Recommender Systems Evaluation

Alejandro Bellogín and Alan Said

Synonyms

Evaluation ; Methods ; Metrics ; Recommendation systems ; Reproducibility ;

Glossary

AUC Area under the curve
CF Collaborative Filtering
CTR Click-Through Rate
DCG Discounted Cumulative Gain
ILD Intra-list diversity
IR Information Retrieval
MAE Mean Absolute Error
MAP Mean Average Precision
ML Machine Learning
RMSE Root Mean Squared Error
ROC Receiver Operating Characteristic
RS Recommender System

Alejandro Bellogín
Universidad Autónoma de Madrid, Ciudad Universitaria de Cantoblanco, 28049 Madrid, Spain
e-mail: alejandro.bellogin@uam.es

Alan Said
School of Informatics, University of Skövde, Högskolevägen, Box 408, 541 28 Skövde, Sweden,
e-mail: alansaid@acm.org

Definition

The evaluation of RSs has been, and still is, the object of active research in the field. Since the advent of the first RS, recommendation performance has been usually equated to the accuracy of rating prediction, that is, estimated ratings are compared against actual ratings, and differences between them are computed by means of the MAE and RMSE metrics. In terms of the effective utility of recommendations for users, there is however an increasing realization that the quality (precision) of a ranking of recommended items can be more important than the accuracy in predicting specific rating values. As a result, precision-oriented metrics are being increasingly considered in the field, and a large amount of recent work has focused on evaluating top-N ranked recommendation lists with the above type of metrics.

Besides that, other dimensions apart from accuracy – such as coverage, diversity, novelty, and serendipity – have been recently taken into account and analyzed when considered what makes a good recommendation (Said et al, 2014b; Cremonesi et al, 2011; McNee et al, 2006; Bellogín and de Vries, 2013; Bollen et al, 2010).

So, what makes a good evaluation? The realization that high prediction accuracy might not translate to a higher perceived performance from the users has brought a plethora of novel metrics and methods, focusing on other aspects of recommendation (Said et al, 2013a; Castells et al, 2015; Vargas and Castells, 2014). Recent trends in evaluation methodologies point towards there being a shift from traditional methods solely based on statistical analyses of static data, i.e., raising precision performance of algorithms on offline data (Ekstrand et al, 2011b) – offline data in this case being recorded user interactions such as movie ratings or product purchases.

Evaluation is the key to identifying how well an algorithm or a system works. Deploying a new algorithm in a new system will have an effect on the overall performance of the system – in terms of accuracy and other types of metrics. Both prior deploying the algorithm, and after the deployment, it is important to evaluate the system performance.

It is in the evaluation of a RS one needs to decide on what should be sought-for, e.g., depending on whether the evaluation is to be performed from the users' perspective (accuracy, serendipity, novelty), the vendor's perspective (catalog, profit, churn), or even from the technical perspective of the system running the RS (CPU load, training time, adaptability). Given the context of the system, there might be other perspectives as well; in summary, what is important is to define the Key Performance Indicator (KPI) that one wants to measure.

Let us imagine an online marketplace where customers buy various goods, an improved recommendation algorithm could result in, e.g., increased numbers of sold goods, more expensive goods sold, more goods from a specific section of the catalog sold, customers returning to the marketplace more often, etc. When evaluating a system like this, one needs to decide on what is to be evaluated – what the sought-for quality is – and how it is going to be measured.

1 Introduction

Recommender Systems have been a popular research topic within personalized systems and information retrieval since the mid nineties. Throughout this time, various models of recommendation have been developed, e.g. approaches using *collaborative filtering*, or *matrix factorization* for purposes such as retrieval of ranked lists of items for consumption, or for the rating prediction task (which was made very popular through the Netflix Prize Bennett et al (2007)). Today, the use of recommender systems has spread to a very wide area of topics, including personalized healthcare (Elsweiler et al, 2015; Luo et al, 2016), online news portals (Said et al, 2014a), food (Elahi et al, 2015, 2014), social networks (Guy, 2015), exercise (Berkovsky et al, 2012), jobs (Abel, 2015), investment (Zhao et al, 2015), transportation (Bistaffa et al, 2015), shopping (Jannach et al, 2015), etc. The list can be made even longer. The very many domains that recommender systems are applied to, give rise to a plethora of recommendation approaches, algorithms, and settings that are tailored to the specific context and domain of the recommendation.

Given the various situations recommendations can be applied to, it follows that evaluation of these systems needs to be tailored to the specific setting, domain, userbase, context, etc. Moreover, RSs have several properties that affect the user experience (Gunawardana and Shani, 2015): accuracy, robustness, scalability, coverage, confidence, novelty, diversity, etc. These properties are associated to one or many evaluation metrics, whose values are, in the end, what will be compared for different recommendation algorithms in comparative studies of RS literature, where different evaluation methodologies are applied depending on the experimental settings (Bellogín, 2012; Said, 2013).

This chapter attempts to give an overview of some of the more commonly used evaluation methods and metrics used for various types of recommendation. The overview should be seen as an introduction to evaluation and not a definitive guide to RS evaluation.

2 Key Points

To accurately measure the efficiency of a recommender system, a proper evaluation that is tailored to the sought-for qualities of the recommender system needs to be created, aiming to adapt metrics, strategies/protocols, user tasks, etc. to the scenario at hand. Given the many situations in which recommender systems can be applied, there are many variations for each evaluation component (metric, protocol, etc.). In this entry, we describe the processes, methods, strategies, and metrics used for the purpose of evaluating recommender systems. We also present the main challenges, applications, and future directions in the area, evidencing that evaluation of recommender systems is a popular and active research problem and, still, an open problem in the field.

3 Historical Background

In 1994, Resnick and colleagues published what is considered the first modern research paper on RSs (Resnick et al, 1994). In the two decades that have passed since, RS has grown exponentially as research topic. In the early stages of RS research and development, evaluation focused on predicting future ratings that users will give to items which have not been rated thus far. Generally, this meant: the closer the predicted rating to the actual rating provided by the user, the better the recommendation.

Over the years, RSs have changed, and along them, so have the evaluation methods used to evaluate them. Historically, evaluation methods and metrics (see the next section for a thorough definition of all these) were adapted from IR, ML, and classification systems (precision and recall, nDCG, ROC, AUC, etc.). RMSE became the focus of RS evaluation after having been used as the default evaluation measure in the Netflix Prize. After the prize concluded, many new methods and metrics were developed, specifically tailored for RSs used within Web 2.0, the interactive web, by integrating and adapting to how users interact with this new generation of RSs.

Let us return to the Netflix Prize to exemplify this. When the service started, customers would order a set of DVDs from the website, wait for a few days until the package arrived, watch the movies, rate them, and return the DVDs. The newly added rating would then be used to generate new recommendations for the customers, for the next round of DVD rentals. The turn-around between recommendations, consumption, and rating was at least a few days. This changed when, instead of ordering DVDs, customers began to streaming content, the turn-around period went from a few days to minutes. This is the context of recommenders today. The feedback loop to the RS is instantaneous, the evaluation methods and metrics have adapted to this. Instead of focusing on predicting the rating given for a movie, a song, or a product, the evaluation focuses on how much of a movie has been watched, how many times a song has been skipped in a playlist, users' dwell-times while reading online newspaper articles, and so on.

Once these *first-generation metrics* (that measure prediction error) became obsolete, metrics from IR were adapted to take into account the ranking presented to the user, instead of the actual rating predicted for the user-item pair (acknowledging the fact that an algorithm that learns better than another the 1's and 2's of a particular user has no practical utility, since those items will never be recommended). Nowadays, these metrics are adapted to many different contexts including very small interfaces (short rankings) and online recommendation. A *third generation* of metrics appears when user data and their associated recommendations move from batch mode (as in standard offline experimentation) to online or streaming mode. In this situation, usually only clicks (CTR) can be measured and additional statistics should be computed if significance of the result has to be estimated (because of a high variance in the population). Finally, side data associated with the interaction between the user and the system has also been considered in the last years. Dwell time, returning visitors, buying events, etc. can be computed in specific domains assuming we can identify the user on a sequence of visits to the system. These metrics are, in

principle, closer to measuring the real value of the recommendation – as we shall discuss in following sections – and therefore are being incorporated in the evaluation of real RSs such as newspapers and e-commerce businesses.

In parallel to the evolution of evaluation metrics and tasks, more datasets have been made publicly available, some of them include new dimensions (social, temporal, clicks, multi-dimensional ratings, personality, etc.) and domains (microblogging, movies, music, e-commerce, recipes, and more). This has caused the evaluation ecosystem to adapt itself: as an example, context-based RSs need their own evaluation metrics that are aware of other dimensions (time, emotion, etc.) (Campos et al, 2014; Tkalcic et al, 2016). A further proof of this evolution are the different challenges and competitions – KDD Cup, RecSys Challenge, and some of the competitions hosted on Kaggle – related to RS where either ad-hoc metrics or others based on rankings have been used. While none of these challenges has had the huge impact the Netflix Prize had, it is true that more and more practitioners are being introduced to the field through these venues, and hence, the experimental methodologies and datasets used there might be considered standard practice.

