A Machine Learning Personalization Flow

Gabriel Bénédict

A Machine Learning Personalization Flow

Academisch Proefschrift

ter verkrijging van de graad van doctor aan de
Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. ir. P.P.C.C. Verbeek
ten overstaan van een door het College voor Promoties ingestelde
commissie, in het openbaar te verdedigen in
de Aula der Universiteit
op vrijdag 8 maart 2024, te 11.00 uur

door

Gabriel Bénédict

geboren te Chavannes-Près-Renens

Promotiecommissie

Promotor:	prof. dr. M. de Rijke	Universiteit van Amsterdam
Co-promotor:	dr. D. Odijk	RTL NL
Overige leden:	prof. dr. K. Balog	University of Stavanger
	prof. dr. N. Helberger	Universiteit van Amsterdam
	dr. M. Lalmas	Spotify
	dr. C.A. Naesseth	Universiteit van Amsterdam
	prof. dr. M. Worring	Universiteit van Amsterdam

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

The research was carried out at the Information Retrieval Lab at the University of Amsterdam, with support from RTL NL & Bertelsmann SE & Co. KGaA.

Printed by Proefschriftspecialist, Zaandam

ISBN: 978-94-93330-58-0

Certains pensent qu'ils font un voyage, en fait, c'est le voyage qui vous fait ou vous défait.

– Nicolas Bouvier

Acknowlegements

The concept of an industry PhD was foreign to me, but it turned out to be a great experience. I joined during COVID, while the lab was still temporarily located in containers. From there on, conditions only improved. We moved on to the beautiful offices at Lab42 that felt more like a tech company than a university building. These great working conditions, combined with colleagues always pushing for a good work-life balance contributed to a great PhD experience.

My supervisors were the central piece in making this a smooth experience. Daan: thank you for giving me so many liberties at RTL and always having our team's back. Maarten: thank you for always asking questions, and encouraging me to do more creative things like organize a workshop!

Krisztian Balog, Natali Helberger, Mounia Lalmas, Christian Andersson Naesseth and Marcel Worring, thank you for agreeing to be part of my PhD committee, and for your valuable time to read and discuss my thesis. Mounia: thank you for listening to my research ideas, from early on as you visited our lab in the container phase. Our later discussions were then essential in orienting my post PhD career. Natali: thank you for supporting the inter-faculty collaboration for our RADio paper. Writing a paper that combines normative and quantitative aspects proved to be hard but was rewarded by the community. Christian: thank you for advising me on diffusion models and especially inpainting. Our discussion was an inspiration for the writing of my final paper and its follow-ups, yet to come.

Sami: thank you for being my paranymph and the other Bayesian in the house.

On the IRLab side, I would like to thank Ali, Ana, Andrew (bringing us up to date with the GenIR literature during reading groups, always with quick clever answers to all our questions. Also, thank you for joining our workshop at the last minute!), Antonios, Arezoo, Barrie, Chang, Chuan, Clara, Clemencia, David, Dan, Evangelos, Gabrielle, Georgios, Harrie, Hongyi, Jin, Jingfen, Jingwei, Julien, Kidist, Maarten, Maartje (you were secretly my mentor during my PhD and while applying for jobs), Maria (being my work-life balance queen), Mariya (probably not the last time we will see each other), Maurits (for being my partner in crime in jokes and loss functions), Ming, Mohammad (for teaching me how to send last minute submissions), Mozhdeh, Olivier, Pablo, Panagiotis, Petra, Philipp (a confidant), Pooya (your dedication to cultural events inspires me), Romain (thank you for inviting me to Grenoble, where I met a lot of brilliant researchers and where we had a good ski session!), Roxana, Ruben, Ruqing (your desire to always be at the edge of research and your dedication to countless submissions are an inspiration to me), Sam (for being my recommender safeguard), Shao, Shashank, Simon (for a memorable ski accident), Svitlana, Thilina (the lab would be nothing without its Wednesday dinners and boardgames), Thong, Vaishali, Weijia (for the endless ping-pong games), Xinyi, Yibin, Yuanna, Zihan.

I also want to thank the students who wrote their thesis with me: Cas, Jordi, Marta, Paula – I learned a lot with you.

On the side of RTL, I would like to thank Aash, Apoorva, Daniella, Denise, Dina, Isabel, Iskaj, Ivan, Jingwei, Jochem, Joel, Lotte, Mateo, Maurits, Maya, Philip, Prajakta, Rashid, Sanne, Tanja; the whole data science team for partnering on many projects and always being supportive of my fluctuating focus between RTL and the PhD. Carsten for helping me navigate the industry-academia maze and motivating me to start a PhD with RTL. Pam and Niki for hiring me and making me feel like a valued asset for RTL all along the way. At Musicflow, I would like to thank, Terence, Vincent, Ren. Nico for becoming a valuable friend. And more broadly at Bertelsmann: thank you Adam, Brenda, Carmine, Caspar, Charles-Edouard, Franziska, Hannes, Isabella, Jendrik, Joan, Lena, Peter, Rick, Rick, Savita, and Vicki.

Outside of my direct colleagues, thank you to Bas, Don, Ivona, Olivier, Rishabh, Sanne, Thiviyan, Thomas, who inspired me, advised me or partnered with me.

I would like to thank my successive roommates for bearing with my PhD and life ups and downs. Maarten you were already a friend before moving in together and Lars, you became one to me.

These PhD years would not have been bearable without the fun times with: Alexis, Anna, Anthony, Antoine, Avi, Bar, Bat-El, Benny, Cory, Emmanuel, Estelle, Gaston, Harry, Ilan, Jamie, Julia, Kai, Magdalena, Maroussia, Maud, Max, Michael, Michelle, Montaga, Natalia, Raphael, Ruben, Sean, Vincent, Yanki, Yannick; and the intemporals elsewhere, Cyrus, Mathieu, Tobi, Zach.

And of course I want to thank my precious family: my parents Alexandra and Pierre whom I owe everything for my accomplishments and who have always supported me in any way they could. My sister Caroline for always having my back in all situations and Simon for the research discussions.

Nathalie: when I would wake up in the middle of the night and say “I got scooped!”, you were the one who had to bear with it and so much more. But you have always taken me seriously and guided me through a lot of perilous situations. Thank you for being my (para)nymph and standing by my side the whole time.

Gabriel Bénédict
Madrid
January 2024

Table of Contents

Chapter 1

Introduction

Video streaming platforms have changed the way people consume and interact with digital media (Gomez-Uribe and Hunt, 2015). One of the key innovations of video streaming platforms with regards to traditional television, is *personalization*, that is, the ability to tailor the experience to each single user and their interests, given their past behavior on the platform (Bennett et al., 2012; Teevan et al., 2005).

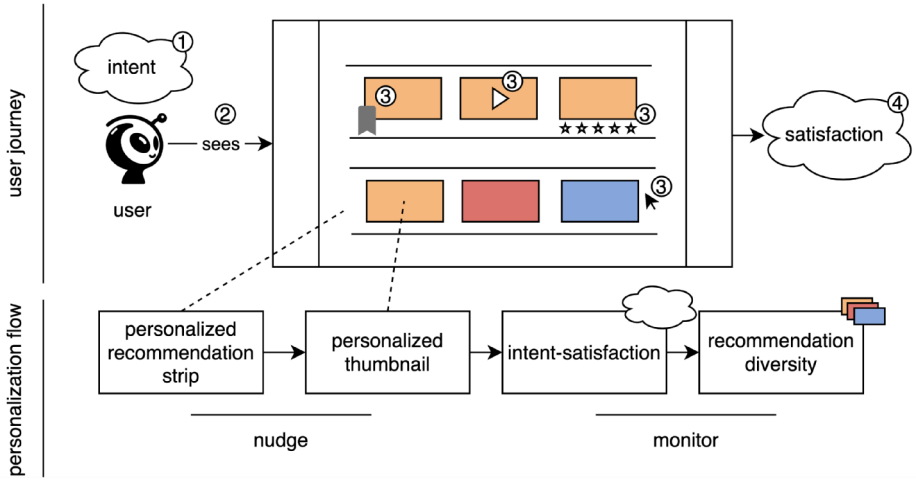


Figure 1.1: Our *user journey* and *recommendation flow* paradigm. Along the *user journey*, a user (1) has an intent (e.g., series catch up), (2) sees the home page with video thumbnails inside recommendation strips, (3) interacts with the platform (clicks, scrolls, bookmarks, video plays, ratings, etc.), and (4) feels a certain level of (dis)satisfaction with the platform over time. With the *recommendation flow*, the platform (i) generates personalized strips, (ii) selects personalized thumbnails (stills from a video), (iii) measures the relationship between intent, interactions and satisfaction, (iv) measures the appropriate level of content diversity. (i) and (ii) nudge the user towards consuming certain content, (iii) and (iv) help monitor the user's resulting actions.

When designing and examining methods for personalization, the concept of *user journey* is helpful. We propose this term in the context of this thesis to describe the user's interactions with a video streaming platform from login to

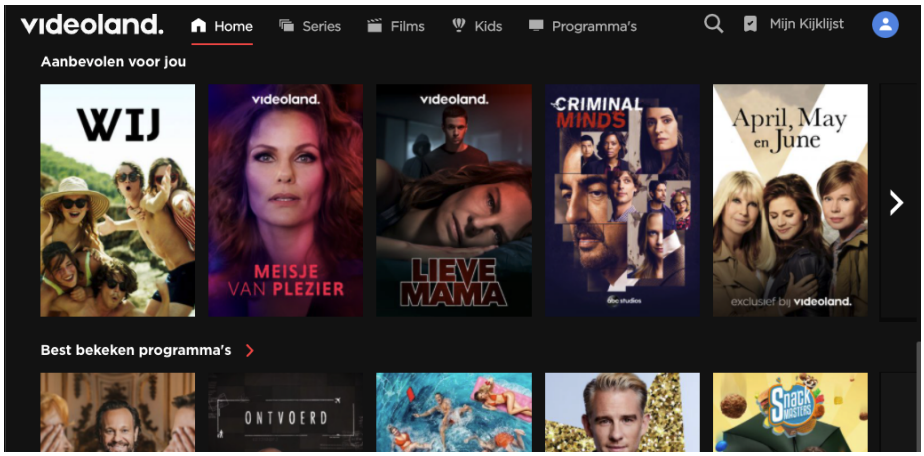


Figure 1.2: Videoland's recommendation strips.

logout. In the setting of video streaming, the user journey consists of several steps (see Figure 1.1). First, users come to the platform with some *intents* (e.g., binge-watching a series, finding a family-friendly movie, discovering new genres) (Bénédict et al., 2023a). Then, they see a customized home page with various horizontal *recommendation* strips. In Figure 1.2 we show how this step in the user journey is realized with the landing page of Videoland, a Dutch video-on-demand service. Each strip contains videos with (sometimes personalized) *thumbnails* (the clickable image that represents the video content). Over time, users interact with the platform and leave *behavioral* signals (e.g., clicks, watch time, bookmarks, ratings, etc.). From the platform's perspective, deciphering how these behavioral signals, prompted by user intents, translate into overall *satisfaction* remains a complex challenge (Mehrotra et al., 2019).

Behavioral signals generated during the user journey have long been drawn on to target users individually (Lai and Yang, 2000). The term *personalization* is commonly used to describe this strategy (Rucker and Polanco, 1997). Initially, in the domain of e-commerce in the late 1990s, personalization was restricted to email newsletters and aimed at user groups (Kohavi and Provost, 2001). Personalization appeared on early video streaming platforms like Netflix in the form of one-dimensional lists, i.e., the recommendation strip mentioned above (Kohavi and Provost, 2001). Today, the user journey on video platforms is steered by the platform's algorithms: personalization is used in the ordering of the strip (i.e., multiple one-dimensional lists), in the choice of the thumbnail, in the font title of the thumbnail and in search. A review of these strategies can be found in (Gomez-Uribe and Hunt, 2015).

In this thesis, *personalization flow* is used to denote a set of a video platform's algorithms for a tailored user journey (see Figure 1.1). In industry, one could link these algorithms with a proper data architecture to retrieve user data, feed it to the algorithms, and serve it back to the users, among other things.

Such a data architecture, together with the algorithms, would then qualify as a *personalization pipeline*. This thesis focuses on the algorithms rather than on the engineering related to the data architecture. We highlight this with our denomination personalization flow. One part of the flow retrieves data that feeds all other algorithms: collecting user data and analyzing user behavior (Ronen et al., 2016). The data granularity can range from the number of items watched (just one data point per user session), all the way to recordings of all mouse movements (thousands of data points per session). With that session-level data, streaming platforms attempt to predict what the user will do and adapt the user journey to the user: the next movie to watch, the subsequent logins, the midterm satisfaction (typically, the amount of time spent on the platform per month), all the way to churn rate (subscription cancellation rate) (Hohnhold et al., 2015).[1] These predictive models are tested first on a simulated platform with simulated users "offline." Models are then evaluated on the platform exposed to users "online" over several iterations and over time. In the evaluation phase, preferences, and interaction patterns are captured again as a feedback loop (Beel and Langer, 2015; Gomez-Uribe and Hunt, 2015).

Aside from the observable user feedback signals (such as clicks, watch time, time on the platform, etc.), a platform can also take hidden signals into consideration. In this thesis, we survey and model user intent ("I want to watch the next episode of my favorite show," "I am looking for new content," "I want to bookmark items for later viewing," etc.).

