Classification of Normative Recommender Systems

Lucien Heitz[1,2]

[1]Department of Informatics, University of Zurich, Zurich, Switzerland
[2]Digital Society Initiative, University of Zurich, Zurich, Switzerland

Abstract

Recommender systems are a primary source for providing user-facing information in a variety of mediums and domains, ranging from movies and news to job advertisements. The potential issues and associated ethical implications have attracted contributions from an interdisciplinary community for studying the normative dimension of recommender systems. However, there has yet to be a shared understanding of the concepts at play and how to operationalize norms and values. We look at normativity from a technical point of view and identify 1.) the pre-processing stage, 2.) the in-processing stage, 3.) the post-processing stage, and 4.) the evaluation stage of a recommender system as the four key areas where normative aspects can be accounted for. Accordingly, four classes of how to implement norms and values in recommender systems are proposed. We proceed with a class-specific comparison of their respective advantages and disadvantages and highlight how such a classification allows us to reason and distinguish between the normative capabilities of recommender systems.

Keywords

operationalization of normative goals, conceptual classification, algorithm design

1. Introduction

Recommender systems (RS) that feature a normative dimension attract a growing interdisciplinary community, ranging from computational linguists [1], legal and political science scholars [2, 3], to computer scientists [4, 5, 6, 7]. This leads to a rich understanding of the normative dimension of RS, which covers a variety of aspects. When speaking of norm-aware systems or normative dimension of RS, we refer to a recommender system that incorporates democratic principles (e.g., social cohesion and autonomy of citizens, cf. [3]) and journalistic values (e.g., transparency and diversity of opinions, cf. [8]). Normative systems follow an optimization goal for recommendations that is shaped by RS-*external* values, as opposed to being optimized to achieve a target score for a "simple" mathematical expression or metric [9], such as accuracy, recall, or click-through rate. RS that make use of such normative values can be located in the domain of beyond-accuracy objectives (BAO). In the RS literature, BAO are operationalized as fairness [10, 11], diversity [12, 13, 14], coverage [15, 12], novelty [5, 12], serendipity [16, 12], or surprise [17], to name but a few of the most prominent examples. In the context of this work, we speak of a norm-aware RS as being a subset of systems that follow one or multiple BAO.

NORMalize 2023: The First Workshop on the Normative Design and Evaluation of Recommender Systems, September 19, 2023, co-located with the ACM Conference on Recommender Systems 2023 (RecSys 2023), Singapore
heitz@ifi.uzh.ch (L. Heitz)
https://github.com/Informfully (L. Heitz)
0000-0001-7987-8446 (L. Heitz)

As the research community in the domain of normative RS is inherently interdisciplinary, there is a plethora of different terms and concepts used when talking about problems and solutions in this area of research. However, recent findings suggest that certain concepts in this domain have almost no overlap between the disciplines. E.g., there is no shared notion of the concept of diversity as an optimization goal in RS research across the interdisciplinary community [18]. Furthermore, there is a gap between descriptive notions (i.e., investigating how current systems that label themselves as normative RS perform) and normative notions (i.e., looking at the tasks that normative systems ought to perform) [7]. We feel this mismatch limits the exchange of ideas and solutions across disciplines.

To tackle this limitation, we propose a classification of norm-aware RS. The classes introduced are anchored in how normative elements are implemented in RS on a technical level. Looking at the RS pipeline, we identify four stages where normative values can be embedded into the system: 1.) at the pre-processing stage (through normative stratification of the dataset), 2.) at the in-processing stage (normativity as optimization goal of the model), 3.) at the post-processing stage (norm-focused re-ranking of candidate items), and finally 4.) the evaluation stage (assessment of normative dimensions of RS through metrics). The advantage of adhering to such an approach is that it allows for an unambiguous way of classifying RS, one that is verifiable through code inspection. It makes explicit the precise way how normative values are accounted for within the RS pipeline. In essence, assigning a class to a system serves as a *label* to quickly communicate the normative capabilities of a RS, how they are implemented, and what types of class comparisons among systems are possible.

We pursue two main goals with the introduction of this classification. The first goal is to contribute towards building a shared vocabulary within this interdisciplinary field of research. By introducing a high-level classification of RS, we aim at creating a common understanding of the different ways of how to operationalize a given normative value (i.e., operationalization of normativity using datasets, models, re-ranking, or evaluation metrics). By using class membership as a label for an RS, researchers are provided an easy and effective means to inform their peers of how normativity was operationalized on a technical level. This is especially valuable in a field where in-depth knowledge of software development and programming is not a given. No inspection of source code needed.

