. 6. We consider calibration as a crucial property of the recommended list, as discussed in Section 5, which hence calls for a rather large value of λ.

Finding the optimal set $\mathcal{I}^*$ of N recommended movies is a combinatorial optimization problem and NP-hard in general. In the Appendix, we outline that the greedy optimization of this optimization problem is equivalent to the greedy optimization of a surrogate *submodular* function. It is well known [17] that the greedy optimization of submodular functions achieves a $(1 - 1/e)$ optimality guarantee, where e is Euler's number. The greedy optimization starts out with the empty set, and iteratively appends one movie i at a time: at step n, when we already have the set $\mathcal{I}_{n-1}$ comprised of $n-1$ movies, the movie i that maximizes Eq. 6 for the set $\mathcal{I}_{n-1} \cup \{i\}$

is added as to obtain $\mathcal{I}_n$. This greedy approach has additional benefits. First, it yields an ordered / ranked list of movies, instead of an (unsorted) list. Second, the resulting list at each step of this greedy approach is $(1 - 1/e)$ optimal among the lists of equal size. Even though we may generate a ranked list of N movies, in the real-world, the user might initially see only the first $n < N$ recommendations, e.g., the remaining movies may become visible in the view-port only after scrolling. Apart from that, the user may scan the list of N movies from top to bottom. In both cases, the greedy optimization of submodular functions automatically ensures that each sub-list of the first n movies ($n < N$) of the recommended list is $(1 - 1/e)$ optimal.

Note that this approach allows for a weighted membership of a movie i to possibly several genres g, as $p(g|i)$ is used in Eqs. 2 and 3. Moreover, if one likes to calibrate the recommended list with respect to several different categories (e.g., genres, subgenres, languages, movie-vs.-TV-show, etc.), a separate calibration-term $C_{\mathrm{KL}}^{(\mathrm{category})}$ may be added to Eq. 6 for each category, with the desired weight/importance $\lambda^{(\mathrm{category})}$. The resulting sum of several submodular functions is still a submodular function, and hence the optimization problem remains efficient.

5 RELATED CONCEPTS

Calibration has long been used in machine learning, mainly in classification, see, e.g., [10, 30], where simple post-processing approaches were often found to be effective. In recent years, calibration received renewed attention, in particular in the context of fairness of machine learning algorithms.

In the literature of recommender systems, the focus has been on various metrics besides accuracy, e.g., see [21] for an overview, among which diversity is closest to calibration.

5.1 Diversity

In this section, we first compare diversity and calibration, followed by a discussion of related work.

Diversity as defined in most papers, i.e., minimal redundancy or similarity among the recommended items, helps avoid recommendations with 100% romance movies in our running example: in a world with only two genres of movies, the most diverse recommendations would contain 50% romance and 50% action movies. In a world with additional movie-genres (where the user has only watched 70 romance and 30 action titles), diversity can be increased by recommending also titles from other genres that the user has not watched yet, like children's movies or documentaries. Diversity is not guaranteed, however, to increase the fraction of recommended action titles from 0% to about 30% as to reflect the user's degree of interest in our example. Only if the trade-off between accuracy and diversity is chosen well, one may obtain well-calibrated recommendations. This may be difficult to achieve in practice, however, as this trade-off may be different for each user. This illustrates that the objective of diversity is not directly aimed at reflecting a user's various interests with the appropriate proportions. This is a main difference to calibrated recommendations.

A second key difference is that diversity, as it may include genres that the user has not played in the past, may help a user escape from a possible filter bubble. This important property is not provided by calibrated recommendations as outlined so far. This motivates a simple *extension to calibrated recommendations*, such that also titles from genres outside of the user's past interests are included into the recommended list: let $p_0(g)$ denote a prior-distribution that takes positive values for all genres g as to promote diversity in the recommendations–two obvious choices are the uniform distribution or the average over all users' genre distributions. The weighted average of this diversity-promoting prior $p_0(g)$ and the calibration-target $p(g|u)$,

$$\tilde{p}(g|u) = \beta \cdot p_0(g) + (1 - \beta) \cdot p(g|u), \tag{7}$$

with tuning parameter $\beta \in [0, 1]$, determines the trade-off between diversity and calibration. This *extended calibration probability* $\tilde{p}(g|u)$ can be used in place of $p(g|u)$ (see Eq. 2).