4 Evaluating Recommender Systems

The evaluation of RSs has been a major object of study in the field since its earliest days, and is still a topic of ongoing research, where open questions remain (Herlocker et al, 2004; Gunawardana and Shani, 2015). It is acknowledged that the evaluation of RSs should take into account the goal of the system itself. For example, (Herlocker et al, 2004) identify two main user tasks: *annotation in context* and *find good items*. In these tasks the users only care about errors in the item rank order provided by the system, not the predicted rating value itself. Based on this consideration, researchers have started to use precision-based metrics to evaluate recommendations, although most works also still report error-based metrics for comparison with state of the art approaches. Moreover, other authors (Herlocker et al, 2004; Gunawardana and Shani, 2015) encourage considering alternative performance criteria, like the novelty of the suggested items and the item coverage of a recommendation method. We describe the above types of evaluation metrics in the subsequent sections.

Not all the evaluation metrics can be computed under any experimental settings. Different evaluation protocols exist and they impose constraints on the type of data that can be measured and analyzed. Two main evaluation protocols are usually considered: offline and online. These protocols present a clear tradeoff between effort (time, users, etc.) and usefulness/trustworthiness of the results, which will be discussed in later sections.

Finally, we present some of the most important challenges that appear when evaluating RSs. As we stated previously, several open questions remain in RS evaluation, and some of the most important ones are related to how replicable the conducted experiments in RSs could be; besides, a deeper understanding of the biases

present in evaluation and how to combine the different dimensions and metrics is still needed in the field – these aspects will be presented and discussed in the last part of this section.

4.1 Evaluation Metrics

In the classical formulation of the recommendation problem, user preferences for items are represented as numeric ratings, and the goal of a recommendation algorithm consists of predicting unknown ratings based on known ratings and, in some cases, additional information about users, items, and the context. In this scenario, the accuracy of recommendations has been commonly evaluated by measuring the error between predicted and known ratings, using error metrics. Although dominant in the literature, some authors have argued this evaluation methodology is detrimental to the field since the recommendations obtained in this way are not the most useful for users (McNee et al, 2006). Acknowledging this, recent work has evaluated top-N ranked recommendation lists with precision-based metrics (Cremonesi et al, 2010; McLaughlin and Herlocker, 2004; Bellogín et al, 2011; Said and Bellogín, 2014), drawing from well studied evaluation methodologies in the Information Retrieval field. We present and discuss the most prominent of these metrics later on.

In a more modern formulation of the recommendation problem, the ratings are no longer important. Instead, the consumption of recommended items by users is key, i.e. whether a recommended movie will be seen, a music track listened to, a product purchased, etc. In this context, it is important to ask oneself what the recommender system should bring the user? If a recommender system recommends an item that the user already knows of, what is the value of the system? Will this recommendation result in the consumption of the item? The assumption is that a recommender system, in this context, should bring the user something she might not yet be familiar with, i.e. something novel, unexpected, or serendipitous (Vargas et al, 2014; Vargas and Castells, 2013). Still, the novel, unexpected, or serendipitous recommendations need to fulfill the requirement of the items being of actual interest to the user. These, so-called, non-accuracy metrics focus on the variety, popularity, novelty and similar aspects of the items, or lists of items, that are recommended (Ziegler et al, 2005; Ziegler and Lausen, 2009). In this section, we present some of the most common non-accuracy metrics and the motivations behind them.

An often forgotten dimension of evaluation, at least in academic research, are *business-related* metrics. Within this domain, there are metrics which are purely based on economy and market growth, e.g. customer churn and profit margin on customer purchases; additionally, there are metrics which are related to the development, and maintenance of the recommender system itself, such as CPU cost per recommendation, storage cost, cost of re-training the model, etc. A brief overview of this metric type ends this section.

4.1.1 Error Metrics

A classic assumption in the RS literature is that a system that provides more accurate predictions will be preferred by the user (Gunawardana and Shani, 2015). Although this has been further studied and refuted by several authors (McNee et al, 2006; Cremonesi et al, 2011; Bollen et al, 2010), the issue is still worth being analyzed. Traditionally, the most popular metrics to measure the accuracy of a RS have been the **Mean Absolute Error** (MAE), and the **Root Mean Squared Error** (RMSE):

$$\text{MAE} = \frac{1}{|\text{Te}|} \sum_{(u,i)\in \text{Te}} |\tilde{r}(u,i) - r(u,i)| \tag{1}$$

$$\text{RMSE} = \sqrt{\frac{1}{|\text{Te}|} \sum_{(u,i)\in \text{Te}} (\tilde{r}(u,i) - r(u,i))^2} \tag{2}$$

where $\tilde{r}$ and r denote the predicted and real rating, respectively, and Te corresponds to the test set. The RMSE metric is usually preferred to MAE because it penalizes larger errors.

Different variations of these metrics have been proposed in the literature. Some authors **normalize MAE** and **RMSE** with respect to the maximum range of the ratings (Goldberg et al, 2001; Gunawardana and Shani, 2015) or with respect to the expected value if ratings are distributed uniformly (Marlin, 2003; Rennie and Srebro, 2005). Alternatively, **per-user** and **per-item** average errors have also been proposed in order to avoid biases from the error on a few very frequent users or items (Massa and Avesani, 2007; Gunawardana and Shani, 2015).

A critical limitation of these metrics is that they do not make any distinction between the errors made on the top items predicted by a system, and the errors made for the rest of the items. Furthermore, they can only be applied when the recommender predicts a score in the allowed range of rating values. Because of that, implicit and some content-based and probabilistic recommenders cannot be evaluated in this way, since $\tilde{r}(u,i)$ would represent a probability or, in general, a preference score, not a rating. Hence, these methods can only be evaluated by measuring the performance of the generated ranking using precision-based metrics.

4.1.2 Ranking Metrics

Ranking or precision-based metrics measure the accuracy of a list of recommendations, usually taking into account the natural browsing order (Gunawardana and Shani, 2015). In contrast to the previous metrics, any algorithm that sorts the items in a particular way could be evaluated – that is, we are not restricted to those predicting an explicit rating. These metrics can be classified into three groups: metrics that only use one ranking, metrics that compare two rankings (typically, one of them is a reference or ideal ranking), and metrics from the ML field.

Table 1 Formulation of ranking metrics[a].

Metric	Definition
P@k	$\frac{1}{\lvert\mathscr{U}\rvert}\sum_{u\in\mathscr{U}}\frac{\lvert\mathrm{Rel}_u@k\rvert}{k}$
R@k	$\frac{1}{\lvert\mathscr{U}\rvert}\sum_{u\in\mathscr{U}}\frac{\lvert\mathrm{Rel}_u@k\rvert}{\lvert\mathrm{Rel}_u\rvert}$
MAP	$\frac{1}{\lvert\mathscr{U}\rvert}\sum_{u\in\mathscr{U}}\frac{1}{\lvert\mathrm{Rel}_u\rvert}\sum_{i\in\mathrm{Rel}_u}P@\mathrm{rank}(u,i)$
nDCG	$\frac{1}{\lvert\mathscr{U}\rvert}\sum_{u\in\mathscr{U}}\frac{1}{\mathrm{IDCG}_u^{p_u}}\sum_{p=1}^{p_u}f_{\mathrm{dis}}(\mathrm{rel}(u,i_p),p)$, usually $f_{\mathrm{dis}}(x,y)=\frac{2^x-1}{\log(1+y)}$
HL	$100\,(\sum_{u\in\mathscr{U}}\mathrm{HL}_u^{\max})^{-1}\sum_{u\in\mathscr{U}}\mathrm{HL}_u$, $\mathrm{HL}_u=\sum_{p=1}^{p_u}\frac{\max(\bar{r}(u,i_p)-d,0)}{2^{(p-1)/(\alpha-1)}}$
MRR	$\sum_{u\in\mathscr{U}}\frac{1}{s_r(u)}$
NDPM	$\frac{1}{\lvert\mathscr{U}\rvert}\sum_{u\in\mathscr{U}}\frac{2C_u^{\mathrm{con}}+C_u^{\mathrm{tie}}}{2C_u}$

[a] Notation: Rel_u represents the set of relevant items for user u, $\mathrm{Rel}_u@k$ is the number of relevant recommended items up to position k, $\mathrm{rank}(u,i)$ outputs the ranking position of item i in the user's u list; IDCG_u^k denotes the score obtained by an ideal or perfect ranking for user u up to position k; p_u denotes the maximum number of items evaluated by each user (sometimes assumed to be a cutoff k, the same for all the users); d is the default rating (or neutral vote), and α is the half-life utility that represents the rank of the item on the list such that there is a 50% chance that the user will view that item (Breese et al (1998) use a value of 5 in their experiments, and note that they did not obtain different results with a half-life of 10); and $s_r(u)$ is a function that returns the position of the first relevant item obtained for user u. Finally, C_u is the number of pairs of items for which the reference ranking asserts an ordering, i.e., the items are not tied. Besides, C_u^{con} denotes the number of discordant item pairs between the method's ranking and the reference ranking, and C_u^{tie} represents the number of pairs where the reference ranking does not tie, but where the method's ranking does.