Second, the distinction between different normative classes allows for a more precise comparison and benchmarking of RS. The inclusion of, e.g., a diversity-optimized target function, can have different outcomes, depending on the stage of the pipeline it is applied to. Applying a diversity target function to the model of a RS will not have the same result as using it for re-ranking of candidate items. The labeling system introduced by the classification, therefore, raises the awareness on the stage-dependent operationalization of normative values for a sound comparison of RS. To this end, we included a list of advantaged and disadvantages of embedding normativity at each of the four stages, together with remarks for class comparisons.

We next give an overview of the structure of the paper. First, Section 2 discusses related work in the domain of normative RS from across scientific disciplines. We identify opportunities and shortcomings. Second, Section 3 presents our main contributions of classifying normative RS, together with the comparison of their respective advantages and disadvantages. We continue with a discussion the benefits and limitations of our classification in Section 4. We end this paper with our concluding remarks in Section 5.

2. Related Work

In the context of social media, personalized online news systems, or online news RS, the discussion of BAO and inclusion of social norms as well as editorial values has become increasingly popular [19, 20, 8]. This is mainly due to the capacity of these RS to impact communities and society, as they promote and provide exposure, e.g., regarding political issues, with the potential to influence people's beliefs and behavior [21]. To this end, Helberger et al. [8] ask developers of RS in the context of digital journalism to be considerate of the real-world impact of the system that they are developing. The goal of doing so is to a.) highlight the societal and ethical dimensions that RS designers should be mindful of [22] and to b.) contribute towards the normative turn in computer science [23]. Unfortunately, proper evaluation, performance benchmarking, and especially understanding of the impact of normative objectives in terms of models and metrics on users are still limited and need closer investigation [24, 25].

BAOs for RS with a normative dimension have a long tradition in RS research [4, 12]. When looking at target function for, e.g., coverage and diversity, there are multiply ways of how to include them within a given RS; they can be feature as part of a re-ranking process of candidate items [26, 15, 5], serve as an evaluation metric for the RS [7]. In addition to that, more recent work highlights the importance of investing into the dataset quality [27].

Looking at the subset of BAO that are normative objectives, e.g., diversity in the domain of news, they can be explicitly designed to "stimulate" certain news items [2] to promote democratic values by exposing the reader to minority voices [3]. This approach is akin to treating normativity as a *desired bias*[1] that we want to introduce or enhance within a system. Investigating such bias mitigation strategies is an important part of machine learning (ML) and artificial intelligence (AI) applications [28]. In the normative BAO domain that is fairness, the literature identifies three key steps where biases can be mitigated: 1.) during the pre-processing state, 2.) during the in-processing stage, and 3.) in the post-processing stage [29].

The introduction of these processing stages for norm-aware systems is not a novelty. Previous works has already extensively discussed in detail the embedding of normative values, such as fairness, in the pre-processing stage [30], the in-processing stage [31], as well as the post-processing stage [32] for algorithms. Rather than focusing on an individual step or normative goal, the aim of this paper is to introduce a more general, light-weight introduction to this stage-based classification. An introduction that is primarily targeting an interdisciplinary audience. And while previous works focus on large domains, such as ML systems [33], the scope of this paper is limited to providing an overview for the normative dimensions within RS research. The advantage of doing so it that this allows to sharpen the focus on the contents of some stages (e.g., focus on re-ranking for the post-processing stage, following [5]) or extending the stages with an evaluation step to account for the domain-specific importance of evaluation metrics (e.g., [7]) to better capture the intricacies of normativity in RS. This all serves the goal of featuring a class-based labeling system to quickly identify normative RS that can be shared and applied across disciplines.

[1]In this context, a *desired bias* is what we outlined in Section 1 to be a *external* value. It is important to note that the classification presented here is value agnostic. I.e., it does not presuppose and normative goal, nor does it provide and guiding principle for finding such a value.