In many papers, a list is considered diverse if there is only a small degree of redundancy or similarity among the recommended items. A multitude of approaches has been proposed to generate such kinds of diverse recommendations, e.g., [4, 15, 31, 32], including determinantal point processes [8, 11], or submodular optimization, e.g., [1, 2, 19].

A second line of research starts out with modeling the user's probability of choosing the n^{th} item from the recommended list after having not selected any of the $n - 1$ items ranked / displayed above, i.e., a browsing model. This idea has resulted in the ranking metric called expected reciprocal rank (ERR) [7], as well as in approaches for generating a more diverse ranked list [20, 27].

Only few papers have addressed the important issue that recommendations should reflect the various interests of the user with the correct proportions [9, 25, 26], which we will discuss in the following.

The idea of proportionality was first proposed in [9] in the context of diversifying search results. In [9], the proposed metric, named DP, is essentially a modified squared difference between the distributions $p(g|u)$ and $q(g|u)$. While it fulfills our property 1 for calibration metrics in Section 3, it does not exhibit the other two properties: as illustrated in Table 1 for the target proportions 60%:40%, the more unbalanced recommendations with 7:3 titles in the two genres receives the same value DP=1 as does the uniform one with 5:5. Given that both deviate from the ideal recommendations of 6:4 by one movie being in the other genre, 5:5 should receive a better calibration score than 7:3 according to property (3) in Section 3. Property (2) is also not fulfilled because DP=1 for a deviation of 1 title is independent of how extreme the target distribution is when 10 titles get recommended in Table 1–ideally the score should be worse for the target distribution 70%:30%, as it is more extreme than 60%:40%. Note that the KL-divergence fulfills these properties in Table 1. In [9], the algorithm for generating a proportional list utilizes a procedure for seat assignment after an election, so that each party's seats are proportional to their received votes. They developed a probabilistic modification of this procedure as to tackle the problem of items belonging to several categories, and found this method to outperform the original one in their experiments. In case that perfect proportionality cannot be achieved, and an approximate solution with some deviations has to be found, their algorithm may treat deviations differently than their metric does, as they are conceptually unrelated. It is hence

Table 1: Comparison of three calibration metrics for a given target distribution $p(g|u)$, and the genre counts $n_{1/2} = N \cdot q(g_{1/2}|u)$ in the recommended list of N movies. See Sections 3 and 5.1 for discussion.

(a) target distribution: 60% : 40%				
N	$n_1 : n_2$	C_{KL} (Eq. 4)	BinomDiv [26]	DP [9]
10	5:5	0.0197	$4.66 \cdot 10^{-4}$	1.0
	6:4	0.0	$5.38 \cdot 10^{-4}$	0.0
	7:3	0.0221	$4.62 \cdot 10^{-4}$	1.0

(b) target distribution: 70% : 30%				
N	$n_1 : n_2$	C_{KL} (Eq. 4)	BinomDiv [26]	DP [9]
10	6:4	0.0212	$1.74 \cdot 10^{-4}$	1.0
	7:3	0.0	$2.04 \cdot 10^{-4}$	0.0
	8:2	0.0275	$1.69 \cdot 10^{-4}$	1.0
100	69:31	$2.31 \cdot 10^{-4}$	$7.78 \cdot 10^{-35}$	1.0
	70:30	0.0	$7.91 \cdot 10^{-35}$	0.0
	71:29	$2.36 \cdot 10^{-4}$	$8.39 \cdot 10^{-35}$	1.0

not obvious if the approximate solution obeys properties desirable in the context of calibrated recommendations.

In [25], personalized diversification is approached from the perspective of submodularity. While they propose a submodular objective function in Eq. 2 in [25] that is comprised of a log-sum term–similar to Eq. 8 in our Appendix–, its connection to the KL-divergence is not outlined in [25]. It hence remains unclear in [25] that the actual goal of this submodular function is to recommend the various item-categories *proportional* to their weights (e.g., CTR in [25]).