Examples of metrics based on one ranking are precision, recall, normalized discounted cumulative gain, mean average precision, and mean reciprocal rank (see Table 1). Each of these metrics captures the quality of a ranking from a slightly different angle. More specifically, **precision** accounts for the fraction of recommended items that are relevant, whereas **recall** is the fraction of the relevant items that has been recommended. Both metrics are inversely related, since an improvement in recall typically produces a decrease in precision. They are typically computed up to a ranking position or cutoff k, being denoted as $P@k$ and $R@k$ (Baeza-Yates and Ribeiro-Neto, 2011). Note that recall has also been referred to as **hit-rate** in (Deshpande and Karypis, 2004). Hit-rate has also been defined as the percentage of users with at least one correct recommendation (Bellogín et al, 2013), corresponding to the **success** metric (or first relevant score), as defined by TREC (Tomlinson, 2012).

The **mean average precision** (MAP) metric provides a single summary of the user's ranking by averaging the precision figures obtained after each new relevant item is obtained (Baeza-Yates and Ribeiro-Neto, 2011). **Normalized discounted cumulative gain** (nDCG) uses graded relevance that is accumulated starting at the top of the ranking and may be reduced, or discounted, at lower ranks (Järvelin and Kekäläinen, 2002). Using a different discount function, the **rank score** or **half-life utility** metric (Breese et al, 1998; Herlocker et al, 2004) can be obtained from the nDCG formulation (see Table 1).

Mean reciprocal rank (MRR) favors rankings whose first correct result occurs near the top ranking results (Baeza-Yates and Ribeiro-Neto, 2011). This metric is similar to the **average rank of correct recommendation** (ARC) proposed in (Burke, 2004) and to the **average reciprocal hit-rank** (ARHR) defined in (Deshpande and Karypis, 2004).

Additionally, specific metrics have been defined in the context of RS evaluation that take as inputs two rankings (ideal vs estimated) instead of just one. A first example is the normalized distance-based performance measure (NDPM), used in (Balabanovic and Shoham, 1997), and proposed in (Yao, 1995). This metric compares two different weakly ordered rankings, considering the number of concordant and discordant pairs. This metric is comparable across datasets since it is normalized with respect to the worst possible scenario. Furthermore, it provides a perfect score of 0 to systems that correctly predict every preference relation asserted by the reference, and a worst score of 1 to methods that contradict every reference preference relation. Besides, a penalization of 0.5 is applied when a reference preference relation is not predicted, whereas predicting unknown preferences (i.e., they are not ordered in the reference ranking) receives no penalization.

Rank correlation metrics such as **Spearman's** ρ and **Kendall's** τ have also been proposed to directly compare the system ranking to a preference order given by the user. These correlation coefficients provide scores in the range of -1 to 1, where 1 denotes a perfect correlation between the two rankings, and -1 represents an inverse correlation. These two metrics, along with NDPM, suffer from the interchange weakness (Herlocker et al, 2004), that is, interchanges at the top of the ranking have the same weight that interchanges at the bottom of the ranking.

Finally, metrics from the ML literature have also been used, although they are not as popular as those described above. For instance, metrics based on the **receiving operating characteristic** (ROC) curve and the **area under the curve** (AUC) provide a theoretically grounded alternative to precision and recall (Herlocker et al, 2004). The ROC model attempts to measure the extent to which an information filtering system can successfully distinguish between signal (relevant items) and noise. Starting from the origin of coordinates at $(0,0)$, the ROC curve is built by considering, at each rank position, whether the item is relevant or not for the user; in the first case, the curve goes one step up, and in the second, one step right. A random recommender is expected to produce a straight line from the origin to the upper right corner; on the other hand, the more leftwards the curve leans, the better is the performance of the system. These facts are related to the area under the ROC curve, a summary metric that is expected to be higher when the recommender performs better, where the expected value of a random recommender is 0.5, corresponding to a diagonal curve in the unit square.

4.1.3 Non-accuracy Metrics

Non-accuracy metrics attempt to measure the quality of the recommendation not in terms of how well the system is able to mimic the users' history. Research and

development of these metrics has grown rapidly in recent years, the reason for this is the realization that the data in a user's prior preferences might not necessarily be enough to generate future models of the user's preference. Additionally, as will be discussed later in this entry, traditional *offline* evaluation methods often suffer from various biases, e.g. popularity, recency, etc.

These metrics attempt to measure a holistic quality of the recommendations, i.e. how well will the recommendation be received by the end user, how will the users' experience of the system be affected by the recommendation; but also how the system performs in terms of e.g. the catalog of items available. Due to the nature of some of these metrics, it is difficult to create a ground truth dataset to use with them. Thus, recommendation algorithms which are specifically tailored towards non-accuracy metrics will often perform badly in terms of accuracy-based metrics (Said et al, 2013a); and symmetrically, if an algorithm is tailored towards accuracy metrics, it will often perform badly in terms of non-accuracy metrics.

Due to the nature of non-accuracy metrics, there are often various definitions of them, each tailored towards the context they are used in. In this chapter, we follow the same definitions as in the Recommender Systems Handbook (Ricci et al, 2015).

Perhaps the most well-known non-accuracy metric, **serendipity**, attempts to model what is often referenced to as a *pleasant and unexpected surprise*, similar to finding a banknote on the pavement. Serendipity is a compound metric, or concept, of amongst others **novelty** and **diversity**. Novelty expresses how new a recommended item is (for a user). The underlying motivation for novelty being an interesting aspect of recommendation is that items which are old, or rather not new, can already have been seen by the user. If this is the case, recommending items which are already known by the user might not be of much value, since the recommendation does not actually present the user with something she could not find herself. Novelty can be directly measured in online experiments by asking users whether they are familiar with the recommended item (Celma and Herrera, 2008). However, it is also interesting to measure novelty in an offline experiment, so as not to restrict its evaluation to costly and hardly reproducible online experiments. While novelty often can have a negative effect on accuracy, diversity can often be increased without necessarily sacrificing accuracy. Diversity expresses, as implied, the variety of the recommended items. There are multiple ways of measuring diversity in a set of recommended items, commonly this is done by measuring the *intra-list diversity* (Smyth and McClave, 2001) which is defined as

$$\text{ILD} = \frac{1}{|R|(|R|-1)} \sum_{i \in R} \sum_{j \in R} (1 - s(i,j)) \tag{3}$$

where $s(i,j)$ is a similarity measure reporting on the similarity of items i and j given some predefined set of item features. When using ILD as a measure, the goal is to generate a list of recommended items that contains items that are both accurate and diverse.

Metrics based on Information Theoretic properties of the items being recommended have also been proposed by several authors. In (Bellogín et al, 2010) the entropy function is used to capture the novelty of a recommendation list, in (Zhou et al, 2010) the authors use the self-information of the user's top recommended items, and in (Filippone and Sanguinetti, 2010) the Kullback-Leibler divergence is used.

Other metrics such as privacy, adaptivity, and confidence have been explored to a lesser extent, but their importance and application to recommender systems have been discussed, making clear their relation with the user's experience and satisfaction, which is the ultimate goal of a "good" recommender system (Herlocker et al, 2004; McNee et al, 2006; Gunawardana and Shani, 2015).

4.1.4 Business Metrics

As stated earlier, business metrics belong to an often neglected dimension in research literature. However, it should be safe to assume that business metrics are among the most important indicators of how well a recommender performs. This contradiction is in part due to the difficulty in measuring the business impact of a recommendation algorithm prior to deployment within a service, which could in turn explain the lesser focus on this from the research community.

Where error metrics, ranking metrics, and non-accuracy metrics attempt to measure an aspect of data interaction such as a user clicking on a recommendation, a user rating an item, or similar, business metrics measure the quality of the recommendation as seen from the vendor's perspective. It should not be forgotten that most systems that utilize recommendation do so with the purpose of increasing revenue, and naturally keeping their users satisfied with the offered service. One common business metric quite simply measures the **coverage** of the list of recommended item, i.e. how well does the list correspond to what is currently in stock or elseways available. Coverage does however not only apply to the catalog of available items, coverage also applies to the users, i.e. an algorithm which can recommend very accurate items, but only for a small portion of users might not be as "good" as a slightly less accurate algorithm with a higher user coverage. In (Gunawardana and Shani, 2015) two metrics are proposed for measuring item coverage: one based on the Gini index, and another based on Shannon's entropy. In (Ge et al, 2010) the authors propose simple ratio quantities to measure such metrics, and to discriminate between the percentage of the items for which the system is able to generate a recommendation (*prediction coverage*), and the percentage of the available items that are effectively ever recommended (*catalog coverage*).