3. Classification of Normative Recommender Systems

In this section, we present our classification for normative RS. For the purpose of building this classification, we adopt the notion of promoting norm-aware optimization goals within a RS pipeline as introducing *desired biases*. We outline the four stages where this can take place within RS pipelines. We then proceed to formalize the recommendation procedure as preparation for the subsequent classification. Finally, we will remark on the advantages and disadvantages of each identified RQ class as well as the performance comparison across classes.

3.1. Stage Overview and Classification

For the task of bias mitigation–and in return with promoting normative values–the following four stages of the RQ pipeline need to be considered:

Pre-processing stage: Mitigation strategies that process the dataset before it is given as input to the RS, applying a transformation to the input data to the model (e.g., stratified sampling to achieve a target distribution).

In-processing stage: This stage includes any operations done on the input data by the model to optimize for the target function. In the domain of RS, this is the process of generating the recommendation lists.

Post-processing stage: These strategies manipulate the output of the model to optimize for a target objective. This process is akin to introducing normativity to a RS pipeline by re-ranking candidate items (cf., [5]).

Evaluation stage: At this stage, the ranking of items is no longer modified. Metrics applied here express certain characteristics of the RS used to generate the recommendations.

These four stages act as a guiding principle for our classification of normative RS. In order to present this classification, we first need to define the following parts of the RS pipeline:

- U = set of all users, UF = set of all user features,
- I = set of all items, IF = set of all item features,
- R = set of all ratings of U for I,
- $\mathbb{N}$ = set of normative target functions (e.g., coverage, diversity, or fairness),
- $\mathbb{T}$ = set of re-ranking target functions, where $\mathbb{N}$ and $\mathbb{T}$ are overlapping,
- $\mathbb{M}$ = set of evaluation metrics, where $\mathbb{N}$ and $\mathbb{M}$ are overlapping.[2]

A normative function $f_{normative}$ can take as input any of the available data points on users UF and items IF to create a ranked item list (recommendation list). We formalize this as follows:

$$f_{normative}(I, U, IF, UF, R, \mathbb{N}) \rightarrow I_{ranked}, \tag{1}$$

[2]Any algorithm used as an evaluation metric $m \in \mathbb{M}$ could be modified in such a way that it serves as a target function $n \in \mathbb{N}$ for a model. The same holds true for pre-processing steps of the stratification procedure; any modification done to the initial dataset can be applied during subsequent steps.

where the values of R are unknown. I_{ranked} can be evaluated against a metric m from $\mathbb{M}$. (As m does not influence I_{ranked}, $\mathbb{M}$ is left out of Equation 1). In this setup, re-ranking on model outputs is allowed *any number* of times. An initial function optimizing for a given relevance criterion $f_{rel}(i)$ (which is not required to be of any normative significance) generates a list of candidate items $I_{candidate}$ (cf. [5]). In a second step, $I_{candidate}$ is re-ranked to satisfy a given optimization objective $f \in \mathbb{N}$ with the available items, e.g, $i* \leftarrow argmax f(i_u), i \in I \setminus I_u$, resulting in $I_{reranked}$. In general, the normative element of a RS is represented by such a target function f. With this formalization in mind, we now present our classification of normative RS:

Class 0 - Normativity at the pre-processing stage: Class 0 approaches take the form of a target function modifying the input dataset of a RS (e.g., stratified sampling of input data). This data processing is done outside of the RS. Nevertheless, if the filtering procedure applied is done by an algorithm sharing a target function $f \in \mathbb{N}$ or metric $m \in \mathbb{N}$.[3]

Class 1 - Normative models at the in-processing stage: Class 1 RS feature models for generating item recommendations that are optimized for normative targets of RS:[4]

- **Class 1.1**: $f \in \mathbb{N}, \mathbb{T} = \emptyset$, norm-aware throughout the entire pipeline.
- **Class 1.2**: $f \in \mathbb{N}, (\forall f_{rerank} \in \mathbb{T}), (\forall f_{rerank} \in \mathbb{N})$, RS that makes exclusive use of norm-aware target functions for the purpose of re-ranking candidate items; norm-aware throughout the entire pipeline.
- **Class 1.3**: $f \in \mathbb{N}, (\exists f_{rerank1} \exists f_{rerank2} \in \mathbb{T}), (f_{rerank1} \in \mathbb{N}, f_{rerank2} \notin \mathbb{N})$, RS featuring at least one normative and one non-normative target function during the process of re-ranking candidate items.