The metric proposed in [26], named BinomDiv, is carefully crafted and fulfills properties (2) and (3) in Section 3: e.g., regarding the target proportions 60%:40% in Table 1, the more extreme recommendations with 7:3 receive a worse (lower) score than the more balanced one with 5:5. These are the important properties for proportionality. Their metric does not fulfill property 1, however, even in a more relaxed sense of taking the same fixed value (instead of 0) in case of perfect calibration: their metric can take on different values if $p(g|u) = q(g|u)$, depending on the length of the recommended list as well as on the distribution of the genres $p(g|u)$, see Table 1. This has two disadvantages: first, a given value of the metric by itself does not provide a sense for how calibrated the recommendations are–it only allows one to make relative comparisons regarding different recommended lists for a fixed user. Second, given that each user tends to have a different distribution of interests/genres, this metric cannot simply be averaged across users as to obtain an aggregate metric. While the transformation of this metric into a z-score for evaluation purposes does not seem to be mentioned in the paper, its use in the algorithm is pointed out. We also found that our computation of their metric suffered from numerical underflow when the number of recommended movies exceeded a couple of hundred–while this may not cause issues in many applications, like top 10 recommendations, there are also scenarios where the number of recommended items is in the hundreds, e.g., on the Netflix homepage. Apart from that, we note that the idea of adding a prior, as we outlined earlier in this section, was mentioned in [26]. Their algorithm is based on maximum marginal relevance [6]. As their metric may not be submodular, however, there may not be an optimality guarantee.

5.2 Fairness

In the field of machine learning, the importance of *fairness* has recently grown dramatically, e.g., see [33] and reference therein for a review. Fairness is concerned with avoiding discrimination against certain persons or groups in the population, e.g., based on gender, race, age, etc. It is typically concerned with the scores or class labels predicted for individual persons in a population.

Various fairness criteria have been proposed in the literature, including *calibration*, *equal(ized) odds*, *equal opportunity*, and *statistical parity* [12, 16, 33]. Using equalized odds as fairness-metric, [12] proposed a post-processing approach, and [28] improved on this by integrating fairness into the training objective.

In the context of collaborative filtering, it was discussed in [29] that small sub-populations in the user-base (i.e., population imbalance), as well as less active sub-populations (i.e., persons who provide fewer ratings) may receive biased recommendations. Apart from that, [29] focused on rating prediction and RMSE, instead of the more relevant scenario of implicit feedback data and ranking metrics.

In this paper, we consider a complementary notion of fairness: instead of fairness regarding *persons*, we consider fairness concerning the *various interests* of a user, with the goal to reflect them according to their corresponding proportions. In the remainder of this section, we outline why we deem the calibration criteria particularly useful for this non-standard notion of fairness.

As shown in [16], calibration and equal(ized) odds / equal opportunity cannot be fulfilled (exactly nor approximately) at the same time–except for two special cases: when the machine-learned model makes perfect predictions (which does typically not hold in real-world applications), or when the different groups of persons (which all should be treated fairly) have the same base rate, i.e., the same fraction of positive classification-labels, which typically does not hold in the real world, either. Given that a user typically played genres with different proportions (like 70% romance and 30% action movies), the base rate of these two genres (or groups in the fairness literature) is obviously different, and so are the average scores predicted for the movies in these two genres. Hence, the fairness criteria equal(ized) odds, equal opportunity and statistical parity are not immediately applicable in our context. This motivated us to focus on *calibration* as a suitable fairness criteria for recommendations.

6 EXPERIMENTS

This section illustrates the proposed calibration metric (see Section 3) and calibration algorithm (see Section 4) in our experiments on the MovieLens 20 Million data [13]. As outlined in [26], the various metrics regarding diversity capture different properties, and the corresponding algorithms perform well regarding the metric they were developed for, but not necessarily with respect to the other metrics. For this reason, we restrict ourselves in the remaining space of this paper to illustrate that the proposed approach works as expected.