Besides satisfying the users, the platform also bears another responsibility, because it is able to steer the user towards certain consumption behaviors. For example, the platform has to ensure that the content it offers is *diverse* and promotes representative voices (e.g., promoting screenwriters of different genres, movies in different languages, a variety of movie genres).

To sum up, the personalization flow is often geared towards optimizing relatively simple metrics like the number of minutes seen (Mehrotra et al., 2019) and the churn rate (Lee and Lee, 2006), but it is also directly linked to a responsibility to balancing multiple, sometimes competing targets, such as long-term user satisfaction (Hohnhold et al., 2015), content diversity, and ethical considerations (Helberger, 2019).

In this thesis, we propose individual tools that use the logged interactions from the user journey above, for the steps of *recommendations*, *thumbnail* selection, *intent-satisfaction* linking, and *diversity* measurement. Together, these tools form our proposed personalization flow. In the thesis we seek to make novel contributions to each of these steps. We adapt diffusion models (Sohl-Dickstein et al., 2015a) from the image domain (continuous-2D-structured-data) to recommendation (binary-1D-unstructured-data). This way, we open the door to the use of priors in recommendation – preferred movie genres, past behavior or incomplete viewing history, etc. – like is seen in diffusion for images (an image description, a previous reference image, a masked image, etc.) (Zhang et al., 2023a). As for personalized

[1] In order of increasing forecast horizon.

thumbnails, existing research in multilabel image classification is highly reliant on variations of the binary cross-entropy loss (Fisher, 1912), but we think that the multilabel setting (as opposed to the binary or multiclass setting) requires its own solution. For the next step, we identify a lack of a systematic approach for *intent-satisfaction* studies, that would provide survey design, code and modern bayesian approaches to the problem. Finally, we argue for the importance of a *diversity* metric for news / movies recommendations that is distribution agnostic – to adapt to any distribution of discrete normative standpoints – and rank-aware – to accommodate for the propensity of a user to scroll up to an item on a ranked list. As such a metric is missing from the literature, we propose a rank-aware adaptation of the Jensen-Shannon Divergence.

In short, in this thesis, we focus on the video streaming platform ecosystem, explore the challenges and opportunities of personalization, recommendation, and user behavior analysis. By combining survey methods, modeling, adaptive testing, and behavioral analysis, this study aims to contribute to the development of video streaming platforms that can provide user satisfaction in a responsible way.

1.1 Research Outline and Questions

We scope the thesis around four research questions, each corresponding to a research chapter in the thesis.

Personalization on streaming platforms is often seen as a way of predicting what users want to watch based on their preferences and behavior. Our first research question addresses the entry point of the user journey on a personalized platform, namely recommendations on the home page of a video platform, that is, the page where a user first lands when visiting the streaming platform.

RQ1 Can we use diffusion to do recommendation in the classical user-item matrix setting?

Traditionally, recommender systems directly retrieve content from the catalog to the user (Dehart, 1966). Alternatively, user instructions and feedback are fed to a generator of personalized content, before retrieving and ranking from that new library of generated content, according to the recent generative recommendation paradigm (Wang et al., 2023). The generative recommendation paradigm covers individual content generated from scratch (like diffusion based image creation) or generative recommendation of existing content (like conversational recommendation). We investigate how diffusion models can be used to do generative recommendation: generate a list of recommended content. Diffusion models are physics inspired neural models, that include a forward (perturbation) and backward (learning) process on each example (Sohl-Dickstein et al., 2015a). Diffusion has been applied to images, music and other modalities. Unlike these, the classical recommendation setting of the user-item matrix (Koren et al., 2009) does not entail spatial relationships between data points: contrary to pixels on an image, there is no information encoded in the allocation of users and item on

a matrix. We illustrate this in RecFusion (Bénédict et al., 2023), where we first use Unets (Ronneberger et al., 2015a) to fit data in a spatial way, before going back to the classical recommendation neural setting of feeding data user-by-user. For this one-dimensional user vector, we provide a proof and first experiments to show that a binomial (Bernoulli) diffusion process is viable.

After recommendation, the next step of our flow caters to the display of recommended videos via thumbnails.

RQ2 Is there a way we can generate personalized thumbnails for each item on a streaming platform?

Personalization can be seen at different levels of granularity: from targeting user segments (into interests, age groups, etc.) to targeting single users differently. For this research question, we are interested in how thumbnails (i.e., static images) can be classified into different categories, more than knowing if we can target each single user. We therefore opt for a least granular option: we assume that each user has a favorite genre. We can provide a thumbnail personalized to that genre (e.g., show a romantic scene from an action movie, if the favorite genre is romance). Given editorial or automatically selected candidates for thumbnails, we wish to display the one that is most closely associated with the user's preferences. This reduces to a multilabel classification problem: given an image, predict one, or many, genre(s). When thinking about classification, the confusion matrix (Stehman, 1997) – with its false positive, true positive, false negative and true positive quadrants – is a classic way to build evaluation metrics. But these metrics are hardly used at training time. We think it is because these quadrant values require counting, which is not differentiable at training time for gradient descent (Robbins and Monro, 1951; Kiefer and Wolfowitz, 1952). We propose a way to build surrogates to these count metrics via sigmoid functions. More precisely, we look at maximizing for the F1 score via our sigmoidF1 surrogate loss function (Bénédict et al., 2022), as a multilabel classification loss over an entire batch. We show that this improves on classical image and text benchmarks with classical backbones (CNNs (Fukushima, 1980) and transformers (Vaswani et al., 2017)).

Recommendations and thumbnails are what primes the users' interactions with the video platform. This relates to the next step in our flow:

RQ3 Are users' intents together with their behavioral data useful signals to predict or explain satisfaction on a video streaming platform?

Streaming platforms have access to users' implicit feedback (such as clicks, scrolls, time on the platform, etc.) and explicit feedback (such as ratings and bookmarks) via their personalization flow. Some of the user behaviors will remain forever hidden from the platform though for privacy or technical reasons (e.g., how many people sit in front of the device, the content the user consumes on other platforms). Among them, we explore user intents of a video streaming platform. Previous work has defined intents for music (Mehrotra et al., 2019); we propose to define them for video and propose a transparent approach by revealing our survey design, code and simulated data. In (Mehrotra et al., 2019), logistic

regression was used to predict satisfaction based on intents and behavioral data. We propose to use random forests and Bayesian hierarchical modeling to enhance accuracy and interpretability, respectively.

Finally, we close off our flow with an approach to diversity.

RQ4 Can we formulate a divergence metric that measures the normative diversity of recommendations?

Videos and especially news platforms serve content that is opinionated. Over time, platforms have been growing their engineering teams to cover more and more of the user journey stages (home page, title fonts, watch/read next etc.) (Gomez-Uribe and Hunt, 2015), with more and more powerful and sometimes generative models. The user is thus influenced by the platform's algorithms and thus the platform's explicit or implicit norms and values. Can we empower a video/news platform to measure its ability to align to its norms and values? We would like to account for how a platform means to properly inform citizens (as defined in (Helberger, 2019)) and any form of diversity metric (topic, presence of alternative voices, complexity of the text, etc.). RADio (Vrijenhoek et al., 2022), the framework we propose caters to these normative aspects but also to the specific recommendation context: RADio is rank aware and caters for any kind of discrete distribution via our proposed rank-aware Jensen-Shannon divergence (Lin, 1991). Chapter 5 is focused on news recommendation but trivially generalizes to any domain that has categories (e.g., video streaming with movie genres).

Our research questions have been outlined in this section. The main contributions of this thesis will be summarized in the next section.

1.2 Main Contributions

In this section, we summarize the main contributions of this thesis. We separate theoretical from artifact (that is, tool and experimental design) contributions.

Theoretical Contributions

- An adaptation of diffusion to unstructured data, where there is no spatial dependency (Chapter 2).
- The use of diffusion for binary and/or 1D data: A demonstration that KL divergence is also suited for binary data and that the Bernouilli Markov process has the same properties as its Gaussian counterpart (Chapter 2).
- A multilabel loss function that ingests all examples in a batch (as opposed to existing Cross-Entropy-like additive losses) (Chapter 3).
- An F1 score surrogate as a loss function (Chapter 3).
- An account of the current limitations and underreporting of thresholding at inference time (Chapter 3).

- A proposal of typical intents for video streaming that we divide into explorative and decisive categories (Chapter 4).
- A diversity metric that adapts to any normative concept and expressed as the divergence between two (discrete) distributions, rank-aware and mathematically grounded in distributional divergence statistics (Chapter 5).

Experimental Contributions

- To model user intent, a Bayesian multilevel models for visualization and explanations, along with random forests for improved accuracy (Chapter 4).
- A reproducibility study from music to video streaming platforms of intent-satisfaction modeling (this time with code and synthetic data) (Chapter 4).
- An in-app survey design for a medium size streaming platform ($\sim$1 million users) and corresponding synthetic data (Chapter 4).
- A metadata enrichment flow (e.g., sentiment analysis, named entity recognition) to extract normative concepts from news articles (Chapter 5).

1.3 Thesis Overview

This first chapter introduces the main topics and goals of this thesis, and suggests some possible ways to read it. The thesis has six chapters in total, and this is the first one. The following four chapters each address one of the research questions that we presented in Section 1.1. Each chapter is based on a previously published paper (see Section 1.4 below), and can be read on its own. If the reader is interested in the entire thesis, we recommend following the original order of chapters, as they follow the *user journey* and its respective *personalization flow* on a streaming platform. The final chapter summarizes the main findings and contributions of this thesis, and proposes future research directions.

1.4 Origins

We list the publications that are the origins of each chapter below.

Chapter 2 is based on the following paper:

> Gabriel Bénédict et al. RecFusion: A Binomial Diffusion Process for 1D Data for Recommendation. *The 1st Workshop on Recommendation With Generative Models on the 32nd ACM International Conference on Information and Knowledge Management*, 2023.
>
> GB wrote the first draft, code, mathematical formulations, designed and ran experiments. GB was helped all along the way via discussion with the coauthors. The coauthors then edited the first draft together with GB. GB did most of the writing.

Chapter 3 is based on the following paper:

> Gabriel Bénédict et al. sigmoidF1: A Smooth F1 Score Surrogate Loss for Multilabel Classification. *Transactions on Machine Learning Research*, 2022.
>
> GB wrote the first draft, code, mathematical formulations, designed and ran experiments. GB was helped all along the way via discussion with the coauthors. The coauthors then edited the first draft together with GB. GB did most of the writing.

Chapter 4 is based on the following paper:

> Gabriel Bénédict et al. Intent-Satisfaction Modeling: From Music to Video Streaming. *ACM Transactions on Recommender Systems*, 1(3), 2023.
>
> GB wrote the first draft, code, mathematical formulations, designed and ran experiments. GB was helped all along the way via discussion with the coauthors. The coauthors then edited the first draft together with GB. GB did most of the writing.

Chapter 5 is based on the following paper:

> Sanne Vrijenhoek et al. RADio – Rank-Aware Divergence Metrics to Measure Normative Diversity in News Recommendations. *Proceedings of the 16th ACM Conference on Recommender Systems*, pages 208–219, 2022.
>
> GB, together with SV and MGG, wrote the first draft, code, mathematical formulations, designed and ran experiments. GB was helped all along the way via discussion with the coauthors. The coauthors then edited the first draft together with GB. GB and SV did most of the writing.

The writing of this thesis also benefited from work on the following publications:

- Garbiel Bénédict et al. Gen-IR@SIGIR 2023: The First Workshop on Generative Information Retrieval. *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3460–3463, 2023.
- Ali Vardasbi et al. The University of Amsterdam at the TREC 2021 Fair Ranking Track. *TREC Fair Ranking*, 2021.
- Gabriel Bénédict. Generative Adversarial Networks. *Spectra ML Review Paper Competition*, 2021. eprint: `09.00009`.

Chapter 2

Generative Recommendations with Diffusion

As a first step in our personalization flow, we look at recommendation – or presenting the right content to the right user algorithmically – in a new generative manner.

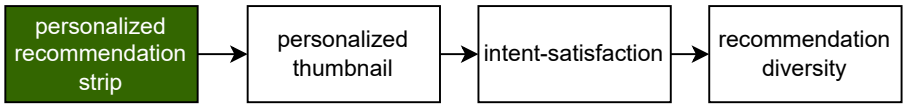


Figure 2.1: The first step of the personalization flow.

Generative models, like diffusion models have been used on modalities like music, image; but can they be used on unstructured data? Beyond the technical aspect of fitting diffusion to yet another problem, we are motivated to use diffusion on the classical user-item matrix situation because it opens doors to diffusion mechanisms like guidance (e.g., conditioning on a movie genre) or inpainting (e.g., guessing what users have watched on other platforms and filling some gaps in the user-item matrix). In other words,

RQ1: Can we use diffusion to do recommendation in the classical user-item matrix setting?

We respond to this question by first formulating different diffusion models for this particular recommendation setup. While the U-Net architectures translated from the image domain (Ho et al., 2020) do not perform well, our own 1D diffusion models based on a simple MLP architecture and a Bernoulli forward and backward process are performing competitively against VAE methods. For the first time, we propose a mathematical derivation of that Bernoulli process first proposed in an appendix of the original diffusion work (Sohl-Dickstein et al., 2015a).

This chapter was published at The 1st Workshop on Recommendation With Generative Models on the 32nd ACM International Conference on Information and Knowledge Management "RecFusion: A Binomial Diffusion Process for 1D Data for Recommendation" (Bénédict et al., 2023).