Class 2 - Normative item re-ranking at the post-processing stage: Class 2 RS feature a target function for norm-aware re-ranking, where the initial set of candidate items is generated by a non-normative model:

- **Class 2.1**: $f \notin \mathbb{N}, (\forall f_{rerank} \in \mathbb{T}), (\forall f_{rerank} \in \mathbb{N})$, RS that makes exclusive use of norm-aware target functions for the purpose of re-ranking candidate items.
- **Class 2.2**: $f \notin \mathbb{N}, (\exists f_{rerank1} \exists f_{rerank2} \in \mathbb{T}), (f_{rerank1} \in \mathbb{N}, f_{rerank2} \notin \mathbb{N})$, RS featuring at least one normative and one non-normative target function during the process of re-ranking candidate items.

Class 3 - Normativity as metric at the evaluation stage: Class 3 RS include a target function as metric for the sole purpose of assessing the normative degree of the recommendation output, with $f \notin \mathbb{N}, (\forall f_{rerank} \in \mathbb{T}), (f_{rerank} \notin \mathbb{N}), m \in \mathbb{N}, m \in \mathbb{M}$. No sub-classes exist, no normative aspects are considered during the recommendation procedure.

Looking at the classification of norm-aware RS, it is important to reiterate that it does not provide, nor does it intend to provide any assessment of the adequacy or quality of any dataset, model, or metric. It simply allows for assessing the stages at which a RS makes use of norm-aware elements. Its main goal is to provide the research community with a structured way of comparing and assessing RS; the optimization objectives are assumed to be a given.

[3]Class 0 make exclusive normative elements during data pre-processing. If a prospective Class 0 RS includes any normative model (Class 1), re-ranking procedure (Class 2), or metric (Class 3), it instead takes on this class.

[4]Inclusion of any normative target metric for evaluating the RS output is optional and not relevant for Class 1.

3.2. Comparison of Advantages and Disadvantages

Having introduced Classes 0, 1, 2, and 3 for normative RS, the next step is the comparison of their advantaged and disadvantages. Table 1 shows the benefits and drawback for operationalizing each class. This is not only intended for analyzing existing solutions, but the table also allows for assessing the viability, i.e., when it comes to operationalizing a given normative value, this overview can help selecting the class most suitable for the given use case.

The advantages and disadvantages of the normative RS classes are systematically analyzed along three dimensions: normative power, ease of implementation, and structural limitations. "Normative Power" describes to what degree it is possible to have this class create normative recommendations. In this dimension "High" means that the class has can have the greatest impact on user recommendation lists, "Low" indicates smallest impact among classes, and "None" identifies classes that do not change the recommendations. Normative power is an inherent limitation of a class. Classes with a higher normative power are more advantageous. "Ease of Implementation" helps assessing the amount of work required to implement a given RS. "Difficult" requires the most time, "Easy" the least amount of time, and "Medium" is in between the two. This ease of implementation is not an inherent limitation of classes. Instead, it is something that can be compensated with having additional resources. The easier the implementation, the more advantageous it is to use a given class. The last dimension is "Structural Limitations," addressing inherent properties of the class that, again, cannot be changed. This dimensions inform about the a pre-requisite for when selecting potential solutions with existing limitations in mind.

Table 1
Overview of the advantages and disadvantages across the different classes along the dimensions of "Normative Power," "Ease of Implementation," and "Structural Limitations."

Class	Normative Power	Ease of Implementation	Structural Limitations
Class 0	High	Medium to difficult	RS-external data gathering
Class 1	High	Difficult	Full access to RS
Class 2	Low	Medium	Require item pool
Class 3	None	Easy	Evaluation only

The next part presents a detailed overview of the data summarized in Table 1. More explanations are provided on the operationalization of a normative value with a given class. Furthermore, each class is listed together with a note on their compatibility when it comes to comparing performance with other classes.

Class 0: The advantage of tackling normativity via Class 0 is that this can have a significant impact throughout the RS and impact the recommendation list. By enrichment and stratification, Class 0 approaches can increase data quality in the normative dimension. The disadvantage is that–depending on the domain–the gathering of additional data can require comparatively more work than with other classes. Class 0 implementations are possible without touching any of the subsequent RS parts. Any Class 0 system, however, is ultimately limited by the available data on items, users, and features, the gathering of which is external to the RS and possible outside the control of the system designer.