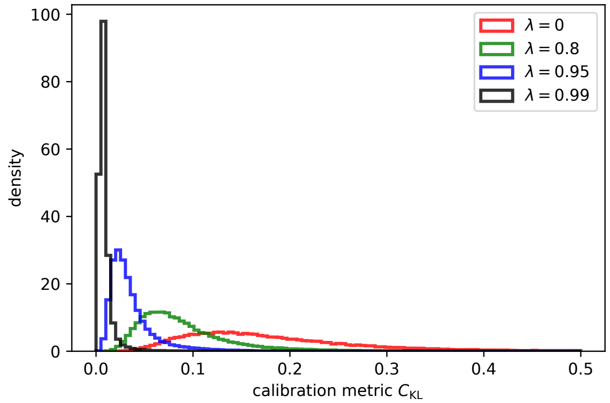

Figure 1: Histograms of calibration scores C_{KL}@50 of all test users, with no ($\lambda = 0$) vs. increased calibration (see Eq. 6). Lower C_{KL}@50 is better.

We used the MovieLens data, as it also contains genre-information besides the rating-data. Implicit feedback data, however, are much more abundant than rating data in most real-world applications. We hence focus on implicit data, and follow the usual procedure of simulating binary implicit feedback data (e.g., user played movie) from the publicly available rating-data by retaining only ratings of 4 stars and higher, while dropping lower ratings. After eliminating movies that had no genre information attached or were not played by a user, the resulting data set was comprised of about 10 million 'plays' with value 1 (instead of ratings) regarding about 21k movies and 140k users. Typically several genres g are assigned to a movie i in this data set–we assigned equal probabilities $p(g|i)$ to each assigned genre g such that $\sum_g p(g|i) = 1$ for each movie i. This is then used to determine the genre-distributions $p(g|u)$ and $q(g|u)$ of user u in Eqs. 2 and 3. We split these data into a training set (99% of the play-data) and a disjoint test set with about 100,000 plays (1% of the play-data), as this split-ratio was also approximately used in the Netflix Prize data [3].

As to generate the baseline-recommendations, we learned a weighted 50-dimensional matrix-factorization model on these training data following the usual approaches described in [14, 18, 23], where we tuned the hyper-parameters, i.e., the L2-norm regularization and the weight for the missing plays (negative sampling), as to maximize recommendation accuracy (recall@50, e.g., see [21] for its definition). In the following experiments, we compare the recommendations produced by this baseline-model that was trained to optimize accuracy, with the re-ranked recommendations generated by our approach outlined in Section 4 for different values of λ, which controls the degree of calibration in Eq. 6. Given that we find calibration an important property of the recommended list, the default value in the presented experiments is $\lambda = 0.99$ unless otherwise noted.

Regarding the baseline-recommendations of each user, we computed the calibration metric C_{KL}.[2] Figure 1 shows the obtained scores C_{KL}@50 as a histogram over all test users: the wide range of calibration scores C_{KL}@50 in the baseline-recommendations ($\lambda = 0$) indicates that different users experience vastly different recommendation-qualities in terms of calibration. Figure 1 also shows the effectiveness of the proposed greedy re-ranking approach: considerably better (lower) calibration scores C_{KL}@50 are achieved as λ is increased. It also illustrates that the degree of calibration can be controlled in a continuous way by changing the value of λ.

Table 2: Trade-off between ranking accuracy and calibration metric C_{KL}, as determined by λ in Eq. 6.

	recall		C_{KL}	
calibration	@10	@50	@10	@50
none ($\lambda = 0$)	0.209	0.464	0.677	0.185
$\lambda = 0.2$	0.209	0.464	0.465	0.171
$\lambda = 0.5$	0.199	0.464	0.274	0.141
$\lambda = 0.8$	0.170	0.463	0.128	0.092
$\lambda = 0.9$	0.146	0.460	0.084	0.061
$\lambda = 0.95$	0.121	0.453	0.065	0.037
$\lambda = 0.99$	0.090	0.417	0.054	0.009
$\lambda = 0.999$	0.082	0.339	0.054	0.005

The average values across all test users are summarized in Table 2: relative to the baseline ($\lambda = 0$), it shows that re-ranking with an increased value of λ in Eq. 6 results in recommendations with better calibration on average, but at the price of reduced accuracy (lower recall), as expected (see Section 2.1). Table 2 also illustrates that, for rather small values of λ, calibration can be improved considerably while accuracy is reduced only slightly. Only for large value of λ, the accuracy drops quickly. Also note that the values C_{KL}@10 are larger than C_{KL}@50 because the genre-distribution is more constraint (and hence less calibrated) if it is based on only 10 rather than 50 recommended movies. Moreover, C_{KL}@10 improves considerably for small values of λ (relative to no calibration) in Table 2, while a larger λ is needed for notable improvements of C_{KL}@50. At the same time, recall@10 deteriorates a lot for larger values of λ, while recall@50 is fairly constant up to around $\lambda = 0.95$. This suggests that a useful strategy in practical application might be to change λ from small to large values during the greedy re-ranking approach. As a result, the top-ranked items of the re-ranked list would be very similar to the original ranking, followed by items that would increasingly cover the lesser areas of interest of a user.