A metric that cannot be measured without a running system with a large number of users is **churn rate**. Churn rate is not specifically linked to RSs, however for a service which focuses on recommendation, it should be an important indicator of the service quality. The churn rate is a measure of the users that are leaving the service for another. One variation of churn rate, revenue churn, measures the changes in revenue.

Additional business-related measures of recommendation quality state the balance between the complexity of an algorithm in terms of development cost, CPU cost, maintenance, versus the benefits it brings to the vendor (Amatriain and Basilico, 2012).

4.2 Evaluation Methods

The metrics presented before can be applied under different experimental conditions. On the one hand, offline evaluation allows to compare a wide range of candidate algorithms at a low cost, it is easy to conduct and does not require any interaction with real users. On the other hand, user studies and online experiments are more trustworthy – since the system is used by real users and interacted with in real time – but care must be taken to consider biases in the experimental design (Gunawardana and Shani, 2015).

4.2.1 Offline Evaluation

An important decision in the experimental configuration of a recommender evaluation is the dataset partition strategy. How the datasets are partitioned into training and test sets may have a considerable impact on the final performance results, and may cause some recommenders to obtain better or worse results depending on how this partition is configured.

First, we have to choose whether or not to take time into account (Gunawardana and Shani, 2015). Time-based approaches naturally require the availability of user interaction data timestamps. A simple approach is to select a time point in the available interaction data timeline to separate training data (all interaction records prior to that point) and test data (dated after the split time point). The split point can be set so as to, for instance, have a desired training/test ratio in the experiment. The ratio can be global, with a single common split point for all users, or user-specific, to ensure the same ratio per user. Time-based approaches have the advantage of more realistically matching working application scenarios, where "'future"' user likes (which would translate to positive response to recommendations by the system) are to be predicted based on past evidence. As an example, the well-known Netflix Prize provided a dataset where the test set for each user consisted on her most recent ratings (Bennett et al, 2007).

If we ignore time, there are at least the following three strategies to select the items to hide from each user: a) sample a fixed number (different) for each user; b) sample a fixed (but the same for all) number for each user, also known as *given n* or *all but n* protocols; c) sample a percentage of all the interactions using cross-validation. The most usual protocol is the last one (Goldberg et al, 2001), although several authors have also used the *all but n* protocol (Breese et al, 1998). Figure 1 shows an example of a random dataset partition.

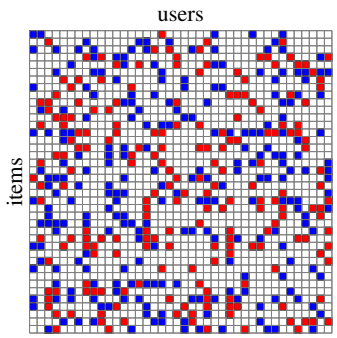

Fig. 1 How the dataset would be split into training (blue) and test (red) sets. White cells denote unknown values in the user-item matrix.

Nonetheless, independently from the dataset partition, it is recognized that the goals for which an evaluation is performed may be different in each situation, and thus, a different setting (and partition protocol) should be developed (Herlocker et al, 2004; Gunawardana and Shani, 2015). If that is not the case, the results obtained in a particular setting would be biased and difficult to use in further experiments, for instance, in an online experimentation.

Regarding the actual evaluation process, there is a relation between the evaluation protocol and the evaluation metrics that can be computed. Error metrics require explicit ground truth values for every user-item pair that needs to be evaluated – that is, only items in the test set of each user will be considered. Ranked recommendations (using the metrics described before), on the other hand, require for a target user u to select two sets of items, namely relevant and not relevant items. The following candidate generation strategies, where L_u denotes the set of target items the recommender ranks (candidate items), have been proposed (we follow the notation presented in (Said and Bellogín, 2014)):

UserTest This strategy takes the same target item sets as standard error-based evaluation: for each user u, the list L_u consists of items rated by u in the test set. The smallest set of target items for each user is selected, including no unrated items. A relevance threshold is used to indicate which of the items in the user's test are considered relevant. Threshold variations can be static for all users (Jambor and Wang, 2010), or per-user (Basu et al, 1998).

TrainItems Every rated item in the system is selected – except those rated by the target user. This strategy is useful when simulating a real system where no test is available, i.e. no need to look into the test set to generate the rankings (Bellogín et al, 2011). The relevant items for each user consist of those included in her test set; the use of a threshold to consider only highly rated items is optional although recommended.

RelPlusN For each user, a set of highly relevant items is selected from the test set. Then, a set of non-relevant items is created by randomly selecting N additional items. In (Cremonesi et al, 2010), N is set to $1,000$ stating that the non-relevant

items are selected from items in the test set not rated by u. Finally, for each highly relevant item i, the recommender produces a ranking of the union between this item and the non-relevant items.

As it was observed in (Bellogín et al, 2011) and (Said and Bellogín, 2014), each of these candidate generation strategies, may produce a different ranking of recommendation performance – TrainItems and RelPlusN produce consistent results (although with different absolute values) whereas UserTest obtains results closer to those from error-based metrics. Hence, we should pay attention to the strategy used when ranking metrics are computed, since the amount of relevant items considered can drastically change the output of the experiment. In contrast to IR, in RS we have to define training and test sets, whereas in IR, we would have the whole dataset available, first, for the indexing task, and then, for the retrieval and evaluation tasks. In RS, we need to separate the data into training and test; the more the training available, the better the algorithm will learn the users' preferences. However, the smaller the test set, the smaller the confidence on the obtained results. This sparsity in the groundtruth dimension may produce biases in the evaluation results, as observed in (Bellogín, 2012).

4.2.2 Online evaluation

Online evaluation, as opposed to traditional offline evaluation is performed through direct involvement of a system's users in order to establish a *qualitative* assessment of the system's quality *as perceived by the end users*.

To illustrate one of the key differences between offline evaluation and online evaluation, consider this top-N recommendation scenario: We have a user-item interaction matrix, as shown in Table 2. The table shows a matrix of 5 users and 6 items and their interactions, e.g. a 1 represents an interaction (rating, purchase, etc.), a 0 the lack of such. The training/test split is illustrated by the the dashed line. In this case, an offline evaluation will only recognize item i_5 as a true positive recommendation for user u_3 and items i_5 and i_6 for user u_4. Users u_1, u_2 and u_5 will not have any true positive recommendations since they have not interacted with any of the items. The evaluation does not consider that the items might actually be liked by the user, if recommended in a real-world situation. Similarly, the fact that u_3 has interacted with i_5 does not need to imply that the item is a good recommendation.

	u_1	u_2	u_3	u_4	u_5
i_1	1	1	0	0	1
i_2	1	0	1	1	1
i_3	0	0	0	1	0
i_4	1	0	1	0	1
i_5	0	0	1	1	0
i_6	0	0	0	1	0

Table 2 A user-item matrix divided into a training set (above the dashed line) and a test set (below the dashed line).

In order to overcome this deficiency, online evaluation attempts to capture the quality of the recommendation as perceived by the users by analyzing their interaction patterns with the system together with explicitly asking questions.

Online evaluation commonly involves a user study. Users can be made aware or encouraged to participate, or participate unknowingly. In real-life systems, the concept of *A/B testing* is readily used to estimate different algorithms' qualities (Kohavi et al, 2009). A/B testing involves assigning a subset of a system's users to the algorithm under evaluation. In studies of real life systems, users are usually not made aware of their participation in tests (Kohavi et al, 2009). The interactions of the users are then analyzed and compared to a baseline.

More elaborate user studies, including questionnaires and other explicitly collected information serve as an alternative to A/B testing. This type of studies commonly involve asking the users questions throughout, or after, their interaction with the system. In studies like this, the participants are naturally aware of their participation in the study. In order to be able to analyze the results quantitatively, the users are asked to agree or disagree with a question in the form of a statement. The scale of (dis)agreement is represented as a *Likert* scale, a basic example of a Likert scale is shown in Figure 2.

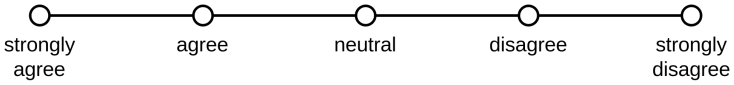

Fig. 2 A Likert scale where users are asked to agree or disagree with a specific question.

Online user-centric evaluation is a means to measure the level of *subjectively perceived quality* by the users of a system. There is a large body of work conducted on how to perform and evaluate user studies and surveys based on statistical significance tests.

There is no default, quality-related, set of questions to ask when performing a recommender systems user study, instead questions are based on the type of quality that is sought for; whether relating to the concepts mentioned above or to rather technical qualities, e.g. time of recommendation, number of items recommended, etc. This type of user studies need to be meticulously planned and executed. If poorly executed, there is a risk of changing the users' opinions, e.g. through suggestive questions, or excessive work load or time involved in answering the questions. Work load and time-related issues can be mitigated by creating an incentive for the users to fulfill the survey, e.g. raffling off vouchers, prizes, etc. If no incentive is given, the time involved in answering the survey creates a decaying effect on the fraction of users who complete the study. When the users are given an incentive, there is a risk that some users will answer the questions quickly (at random) in order to be eligible for the award (Swearingen and Sinha, 2001). In order to mitigate these effects, the number of questions and work load should be kept relatively low.