Compatibility note: Class 0 stratification approaches are ideally compared with another Class 0 RS. Comparison with Class 1 and Class 2 RS are possible. Class 0 approaches cannot be compared with Class 3 approaches.

Class 1: The advantage of having norm-aware target function implemented as a model within a RQ is that it offers one of the greatest levels of freedom in terms of serving norm-aware recommendations to users. The main disadvantage is, however, that a Class 1 system can require significantly more work to implement compared with the other classes. From a limitation point of view, Class 1 does require full access to the RS pipeline.

Compatibility note: Class 1 systems are ideally compared in terms of their performance with other Class 1 systems and with Class 2 systems. Comparisons with Class 0 RS are possible. Class 1 approaches cannot be compared with Class 3 approaches.

Class 2: The main advantage of a re-ranking approaches to normativity is that it offers a light-weight implementation for introducing norm-aware principles (compared to Classes 0 and 1). Re-ranking allows for a fine-tuned adjustment of existing recommendations lists. The main disadvantage of re-ranking is that the pool of items is limited through the dataset and the underlying model. It therefore has not the highest normative power. Looking at the structural limitations, a sufficiently large pool of candidate items is required.

Compatibility note: Class 2 systems can be compared in terms of their performance with other Class 2 systems and with Class 1 systems. Comparisons with Class 0 RS are possible. Class 2 approaches cannot be compared with Class 3 approaches.

Class 3: Class 3 RS have the disadvantage that they are the least norm-aware RS from among all classes. Following the presented classification, any Class 3 system uses normative elements to solely assess the output. Using normativity as metric in this way comes with the limitation that any norm-aware Class 3 system is unable to influence the recommendations. The selection of items happens before normative values are considered. The advantage of these solutions, however, is that normativity expressed as metrics requires the least amount of work to implement. The structural limitation, again, is that it only supports the assessment and evaluation of an RS for comparison purposes.

Compatibility note: Class 3 approaches cannot be compared or benchmarked against other classes. The limitation that applies here is that when comparing Class 3 systems, one and the same optimization goal must be selected. E.g., when measuring the diversity of a recommendation list, it must be compared against the *same* diversity measurement applied to another RS.

3.3. Applying the Classification

Up to this point, the discussion of the RS classes has been on a general and theoretical level. The goal of the following part is to complement this discussion with examples on how to apply the classification to existing systems. For the purpose of providing an example of the application of the classification, we pick one specific use case within the normative RS domain. The chosen example of norm-aware RS is diversity optimization for news recommendations.

Stating again the initial goal of the classification, it is a means to help labeling different norm-aware approaches. It is to effectively and precisely communicate how a normative value was embedded within the RS and to facilitate meaningful comparison and benchmarking across different RS. To do so and in order to properly apply the RS classification outline here, the default assumption when approaching a RS is that it does not feature any normative dimensions. Step by step, the four main components of the RS are then analyzed: the dataset, the model, the re-ranking approaches, and the evaluation metrics (in that exact order). Based on their inclusion of normative principles, a class label is assigned.

Class 0: Starting with the data, there are multiple ways in which the dataset can embed normative values. In the chosen example that is diversity of news recommendations, the dataset can satisfy the normative value by featuring, e.g., a diverse selection of topics [34], or it is a dataset that has been pre-processed/stratified [30, 35] to ensure the data meets certain diversity requirements. Assuming that this is the only step where normativity is introduced, such an RS would be labeled a Class 0 system.

Class 1: The next part to investigate is the model of the RS. Looking at existing systems or at proposed solutions in the literature, a Class 1 RS is one that embeds normativity as part of the core recommendation procedure. For diversity in news, this can be achieved by tweaking existing solutions to optimize for a diversity goal functions (e.g., optimizing for topic or viewpoint diversity by adapting existing solutions like [36, 37, 38]). Regardless of the inclusion of a data pre-processing/stratification step, if a system features such a normative model, the classification calls it a Class 1 model.[5]

Class 2: To be a Class 2 RS means that no pre-processing/stratification step was introduced, and that no norm-aware model is in place. The literature on diversity features offers a multitude of approaches that can be applied to the domain of news (see [39, 26, 5]). What these approaches all have in common is that they take as input a list of candidate items generated by an underlying model and try to embed normative values through re-ranking of the item list. When doing so, such a RS would be labeled a Class 2 RS.[6]

Class 3: The goal of Class 3 systems is not to primarily provide normative recommendations. Instead, their aim is the assessment of, e.g., diversity, within an existing RS. Given the popularity of these metrics (see [40, 12, 41, 7]) a dedicated evaluation state was introduced. The main property of Class 3 RS is that they do not feature any norm-aware elements in previous steps. Any norm-aware metric implemented in an otherwise non-normative RS makes it a Class 3 RS. Given that metrics do not impact the recommendations, the presence of non-normative metrics used to assess the RS output does not change the label assigned by this classification.