2.1 Introduction

Diffusion models have been profusely used in the image domain (Croitoru et al., 2023). Next to the 2D setup, an increasing amount of research is focused on the 3D domain (Ho et al., 2022), as well as diffusion on the embedding space (Gao et al., 2023).[1] Typical image diffusion models rely on a U-Net (Ronneberger et al., 2015b) architecture with attention layers and process entire images at once. However, image diffusion models rely on and exploit spatial correlations (i.e., between pixels in localized regions). Unstructured data settings, where these assumptions do not hold, are under-explored. In this chapter, we consider the recommendation systems domain and, more specifically, 1D binary data in the classical recommendation setting. The recommendation setting is characterized by the following conditions: (i) a user's interaction history is organized like 1D binary data, where columns represent items; and (ii) organized as a matrix for multiple users, each entry in this *interaction matrix* corresponds to the type of interaction between a specific user and item. These interactions can either be *explicit* (ratings, 'likes' or dislikes), or *implicit* (dwell time, clicks, purchases, etc.).

Most modern recommender systems leverage the implicit feedback paradigm, which utilizes data that is not explicitly provided by the user, such as click data, purchase history, browsing behavior. Research in recommender systems employs simpler linear models (Jannach et al., 2020; Rendle et al., 2019; Rendle et al., 2022; Klingler et al., 2022), or neural models, many of which employ the variational autoencoder (Kingma and Welling, 2013) framework, e.g., cVAE (Chen and de Rijke, 2018), RecVAE (Shenbin et al., 2020) or MultVAE (Liang et al., 2018). Neural models have benefits beyond recommendation performance (e.g., in controllability / critiquing (Wu et al., 2019e; Li et al., 2020; Luo et al., 2020; Yang et al., 2021)), with some models utilizing disentanglement (Bengio et al., 2013; Higgins et al., 2017) for this purpose (Ma et al., 2019; Nema et al., 2021; Bhargav and Kanoulas, 2021). Beyond controllability, neither non-neural nor VAE-based models can handle time information directly. Making it hard, for example, to deal with preference drift (Huang et al., 2022), where more recent items may be more relevant for future recommendations. In principle, diffusion models should be able to deal with these recommendation conditions. There have been some initial attempts at modeling recommendation problems using diffusion; CODIGEM (Walker et al., 2022) defines a diffusion model akin to early attempts in diffusion, with one neural network per diffusion step. We propose RecFusion, a diffusion model inspired by the DDPM (Ho et al., 2020) architecture adapted for 1D data. We also propose the Bernouilli diffusion process, specifically designed for binary data. We experiment with different common diffusion techniques, such as noise timestep embeddings, modelling the mean and variance and different noise schedules.

Setup. We assume a binary non-sequential top-n implicit feedback setting

[1] This line of work on the embedding space is still emergent, we could only find a work in the text domain.

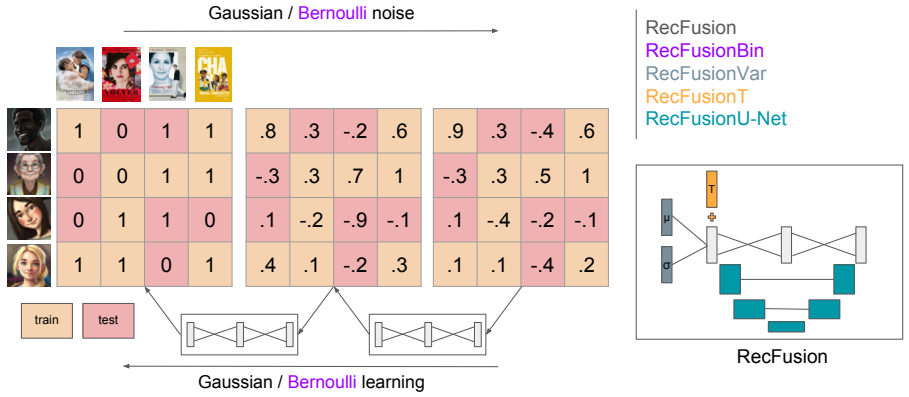


Figure 2.2: The RecFusion architecture and its variations (user images generated with DALL · E 2).

(see explicit assumptions in Section 2.2): we seek to predict only the immediate next best item(s) for each user and the time is unknown for any past user-item interaction. The reason for choosing this standard setup is two-fold: (i) by using binary data, we can study the use case of diffusion for binary data, and (ii) we remain comparable with the overwhelming majority of recommendation literature. Indeed, it is common for Assumptions 1–5 that we specify in Section 2.3.1 to be used.

Main results. As previously shown in the literature, VAE-based models and non-neural models outperform more complex methods in the standard recommendation setting. RecFusion outperforms existing diffusion models for recommendation and opens the way to use guidance and conditioning.

Key contributions of the chapter include: (i) we demonstrate uses of diffusion where there is no spatial dependency, (ii) we offer a simple implementation of diffusion that can accomodate any binary and/or 1D data, (iii) we propose modern Variational Auto Encoder (VAE) architectures for recommendation (diffusion models are hierarchical VAEs (Kingma et al., 2021)), and (iv) our code is open and available at `https://github.com/gabriben/recfusion`, implemented using a reproducible and well-tested framework to facilitate follow-up work.

2.2 Method

2.2.1 Diffusion models

Diffusion models (Sohl-Dickstein et al., 2015b) are latent variable models that strive to address the issues of tractability and flexibility by gradually converting a distribution into another using a Markov chain. The gradual change from the intractable data distribution to a known one allows for the reverse process to

be learned (Feller, 1949). In simple terms, this solves traceability, as now one can obtain samples of the data distribution by starting from the known one and using the learned reverse process. At the same time, it introduces flexibility, thanks to the compounding effect of many small simple steps, which allow for the complexity of the target distribution.

Given the starting binary variable $\mathbf{X}^0$, the forward diffusion process is a Markov chain used to sample latent variables $\mathbf{X}^1, \ldots, \mathbf{X}^t, \ldots, \mathbf{X}^T$. We try to match the original notation from (Sohl-Dickstein et al., 2015b), and make a few simplifying assumptions that are clear from context. In order to generalize, we use the notation $\mathbf{X}$ to indicate a 2D user-item matrix composed of user vectors $\mathbf{x}_u$,[2] themselves composed of individual interactions $x_{u,i}$. The *forward diffusion process* gradually adds noise using a Markov kernel $\kappa_p(\mathbf{X}|\mathbf{X}'; \beta)$, until the target distribution $p(\mathbf{X})$ is reached using diffusion rate β. The kernel used is constructed such that it guarantees that the target distribution is reached in the limit of $T \to \infty$.

The forward diffusion process is factorized as follows:

$$q\left(\mathbf{X}^{1:T} \mid \mathbf{X}^0\right) = \prod_{t=1}^{T} q\left(\mathbf{X}^T \mid \mathbf{X}^{t-1}\right) \tag{2.1}$$

$$q\left(\mathbf{X}^T \mid \mathbf{X}^{t-1}\right) = \kappa_p\left(\mathbf{X}^t|\mathbf{X}^{t-1}; \beta_t\right). \tag{2.2}$$

In this work, the kernel κ_p is either given by the original gaussian diffusion (Sohl-Dickstein et al., 2015b) or binomial diffusion (see next section). Then, we can define the *reverse diffusion process* using an analogous formulation:

$$p_\theta\left(\mathbf{X}^{0:T}\right) = p\left(\mathbf{X}^T\right) \prod_{t=1}^{T} p_\theta\left(\mathbf{X}^{t-1} \mid \mathbf{X}^t\right). \tag{2.3}$$

Thanks to the use of very small diffusion steps, the functional form of the reverse process is the same as the forward process (Feller, 1949).

The noise schedule of the forward diffusion process $\beta_1, \ldots, \beta_T$ is either learned or follows a predetermined schedule (increasing, decreasing or constant). The optimization of the backwards diffusion process follows the classical Evidence Lower Bound (ELBO) formulation:

$$\mathbb{E}_q[\underbrace{D_{\mathrm{KL}}\left(q\left(\mathbf{X}^T \mid \mathbf{X}^0\right) \mid p\left(\mathbf{X}^T\right)\right)}_{L_T} + \sum_{t>1} \underbrace{D_{\mathrm{KL}}\left(q\left(\mathbf{X}^{t-1} \mid \mathbf{X}^T, \mathbf{X}^0\right) \mid p_\theta\left(\mathbf{X}^{t-1} \mid \mathbf{X}^T\right)\right)}_{L_{t-1}} - \underbrace{\log p_\theta\left(\mathbf{X}^0 \mid \mathbf{X}^1\right)}_{L_0}], \tag{2.4}$$

where D_{KL} is the KL divergence between each forward process step and its reconstructed representation in the backwards process.

[2] In most neural recommendation algorithms, a single vector $\mathbf{x}_u$ is fed to the model, see Section 2.2.3 for a discussion on that topic.

2.2.2 A binomial forward process

We use a binomial (single trial Bernoulli) Markov diffusion process to fit the binomial input data (see assumptions in Section 2.3.1). Intuitively, this corresponds to performing bit flips over diffusion time steps in the forward process and predicting these bit flips in the reverse process. The latter is defined with

$$p_\theta\left(\mathbf{X}^{t-1} \mid \mathbf{X}^t\right) := \mathcal{B}\left(\mathbf{X}^{t-1}; \pi_\theta\left(\mathbf{X}^t, t\right)\right), \tag{2.5}$$

where $\mathcal{B}(u;\pi)$ is the distribution for a single Bernoulli trial (bit flip), with $u = 1$ occurring with probability π, and $u = 0$ occurring with probability $1 - \pi$. The forward process is a flip of the original $\{0, 1\}$ bits with increasing chance, determined by the schedule β_t:

$$q\left(\mathbf{X}^t \mid \mathbf{X}^{t-1}\right) := \mathcal{B}\left(\mathbf{X}^t; \mathbf{X}^{t-1}\left(1 - \beta_t\right) + 0.5\beta_t\right). \tag{2.6}$$

Let $\alpha_t = 1 - \beta_t$ and $\bar{\alpha_t} = \prod_{\tau=1}^{t} \alpha_\tau$, we get (see Appendix):

$$q\left(\mathbf{X}^t|\mathbf{X}^0\right) = \mathcal{B}\left(\mathbf{X}^t; \bar{\alpha}_t\mathbf{X}^0 + \frac{1}{2}\left(1 - \bar{\alpha}_t\right)\bar{\alpha}_t\right). \tag{2.7}$$

This means that, when we use a binomial forward kernel the reverse diffusion process will also be binomial. Then, we can simply use a network to predict the bit flip probability for the reverse process to be modeled accurately.

For the loss function we use a Binary Cross Entropy (BCE) ELBO formulation (Eq. 2.4) for L_0:

$$L_0 = -\log p_\theta\left(\mathbf{X}^0 \mid \mathbf{X}^1\right) = \mathbf{X}^1 \odot \log \mathbf{X}^0 + \left(1 - \mathbf{X}^1\right) \odot \log\left(1 - \mathbf{X}^0\right). \tag{2.8}$$

We use $\odot$ as the sign for element-wise multiplication. The traditional ELBO loss relies on the KL divergence. We demonstrate that the KL divergence is also suited for binary data (see Appendix A.1). Additionally, we derive the Bernoulli Markov process and verify that a combination (multiplication) of Bernoulli distributions is still a Bernoulli distribution (see Appendix A.2). This guarantees that we can use the Gaussian Markov process properties.

2.2.3 RecFusion – Recommendation systems as diffusion models

In the image domain, $\mathbf{X}^0$ is of dimension corresponding to the image resolution.[3] Instead, in the recommendation setup, $\mathbf{X}^0$ is the full user-item matrix.

Full-matrix. In our recommendation setting, we can consider the entire user-item matrix as $\mathbf{X}^0$ of dimension $U \times I$, where U is the number of users and I is the number of items. Each cell in that matrix is a binary representation of the feedback of a user on an item (e.g., for MovieLens (Harper and Konstan, 2015), x_{ui} is 1 for ratings above 3 stars and 0 otherwise, following (Liang et al.,

[3]See Section 2.2.1 for our choice of notation.

2018)). We are thus framing our setting as non-sequential recommendation with binary feedback (see Section 2.3)). With ever-growing user-item matrices, it quickly becomes infeasible to perform in-memory computations. The solution for image diffusion models is to use a first diffusion model for a say 32×32 image and then use several super-resolution models to upscale it (Saharia et al., 2022). We consider that a user-item matrix cannot be downscaled by blurring it, because it does not contain hierarchical features, unlike for an image (e.g. an image of a dog probably contains the dog's head, which contains eyes, etc.).

User-batch. Instead, we could think of batches of users $\mathbf{X}^0_{u \in b_j} \ \forall b_j \in \mathbf{b}$.[4] In that case, the input matrix is still big. For example on MovieLens1M, a batch size 200 users leads to a 200×10000 matrix compared to a 32×32 image, but possible to fit in memory. There are two more advantages to feeding by batch. (I) We can now perform gradient descent over several examples of the data instead of just one matrix example, and (II) we can form batches of items of the same category and use that as a downstream task (a.k.a. diffusion guidance (Ho and Salimans, 2021)). For example, we could batch by movie genre in the case of the MovieLens dataset (Harper and Konstan, 2015). This user-batch formulation is still similar to the original 2 dimensional image setting, but assumes relationships between users close together in the matrix if convolution-based models are used. This assumption is unrealistic and we thus think that this is not a viable approach theoretically. We nonetheless verify that assumption empirically with *RecFusionU-Net2D*.