Figure 2 shows the distribution of genres in the top 50 recommendations for a test user chosen from the 10% sub-population where the baseline-recommendations were very uncalibrated: while the genres Drama and Adventure are over-represented in the user's baseline-recommendations (i.e., before calibration) relative to the user's play history, the user's lesser interests, including the genres Musical, Mystery, Thriller and Western, are essentially missing from the baseline-recommendations. This illustrates that the recommender system may amplify the main areas of interest of a user, while crowding out the user's lesser interests. Figure 2 also shows that this can be largely prevented by the proposed greedy approach (with $\lambda = 0.99$ in Eq. 6): the re-ranked recommendations reflect the

[2] In all our experiments, for simplicity we use no prior p_0 in Eq. 7, and no weights in the averages for p and q in Eqs. 2 and 3.

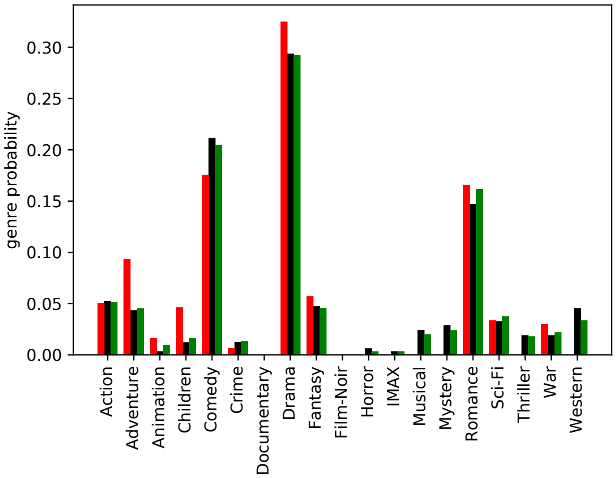

Figure 2: Genre-distribution of a user's play-history (black), and of the recommendations before (red/left) and after calibration (green/right).

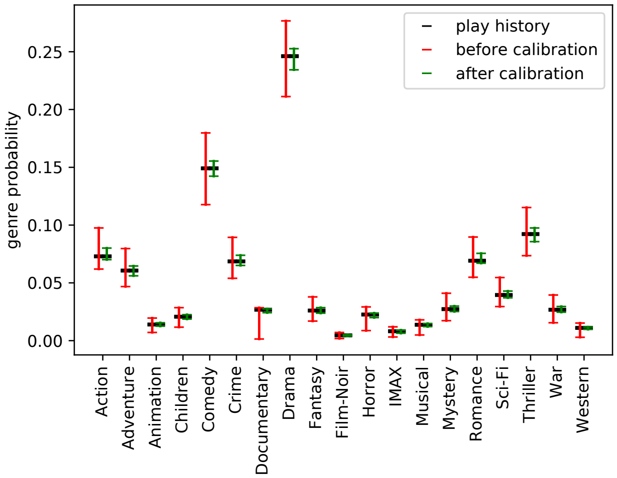

Figure 3: Average over the 10% sub-population of test users with the most uncalibrated recommendations: the mean probability of each genre in the users' play-histories (black); the intervals reflect the average difference between the genre's proportion in the recommended list vs. in the play-history of each user before (red/left) and after (green/right) calibration. See text for details.

user's various interests with their proportions much more closely. Note, however, that recommendations that reflect a user's past interests may still keep this user in their personal filter bubble. For this reason, it is important to use the *extended calibration probability* in practical applications (see Section 5.1 and Eq. 7), which results in recommendations that also cover the areas outside of a user's past areas of interest.