[5]A more fine-grained assessment is possible. I.e., if there is no re-ranking step, the RS received a Class 1.1 label. If all re-ranking steps follow normative principles, it is a Class 1.2 RS. It is a Class 1.3 RS if at least one re-ranking step features normative values among other non-normative re-ranking steps.

[6]Similar to Class 1, the Class 2 RS can be further differentiated. As re-ranking steps can be done any number of times, a system that features exclusively norm-aware re-ranking is called a Class 2.1 system. If there norm-aware re-ranking is complemented by non-normative re-ranking, then it is a Class 2.2 RS.

4. Discussion and Limitations

The classification presented in Section 3 rests on the assumption that any target function f or metric m can be identified to be a member of set $\mathbb{N}$, i.e., the set of *norm-aware and/or norm-relevant* models and metrics. However, what precisely means to be of normative relevance has not been defined. Models and metrics for multi-objective optimization, for example, make this classification even more difficult. This discussion of the normative nature is something we propose to have on a case-by-case basis. A general discussion is difficult due to the fact that 1.) each norm-aware value must be carefully designed to consider the respective user needs and topics [3, 24], and 2.) it remains a *normative* definition, meaning that it is influenced by the norms and convictions of its authors [9].

As such, there is no *one* understanding of the content of normativity. This is made even more evident by previous studies highlighting cultural differences in the perception of recommendations [42, 41] and user-dependent differences and effects (e.g. making sure the user interface is adaptable to personal preferences and needs [43]), identifying yet further dimensions to control for when adapting normative elements in RS. Another limitation is that the current typification does not include any visualization aspects of the recommended items. Earlier works showed importance of controlling for the visualization of the results for properly assessing their impact on users [44, 13, 45].

5. Conclusion

In this paper, we presented a classification of normative approaches to RS research. We identified data stratification, target functions for models, re-ranking, and metrics as key factors for introducing norm-aware dimensions to the RS pipeline. Using these elements, we proposed four different classes for assessing the normative capacity of a RS. This is done by looking at the extent to which the pre-processing, in-processing, post-processing, and evaluation phase of a recommender system pipeline account for societal values, guiding its curation procedure. By presenting this classification, we hope to help in aligning the different notions of normativity and its operationalization within the interdisciplinary research community of RS.

Acknowledgments

This work was funded by the Digital Society Initiative (DSI) of the University of Zurich under a grant of the DSI Excellence Program.

References

[1] M. Reuver, A. Fokkens, S. Verberne, H. Toivonen, M. Boggia, No nlp task should be an island: multi-disciplinarity for diversity in news recommender systems, Proceedings of the EACL Hackashop on news media content analysis and automated report generation (2021) 45–55.