User-by-user. Alternatively, we can use a 1D formulation (batch size of 1), with $\mathbf{x}^0_u$, the vector of all item feedbacks for user u. In that case, we assume no relationships between users. This corresponds to the formulation of MultVAE (Liang et al., 2018). With this formulation, the advantages of the user-batch formulation are kept and spatial dependence between users does not need to be assumed. This setting applies to *RecFusion, RecFusionT, RecFusionVar, RecFusionBin, RecFusionU-Net1D*.

We use the vector notation $\mathbf{x}$ for the rest of the chapter, to refer to $\mathbf{x}_u$, a user vector. Below we look at two practicalities, conditional generation and the fully perturbed recommendation matrix.

Generate from $\mathbf{x}^1$, a simple alternative to conditional generation

In practice, a recommendation platform is interested in finding the top K next items for users (see Assumption 4 in Section 2.3.1). In a traditional diffusion inference setup, we would start with a completely random recommendation matrix $\mathbf{x}^T$ and generate $\mathbf{x}^0$ iteratively via the backward diffusion pass through the neural network $p_\theta\left(\mathbf{x}^{t-1} \mid \mathbf{x}^t\right)$ over each diffusion time step t. Without any conditioning / guidance / inpainting techniques, the generated matrix $\mathbf{x}^0$ remains the same, given a particular random $\mathbf{x}^T$. We propose a simpler approach: at inference time, we feed the validation/test recommendation matrix as $\mathbf{x}^1$ and perform a single backward diffusion step to $\mathbf{x}^0$. One question remains: what is a

[4] Batching by items is also possible, but would rather fit the domain of item-based collaborative filtering (Sarwar et al., 2001).

completely perturbed matrix?

What is a completely perturbed matrix $\mathbf{X}^T$ in the recommendation setting?

Strictly speaking, the kernel κ_p in the Benoulli markov chain forward process $q\left(\mathbf{X}^T \mid \mathbf{X}^{t-1}\right) = \kappa_p\left(\mathbf{X}^t|\mathbf{X}^{t-1}; \beta_t\right)$ determines the final state $p(\mathbf{X}^T)$. Instead, as an experiment, we propose here to start from a desired final state $p(\mathbf{X}^T)$ and determine a markov chain that leads to it.

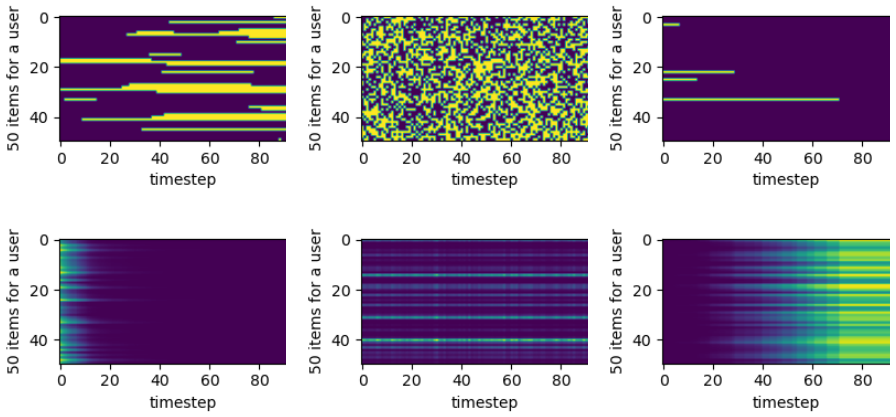
More concretely, we ask what is a completely diffused matrix $\mathbf{X}^T$? Is it (a) a matrix with the same mean activity as the input data $p(\mathbf{X}^T = 1) = E(x^0) = \bar{\mathbf{X}}^0$ (as proposed by Sohl-Dickstein et al. (2015b)) (b) a matrix with a fair coin flip activity $p(\mathbf{X}^T = 1) = 0.5$ – in the binomial case $\mathcal{B}(\mathbf{X}^T; 0.5)$ – or (c) a matrix full of zero values $p(\mathbf{X}^T = 1) = 0$. We show these three alternatives in Figure 2.3 with a Bernouilli diffusion example.

With (a) and (b), we experiment with allowing bit flips from 0 to 1 and from 1 to 0, by formulating $p_\theta = \mathcal{B}\left(x^t; \beta_t\right)$ and $x^t = (1 - p_\theta) \cdot x^{t-1} + p_\theta \cdot (1 - x^{t-1})$. For (a), we use a fixed schedule of $\beta_t = 0.01\ \forall t$. The reverse diffusion process is able to pick up a signal. For (b), we use a fixed schedule of $\beta_t = 0.5\ \forall t$. Right away, the user vector becomes chaotic and no real signal is picked up by the reverse diffusion process. With (c), we only allow bit flips from 1 to 0 and end up with with a matrix full of zeroes. For that we let $p_\theta = \mathcal{B}\left(\mathbf{X}^t; \mathbf{X}^{t-1}\left(1 - \beta_t\right) + 0.5\beta_t\right)$ and $x^t = p_\theta \cdot x^t$. Again, the reverse diffusion process picks up a signal. We found (c) to work best in practice. We conjecture that this is because bit flips only go in one direction and that this information flows more smoothly in the gradient descent steps.

Architecture

We propose a few different architectures for RecFusion, in order of complexity. *RecFusion*, a three layer fully connected network with tanh activation. *RecFusionT* with a time step embedding: we first tried to use the time embedding as in the original attention paper (Vaswani et al., 2017), namely feeding the time embedding to the output of the MLP $f(x) + Z_t$. This was not very successful. Instead we fed the time embedding in a DDPM (Ho et al., 2020) manner (we are not sure if this practice emerged in DDPM or before): $f(x + Z_t)$. *RecFusionVar*, which predicts mean and variance/error of diffusion steps like in DDPM (Ho et al., 2020). *RecFusionBin* our own 1D Bernoulli diffusion model: forward steps as described in Section 2.2.3 (c), reverse steps with a *RecFusion* architecture but a sigmoid final activation and our own BCE ELBO loss (Eqn. 2.4 and 2.8). *RecFusionU-Net1D* is the original DDPM (Ho et al., 2020) architecture simplified, with only one channel and flattened on one dimension to allow for user vector input $\mathbf{x}$. *RecFusionU-Net2D* is the original DDPM (Ho et al., 2020) architecture simplified and only one channel to allow for a user-batch matrix input $\mathbf{X}^0_{u \in b_j}\ \forall b_j \in \mathbf{b}$. Both U-Nets have a time embedding.

Our two Unet architectures are more as proof-of-concepts than theoretically



(a) a user vector with same mean activity as the input data $p(x^T = 1) = E(x^0) = \bar{\mathbf{X}}^0$

(b) a user vector with a coin flip activity $p(x^T = 1) = 0.5$

(c) a user vector full of zero values $p(x^T = 1) = 0$

Figure 2.3: Binomial diffusion on MovieLens100k after 20 epochs. Top row is the Bernouilli forward process and the bottom row is the learned reverse process. Blue is closer to 0; yellow is closer to 1.

grounded architectures. Some elements of the U-Net architecture make it rather impractical, such as the necessary spatial relationships in the matrix/vector and the necessity for an even-sized matrix/vector input for the up-downsizing steps in the Unet. For some datasets, we removed the least popular item from the data altogether, in order to be able to fit an even number of items as a vector / matrix.

2.3 Experimental Setup

Our experimental setup focuses on the classical recommendation task, where the task is to predict items which a user would enjoy / interact with, based on historical interactions (Steck, 2013). For instance, prior models like the MultVAE (Liang et al., 2018) are fed the user history, and tasked to rank items, where each dimension of the input and output correspond to an item – in the case of the MultVAE, the predicted likelihood can be used to rank recommended items.

Given original binary input (feedback of whether or not someone liked / consumed an item), it is a bit harder to argue for a regular forward diffusion process with Gaussian noise. The forward process is either Gaussian or Binomial.

2.3.1 Assumptions

Our experiments make the following set of standard assumptions, following prior work: we assume a Top-K recommendation setup for binary implicit feedback, and evaluate using a strong generalization split. These, and other assumptions are explicitly described below:

Assumption 1 Top-K recommendation: We consider the Top-K recommendation problem, reflected primarily in the evaluation metrics we utilize: Recall@20, Recall@50 and NDCG@100.

Assumption 2 Binary feedback: If a rating is non-binary, we binarize it. We experiment with two datasets: for MovieLens (Harper and Konstan, 2015) and Netflix (Bennett and Lanning, 2007), we convert ratings of 4 and higher to 1, and use 0 otherwise, following prior work (Liang et al., 2018; Shenbin et al., 2020; Ma et al., 2019).

Assumption 3 Missing or negative interactions cannot be easily distinguished. Note that this is assumption is easily flawed, but widepsread in the research literature (Verstrepen et al., 2017). Existing schemes to deal with this typically either (i) adopt heuristics to classify user-item interactions as either missing or negative, and weight such instances appropriately in the loss function (Pan et al., 2008; Hu et al., 2008), or (ii) leverage information about users' exposure to recommendations to model this probabilistically (Liang et al., 2016). We envision that such methods can be straightforwardly extended to general classes of diffusion models as well, providing an interesting avenue for future work.

Assumption 4 In contrast to sequential recommendation (Wang et al., 2019), we do not explicitly consider the order in which items are viewed, an assumption consistent with prior work (Liang et al., 2018; Shenbin et al., 2020; Ma et al., 2019), which RecFusion builds on. For the validation and test sets splits, we randomly sample items independently of item consumption time.

Assumption 5 We filter out users with fewer than five items, and items with fewer than five interactions, as is common practice (Liang et al., 2018; Beel and Brunel, 2019).

Assumption 6 Strong generalization (Marlin, 2004): Users are split into train/-validation and test sets, with the training employing the entire history. For validation and test sets, a partial history is fed to the recommender, with a held-out set being used to evaluate the resulting recommendation.

2.3.2 Baselines

We benchmark our methods against the following *non-neural* baselines: (i) **Random**: Recommendations are generated by uniformly sampling without replacement from the set of items that have been interacted with. (ii) **Popularity**: The

frequency of each item is calculated and subsequently normalized by dividing the individual count by the maximum count among all items. Consequently, every user receives identical recommendations with scores ranging from zero to one. (iii) **SLIM**: Linear model with a sparse item-to-item similarity matrix; solved using a constrained ℓ_1, ℓ_2 regularized optimization problem (Ning and Karypis, 2011). (iv) **EASE**: A variant of SLIM with a closed-form solution, obtained by dropping the non-negativity and ℓ_1 constraint, which simplifies to ridge regression (Steck, 2019).

We also consider the following *neural* baselines: (i) **MultVAE**: Variational autoencoder (Kingma and Welling, 2013) with a multinomial likelihood (Liang et al., 2018). (ii) **RecVAE**: Improves upon the MultVAE with a composite prior, newer architecture and a training schedule which alternates between training the encoder/decoder (Shenbin et al., 2020). (iii) **CODIGEM**: We took the original CODIGEM code and had to to fix some bugs to make it run. Once it ran in the original bare repo, we transferred the modelling code to the RecPack framework (Walker et al., 2022).

2.3.3 Implementation and parameters

We provide a model card in Appendix A.3. We use RecPack (Michiels et al., 2022), a reproducibility framework for our experiments. We reproduce baselines ourselves, given the ambiguity over the aforementioned assumptions in existing literature. We promote a reproducible setup with the above assumptions.

We utilize the following datasets in our experiments (i) MovieLens (Harper and Konstan, 2015) (we use 1M, 25M) and (ii) Netflix (Bennett and Lanning, 2007). Dataset statistics are reported in Appendix A.4. As mentioned before, we evaluate on the test set using the following metrics: Recall@20, Recall@50 and NDCG@100. We report *calibrated recall*, which adjusts for the number of true positive interactions and ensures that optimal recommendations map to a perfect recall value of 1, as is commonly done in previous work (Liang et al., 2018). We train on single NVIDIA V100 GPUs.

For hyperparameter tuning, we use Hyperopt (Bergstra et al., 2013) and its Tree of Parzen Estimators (Bergstra et al., 2011) algorithm and Sparktrial[5] to coordinate GPUs. We use the validation set NDCG@100 to navigate the hyperparameter space. Once the best hyperparameter combination is found, we run the model on the test split (train/val/test – 0.8/0.1/0.1). For MovieLens1M, we bootstrap predictions and run on 10 different splits to obtain error bars on out-of-sample prediction (see Figure 2.4).

2.4 Results

Our results show that between the diffusion models, RecFusion outperforms CODIGEM on two of three datasets. However non-neural baselines, and EASE in particular, outperform both neural and diffusion-based models on all datasets.

[5] `http://hyperopt.github.io/hyperopt/scaleout/spark/`

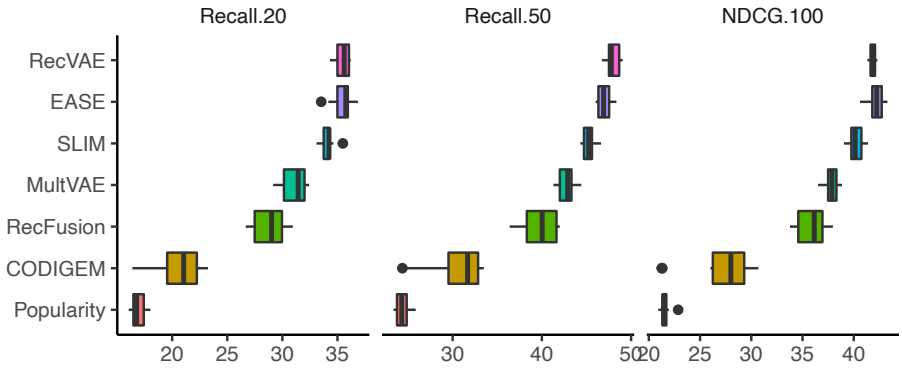


Figure 2.4: Experimental results on the MovieLens1M dataset. All results reproduced by us. Our method is RecFusion. Boxplots show median and IQR over 10 train/test splits.