4.3 Challenges

Evaluating recommender systems is not an easy task. In all fairness, it is not *a* task, it is a set of several interconnected and standalone tasks that, when viewed together, should result in one (or several) measure(s) which then state the quality of the recommender. So far, we have presented measures, metrics, methods, and a general history of recommender system evaluation. In this section we focus on the challenges, the pitfalls, and other related concepts that make evaluation difficult.

4.3.1 Technical Challenges

Earlier in this chapter, evaluation measures and evaluation methods have been presented. In order to accurately evaluate a RS, a combination of metrics and methods are used. The challenge becomes, for instance, what metrics to use when evaluating, or how many metrics to use. This poses the problem of how to combine the various metrics into something that can be optimized. Even though there is a significant body of work on multi-criteria RS optimization, there is no clear guideline on how to perform this type of evaluation (Jambor and Wang, 2010; Ge et al, 2010; Said et al, 2013b). State of the art methods in multi-criteria evaluation require a trade-off between the different metrics to be evaluated, this can then be applied to offline evaluation. Online evaluation using several criterions requires elaborate qualitative analyses of long term results of recommendation approaches.

Additional challenges related to offline evaluation focus on aspects of the underlying data which is used for both training and evaluation of RSs. The concept known as the *magic barrier* of RS (Herlocker et al, 2004) concerns the fact that user interactions are not concise, i.e. they contain noise and other irregularities which make the data not fully trustworthy. The effect of this is that when optimizing towards a certain metric, there is an upper level, a threshold beyond which optimization is useless (Said et al, 2012). This threshold is unknown; at best, it can be estimated assuming there are enough interactions provided by each user (Bellogín et al, 2014).

There exist multiple other factors making evaluation difficult, and at some points even flawed. Most RSs are used for the purpose of alleviating users in finding interesting information, this usually creates a bias in terms of popularity, i.e. the fact that some items are more popular than others create synthetic similarities between users based solely on their interaction with popular items. If not taken into consideration, the evaluation of a RS can point to an algorithm performing significantly better than it would in a real world situation simply due to the fact that it will recommend mostly popular items (Zhao et al, 2013), i.e. items that the user does not need to get recommended as there is a high probability that she already know of the item.

4.3.2 Replication and Reproducibility

Looking back on the accumulated amount of work in the research community, there have emerged several recommendation frameworks: *Apache Mahout* (Mahout) (Owen et al, 2011), *LensKit* (Ekstrand et al, 2011a), *MyMediaLite* (Gantner et al, 2011), *RankSys* (Vargas, 2015), etc. Even though the frameworks provide basically the same recommendation algorithms, they have differences in their implementations, data management, and evaluation methods. Additionally, the frameworks provide basic evaluation packages able to calculate some of the most common evaluation metrics; however, due to the differences in implementation of the same algorithm across frameworks, even when using the same dataset, it becomes uncertain whether a comparison of results from different frameworks is possible.

In (Said and Bellogín, 2014), a standalone evaluation of recommendation results was performed, allowing fine-grained control of a wide variety of parameters within the evaluation methodology. In that work, a cross-system and cross-dataset benchmark of some of the most common recommendation and rating prediction algorithms was presented. The results obtained there highlights the differences in recommendation accuracy between implementations of the same algorithms on different frameworks, distinct levels of accuracy in different datasets, and variations of evaluation results on the same dataset and in the same framework when employing various evaluation strategies. It is important to note, thus, that inter-framework comparisons of recommendation quality can potentially point to incorrect results and conclusions, unless performed with great caution and in a controlled, framework independent, environment.

In summary, the issues of replication – obtaining the exact same results in the same setting – and reproducibility – obtaining comparable results using a different setting – are very difficult challenges at the moment. They force researchers to reimplement the baseline algorithm they want to compare their approach against, or to pay extra attention to every algorithmic and evaluation detail, ignoring if the observed discrepancies with respect to what already was published come from omitted details from the original papers (parameters, methodologies, protocols, etc.) or a wrong interpretation of any of these intermediate steps.

5 Key Applications

In general, evaluation of recommender systems is applied whenever a RS is used. Services that use recommender systems, whether for entertainment (Netflix, Spotify, Pandora), e-learning (Coursera, EdX), networking (LinkedIn, Facebook, Twitter), shopping (Amazon, eBay), etc. needs to be evaluated in order to identify whether the recommender system in use is of practical utility to the users and to the service operators. However, evaluation is not only applied in real-world systems. Recommender systems research is contingent on the evaluation of these systems. When a novel recommendation algorithm is created by researchers, the usefulness of the

approach needs to be established. In the vast majority of the cases this is done by comparing the evaluation of the new approach to a state of the art baseline algorithm. Evaluation is thus applied in order to benchmark recommender systems in academia as well as in industry.

There are pitfalls related to evaluation. In a closely related research topic, Information Retrieval, Armstrong et al (2009) reported that there was little measurable improvement in certain types of retrieval tasks over the course of ten years, even though there was a large body of scientific work suggesting so. The reason for this appears to have been lack of guidelines, use of weak baselines, and not enough comparisons to results obtained in previous works. Drawing a lesson from this, we stress the importance of proper evaluation, in terms of methods, metrics, and general execution.

6 Future Directions

The empirical evaluation of RS is acknowledged to be an open problem in the field, with open issues yet to be addressed (Gunawardana and Shani, 2015). Many experimental approaches and metrics have been developed over the years, which the community is well acquainted with, but key aspects and details in the design and application of available methodologies are open to configuration and interpretation, where even apparently subtle details may create a considerable difference. This results in a significant divergence in experimental practice, hindering the comparison and proper assessment of contributions and advances to the field.

In this context, the replication and reproduction of experiments is one of the desirable requirements for experimental research still to be met in the field, as it was already described in the technical challenges section. The discussion and definition of the basic elements of the experimental conditions (and their requirements) are critical to support continuous innovation in any discipline. The offline evaluation of RS requires an implementation of the algorithm or technique to be evaluated, a set of quality measures for comparative evaluation, and an experimental protocol establishing how to handle the data and compute metrics in detail. Online evaluation similarly requires an algorithm implementation and a population of users to survey (by means of an A/B test, for instance). Here again, perhaps even more importantly than in offline evaluation, an experimental protocol needs to be established and adhered to. However, even when a set of publicly available resources (data and algorithm implementations) exists in the RS community, very often research studies do not report comparable results for the same methods *under the same conditions* (Said and Bellogín, 2014). This is due to the high number of experimental design options and parameters in RS evaluation, and the huge impact of the experiment configuration on the outcomes. In order to seek reproducibility and replication, several strategies can be considered, such as source code sharing, standardization of agreed evaluation metrics and protocols, or releasing public experimental design software, all of which have difficulties of their own. Furthermore, for online evaluation, an exten-

sive analysis of the population of test users should be provided. While the problem of reproducibility and replication has been recognized in the community, the need for a solution remains largely unmet.

Another open problem when evaluating RS is the relation between online and offline experiments. Several authors have explored this issue in some domains but no conclusive results have been obtained (Garcin et al, 2014; Beel et al, 2013; de Souza Pereira Moreira et al, 2015). The main question is how to align the offline evaluation to the online (usually, with A/B tests) results? Some possibilities include designing a good evaluation methodology or using a sensible evaluation metric to the problem at hand. An interesting output that could be produced by a better understanding of this issue is that offline evaluation would predict which methods, parameters, or configurations will work better when integrated and tested in an online evaluation.

Additionally, there should be an increased effort in making metrics meaningful. Evaluation should be realistic and, at the same time, provide test setups and reproducible baselines. For this, we need an honest measure of the preference (predicted items may not be correct just because they were consumed), we should capture the value of the recommendation (it is hard to say if a recommender that is useful in the short term may be just too obvious) and not neglect contextual side-data. With this goal in mind, evaluation metrics from other areas – specially from Information Retrieval, considering the increasing importance item rankings have in recommendation nowadays – could be incorporated and its potential analyzed. As an example, the bpref metric (Buckley and Voorhees, 2004) was specifically defined to be robust to incomplete judgments sets – a very typical scenario in recommendation; however, this metric is seldom use in recommendation tasks. An interesting problem that is still unsolved is how to properly combine evaluation metrics measuring orthogonal dimensions. A paradigmatic example is the tradeoff between accuracy and coverage (Gunawardana and Shani, 2015), where we may prefer recommenders that can provide recommendations to a wider range of users, despite their lower global accuracy. Another example arises when several dimensions are considered and optimized at the same time (see, e.g., (Ribeiro et al, 2012)); in these cases, multi-objective evaluation measures should be used, like the one proposed in (Said et al, 2013b).