While it is crucial to ensure calibration for each individual user (as exemplified in Figure 2), we now show aggregate results regarding the 10% sub-population of test users who received recommendations with the worst calibration. Figure 3 is obtained as follows: for each user, we calculate the difference in the genre-probabilities between the recommended list of 50 movies and the user's play-history. This difference is then averaged across all test users, separately for positive and negative differences. In Figure 3, the average genre-probability across all users' play histories serves as reference point, while the average positive and negative differences determine the lengths of the intervals above and below each reference point, respectively. If the recommendations are perfectly calibrated to each individual user's play-history, then the length of the intervals is zero. If the lower interval is larger than the upper interval, this genre is under-represented in the recommendations on average. Analogously, the upper interval is larger for over-represented genres. If both intervals are of equal length, then the genre is represented with the correct proportions *on average* across all test users–however, the length of the intervals indicate the average deviation for an *individual* user. Given that recommendations should be calibrated for each individual user, it is desirable that the intervals are small.

Figure 3 shows that the users played about 3% documentaries on average, but the top-50 baseline-recommendations essentially missed this genre completely (i.e., the lower red interval extends all the way to zero, while the upper red interval is essentially of zero length). Similarly, the users' lesser interests, i.e., the genres IMAX, Musical and Western, are also severely underrepresented on average (the lower red interval extends to about zero as well). At the same time, genres like Action, Adventure, Crime, Mystery and Sci-Fi are mostly over-represented (the upper red interval is larger than the lower red one).

Figure 3 also illustrates the results of the proposed greedy algorithm (with $\lambda = 0.99$ in Eq. 6): now the various genres, including the less popular ones, are well calibrated for each individual user (the green intervals are smaller than the red ones). The various interests of each user are hence reflected by the re-ranked recommendations. Drama is the genre with the largest interval: it now is slightly under-represented in the re-ranked recommendations on average (the lower green interval is larger than the upper green interval)–which allows for space in the recommended list as to slightly over-represent the genres that pertain to the lesser areas of interest of each user. This illustrates the effectiveness of the proposed algorithm in generating calibrated recommendations that reflect the various interests of each individual user.

7 CONCLUSIONS

In this paper, we showed that recommender systems that are trained toward accuracy in the typical offline-setting may generate unbalanced recommendations, especially when the available training data are limited or noisy. We motivated the importance of *calibration* as an additional objective besides recommendation-accuracy. We outlined established metrics for quantifying the degree of calibration. It is desirable that they are particularly sensitive to discrepancies regarding the lesser areas of interest of a user, especially when such an area of interest is completely missing from the recommended list. Moreover, we presented a simple yet effective greedy algorithm, and outlined an optimality-guarantee due to submodular functions. These approaches can be applied for post-processing the recommendation-lists generated by recommender systems. We also discussed the difference to *diversity* in its typical sense of minimal similarity or redundancy among the recommended items. Given that calibration is a property of the entire recommended list, future improvements may be achieved by integrating calibration in the objective of listwise learning-to-rank approaches, and by going beyond the typical offline-setting of training and testing recommender systems, e.g., [24].

This paper took a user-centric view, i.e., the recommendations were calibrated for each user. The complementary perspective is the item-centric view, which we leave for future work: as to calibrate the recommendations with respect to each item, one may consider, for instance, whether an item that is recommended twice as often as another item, is also consumed twice as often (across all the users).

ACKNOWLEDGMENTS

I would like to thank Tony Jebara, Aish Fenton, Dawen Liang, and Ehtsham Elahi for their useful feedback.

APPENDIX

Here we show that the greedy optimization of Eq. 6 is equivalent to the greedy optimization of a surrogate function that is submodular. Loosely speaking, the concept of submodular functions may be viewed as a generalization of non-decreasing concave functions to set-functions (on a matroid).