Non-neural baselines. Across datasets and metrics, the best performance is often obtained by EASE (see Table 2.4 and Table 2.1). EASE even outperforms MultVAE, a popular neural baseline, on most datasets and metrics. This is in line with prior research that demonstrates the efficacy of linear models for recommendation over some neural methods (Steck, 2019; Ferrari Dacrema et al., 2019; Dacrema et al., 2021). Despite this, neural methods make other tasks within recommendation viable (e.g., using user or item metadata), as highlighted in Section 2.5.

Comparing diffusion models and neural methods. From Figure 2.4 and Table 2.1, we observe a consistent trend: MultVAE outperforms both diffusion models, CODIGEM and RecFusion, across all datasets and for all metrics. One reason might be the difference in the number of parameters employed by the two networks: MultVAE uses two three-layer networks, one each for the encoder and decoder, whereas RecFusion employs a single three-layer network, which is reflected in the number of parameters reported in Table 2.6. For MovieLens 1M, we observe that RecVAE outperforms MultVAE and both diffusion models. However, RecVAE uses a somewhat complicated (per-user) prior, along with a complicated training schedule where only the encoder (or decoder) is trained with the decoder (or encoder) frozen. RecFusion employs none of these heuristics.

We highlight that RecFusion outperforms, or is on par with CODIGEM. We keep this time the most popular model in each of VAEs and non-neural classifications in Table 2.5.

Ablation study. In Table 2.2, we perform an ablation study: we start with models that integrate the most diffusion methods and remove elements, one-by-one. Perhaps unsurprisingly for recommendations where linear models dominate, we discover that the most bare-bone (close to linear) diffusion model works best. RecFusionU-Net1D and RecFusionU-Net2D drastically under-perform, even scoring below the Popularity baselines. For RecFusionUNet-2D, this is expected because of the lack of spatial correlations that the model was originally

Table 2.1: Experimental results on the MovieLens25M and Netflix datasets. All results are reproduced by us. Our method in bold.

Dataset	Model	Recall@20	Recall@50	NDCG@100
MovieLens25M	Random	0.13	0.30	0.24
	Popularity	16.63	24.43	19.69
	RecFusion	33.21	45.44	37.31
	CODIGEM	34.05	45.84	37.90
	MultVAE	35.12	48.09	39.12
	EASE	40.02	52.71	43.84
Netflix	Random	0.18	0.32	0.31
	Popularity	11.73	17.48	15.89
	CODIGEM	25.54	33.48	29.08
	RecFusion	29.68	37.63	32.87
	MultVAE	31.61	40.61	35.23
	EASE	36.19	44.49	39.35

Table 2.2: Experimental results for different RecFusion methods on MovieLens1M.

Model	Recall@20	Recall@50	NDCG@100
RecFusionU-Net1D	4.45	7.77	6.99
RecFusionU-Net2D	6.47	9.08	9.03
RecFusionT	14.03	17.80	16.59
RecFusionVar	16.71	24.73	21.63
RecFusionBin	17.59	23.53	21.94
RecFusion	30.91	41.76	37.44

designed for.

Adding typical diffusion elements like time embeddings (RecFusionT), mean/-variance (RecFusionVar) also underperforms compared to the base RecFusion model. We hypothesize that RecFusionBin underperforms due to the noise schedule employed: adding noise to images (256 colors) is more meaningful than adding noise to binary data. We exacerbate this problem by explicitly modelling it as a binomial Markov diffusion process.

Summary of results and discussion. Our results show that existing VAE (MultVAE, RecVAE) and non-neural baselines (EASE, SLIM) outperform more complicated architectures, like diffusion models. RecFusion, however, outperforms the diffusion baseline, CODIGEM, on two of three datasets across all metrics. RecFusion is the simplest form of our approach, with a VAE akin to MultVAE (in terms of mean-variance estimation and loss function). We stress that our proposed method, RecFusion is a simpler and more elegant way to model the recommendation problem than CODIGEM. Our experiments, however, highlight the difficulty of utilizing generative models for real-world problems in which (close to) linear models dominate.

2.5 Related Work

This work should not be confused with diffusion models in social recommendation (e.g., (Wu et al., 2019f; Jin et al., 2020)), an orthogonal field. We briefly review the diffusion and the recommmendation literature.

Diffusion models. Diffusion probabilistic models were first introduced by (Sohl-Dickstein et al., 2015b), where the specific implementation and optimization objectives failed to surpass the state-of-the-art. A few years later, the denoising diffusion model (DDPM) was introduced by (Ho et al., 2020), where the loss function is simplified and the architecture proposed manages to achieve strong state-of-the-art performance. The rich literature that follows would be impossible to summarize in a single paragraph. The most relevant work is denoising diffusion implicit models which changes the parametrization of the sampling to make it deterministic instead of stochastic (Song et al., 2021). Diffusion is first often used in the 2D domain, it can also have a 3D interpretation (Ho et al., 2022; Chen et al., 2023a; Yang et al., 2023; Khader et al., 2023).

Diffusion for recommendation is already present in early work. CODIGEM is probably the first attempt at using diffusion models for recommendation (Walker et al., 2022). They take inspiration from diffusion models and generate recommendations through iterative denoising. Although Diffusion models inspire CODIGEM, it is implemented effectively as a simple hierarchical variational autoencoder. The first reason is that the model does not share weights across timesteps. Also, diffusion models are based on the assumption that the forward process is performed in sufficiently small steps to guarantee that the reverse will have the same functional form (Sohl-Dickstein et al., 2015b; Feller, 1949).

Recommender systems. Non-neural MF methods can solve minimization problems on single user-item rating matrices (see Figure 2.2). But (i) user-item metadata, (ii) time representation, (iii) and controllability / guidance (e.g., a movie recommendation set that must be action-comedy oriented) are harder to model in a closed form or iterative manner (e.g., Gibbs sampler for ALS (Menzen et al., 2021)). This is where neural models can help. Within neural models, probabilistic models and especially Variational Auto Encoders (VAEs) are omnipresent, including MultVAE (Liang et al., 2018) and RecVAE (Shenbin et al., 2020).

Recommendation (together with, arguably, time series and tabular data) is one of only few areas where neural models do not seem to have gained supremacy yet. This has been shown in the settings of general recommendation (Ferrari Dacrema et al., 2019; Jannach et al., 2020; Rendle et al., 2019; Rendle et al., 2022), sparse interactions (Klingler et al., 2022), session-based (Ludewig and Jannach, 2018) and next basket recommendation (Hu et al., 2020; Latifi et al., 2021; Li et al., 2021). In these benchmarks, winning methods are variations of matrix factorization (MF) techniques (SVD++, (i)ALS, EASE (Steck, 2019), and SLIM (Ning and Karypis, 2011)) or even the *most popular* benchmark. Neural models are a popular choice for recommender systems, with early models like AutoRec (Sedhain et al., 2015) or CDAE (Wu et al., 2016) employing autoencoder architectures. Despite limited reproducibility of some neural models,

(Ferrari Dacrema et al., 2019; Dacrema et al., 2021; Li et al., 2021) or the superior performance of non-neural methods in certain settings (Steck, 2019; Ning and Karypis, 2011), e.g., competitions (Jannach et al., 2020), neural methods have comparable or better performance in several settings. Of these, probabilistic methods employing *variational* inference, i.e., variational auto-encoders (VAE) (Kingma and Welling, 2013), like the MultVAE (Liang et al., 2018) or RecVAE (Shenbin et al., 2020) are notable, with the latter being the only neural model successfully reproduced in a large-scale reproducibility study (Ferrari Dacrema et al., 2019; Dacrema et al., 2021).

Contemporaneous work. In April 2023, while the table of results above was being finalized, three papers were published on diffusion for recommendation and one on Bernoulli diffusion (some peer-reviewed). *BSPM* (Jeongwhan et al., 2023) uses score-based models as a testbed for generative models for the recommendation. *DiffuRec* (Li et al., 2023) is the first attempt at diffusion for sequential recommendation and thus not comparable to our non-sequential setup. *DiffRec* is a similar paper to ours on smaller datasets (Wenjie et al., 2023). *DiffRec* corresponds to one of our Recfusion formulations (RecFusionVar) but results on ML1M are significantly (3X) lower compared to our computations or (Sachdeva et al., 2022). This highlights the difficulty of comparisons in the recommendation literature, due to different experimental setups and (at times, unstated) data pre-processing assumptions. We make these assumptions explicit in our chapter and code (see Section 2.3.1). Similarly, it was a challenge to bring the CODIGEM code to work in general (the code does not run as-is) and within our framework in particular (see Section 2.3.2). However, we come fairly close to the original numbers reported in the paper. Finally, *BerDiff* (Chen et al., 2023b) is the first attempt we could find to explicitly model binary data with a Bernoulli Markov diffusion process.[6] *BerDiff* focuses on 2D CT scan and MRI data and thus relies heavily on the Unet architecture. In our chapter, we show theoretically and empirically that we face more of a 1D problem and thus define our own 1D diffusion model for binary data.

2.6 Conclusion

We position this chapter as a first attempt at designing diffusion models for unstructured binary 1D data in the context of recommendation and beyond. With RecFusion, our simple diffusion model (hierarchical VAE) is on par with popular VAE methods. We conjecture that extensions (techniques like composite priors, etc. as in RecVAE (Shenbin et al., 2020)) can further improve performance. We first argue that we need to tackle limitations in our existing implementation and lay out some proposals for future improvements. We can summarize our contributions as follows. First, we show theoretically and empirically that the lack of spatial relationship between users and items is a hindrance to using any image-inspired models, including even a 1D U-Net. We then implemented our

[6] Sohl-Dickstein et al. (2015b) already hinted at diffusion as a 1-dimensional idea as a proof of concept. We could not find the full mathematical derivation or code for it, however.

binary (Bernoulli) Markov process, as a model adapted to the problem at hand.

Broader impact. The image domain sometimes still requires the simplicity of binary settings, like segmentation masks on MRI, CT scans (Chen et al., 2023b; Ma and Wang, 2023) or for conventional object detection techniques (Kirillov et al., 2023). This is potentially fruitful ground for applying our proposed diffusion models for binary 1D data.

2.7 Limitations

Our setup relies on weak (even if common) recommendation setup assumptions. To these assumptions we have to add that the items list is fixed: our model can not account for new items in the catalogue after training. This is a limitation shared with CODIGEM, but also with VAE-based models.

We have yet to test how robust our diffusion models are to a relaxations of these assumptions. Diffusion can also be applied to further domains of recommendation like sequential recommendation with diffusion + RNN, or explicitly model count data input with star ratings instead of binarized feedback.

RecFusion does not yet use further diffusion methods, such as inpainting, guidance (e.g., to predict the user preference distribution or use a prior on movie genre a.k.a controllable recommendations), diffusion on the embedding space (Gao et al., 2023) (in particular, user-item matrix embeddings), or multinomial likelihood to model the dependencies of item feedbacks for a user (Hoogeboom et al., 2021), input masking. We believe these are fruitful areas for future work.

2.8 Upshots for the Personalization Flow

In this chapter, we formulated and trained a diffusion model for recommendation. We answer the research question on two levels. (i) Mathematically, our 1D Bernoulli or Gaussian diffusion processes fit the user-item matrix setting. (ii) RecFusion outperforms other existing diffusion models for recommendation and remains inferior to VAE-based and non-neural models. More work is needed towards beating these baselines and unlocking the diffusion tricks like guidance and inpainting.

Thanks to RecFusion, we can distribute recommendation strips to each user. In the next chapter, we are interested in how these strips look. More precisely, how to illustrate each video in the strip for each type of user in an enticing way. If users can be organized into what movie genres they like the most, this becomes a multilabel classification problem: for each candidate video thumbnails, what is (are) the genre(s) they most relate to?

Reproducibility

To facilitate the reproducibility of the work in this chapter, our code is available at `https://github.com/gabriben/recfusion`.

A Appendices

A.1 The ELBO is also suited for Bernoulli samples

According to the classic definition of the ELBO in (Sohl-Dickstein et al., 2015b) and (Ho et al., 2020), there are no assumptions regarding the distributions p_θ or q_θ. We reproduce here for completeness the derivation from (Sohl-Dickstein et al., 2015b) on why the ELBO satisfies any distribution given Jensen's inequality:

$$\begin{aligned} L &= \int d\mathbf{X}^0 q\left(\mathbf{X}^0\right) \log p\left(\mathbf{X}^0\right) \\ &= \int d\mathbf{X}^0 q\left(\mathbf{X}^0\right) \log \left[\begin{array}{c} \int d\mathbf{X}^{1:T} q\left(\mathbf{X}^{1:T} \mid \mathbf{X}^0\right) \\ p\left(\mathbf{X}^T\right) \prod_{t=1}^T \frac{p\left(\mathbf{X}^{t-1} \mid \mathbf{X}^T\right)}{q\left(\mathbf{X}^T \mid \mathbf{X}^{t-1}\right)} \end{array} \right]. \end{aligned}$$

The latter has a lower bound given Jensen's inequality that also applies to the bernoulli distribution.

$$L \geq \int d\mathbf{X}^{0:T} q\left(\mathbf{X}^{0:T}\right) \log \left[p\left(\mathbf{X}^T\right) \prod_{t=1}^T \frac{p\left(\mathbf{X}^{t-1} \mid \mathbf{X}^T\right)}{q\left(\mathbf{X}^T \mid \mathbf{X}^{t-1}\right)} \right].$$

In practice, the product term is computed with a KL divergence. It can be shown with Fano's inequality (Scarlett and Cevher, 2021) that our cross-entropy loss also aims for a lower bound like KL divergence.