Finally, the evaluation of RSs deserves a closer analysis and more attention to the differences in the experimental conditions, and their implications on the explicit and implicit principles and assumptions on which the metrics are built. As it was detected and examined in (Bellogín, 2012), different statistical biases – e.g., test sparsity and item popularity – do interfere in recommendation experiments, even though they do not arise in evaluation scenarios on other domains (like IR).

Cross references

Community Detection and Recommender Systems ; Emotions and Personality in Recommender Systems ; Friends Recommendations in Dynamic Social Networks ;

Recommender Systems based on Linked Open Data ; Recommender Systems based on Social Networks ; Recommender Systems in Crowd Sourcing ; Recommender Systems Using Social Network Analysis: Challenges and Future Trends ; Recommender Systems, Basics of ; Recommender Systems, Semantic-Based ; Recommender Systems: Models and Techniques ; Social Recommendation in Dynamic Networks ; Social-based Collaborative Filtering ; Spatio-Temporal Recommendation in Geo-Social Networks ; Spatiotemporal Personalized Recommendation of Social Media Content ; User Behavior Prediction in Social Networks ;

References

Abel F (2015) We know where you should work next summer: Job recommendations. In: Werthner et al (2015), p 230, DOI 10.1145/2792838.2799496, URL http://doi.acm.org/10.1145/2792838.2799496

Amatriain X, Basilico J (2012) Netflix recommendations: Beyond the 5 stars (part 1) – the netflix tech blog. http://techblog.netflix.com/2012/04/netflix-recommendations-beyond-5-stars.html (retrieved July 27, 2016), URL http://techblog.netflix.com/2012/04/netflix-recommendations-beyond-5-stars.html

Amatriain X, Torrens M, Resnick P, Zanker M (eds) (2010) Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, Barcelona, Spain, September 26-30, 2010, ACM

Armstrong TG, Moffat A, Webber W, Zobel J (2009) Improvements that don't add up: ad-hoc retrieval results since 1998. In: Cheung DW, Song I, Chu WW, Hu X, Lin JJ (eds) Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM 2009, Hong Kong, China, November 2-6, 2009, ACM, pp 601–610, DOI 10.1145/1645953.1646031, URL http://doi.acm.org/10.1145/1645953.1646031

Baeza-Yates RA, Ribeiro-Neto BA (2011) Modern Information Retrieval - the concepts and technology behind search, Second edition. Pearson Education Ltd., Harlow, England, URL http://www.mir2ed.org/

Balabanovic M, Shoham Y (1997) Content-based, collaborative recommendation. Commun ACM 40(3):66–72, DOI 10.1145/245108.245124, URL http://doi.acm.org/10.1145/245108.245124

Basu C, Hirsh H, Cohen WW (1998) Recommendation as classification: Using social and content-based information in recommendation. In: Mostow J, Rich C (eds) AAAI/IAAI, AAAI Press / MIT Press, pp 714–720

Beel J, Genzmehr M, Langer S, Nürnberger A, Gipp B (2013) A comparative analysis of offline and online evaluations and discussion of research paper recommender system evaluation. In: Bellogín A, Castells P, Said A, Tikk D (eds) Proceedings of the International Workshop on Reproducibility and Replication in Recommender Systems Evaluation, RepSys 2013, Hong Kong, China, Oc-

tober 12, 2013, ACM, pp 7–14, DOI 10.1145/2532508.2532511, URL http://doi.acm.org/10.1145/2532508.2532511

Bellogín A (2012) Recommender system performance evaluation and prediction: An information retrieval perspective. PhD thesis, Universidad Autónoma de Madrid

Bellogín A, de Vries AP (2013) Understanding similarity metrics in neighbour-based recommender systems. In: Kurland O, Metzler D, Lioma C, Larsen B, Ingwersen P (eds) International Conference on the Theory of Information Retrieval, ICTIR '13, Copenhagen, Denmark, September 29 - October 02, 2013, ACM, p 13, DOI 10.1145/2499178.2499186, URL http://doi.acm.org/10.1145/2499178.2499186

Bellogín A, Cantador I, Castells P (2010) A study of heterogeneity in recommendations for a social music service. In: Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems, ACM, New York, NY, USA, HetRec '10, pp 1–8, DOI 10.1145/1869446.1869447, URL http://doi.acm.org/10.1145/1869446.1869447

Bellogín A, Castells P, Cantador I (2011) Precision-oriented evaluation of recommender systems: an algorithmic comparison. In: Mobasher B, Burke RD, Jannach D, Adomavicius G (eds) Proceedings of the 2011 ACM Conference on Recommender Systems, RecSys 2011, Chicago, IL, USA, October 23-27, 2011, ACM, pp 333–336, DOI 10.1145/2043932.2043996, URL http://doi.acm.org/10.1145/2043932.2043996

Bellogín A, Cantador I, Díez F, Castells P, Chavarriaga E (2013) An empirical comparison of social, collaborative filtering, and hybrid recommenders. ACM TIST 4(1):14, DOI 10.1145/2414425.2414439, URL http://doi.acm.org/10.1145/2414425.2414439

Bellogín A, Said A, de Vries AP (2014) The magic barrier of recommender systems - no magic, just ratings. In: Dimitrova V, Kuflik T, Chin D, Ricci F, Dolog P, Houben G (eds) User Modeling, Adaptation, and Personalization - 22nd International Conference, UMAP 2014, Aalborg, Denmark, July 7-11, 2014. Proceedings, Springer, Lecture Notes in Computer Science, vol 8538, pp 25–36, DOI 10.1007/978-3-319-08786-3_3, URL http://dx.doi.org/10.1007/978-3-319-08786-3_3

Bennett J, Lanning S, Netflix N (2007) The netflix prize. In: In KDD Cup and Workshop in conjunction with KDD

Berkovsky S, Freyne J, Coombe M (2012) Physical activity motivating games: Be active and get your own reward. ACM Trans Comput-Hum Interact 19(4):32, DOI 10.1145/2395131.2395139, URL http://doi.acm.org/10.1145/2395131.2395139

Bistaffa F, Filippo A, Chalkiadakis G, Ramchurn SD (2015) Recommending fair payments for large-scale social ridesharing. In: Werthner et al (2015), pp 139–146, DOI 10.1145/2792838.2800177, URL http://doi.acm.org/10.1145/2792838.2800177

Bollen DGFM, Knijnenburg BP, Willemsen MC, Graus MP (2010) Understanding choice overload in recommender systems. In: Amatriain et al (2010), pp

63–70, DOI 10.1145/1864708.1864724, URL http://doi.acm.org/10.1145/1864708.1864724

Breese JS, Heckerman D, Kadie CM (1998) Empirical analysis of predictive algorithms for collaborative filtering. In: Cooper GF, Moral S (eds) UAI '98: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, University of Wisconsin Business School, Madison, Wisconsin, USA, July 24-26, 1998, Morgan Kaufmann, pp 43–52, URL https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=231&proceeding_id=14

Buckley C, Voorhees EM (2004) Retrieval evaluation with incomplete information. In: Sanderson et al (2004), pp 25–32, DOI 10.1145/1008992.1009000, URL http://doi.acm.org/10.1145/1008992.1009000

Burke RD (2004) Hybrid recommender systems with case-based components. In: Funk P, González-Calero PA (eds) Advances in Case-Based Reasoning, 7th European Conference, ECCBR 2004, Madrid, Spain, August 30 - September 2, 2004, Proceedings, Springer, Lecture Notes in Computer Science, vol 3155, pp 91–105, DOI 10.1007/978-3-540-28631-8_8, URL http://dx.doi.org/10.1007/978-3-540-28631-8_8

Campos PG, Díez F, Cantador I (2014) Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. User Model User-Adapt Interact 24(1-2):67–119, DOI 10.1007/s11257-012-9136-x, URL http://dx.doi.org/10.1007/s11257-012-9136-x

Castells P, Hurley NJ, Vargas S (2015) Novelty and diversity in recommender systems. In: Ricci et al (2015), pp 881–918, DOI 10.1007/978-1-4899-7637-6_26, URL http://dx.doi.org/10.1007/978-1-4899-7637-6_26

Celma Ò, Herrera P (2008) A new approach to evaluating novel recommendations. In: Pu P, Bridge DG, Mobasher B, Ricci F (eds) Proceedings of the 2008 ACM Conference on Recommender Systems, RecSys 2008, Lausanne, Switzerland, October 23-25, 2008, ACM, pp 179–186, DOI 10.1145/1454008.1454038, URL http://doi.acm.org/10.1145/1454008.1454038

Cremonesi P, Koren Y, Turrin R (2010) Performance of recommender algorithms on top-n recommendation tasks. In: Amatriain et al (2010), pp 39–46, DOI 10.1145/1864708.1864721, URL http://doi.acm.org/10.1145/1864708.1864721

Cremonesi P, Garzotto F, Negro S, Papadopoulos AV, Turrin R (2011) Comparative evaluation of recommender system quality. In: Tan DS, Amershi S, Begole B, Kellogg WA, Tungare M (eds) Proceedings of the International Conference on Human Factors in Computing Systems, CHI 2011, Extended Abstracts Volume, Vancouver, BC, Canada, May 7-12, 2011, ACM, pp 1927–1932, DOI 10.1145/1979742.1979896, URL http://doi.acm.org/10.1145/1979742.1979896