[2] N. Helberger, K. Karppinen, L. D'acunto, Exposure diversity as a design principle for recommender systems, Information, Communication & Society 21 (2018) 191–207.
[3] N. Helberger, On the democratic role of news recommenders, Digital Journalism 7 (2019) 993–1012.
[4] M. Karimi, D. Jannach, M. Jugovac, News recommender systems–survey and roads ahead, Information Processing & Management 54 (2018) 1203–1227.
[5] P. Castells, N. Hurley, S. Vargas, Novelty and diversity in recommender systems, in: Recommender systems handbook, Springer, 2021, pp. 603–646.
[6] L. Heitz, J. A. Lischka, A. Birrer, B. Paudel, S. Tolmeijer, L. Laugwitz, A. Bernstein, Benefits of diverse news recommendations for democracy: A user study, Digital Journalism 10 (2022) 1710–1730.
[7] S. Vrijenhoek, G. Bénédict, M. Gutierrez Granada, D. Odijk, M. De Rijke, Radio–rank-aware divergence metrics to measure normative diversity in news recommendations, in: Proceedings of the 16th ACM Conference on Recommender Systems, 2022, pp. 208–219.
[8] N. Helberger, M. van Drunen, J. Moeller, S. Vrijenhoek, S. Eskens, Towards a normative perspective on journalistic ai: Embracing the messy reality of normative ideals, 2022.
[9] J. Grosman, T. Reigeluth, Perspectives on algorithmic normativities: engineers, objects, activities, Big Data & Society 6 (2019) 2053951719858742.
[10] Y. Deldjoo, D. Jannach, A. Bellogin, A. Difonzo, D. Zanzonelli, Fairness in recommender systems: research landscape and future directions, User Modeling and User-Adapted Interaction (2023) 1–50.
[11] Y. Wang, W. Ma, M. Zhang, Y. Liu, S. Ma, A survey on the fairness of recommender systems, ACM Transactions on Information Systems 41 (2023) 1–43.
[12] M. Kaminskas, D. Bridge, Diversity, serendipity, novelty, and coverage: a survey and empirical analysis of beyond-accuracy objectives in recommender systems, ACM Transactions on Interactive Intelligent Systems (TiiS) 7 (2016) 1–42.
[13] M. Mulder, O. Inel, J. Oosterman, N. Tintarev, Operationalizing framing to support multiperspective recommendations of opinion pieces, in: Proceedings of the 2021 ACM conference on fairness, accountability, and transparency, 2021, pp. 478–488.
[14] R. Hada, A. Ebrahimi Fard, S. Shugars, F. Bianchi, P. Rossini, D. Hovy, R. Tromble, N. Tintarev, Beyond digital" echo chambers": The role of viewpoint diversity in political discussion, in: Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, 2023, pp. 33–41.
[15] S. Vargas, L. Baltrunas, A. Karatzoglou, P. Castells, Coverage, redundancy and size-awareness in genre diversity for recommender systems, in: Proceedings of the 8th ACM Conference on Recommender systems, 2014, pp. 209–216.
[16] M. Ge, C. Delgado-Battenfeld, D. Jannach, Beyond accuracy: evaluating recommender systems by coverage and serendipity, in: Proceedings of the fourth ACM conference on Recommender systems, 2010, pp. 257–260.
[17] M. Kaminskas, D. Bridge, Measuring surprise in recommender systems, in: Proceedings of the workshop on recommender systems evaluation: dimensions and design (Workshop programme of the 8th ACM conference on recommender systems), Citeseer, 2014.
[18] F. Loecherbach, J. Moeller, D. Trilling, W. van Atteveldt, The unified framework of media diversity: A systematic literature review, Digital Journalism 8 (2020) 605–642.