For this assumption regarding p_θ or q_θ to be valid we make sure that the forward steps (i.e. β_t) are small enough, following (Sohl-Dickstein et al., 2015b).

A.2 Proof of why Bernoulli diffusion is multiplicative

Given our Bernoulli diffusion formulation

$$q\left(\mathbf{X}^t \mid \mathbf{X}^{t-1}\right) := \mathcal{B}\left(\mathbf{X}^t; \mathbf{X}^{t-1}\left(1-\beta_t\right) + \tfrac{1}{2}\beta_t\right),$$

we would like to make sure that $q\left(\mathbf{X}^t \mid \mathbf{X}^{t-1}\right)$ can still be sampled at an arbitrary step t in closed form, as with traditional gaussian diffusion (Sohl-Dickstein et al., 2015b). Without loss of generalization – since we sample independently from a Bernoulli distribution – we show that this is true for a single user-item combination. Let x^t be the random variable that represents the t-th forward diffusion step for a specific user-item combination. Then:

$$x^t = \left(1-\beta_t\right) x^{t-1} + \tfrac{1}{2}\beta_t.$$

By substituting x^{t-1}, we get

$$\begin{aligned} x^t &= \left(1-\beta_t\right)\left[\left(1-\beta_{t-1}\right) x_{t-2} + \tfrac{1}{2}\beta_{t-1}\right] + \tfrac{1}{2}\beta_t \\ &= \left(1-\beta_t\right)\left(1-\beta_{t-1}\right) x_{t-2} + \tfrac{1}{2}\left(1-\beta_t\right)\beta_{t-1} + \tfrac{1}{2}\beta_t. \end{aligned}$$

If we keep on substituting the previous diffusion step, we arrive at the original data x_0. By factorizing the above and by induction, it is trivial to show that

$$x^t = \prod_{i=1}^{t} (1-\beta_i)\, x_0 + \frac{1}{2} \sum_{j=1}^{t-1} \left[\prod_{i=j+1}^{t} (1-\beta_i) \right] \beta_j + \frac{1}{2}\beta_t. \tag{2.9}$$

We can actually express the middle term with a common index by using $\beta_j = 1-(1-\beta_j)$.[7] We then obtain a telescoping sum:

$$\begin{aligned}\sum_{j=1}^{t-1} \left[\prod_{i=j+1}^{t} (1-\beta_i) \right] \beta_j &= \sum_{j=1}^{t-1} \left[\prod_{i=j+1}^{t} (1-\beta_i)\,(1-(1-\beta_j)) \right] \\ &= \sum_{j=1}^{t} \left[\prod_{i=j+1}^{t} (1-\beta_i) - \prod_{i=j}^{t} (1-\beta_i) \right] = 1 - \prod_{i=1}^{t} (1-\beta_i)\,.\end{aligned}$$

Substituting this term back into Equation 2.9,

$$x^t = \prod_{i=1}^{t} (1-\beta_i)\, x_0 + \tfrac{1}{2} \left[1 - \prod_{i=1}^{t} (1-\beta_i) \right] + \tfrac{1}{2}\beta_t.$$

Finally, by defining $\alpha_t = 1-\beta_t$ and $\bar{\alpha_t} = \prod_{\tau=1}^{t} \alpha_\tau$, we get

$$x^t = \bar{\alpha}_t x_0 + \frac{1}{2}\left(1-\bar{\alpha}_t\right)\bar{\alpha}_t \tag{2.10}$$

$$q\left(\mathbf{X}^t|\mathbf{X}^0\right) = \mathcal{B}\left(\mathbf{X}^t; \bar{\alpha}_t \mathbf{X}^0 + \frac{1}{2}\left(1-\bar{\alpha}_t\right)\bar{\alpha}_t\right). \tag{2.11}$$

We showed that $\mathbf{X}^t$ can be sampled directly from $\mathbf{X}^0$ in a single Bernoulli sample. ■

A.3 Model card

See
`https://github.com/gabriben/recfusion/blob/master/model_card.md`.

A.4 Descriptive statistics

In Table 2.3, we show counts of users items and interactions on the train, val and test sets. We provide this as an extra step for data preprocessing transparency.

A.5 Results on MovieLens1M as a table

Table 2.5 is the pendant of Figure 2.4 for the MovieLens 1M dataset. We added the Random baseline here, but left it out of the figure for aesthetics.

[7] We borrow this trick from `https://math.stackexchange.com/questions/4467894/does-a-markov-chain-with-gaussian-transitions-px-tx-t-1-mathcal-n-sqrt1`

Table 2.3: Descriptive statistics: Counts of active (non-zero) users and items after preprocessing and under train / val / test (0.8 / 0.1 / 0.1) splitting regime over users.

	No. users			No. items		
Dataset	train	val	test	train	val	test
ML1M	4,832	604	604	3,416	3,158	3,282
ML25M	130,032	16,254	16,255	32,718	24,818	25,597
Netflix	378,389	47,299	47,299	17,769	17,761	17,759

Table 2.4: Descriptive statistics: Counts of active (non-zero) interactions after preprocessing and under train / val / test (0.8 / 0.1 / 0.1) splitting regime over users.

	No. interactions		
Dataset	train	val	test
ML1M	798,608	76,513	84,772
ML25M	19,924,515	1,999,297	2,030,221
Netflix	80,418,808	8,011,940	8,060,214

A.6 Number of parameters

One of our arguments is that our model is more efficient than existing neural baselines (see Table 2.6).

Table 2.5: Experimental results on the MovieLens1M dataset. All results reproduced by us over 10 train/test splits, we report median results. Our method in bold.

Type	Model	Recall@20	Recall@50	NDCG@100
Baselines	Random	0.87	1.71	1.73
	Popularity	16.77	24.30	21.44
Diffusion	CODIGEM	21.04	31.67	28.00
	RecFusion	29.02	40.03	36.14
VAE	MultVAE	31.43	42.89	37.87
	RecVAE	35.61	47.79	41.81
Non-neural	SLIM	34.23	45.25	40.14
	EASE	35.76	46.92	42.24

Table 2.6: Number of parameters for different neural architectures on the ML1M dataset.

MultVAE	CODIGEM	RecFusion
446,421,6	560,388,6	141,021,8

Metrics as Losses

After a recommendation model outputs video titles in the first step of the personalization flow, we still have to represent each video via an image.

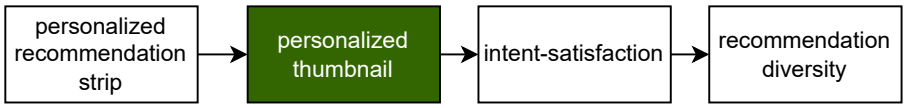


Figure 3.1: The second step of the personalization flow.

For that we revisit a classical classification problem. Namely, we take video recommendations generated via a diffusion model and perform multilabel classification on thumbnails of these videos with our custom loss sigmoidF1. Through that process, we hope to provide personalized thumbnails of the video for each user. There is little work beyond (informal or industry) blog posts on how to personalize thumbnails on recommendation platforms. Under the pretense of handling this in an algorithmic way, we point out the lack of loss functions speficially designed for the multilabel classification problem in general. This part of the personalization flow is focused on the following question:

RQ2: Is there a way we can generate personalized thumbnails for each item on a streaming platform?

We tackle this question, by simplifying the problem first. Let's assume each user has a preferred movie genre. Then we would like to serve each user with that preferred genre. Given a set of candidate thumbnails, how can we determine which genre(s) they relate to the most? At inference time, we try to optimize for rather comprehensive metrics like F1: it can balance true positives, false positives and false negatives. At training time, we use classical multilabel classification, but propose our own loss function, sigmoidF1. This loss function is a surrogate of the F1 classification metric. As such, we propose a metric as a loss.

3.1 Introduction

Many real-world classification problems are challenging because of unclear (or overlapping) class-boundaries, subjectivity issues, and disagreement between

This chapter was published in the Transactions on Machine Learning Research (TMLR) under the title "sigmoidF1: A Smooth F1 Score Surrogate Loss for Multilabel Classification" (Bénédict et al., 2022).

annotators. Multilabel learning tasks are common, e.g., document and text classification often deal with multilabel problems (Hull, 1994; Bruno et al., 2013; Yang, 2004; Blosseville et al., 1992), as do query classification (Kang and Kim, 2003; Manning et al., 2008), image classification (Shen et al., 2017; Xiao et al., 2010) and product classification (Amoualian et al., 2020). Existing optimization frameworks typically split the task into known problems and sum over existing losses $\sum \mathcal{L}_{\text{MC}}$, with $\mathcal{L}_{\text{MC}}$ any multiclass classification loss – oftentimes variations of the cross-entropy or logistic loss. Wydmuch et al. (2018) define these frameworks as *multilabel reduction* techniques; Menon et al. (2019) put an emphasis on two: One-Versus-All (OVA)[1] and Pick-All-Labels (PAL). For example, if C is the number of possible classes, OVA and PAL reformulate the multilabel problem to C binary classification and C multiclass classification problems, respectively (see Section 3.2.3). These methods assume that, for one example, label probabilities (a.k.a. Bayes Optimal Classifier (Dembczyński et al., 2010)) are marginally independent of other label probabilities. Menon et al. (2019) show mathematically and empirically that reduction methods (OVA and PAL) can optimize for precision or recall, but not for both precision and recall at once. More generally, a shortcoming shared by OVA and PAL is their reliance on the binary or multiclass classification setting and the lack of a pure multilabel approach – inspired by binary classification literature (see most recently (Gai et al., 2019) and their F1 surrogate loss functions on 3-layer neural networks). We are not aware of a metric surrogate loss function that deals with multilabel classification in a modern deep learning setting in a single task. Figure 3.2 illustrates our approach with a concrete example of classifying a movie poster into movie genres with a single loss function: *sigmoidF1*.

Proposed solution to multilabel problems. We propose a loss function $\mathcal{L}_{\widetilde{F1}}$ that (i) naturally approximates the macro $F1$ classification metric (see Table 3.3), (ii) estimates label probabilities and label counts (see Eq. 3.7), and (iii) is decomposable for stochastic gradient descent at training time (see Section 3.4.1 and Figure 3.3). Our proposed solution is to minimize a surrogate of the F1 metric as a loss. Strictly speaking, we minimize $1 - \widetilde{F1}$, where $\widetilde{F1}$ is a smooth version of $F1$. Using a metric as a loss function is unpopular for metrics that require a form of thresholding (e.g., counting the number of true positives), as minimizing a step loss function (a.k.a. 0-1 loss) is intractable. The soft margin for support vector machines is an early example, where the intractability of the direct 0-1 loss optimization is overcome with the hinge loss (Cortes and Vapnik, 1995). We resolve this by approximating the step function by a sigmoid curve (see Figure 3.2).

Main contributions. We introduce *sigmoidF1*, an F1 score surrogate, with a sigmoid function acting as a surrogate thresholding step function. *sigmoidF1* allows for the use of the F1 metric that simultaneously optimizes for label

[1] This was already described in (Dembczyński et al., 2010) and further formalized in (Wydmuch et al., 2018).

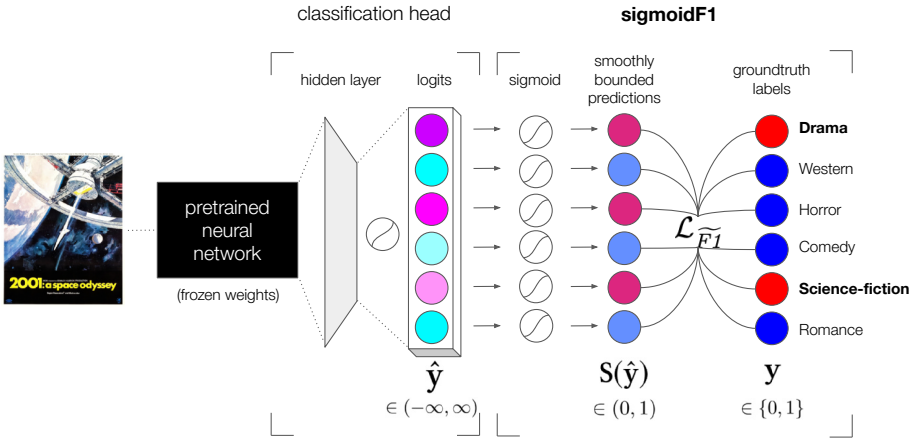


Figure 3.2: Experimental setup for *sigmoidF1* as a loss function for **multilabel classification**. Here, a movie poster image is fed to a pre-trained network with a custom classification head that outputs logits (i.e., unbounded values) for each class (i.e., movie genre). At training time, a sigmoid function forces logits towards either -1 or 1, respectively negative and positive predictions (illustrated by the darker colors). Confusion matrix metrics and macro F1 can then subsequently be computed. Here, $S(\hat{y}_{horror})$ is close to 1, but the ground truth data claims that *2001: a space odyssey* is not a horror movie; this approximately corresponds to a false positive. Note that $\mathcal{L}_{\widetilde{F1}}$ is computed over a whole batch at training time as a macro measure with the formulas in Sections 3.4.3 and 3.4.4. With this setup, one can optimize directly for the metric of interest at training time. Our image and text classification tasks below show improved results when compared to existing losses.

prediction and label counts in a single task. *sigmoidF1* is benchmarked against loss functions commonly used in multilabel learning and other existing multilabel models. We show that our custom losses improve predictions over current solutions on several different metrics, across text and image classification tasks. PyTorch and TensorFlow source code are made available.[2]

3.2 Background

We use a traditional statistical framework as a guideline for multilabel classification methods (Tukey, 1977). We distinguish the desired theoretical statistic (the **estimand**), its functional form (the **estimator**) and its approximation (the **estimate**); estimates can be benchmarked with **metrics**. We show how multilabel reduction estimators tend to reformulate the estimand and treat labels as marginally independent. For example, by treating a multilabel problem as a succession of binary classification tasks. However, with a proper estimator, it is

[2] `https://github.com/gabriben/metrics-as-losses`

possible to directly model the estimand. If F1 score is indeed the statistic of interest (i.e. estimand), our proposed loss function, *sigmoidF1*, accommodates for the true estimand.