We first split the calibration metric (Eq. 4) into its parts:

$$\begin{aligned} & C_{\mathrm{KL}}(p,q) \\ = \;& \mathrm{KL}(p||\tilde{q}) = \sum_g p(g|u) \log \frac{p(g|u)}{\tilde{q}(g|u)} \\ = \;& \sum_g p(g|u) \log p(g|u) - \sum_g p(g|u) \log \tilde{q}(g|u) \qquad (8) \\ = \;& \underbrace{-H(p)}_{=\text{const.}} + \underbrace{\log \sum_{i\in\mathcal{I}} w_{r(i)}}_{=\log \sum_{r=1}^{|\mathcal{I}|} w_r} - \sum_g p(g|u) \log \sum_{i\in\mathcal{I}} w_{r(i)} \tilde{q}(g|i). \end{aligned}$$

Regarding the last line, several remarks are in order. First, the entropy $H(p)$ refers to the user's past plays, and hence is a constant when optimizing for the set $\mathcal{I}$ of recommended movies. Second, and equivalent to Eqs. 3 and 5, we here absorbed the regularization of $\tilde{q}(g|u)$ into each individual movie's genre distribution $\tilde{q}(g|i) = (1-\alpha)\cdot p(g|i) + \alpha \cdot p(g|u)$. This results in the last two terms in the last line in Eq. 8. Given that the weights $w_{r(i)}$ depend merely on the rank $r(i)$ of movie i, and hence not directly on i, we have that $\sum_{i\in\mathcal{I}} w_{r(i)} = \sum_{r=1}^{|\mathcal{I}|} w_r$, which thus is also a constant in the optimization problem when the size of $\mathcal{I}$ is fixed. When the size $|\mathcal{I}|$ is not fixed, then $\log \sum_{r=1}^{|\mathcal{I}|} w_r$ is a non-decreasing concave function of its size, and hence a submodular function regarding sets. Moreover, also the last term in Eq. 8 is a submodular function, see [22]. Hence, the KL-divergence can be expressed as a difference of two submodular functions [22].

In [22], it was proposed to use this last term in place of the KL-divergence. This results in the new optimization problem:

$$\mathcal{I}^* = \underset{\mathcal{I},|\mathcal{I}|=N}{\arg\max}\; (1-\lambda)\cdot s(\mathcal{I}) + \lambda \cdot \sum_g p(g|u) \log \sum_{i\in\mathcal{I}} w_{r(i)} \tilde{q}(g|i), \quad (9)$$

which now is submodular, given that $s(I) = \sum_{i\in\mathcal{I}} s(i)$ is modular, and the sum of modular and submodular functions is submodular. With the well-known $(1-1/e)$ optimality guarantee [17], we can now greedily optimize this objective function: at each iterative step, when we determine the next movie i to append, only sets of the same size are considered, and hence the first two terms in Eq. 8 are constants, while only the last term depends on i. Hence, at each greedy iterative step, both Eqs. 6 and 9 yield the same optimal movie i^*. The only subtle point here is that the $(1-1/e)$ optimality guarantee refers to Eq. 9 due to its submodularity.

REFERENCES

[1] A. Ashkan, B. Kveton, S. Berkovsky, and Z. Wen. 2014. Diversified Utility Maximization for Recommendations. In *ACM Conference on Recommender Systems (RecSys)*.

[2] A. Ashkan, B. Kveton, S. Berkovsky, and Z. Wen. 2015. Optimal Greedy Diversity for Recommendation. In *Int. Joint Conf. on Artificial Intelligence (IJCAI)*.

[3] J. Bennet and S. Lanning. 2007. The Netflix Prize. In *Workshop at SIGKDD-07, ACM Conference on Knowledge Discovery and Data Mining*.

[4] A. Bhaskara, M. Ghadiri, and V. Mirrokni. 2016. Linear Relaxations for Finding Diverse Elements in Metric Spaces. In *Advances in Neural Information Processing Systems (NIPS)*.

[5] D.M. Blei, A.Y. Ng, and M. Jordan. 2001. Latent Dirichlet Allocation. In *Advances in Neural Information Processing Systems (NIPS)*.

[6] J. Carbonell and J. Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *ACM Conference on Research and development in information retrieval (SIGIR)*.

[7] O. Chapelle, D. Metzler, Y. Zhang, and P. Grinspan. 2009. Expected Reciprocal Rank for Graded Relevance. In *CIKM*.

[8] L. Chen, G. Zhang, and H. Zhou. 2017. Improving the Diversity of Top-N Recommendation via Determinantal Point Process. arXiv:1709.05135.

[9] V. Dang and W. B. Croft. 2012. Diversity by Proportionality: An Election-based Approach to Search Result Diversification. In *ACM Conference on Research and development in information retrieval (SIGIR)*.