[19] L. Udris, M. Rivière, D. Vogler, M. Eisenegger, Reuters institute digital news report 2022: Länderbericht schweiz (2022).
[20] L. A. Møller, Recommended for you: how newspapers normalise algorithmic news recommendation to fit their gatekeeping role, Journalism Studies 23 (2022) 800–817.
[21] R. K. Garrett, N. J. Stroud, Partisan paths to exposure diversity, Journal of Communication 64 (2014) 680–701. doi:10.1111/jcom.12105.
[22] A. Bernstein, C. De Vreese, N. Helberger, W. Schulz, K. Zweig, L. Heitz, S. Tolmeijer, et al., Diversity in news recommendation, Dagstuhl Manifestos 9 (2021) 43–61.
[23] J. Harambam, D. Bountouridis, M. Makhortykh, J. van Hoboken, Designing for the better by taking users into account: A qualitative evaluation of user control mechanisms in (news) recommender systems, in: Proceedings of the 13th ACM Conference on Recommender Systems, RecSys '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 69–77. URL: https://doi.org/10.1145/3298689.3347014. doi:10.1145/3298689.3347014.
[24] C. Treuillier, S. Castagnos, E. Dufraisse, A. Brun, Being diverse is not enough: Rethinking diversity evaluation to meet challenges of news recommender systems, in: Adjunct Proceedings of the 30th ACM Conference on User Modeling, Adaptation and Personalization, 2022, pp. 222–233.
[25] H. Sargeant, E. Pirkova, M. C. Kettemann, M. Wisniak, M. Scheinin, E. Bevensee, K. Pentney, L. Woods, L. Heitz, B. Kostic, et al., Spotlight on artificial intelligence and freedom of expression: A policy manual, Organization for Security and Co-operation in Europe (2022).
[26] S. Vargas, New approaches to diversity and novelty in recommender systems, in: Fourth BCS-IRSG Symposium on Future Directions in Information Access (FDIA 2011) 4, 2011, pp. 8–13.
[27] N. Sambasivan, S. Kapania, H. Highfill, D. Akrong, P. Paritosh, L. M. Aroyo, "everyone wants to do the model work, not the data work": Data cascades in high-stakes ai, in: proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, 2021, pp. 1–15.
[28] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, A. Galstyan, A survey on bias and fairness in machine learning, ACM Computing Surveys (CSUR) 54 (2021) 1–35.
[29] A. Singh, J. Singh, A. Khan, A. Gupta, Developing a novel fair-loan classifier through a multi-sensitive debiasing pipeline: Dualfair, Machine Learning and Knowledge Extraction 4 (2022) 240–253.
[30] F. Kamiran, T. Calders, Data preprocessing techniques for classification without discrimination, Knowledge and information systems 33 (2012) 1–33.
[31] M. Wan, D. Zha, N. Liu, N. Zou, In-processing modeling techniques for machine learning fairness: A survey, ACM Transactions on Knowledge Discovery from Data 17 (2023) 1–27.
[32] F. Petersen, D. Mukherjee, Y. Sun, M. Yurochkin, Post-processing for individual fairness, Advances in Neural Information Processing Systems 34 (2021) 25944–25955.
[33] S. Caton, C. Haas, Fairness in machine learning: A survey, ACM Computing Surveys (2020).
[34] F. Wu, Y. Qiao, J.-H. Chen, C. Wu, T. Qi, J. Lian, D. Liu, X. Xie, J. Gao, W. Wu, et al., Mind: A large-scale dataset for news recommendation, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 3597–3606.
[35] Z. Gong, P. Zhong, W. Hu, Diversity in machine learning, Ieee Access 7 (2019) 64323–64350.

[36] D. Liang, R. G. Krishnan, M. D. Hoffman, T. Jebara, Variational autoencoders for collaborative filtering, in: Proceedings of the 2018 world wide web conference, 2018, pp. 689–698.
[37] H. Wang, N. Wang, D.-Y. Yeung, Collaborative deep learning for recommender systems, in: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, 2015, pp. 1235–1244.
[38] C. Chen, M. Zhang, Y. Zhang, Y. Liu, S. Ma, Efficient neural matrix factorization without sampling for recommendation, ACM Transactions on Information Systems (TOIS) 38 (2020) 1–28.
[39] G. Adomavicius, Y. Kwon, Improving aggregate recommendation diversity using ranking-based techniques, IEEE Transactions on Knowledge and Data Engineering 24 (2011) 896–911.
[40] S. Vargas, P. Castells, Rank and relevance in novelty and diversity metrics for recommender systems, in: Proceedings of the fifth ACM conference on Recommender systems, 2011, pp. 109–116.
[41] A. Ferraro, G. Ferreira, F. Diaz, G. Born, Measuring commonality in recommendation of cultural content: Recommender systems to enhance cultural citizenship, in: Proceedings of the 16th ACM Conference on Recommender Systems, 2022, pp. 567–572.
[42] A. Starke, M. Willemsen, C. Snijders, Promoting energy-efficient behavior by depicting social norms in a recommender interface, ACM Transactions on Interactive Intelligent Systems (TiiS) 11 (2021) 1–32.
[43] L. Rossetto, M. Baumgartner, N. Ashena, F. Ruosch, R. Pernisch, L. Heitz, A. Bernstein, Videograph–towards using knowledge graphs for interactive video retrieval, in: International Conference on Multimedia Modeling, Springer, 2021, pp. 417–422.
[44] C. He, D. Parra, K. Verbert, Interactive recommender systems: A survey of the state of the art and future research challenges and opportunities, Expert Systems with Applications 56 (2016) 9–27.
[45] L. Rossetto, M. Baumgartner, R. Gasser, L. Heitz, R. Wang, A. Bernstein, Exploring graph-querying approaches in lifegraph, in: Proceedings of the 4th Annual on Lifelog Search Challenge, 2021, pp. 7–10.