We define a learning algorithm $\mathcal{F}$ (i.e., a class of estimators) that maps inputs to outputs given a set of hyperparameters $\mathcal{F}(\cdot;\Theta) : \mathcal{X} \to \mathcal{Y}$. We consider a particular case, with the input vector $\mathbf{x} = \{x_1, \ldots, x_n\}$ and each observation is assigned k labels (one or more) $\mathbf{l} = \{l_1, \ldots, l_C\}$ out of a set of C classes. y_i^j are binary variables, indicating presence of a label for each observation i and class j. Together, they form the matrix output $\mathbf{Y}$. This is our multilabel setting. Note that multiclass classification can be considered as an instance of multilabel classification, where a single label is attributed to an example.

3.2.1 Estimand and definition of the risk

We distinguish between two scenarios: the *multiclass* and the *multilabel* scenario. In the multiclass scenario, a single example is attributed one class label (e.g., classification of an animal on a picture). In the multilabel scenario, a single example can be assigned more than one class label (e.g., movie genres). We focus on the latter. For a particular set of inputs $\mathbf{x}$ (e.g., movie posters) and outputs $\mathbf{Y}$ (e.g., movie genre(s)), the risk formulation is the same as in (Menon et al., 2019):

$$R_{\mathrm{ML}}(\mathcal{F}) = \mathbb{E}_{(\mathbf{x},\mathbf{Y})} \left[\mathcal{L}_{\mathrm{ML}}(\mathbf{Y}, \mathcal{F}(\mathbf{x})) \right]. \tag{3.1}$$

The learning algorithm $\mathcal{F}$ is the estimand, the theoretical statistic. For one item x_i, the theoretical risk defines how close the estimand can get to that deterministic output vector $\mathbf{y}_i$. In practice, statistical models do output probabilities $\hat{\mathbf{y}}_i$ via an estimator and its estimate (also called propensities or suitabilities (Menon et al., 2019)). The solution to that stochastic-deterministic incompatibility is either to convert the estimator to a deterministic measure via decision thresholds (e.g., traditional cross-entropy loss), or to treat the estimand as a stochastic measure (our *sigmoidF1* loss proposal).

3.2.2 Estimator: The functional form

The estimator $f \in \mathcal{F}$ is any minimizer of the risk R_{ML}. Predicting multiple labels per example comes with the assumption that labels are non mutually-exclusive.

Definition. *The multilabel estimator of y_i^j is dependent on the input and other ground truth labels for that example, $\hat{y}_i^j = f(x, y_i^1, \ldots, y_i^{j-1}) = P(y_i^j = 1 | y_i^1, \ldots, y_i^{j-1}, x_i)$.*

By proposing this general formulation, we entrench that mutually-inclusive characteristic in the estimator. Contrary to (Menon et al., 2019), our definition above models interdependence between labels and deals with thresholding for the estimate at training time for free. Waegeman et al. (2014) show that an estimator of an F-score can be used at inference time for multilabel classification, when using probabilistic models where parameter estimation is possible (e.g.,

decision trees, probabilistic classifier chains). When it is not possible, we resort to defining a loss function.

3.2.3 Estimate: Approximation via a loss function

Most of the literature on multilabel classification can be characterized as multilabel reductions (Menon et al., 2019): an approximation of the original multilabel problem via a loss function $\mathcal{L}(\mathbf{y}_i, f)$. It can take different forms.

One-versus-all (OVA) is a reformulation of the multilabel classification task to a sequence of C binary classifications $(f^1, \ldots, f^C)$, with C the number of classes, $\mathcal{L}_{\text{OVA}}(\mathbf{y}_i, f) = \sum_{c=1}^{C} \mathcal{L}_{\text{BC}}\left(y_i^c, f^c\right)$ where $\mathcal{L}_{\text{BC}}$ is a binary classification loss (binary relevance (Brinker et al., 2006; Tsoumakas and Katakis, 2007; Dembczyński et al., 2010)), most often logistic loss. Minimizing binary cross-entropy is equivalent to maximizing for log-likelihood Bishop, 2007, §4.3.4.

Pick-all-labels (PAL) gives the loss function $\mathcal{L}_{\text{PAL}}(\mathbf{y}_i, f) = \sum_{c=1}^{C} y_i^c \cdot \mathcal{L}_{\text{MC}}(y_i^c, f)$, with $\mathcal{L}_{\text{MC}}$ a multiclass loss (e.g., softmax cross-entropy). In this formulation, each example $(x_i, \mathbf{y}_i)$ is converted to a multiclass framework, with one observation per positive label. The sum of inherently multiclass losses is used to represent the multilabel estimand.

Multilabel reduction methods are characterized by their way of reformulating the estimand, the resulting estimator, and the estimate. This allows the use of existing losses: logistic loss (for binary classification formulations), sigmoid or softmax cross-entropy loss (for multiclass formulations). These reductions imply a reformulation of the estimator (a.k.a. Bayes Optimal) as follows:

$$\hat{y}_i^j = f(x) = P(y_i^j = 1 | x_i). \tag{3.2}$$

Contrary to our definition of the original multilabel estimator (Section 3.2.2), marginal independence of label propensities is assumed. In other words, the loss function becomes any monotone transformation of the marginal label probabilities $P(y_i^j = 1|x)$ (Dembczyński et al., 2010; Koyejo et al., 2015; Wu and Zhou, 2017). In literature reviews, the multilabel reductions OVA and PAL have been coined as *fit-data-to-algorithm*, as opposed to *fit-algorithm-to-data* (Zhang and Zhou, 2014), originally framed as *problem transformation* and *algorithm adaptation* respectively (Tsoumakas and Katakis, 2007)). For the purpose of our narrative, we propose the following formalization of this dichotomy: *fit-data-to-algorithm* formulates an additive loss over existing losses $\sum \mathcal{L}_{\mathcal{C}}$, with $\mathcal{L}_{\mathcal{C}}$ any classification loss and oftentimes a sum over all classes. This can be contrasted with *fit-algorithm-to-data*, where a custom loss $\mathcal{L}^*$ is built for the multilabel task. We further discuss this in Section 3.3 and Table 3.1.

3.2.4 Metrics: Evaluation at inference time

There is consensus on the usefulness of a confusion matrix and ranking metrics to evaluate multilabel classification models at inference time (Koyejo et al., 2015; Behera et al., 2019; Wu and Zhou, 2017). Confusion matrix metrics come with caveats: most of these measures (i) require hard thresholding, which makes them

non-differentiable for stochastic gradient descent; (ii) they are very sensitive to the number top labels to include k (Chen et al., 2006); and (iii) they require aggregation choices to be made in terms of micro/macro/weighted metrics. Common confusion matrix metrics are Precision, Recall, F1-score or one-error-loss; see (Wu and Zhou, 2017) for others.

3.2.5 Multilabel estimate: F1 metric as a loss

A model's out-of-sample accuracy is commonly measured on metrics such as AUROC, F1 score, etc. These reflect an objective catered towards evaluating the model over an entire ranking. Due to the lack of differentiability, these metrics cannot be directly used as loss functions at training time (in-sample). (Eban et al., 2017) propose a theoretical framework for deriving decomposable surrogates to some of these metrics. We propose our own decomposable surrogates tailored for multilabel classification (see Appendix B.1).

In a typical machine learning classification task, ground truth binary labels are compared to a probabilistic measure (or a reversible transformation of a probabilistic measure such as a sigmoid or a softmax function) (Bishop, 2007). If the number n_i of labels to be predicted per example is known a priori, it is natural at training time to assign the top_{n_i} predictions to that example (Lapin et al., 2016; Lapin et al., 2015). If the number of labels per example is not known a priori, the question remains at both training and at inference time as to how to decide on the number of labels to assign to each example. This is generally done via a *decision threshold*, that can be set globally for all examples (Lipton et al., 2014). This threshold can optimize for specificity or sensitivity (Chen et al., 2006) – for per-class thresholding see (Chu and Guo, 2017). In Section 3.4, we propose an approach where this threshold is implicitly defined at training time, by using a loss function that penalizes explicitly for wrong label counts and fits to the original estimand in Definition 3.2.2: the F1 metric. In Section 3.4, we show how F_1 is formulated into a surrogate loss $\mathcal{L}_{\widetilde{F1}}$. Our contribution is thus in the continuation of the *fit-algorithm-to-data* trend, because we propose a custom loss function. That loss function is also the first to directly approximate the F1 score with non-divergent estimates (see Sections 3.4.1 and 3.4.2 on *boundedness*).

3.3 Related Work

In Section 3.2.3, we mentioned how existing solutions for multilabel tasks can be divided into *fit-data-to-algorithm* solutions, which map multilabel problems to a known problem formulation like multiclass classification, and *fit-algorithm-to-data* solutions, which adapt existing classification algorithms to the problem at hand (Madjarov et al., 2012). In most of this work, the term *multilabel classification* excludes *extreme* (tens of thousands of labels) (e.g., Jernite et al., 2017; Agrawal et al., 2013; Jain et al., 2019), *hierarchical* (parent and children labels) (e.g., Lehmann et al., 2015; Yang et al., 2019; Howard and Ruder, 2018) or *multiclass* (single label per example) subfields. These subfields call for their

own solutions, including label embeddings (Bhatia et al., 2015) or negative mining (Reddi et al., 2019) for the *extreme* usecase.

Fit-data-to-algorithm. In fit-data-to-algorithm solutions, cross-entropy losses (Fisher, 1912; Good, 1952) are used at training time and thresholding is done at inference time to determine how many labels should be assigned to an instance. This has also been called multilabel reduction (Menon et al., 2019) and differs from multiclass-to-binary classifications (Zhang, 2004; Tewari and Bartlett, 2005; Ramaswamy et al., 2014). We can further distinguish between One-versus-all (OVA) and Pick-all-labels (PAL) solutions (Menon et al., 2019) (see Section 3.2). In OVA, one reduces the classification problem to independent binary classifications (Brinker et al., 2006; Tsoumakas and Katakis, 2007; Dembczyński et al., 2010; Wydmuch et al., 2018). In PAL, one reformulates the task to independent multiclass classifications (Boutell et al., 2004; Jernite et al., 2017; Joulin et al., 2017). The *label powerset* approach considers each set of labels as a class (Boutell et al., 2004). In Pick-One-Label (POL), a single multiclass example is created by randomly sampling a positive label (Joulin et al., 2017; Jernite et al., 2017). Alternatively, *ranking by pairwise comparison* is a solution where the dataset is duplicated for each possible label pair. Each duplicated dataset has therefore two classes and only contains instances that have at least one of the labels in the label pair. Different ranking methods exist (Zhang and Zhou, 2006; Hüllermeier et al., 2008; Loza Mencia and Furnkranz, 2008). Ranking loss has been shown to optimize for two Learning To Rank metrics (Chen et al., 2009). More recently, hierarchical datasets such as DBpedia (Lehmann et al., 2015) are used to fine-tune BERT-based models (Yang et al., 2019; Zaheer et al., 2020); the latter publications use cross-entropy to predict the labels.

Table 3.1: *SigmoidF1* and related loss formulations ordered by publication date. The solution column refers to our proposed formalization of the literature review on how to conduct multilabel classification: *D2A* refers to *fit-data-to-algorithm* (sum over existing or cross-entropy-like, CE-like, classification losses $\sum \mathcal{L}_C$) and *A2D* refers to *fit-algorithm-to-data* (custom loss $\mathcal{L}^*$).