Deshpande M, Karypis G (2004) Item-based top-*N* recommendation algorithms. ACM Trans Inf Syst 22(1):143–177, DOI 10.1145/963770.963776, URL http://doi.acm.org/10.1145/963770.963776

Ekstrand MD, Ludwig M, Konstan JA, Riedl J (2011a) Rethinking the recommender research ecosystem: reproducibility, openness, and lenskit. In: RecSys, pp 133–140

Ekstrand MD, Riedl J, Konstan JA (2011b) Collaborative filtering recommender systems. Foundations and Trends in Human-Computer Interaction 4(2):175–243, DOI 10.1561/1100000009, URL http://dx.doi.org/10.1561/1100000009

Elahi M, Ge M, Ricci F, Massimo D, Berkovsky S (2014) Interactive food recommendation for groups. In: Chen L, Mahmud J (eds) Poster Proceedings of the 8th ACM Conference on Recommender Systems, RecSys 2014, Foster City, Silicon Valley, CA, USA, October 6-10, 2014, CEUR-WS.org, CEUR Workshop Proceedings, vol 1247, URL http://ceur-ws.org/Vol-1247/recsys14_poster2.pdf

Elahi M, Ge M, Ricci F, Fernández-Tobías I, Berkovsky S, Massimo D (2015) Interaction design in a mobile food recommender system. In: O'Donovan J, Felfernig A, Tintarev N, Brusilovsky P, Semeraro G, Lops P (eds) Proceedings of the Joint Workshop on Interfaces and Human Decision Making for Recommender Systems, IntRS 2015, co-located with ACM Conference on Recommender Systems (RecSys 2015), Vienna, Austria, September 19, 2015., CEUR-WS.org, CEUR Workshop Proceedings, vol 1438, pp 49–52, URL http://ceur-ws.org/Vol-1438/paper9.pdf

Elsweiler D, Harvey M, Ludwig B, Said A (2015) Bringing the "healthy" into food recommenders. In: Ge M, Ricci F (eds) Proceedings of the 2nd International Workshop on Decision Making and Recommender Systems, Bolzano, Italy, October 22-23, 2015., CEUR-WS.org, CEUR Workshop Proceedings, vol 1533, pp 33–36, URL http://ceur-ws.org/Vol-1533/paper8.pdf

Filippone M, Sanguinetti G (2010) Information theoretic novelty detection. Pattern Recognition 43(3):805–814, DOI 10.1016/j.patcog.2009.07.002, URL http://dx.doi.org/10.1016/j.patcog.2009.07.002

Gantner Z, Rendle S, Freudenthaler C, Schmidt-Thieme L (2011) Mymedialite: A free recommender system library. In: RecSys, DOI 10.1145/2043932.2043989, URL http://doi.acm.org/10.1145/2043932.2043989

Garcin F, Faltings B, Donatsch O, Alazzawi A, Bruttin C, Huber A (2014) Offline and online evaluation of news recommender systems at swissinfo.ch. In: Kobsa et al (2014), pp 169–176, DOI 10.1145/2645710.2645745, URL http://doi.acm.org/10.1145/2645710.2645745

Ge M, Delgado-Battenfeld C, Jannach D (2010) Beyond accuracy: evaluating recommender systems by coverage and serendipity. In: Amatriain et al (2010), pp 257–260, DOI 10.1145/1864708.1864761, URL http://doi.acm.org/10.1145/1864708.1864761

Goldberg KY, Roeder T, Gupta D, Perkins C (2001) Eigentaste: A constant time collaborative filtering algorithm. Inf Retr 4(2):133–151, DOI 10.1023/A:1011419012209, URL http://dx.doi.org/10.1023/A:1011419012209

Gunawardana A, Shani G (2015) Evaluating recommender systems. In: Ricci et al (2015), pp 265–308, DOI 10.1007/978-1-4899-7637-6_8, URL http://dx.doi.org/10.1007/978-1-4899-7637-6_8

Guy I (2015) Social recommender systems. In: Ricci et al (2015), pp 511–543, DOI 10.1007/978-1-4899-7637-6_15, URL http://dx.doi.org/10.1007/978-1-4899-7637-6_15

Herlocker JL, Konstan JA, Terveen LG, Riedl J (2004) Evaluating collaborative filtering recommender systems. ACM Trans Inf Syst 22(1):5–53, DOI 10.1145/963770.963772, URL http://doi.acm.org/10.1145/963770.963772

Jambor T, Wang J (2010) Optimizing multiple objectives in collaborative filtering. In: RecSys, ACM, New York, NY, USA, pp 55–62, DOI 10.1145/1864708.1864723, URL http://doi.acm.org/10.1145/1864708.1864723

Jannach D, Lerche L, Jugovac M (2015) Adaptation and evaluation of recommendations for short-term shopping goals. In: Werthner et al (2015), pp 211–218, DOI 10.1145/2792838.2800176, URL http://doi.acm.org/10.1145/2792838.2800176

Järvelin K, Kekäläinen J (2002) Cumulated gain-based evaluation of IR techniques. ACM Trans Inf Syst 20(4):422–446, DOI 10.1145/582415.582418, URL http://doi.acm.org/10.1145/582415.582418

Kobsa A, Zhou MX, Ester M, Koren Y (eds) (2014) Eighth ACM Conference on Recommender Systems, RecSys '14, Foster City, Silicon Valley, CA, USA - October 06 - 10, 2014, ACM, URL http://dl.acm.org/citation.cfm?id=2645710

Kohavi R, Longbotham R, Sommerfield D, Henne RM (2009) Controlled experiments on the web: survey and practical guide. Data Min Knowl Discov 18(1):140–181, DOI 10.1007/s10618-008-0114-1, URL http://dx.doi.org/10.1007/s10618-008-0114-1

Luo L, Li B, Berkovsky S, Koprinska I, Chen F (2016) Who will be affected by supermarket health programs? tracking customer behavior changes via preference modeling. In: Bailey J, Khan L, Washio T, Dobbie G, Huang JZ, Wang R (eds) Advances in Knowledge Discovery and Data Mining - 20th Pacific-Asia Conference, PAKDD 2016, Auckland, New Zealand, April 19-22, 2016, Proceedings, Part I, Springer, Lecture Notes in Computer Science, vol 9651, pp 527–539, DOI 10.1007/978-3-319-31753-3_42, URL http://dx.doi.org/10.1007/978-3-319-31753-3_42

Marlin BM (2003) Modeling user rating profiles for collaborative filtering. In: Thrun S, Saul LK, Schölkopf B (eds) Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems, NIPS 2003, December 8-13, 2003, Vancouver and Whistler, British Columbia, Canada], MIT Press, pp 627–634, URL http://papers.nips.cc/paper/2377-modeling-user-rating-profiles-for-collaborative-filtering

Massa P, Avesani P (2007) Trust-aware recommender systems. In: Konstan JA, Riedl J, Smyth B (eds) Proceedings of the 2007 ACM Conference on Recommender Systems, RecSys 2007, Minneapolis, MN, USA, October 19-20, 2007,

ACM, pp 17–24, DOI 10.1145/1297231.1297235, URL http://doi.acm.org/10.1145/1297231.1297235

McLaughlin MR, Herlocker JL (2004) A collaborative filtering algorithm and evaluation metric that accurately model the user experience. In: Sanderson et al (2004), pp 329–336, DOI 10.1145/1008992.1009050, URL http://doi.acm.org/10.1145/1008992.1009050

McNee SM, Riedl J, Konstan JA (2006) Being accurate is not enough: how accuracy metrics have hurt recommender systems. In: Olson GM, Jeffries R (eds) Extended Abstracts Proceedings of the 2006 Conference on Human Factors in Computing Systems, CHI 2006, Montréal, Québec, Canada, April 22-27, 2006, ACM, pp 1097–1101, DOI 10.1145/1125451.1125659, URL http://doi.acm.org/10.1145/1125451.1125659

Owen S, Anil R, Dunning T, Friedman E (2011) Mahout in Action. Manning Publications Co., Greenwich, CT, USA

Rennie JDM, Srebro N (2005) Fast maximum margin matrix factorization for collaborative prediction. In: Raedt LD, Wrobel S (eds) Machine Learning, Proceedings of the Twenty-Second International Conference (ICML 2005), Bonn, Germany, August 7-11, 2005, ACM, ACM International Conference Proceeding Series, vol 119, pp 713–719, DOI 10.1145/1102351.1102441, URL http://doi.acm.org/10.1145/1102351.1102441

Resnick P, Iacovou N, Suchak M, Bergstrom P, Riedl J (1994) Grouplens: An open architecture for collaborative filtering of netnews. In: Smith JB, Smith FD, Malone TW (eds) CSCW '94, Proceedings of the Conference on Computer Supported Cooperative Work, Chapel Hill, NC, USA, October 22-26, 1994, ACM, pp 175–186, DOI 10.1145/192844.192905, URL http://doi.acm.org/10.1145/192844.192905