[10] D.P. Foster and R.V. Vohra. 1998. Asymptotic calibration. *Biometrika* 85 (1998), 379–90. Issue 2.

[11] M. Gartrell, U. Paquet, and N. Koenigstein. 2016. Bayesian low-rank determinantal point processes. In *ACM Conference on Recommender Systems (RecSys)*. 349–56.

[12] M. Hardt, E. Price, and N. Srebro. 2016. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems (NIPS)*.

[13] F. M. Harper and J. A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 5 (2015). Issue 4.

[14] Y. Hu, Y. Koren, and C. Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *IEEE International Conference on Data Mining (ICDM)*.

[15] Neil Hurley and Mi Zhang. 2011. Novelty and diversity in top-N recommendation–analysis and evaluation. *ACM Transactions on Internet Technology (TOIT)* 10 (2011). Issue 4.

[16] J. Kleinberg, S. Mullainathan, and M. Raghavan. 2016. Inherent Trade-Offs in the Fair Determination of Risk Scores. In *Advances in Neural Information Processing Systems (NIPS)*.

[17] G. L. Nemhauser and L.A. Wolsey. 1978. An Analysis of Approximations for Maximizing Submodular Set Functions. *Mathematical Programming* 14 (1978).

[18] 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 (ICDM)*.

[19] L. Qin and X. Zhu. 2013. Promoting Diversity in Recommendation by Entropy Regularizer. In *Int. Joint Conf. on Artificial Intelligence (IJCAI)*.

[20] R. L. T. Santos, C. Macdonald, and I. Ounis. 2010. Exploiting Query Reformulation for Web Search Result Diversification. In *International World Wide Web Conference (WWW)*.

[21] G. Shani and A. Gunawardana. 2011. Evaluating Recommendation systems. In *Recommender Systems Handbook*. Springer, 257–97.

[22] Y. Shinohara. 2014. A submodular optimization approach to sentence set selection. In *IEEE International Conference on Acoustic, Speech and Signal processing (ICASSP)*.

[23] H. Steck. 2010. Training and Testing of Recommender Systems on Data Missing Not at Random. In *ACM Conference on Knowledge Discovery and Data Mining (KDD)*. 713–22.

[24] A. Swaminathan, A. Krishnamurthy, A. Agarwal, M. Dudik, J. Langford, D. Jose, and I. Zitouni. 2017. Off-policy evaluation for slate recommendation. In *Advances in Neural Information Processing Systems (NIPS)*. 3635–45.

[25] C. H. Teo, H. Nassif, D. Hill, S. Srinivasan, M. Goodman, V. Mohan, and S. V. N. Vishwanathan. 2016. Adaptive, Personalized Diversity for Visual Discovery. In *ACM Conference on Recommender Systems (RecSys)*. 35–8.

[26] S. Vargas, L. Baltrunas, A. Karatzoglou, and P. Castells. 2014. Coverage, Redundancy and Size-Awareness in Genre Diversity for Recommender Systems. In *ACM Conference on Recommender Systems (RecSys)*.

[27] S. Vargas, P. Castells, and D. Vallet. 2012. Explicit Relevance Models in Intent-Oriented Information Retrieval Diversification. In *ACM Conference on Research and Development in Information Retrieval (SIGIR)*.

[28] B. Woodworth, S. Gunasekar, M.I. Ohannessian, and N. Srebro. 2017. Learning Non-Discriminatory Predictors. arXiv:1702.06081.

[29] S. Yao and B. Huang. 2017. Beyond Parity: Fairness Objectives for Collaborative Filtering. In *Advances in Neural Information Processing Systems (NIPS)*.

[30] B. Zadrozny and C. Elkan. 2001. Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In *International Conference on Machine Learning (ICML)*. 609–16.

[31] M. Zhang and N. Hurley. 2008. Avoiding monotony: improving the diversity of recommendation lists. In *ACM Conference on Recommender Systems (RecSys)*. 123–30.

[32] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. 2005. Improving recommendation lists through topic diversification. In *International World Wide Web Conference (WWW)*. 22–32.

[33] I. Zliobaite. 2015. A survey on measuring indirect discrimination in machine learning. arXiv:1511.00148.