Ribeiro MT, Lacerda A, Veloso A, Ziviani N (2012) Pareto-efficient hybridization for multi-objective recommender systems. In: Cunningham P, Hurley NJ, Guy I, Anand SS (eds) Sixth ACM Conference on Recommender Systems, RecSys '12, Dublin, Ireland, September 9-13, 2012, ACM, pp 19–26, DOI 10.1145/2365952.2365962, URL http://doi.acm.org/10.1145/2365952.2365962

Ricci F, Rokach L, Shapira B (eds) (2015) Recommender Systems Handbook. Springer, DOI 10.1007/978-1-4899-7637-6, URL http://dx.doi.org/10.1007/978-1-4899-7637-6

Said A (2013) Evaluating the accuracy and utility of recommender systems. PhD thesis, Technische Universität Berlin

Said A, Bellogín A (2014) Comparative recommender system evaluation: benchmarking recommendation frameworks. In: Kobsa et al (2014), pp 129–136, DOI 10.1145/2645710.2645746, URL http://doi.acm.org/10.1145/2645710.2645746

Said A, Jain BJ, Narr S, Plumbaum T (2012) Users and noise: The magic barrier of recommender systems. In: Masthoff J, Mobasher B, Desmarais MC, Nkambou R (eds) User Modeling, Adaptation, and Personalization - 20th International Conference, UMAP 2012, Montreal, Canada, July 16-20, 2012. Proceedings, Springer, Lecture Notes in Computer Science, vol 7379, pp 237–

248, DOI 10.1007/978-3-642-31454-4_20, URL http://dx.doi.org/10.1007/978-3-642-31454-4_20

Said A, Fields B, Jain BJ, Albayrak S (2013a) User-centric evaluation of a k-furthest neighbor collaborative filtering recommender algorithm. In: Bruckman A, Counts S, Lampe C, Terveen LG (eds) Computer Supported Cooperative Work, CSCW 2013, San Antonio, TX, USA, February 23-27, 2013, ACM, pp 1399–1408, DOI 10.1145/2441776.2441933, URL http://doi.acm.org/10.1145/2441776.2441933

Said A, Jain BJ, Albayrak S (2013b) A 3d approach to recommender system evaluation. In: Bruckman A, Counts S, Lampe C, Terveen LG (eds) Computer Supported Cooperative Work, CSCW 2013, San Antonio, TX, USA, February 23-27, 2013, Companion Volume, ACM, pp 263–266, DOI 10.1145/2441955.2442017, URL http://doi.acm.org/10.1145/2441955.2442017

Said A, Bellogín A, Lin JJ, de Vries AP (2014a) Do recommendations matter?: news recommendation in real life. In: Fussell SR, Lutters WG, Morris MR, Reddy M (eds) Computer Supported Cooperative Work, CSCW '14, Baltimore, MD, USA, February 15-19, 2014, Companion Volume, ACM, pp 237–240, DOI 10.1145/2556420.2556510, URL http://doi.acm.org/10.1145/2556420.2556510

Said A, Tikk D, Cremonesi P (2014b) Benchmarking - A methodology for ensuring the relative quality of recommendation systems in software engineering. In: Robillard MP, Maalej W, Walker RJ, Zimmermann T (eds) Recommendation Systems in Software Engineering, Springer, pp 275–300, DOI 10.1007/978-3-642-45135-5_11, URL http://dx.doi.org/10.1007/978-3-642-45135-5_11

Sanderson M, Järvelin K, Allan J, Bruza P (eds) (2004) SIGIR 2004: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield, UK, July 25-29, 2004, ACM

Smyth B, McClave P (2001) Similarity vs. diversity. In: Aha DW, Watson ID (eds) Case-Based Reasoning Research and Development, 4th International Conference on Case-Based Reasoning, ICCBR 2001, Vancouver, BC, Canada, July 30 - August 2, 2001, Proceedings, Springer, Lecture Notes in Computer Science, vol 2080, pp 347–361, DOI 10.1007/3-540-44593-5_25, URL http://dx.doi.org/10.1007/3-540-44593-5_25

de Souza Pereira Moreira G, de Souza GA, da Cunha AM (2015) Comparing offline and online recommender system evaluations on long-tail distributions. In: Castells P (ed) Poster Proceedings of the 9th ACM Conference on Recommender Systems, RecSys 2015, Vienna, Austria, September 16, 2015., CEUR-WS.org, CEUR Workshop Proceedings, vol 1441, URL http://ceur-ws.org/Vol-1441/recsys2015_poster4.pdf

Swearingen K, Sinha R (2001) Beyond algorithms: An hci perspective on recommender systems. In: ACM SIGIR. Workshop on Recommender Systems, vol Vol. 13, Numbers 5-6, pp 393–408

Tkalcic M, Quercia D, Graf S (2016) Preface to the special issue on personality in personalized systems. User Model User-Adapt Interact 26(2-3):103–107,

DOI 10.1007/s11257-016-9175-9, URL http://dx.doi.org/10.1007/s11257-016-9175-9

Tomlinson S (2012) Measuring robustness with first relevant score in the TREC 2012 microblog track. In: Voorhees EM, Buckland LP (eds) Proceedings of The Twenty-First Text REtrieval Conference, TREC 2012, Gaithersburg, Maryland, USA, November 6-9, 2012, National Institute of Standards and Technology (NIST), vol Special Publication 500-298, URL http://trec.nist.gov/pubs/trec21/papers/OpenText.microblog.final.pdf

Vargas S (2015) Novelty and diversity evaluation and enhancement in recommender systems. PhD thesis, Universidad Autónoma de Madrid

Vargas S, Castells P (2013) Exploiting the diversity of user preferences for recommendation. In: Ferreira J, Magalhães J, Calado P (eds) Open research Areas in Information Retrieval, OAIR '13, Lisbon, Portugal, May 15-17, 2013, ACM, pp 129–136, URL http://dl.acm.org/citation.cfm?id=2491776

Vargas S, Castells P (2014) Improving sales diversity by recommending users to items. In: Kobsa et al (2014), pp 145–152, DOI 10.1145/2645710.2645744, URL http://doi.acm.org/10.1145/2645710.2645744

Vargas S, Baltrunas L, Karatzoglou A, Castells P (2014) Coverage, redundancy and size-awareness in genre diversity for recommender systems. In: Kobsa et al (2014), pp 209–216, DOI 10.1145/2645710.2645743, URL http://doi.acm.org/10.1145/2645710.2645743

Werthner H, Zanker M, Golbeck J, Semeraro G (eds) (2015) Proceedings of the 9th ACM Conference on Recommender Systems, RecSys 2015, Vienna, Austria, September 16-20, 2015, ACM, URL http://dl.acm.org/citation.cfm?id=2792838

Yao YY (1995) Measuring retrieval effectiveness based on user preference of documents. JASIS 46(2):133–145, DOI 10.1002/(SICI)1097-4571(199503)46:2⟨133::AID-ASI6⟩3.0.CO;2-Z, URL http://dx.doi.org/10.1002/(SICI)1097-4571(199503)46:2<133::AID-ASI6>3.0.CO;2-Z

Zhao X, Niu Z, Chen W (2013) Opinion-based collaborative filtering to solve popularity bias in recommender systems. In: Decker H, Lhotská L, Link S, Basl J, Tjoa AM (eds) Database and Expert Systems Applications - 24th International Conference, DEXA 2013, Prague, Czech Republic, August 26-29, 2013. Proceedings, Part II, Springer, Lecture Notes in Computer Science, vol 8056, pp 426–433, DOI 10.1007/978-3-642-40173-2_35, URL http://dx.doi.org/10.1007/978-3-642-40173-2_35

Zhao X, Zhang W, Wang J (2015) Risk-hedged venture capital investment recommendation. In: Werthner et al (2015), pp 75–82, DOI 10.1145/2792838.2800181, URL http://doi.acm.org/10.1145/2792838.2800181

Zhou T, Kuscsik Z, Liu JG, Medo M, Wakeling JR, Zhang YC (2010) Solving the apparent diversity-accuracy dilemma of recommender systems. Proceedings of the National Academy of Sciences 107(10):4511–4515, DOI 10.1073/pnas.1000488107, URL http://www.pnas.org/content/107/10/4511.abstract, http://www.pnas.org/content/107/10/4511.full.pdf

Ziegler C, Lausen G (2009) Making product recommendations more diverse. IEEE Data Eng Bull 32(4):23–32, URL http://sites.computer.org/debull/A09dec/ziegler-paper1.pdf

Ziegler C, McNee SM, Konstan JA, Lausen G (2005) Improving recommendation lists through topic diversification. In: Ellis A, Hagino T (eds) Proceedings of the 14th international conference on World Wide Web, WWW 2005, Chiba, Japan, May 10-14, 2005, ACM, pp 22–32, DOI 10.1145/1060745.1060754, URL http://doi.acm.org/10.1145/1060745.1060754