Method	Solution	Model type	Context
ACE (Fisher, 1912)	*D2A*	Any	Any
rankingLoss (Zhang and Zhou, 2006)	*D2A*	Any	Any
MFC (Huang et al., 2015)	–	Gaussian mixtures	Mispronunciation detection
optLosses (Eban et al., 2017)	*A2D*	Any	Any
focalLoss (Lin et al., 2017)	*D2A*	Neural net	Imbalanced-multiclass
deepF (Decubber et al., 2018)	*A2D*	Neural net	Multilabel
softF1 (Chang et al., 2019)	*A2D*	Neural net	Multilabel
ASL (Baruch et al., 2020)	*D2A*	Neural net	Multilabel
RS@k (Patel et al., 2022)	*A2D*	Neural net	Similarity
polyLoss (Leng et al., 2022)	*A2D*	Neural net	Imbalanced-multiclass, ...
sigmoidF1 [ours]	*A2D*	Neural net	Multilabel

Method	Implementation	Surrogated metric	Modality
ACE (Fisher, 1912)	CE-like	–	Any
rankingLoss (Zhang and Zhou, 2006)	pair-rank	–	tabular
MFC (Huang et al., 2015)	sigmoid	F_1	Text
optLosses (Eban et al., 2017)	–	F_1	Theoretical
focalLoss (Lin et al., 2017)	CE-like	–	Image
deepF (Decubber et al., 2018)	CE-like	F_1	Image
softF1 (Chang et al., 2019)	Unbounded	F_1	Image
ASL (Baruch et al., 2020)	CE-like	–	Image
RS@k (Patel et al., 2022)	sigmoid	Recall	Image
polyLoss (Leng et al., 2022)	CE-like	–	Image
sigmoidF1 [ours]	sigmoid	F_1	Text & Image

Fit-algorithm-to-data. In fit-algorithm-to-data solutions, elements of the learning algorithm are changed (e.g., the back propagation procedure). Before focusing on the multilabel case, the multiclass literature has some examples of F1 surrogate loss functions in particular: in the context of SVMs, via pseudo linear functions (Narasimhan et al., 2015) or by learning a feasible confusion matrix (Narasimhan et al., 2015); in the context of deep networks, by learning the surrogate loss function via a dedicated neural network in the binary classification case (Grabocka et al., 2019), by optimizing performance measures composed of true positive and true negative rates (Sanyal et al., 2018) or via empirical utility maximization of F1 on 3-layer neural networks (Gai et al., 2019). Early representatives of multilabel fit-algorithm-to-data solutions stem from heterogenous domains of machine learning. MultiLabel k-Nearest Neighbors (Zhang and Zhou, 2007), MultiLabel Decision Tree (Clare and King, 2001), Ranking Support Vector Machine (SVM) (Elisseeff and Weston, 2001) and backpropagation for multiLabel learning with a ranking loss (Zhang and Zhou, 2006). More recently, the idea of multi-task learning for *label prediction* and *label count prediction* was introduced ML_{NET}, Du et al., 2019; Li et al., 2017; Wu et al., 2019a. The literature has been clearly hinting at the usefulness of a single task loss function that approximates a metric. A formulation similar to our loss *unboundedF1* was proposed in an unpublished blog post, which was referred to as *softF1* (Chang et al., 2019). A similar proposal was to use the hinge loss as a decomposable surrogate for confusion matrix entries for binary classification (Eban et al., 2017). Outside of the context of neural networks, the *Maximum F1-score criterion* for automatic mispronunciation detection was proposed as an objective function to a Gaussian Mixture Model-hidden Markov model (GMM-HMM) (Huang et al., 2015). A recent paper used recall as a loss function for image similarity (Patel et al., 2022). In parallel, there is a growing consensus that the original cross-entropy loss (*fit-data-to-algorithm*) cannot solve all our problems. A variation of the cross-entropy loss adapted to multilabel classification has been proposed (Baruch et al., 2020; Wu et al., 2019a); it extends the multiclass sparse class representation setting (Lin et al., 2017; Leng et al., 2022). In the ranking domain, LambdaLoss has been proposed to optimize directly for the lambdaRank metric (Wang et al., 2018). In the theoretical space, (Eban et al., 2017) have proposed a generic framework for decomposable metrics, including $F1$ as a theoretical fractional linear program. Table 3.1 illustrates how *sigmoidF1* differs from the methods listed in this paragraph.

An important limitation shared by existing *fit-data-to-algorithm* and *fit-algorithm-to-data* approaches is the lack of a unified loss framework that deals with multilabel classification and can approximate a metric of interest. *sigmoidF1* computes an F1 surrogate loss over the aggregation of examples in a batch at training time.

3.4 Method

We introduce our approach for multilabel problems, with a smoothed confusion matrix metric as a loss (the original confusion matrix metrics rely on step functions and are therefore intractable, see for example the blue step function in Figure 3.3). We first briefly define our learning setting and define the confusion matrix metrics in this setting more formally.

We use the binary classification setting (two classes) to simplify notation, without loss of generalization to the multilabel case. In a typical binary classification problem with the label vector $\mathbf{y} = \{y_1, \ldots, y_n\}$, predictions are probabilistic and it is necessary to define a threshold t, at which a prediction is binarized. With $\mathbb{1}$ as an indicator function, $\mathbf{y}^+ = \sum \mathbb{1}_{\hat{\mathbf{y}} \geq t}$, $\mathbf{y}^- = \sum \mathbb{1}_{\hat{\mathbf{y}} < t}$ are thus the count of positive and negative predictions at threshold t. Let *tp*, *fp*, *fn*, *tn* be number of true positives, false positives, false negatives and true negatives respectively:

$$\begin{aligned} tp &= \sum \mathbb{1}_{\hat{\mathbf{y}} \geq t} \odot \mathbf{y} \quad fp = \sum \mathbb{1}_{\hat{\mathbf{y}} \geq t} \odot (\mathbb{1} - \mathbf{y}) \\ fn &= \sum \mathbb{1}_{\hat{\mathbf{y}} < t} \odot \mathbf{y} \quad tn = \sum \mathbb{1}_{\hat{\mathbf{y}} < t} \odot (\mathbb{1} - \mathbf{y}), \end{aligned} \tag{3.3}$$

with $\odot$ the component-wise multiplication sign. For simplicity, in the formulation above and the ones that follow scores are calculated for a single class, therefore the sum is implicitly over all examples $\sum_i$. This applies to the binary classification problem but also to our multilabel setting, when micro metrics are calculated (i.e., compute the metric value for each class, and then averaged over all classes). In the multilabel setting $\mathbf{y}$ can be substituted by $\mathbf{y}^j$ for each class j. Note that vectors could be trivially substituted by matrices ($\mathbf{Y}$) in Eq. 3.3 to obtain the macro formulation. Given the four confusion matrix quadrants, we can generate further metrics like precision and recall (see Table 3.5 in Appendix B.1). However, none of these metrics are decomposable due to the hard thresholding, which is, in effect, a step function (see Figure 3.3).

Next, we define desirable properties for decomposable thresholding, unbounded confusion matrix entries, and a sigmoid transformation that renders confusion matrix entries decomposable. Finally, we focus on a smooth F1 score.

3.4.1 Desirable properties of decomposable thresholding

We define desirable properties for a decomposable sign function $f(u)$ as a surrogate of the above indicator function $\mathbb{1}_{\hat{\mathbf{y}} < t}$.

Property 1. *Boundedness:* $|f(u)| < M$*, where* M *is an upper and lower bound.*

The ground truth $\mathbf{y}$ is bounded between $[0, 1]$ and thus it must be compared to a bounded prediction $\hat{\mathbf{y}}$, preferably bounded by $[0, 1]$, to avoid further scaling.

Property 2. *Saturation:* $\int_s^\infty f^{-1}(u) = \int_{-\infty}^{-s} f(u) = \epsilon$*, with* ϵ *a number close to zero and* s *a saturation bound.*

For the surrogate to be a proper sign function substitute, it is important to often return values close to 1 or 0. Saturation is defined in the context of

neural network activation functions and refers to the propensity of iterative backpropagation to progressively lead to values very close to 0 or 1 after a long enough training period. Our aim is to reach that convergence quickly in order to force decisions towards 0 or 1 in order to be comparable to a step function. This highlights a tension: the sigmoid function should contrast outputs towards 0 or 1 but should not be too saturated, in order for the derivative at point u to be non-null and information to flow back to the network (Krizhevsky et al., 2017).

Property 3. *Dynamic Gradient:* $f'(u) \gg 0 \quad \forall\ u \in [-s, s]$*, where* s *is the saturation bound.*

Inside the saturation bounds $[-s, s]$, the derivative should be significantly higher than zero in order to facilitate stochastic gradient descent and backpropagation. Note that the upper and lower limits of $f(u)$ are interchangeably $[-1, 1]$ or $[0, 1]$ in this chapter and in the literature. The conditions above still apply after linear transformation. Next, we show how our formalization of an unbounded F1 surrogate would not fulfill these properties and how our proposition of a smooth bounded alternative does.

3.4.2 Unbounded confusion matrix entries

A first trivial remedy to allow for derivation of the sign function $f(u)$, is to define *unbounded* confusion matrix entries by retaining the original logits (scores) when counting true positives, false negatives, etc. Countrary to the original confusion matrix definition in Eq. 3.3, $\overline{tp}$, $\overline{fp}$, $\overline{fn}$ and $\overline{tn}$ are not natural numbers anymore:

$$\begin{aligned} \overline{tp} &= \sum \hat{\mathbf{y}} \odot \mathbf{y} & \overline{fp} &= \sum \hat{\mathbf{y}} \odot (\mathbb{1} - \mathbf{y}) \\ \overline{fn} &= \sum (\mathbb{1} - \hat{\mathbf{y}}) \odot \mathbf{y} & \overline{tn} &= \sum (\mathbb{1} - \hat{\mathbf{y}}) \odot (\mathbb{1} - \mathbf{y}), \end{aligned} \tag{3.4}$$

where *tp*, *fp*, *fn* and *tn* are now replaced by rough surrogates. The disadvantages are that the desirable properties mentioned above are not fulfilled, namely (i) $\hat{\mathbf{y}}$ is unbounded and thus certain examples can have over-proportional effects on the loss; (ii) it is non-saturated; while non-saturation is desirable for activation functions (Krizhevsky et al., 2017), here it would be desirable to tend towards saturation (i.e., tend to values close to 0 or 1, so as to give the most accurate predictions at any thresholding values at inference time); and (iii) the gradient of that linear function is 1 and therefore backpropagation will not learn depending on different inputs at this stage of the loss function. However, this method has the advantage of resulting in a linear loss function that avoids the concept of thresholding altogether and is trivial to decompose for stochastic gradient descent.

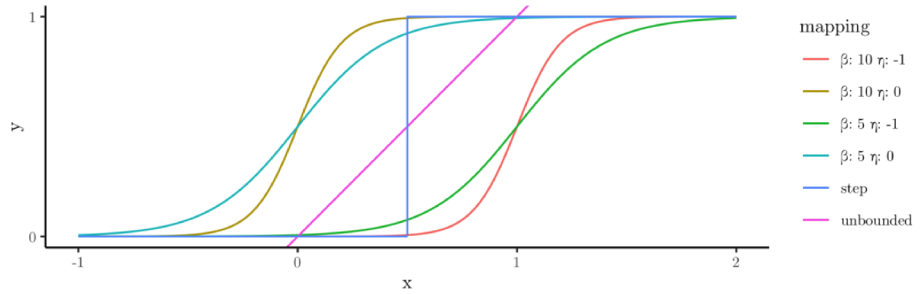


Figure 3.3: Different thresholding regimes: the step function (original *F1* metric) is not decomposable, the linear function is unbounded ($\mathcal{L}_{\overline{F1}}$) and tends to produce divergent gradients, whereas the sigmoid function ($\mathcal{L}_{\widetilde{F1}}$) is bounded and allows for differentiation due to its smooth curvature, tunable at different parametrizations.

3.4.3 Smooth confusion matrix entries

We propose a sigmoid-based transformation of the confusion matrix that renders its entries decomposable and fulfills the three desirable properties above:

$$\begin{aligned} \widetilde{tp} &= \sum \mathbf{S}(\hat{\mathbf{y}}) \odot \mathbf{y} \qquad \sim \widetilde{fp} = \sum \mathbf{S}(\hat{\mathbf{y}}) \odot (\mathbb{1} - \mathbf{y}) \\ \widetilde{fn} &= \sum (\mathbb{1} - \mathbf{S}(\hat{\mathbf{y}})) \odot \mathbf{y} \sim \widetilde{tn} = \sum (\mathbb{1} - \mathbf{S}(\hat{\mathbf{y}})) \odot (\mathbb{1} - \mathbf{y}), \end{aligned} \tag{3.5}$$

with $\mathbf{S}(\cdot)$ the vectorial form of the sigmoid function $S(\cdot)$:

$$S(u; \beta, \eta) = \frac{1}{1 + \exp(-\beta(u + \eta))}, \tag{3.6}$$

with β and η tunable parameters for slope and offset, respectively. Higher β results in steeper slope at the center of the sigmoid and thus more stringent thresholding. At its extreme, $\lim_{\beta \to \infty} S(u; \beta, \eta)$ corresponds to the step function used in Eq. 3.3. Note that negative values of β geometrically reflect the sigmoid function across the horizontal line at 0.5 and thus invert predictions. These smooth confusion matrix entries allow us to build any related metric (see Table 3.5 in Appendix B.1). Furthermore, the surrogate entries are decomposable, bounded, saturated and have a dynamic gradient.

3.4.4 Smooth macro F1 scores

F1 scores can be calculated on a macro and micro level. Macro-averaging regards all classes as equally important, whereas micro-averaging reflects within-class frequency. *unboundedF1* and *sigmoidF1* below are thought of as macro scores (aggregated over all classes). These scores require a high enough number of representatives in the four confusion matrix quadrants to learn from batch to batch. Ideally, each training epoch would have only one batch, so as to have the

most representatives. Following Eq. 3.4, it is possible to define an *unbounded F1* score:

$$\mathcal{L}_{\overline{F1}} = 1 - \overline{F1}, \quad \text{where} \quad \overline{F1} = \frac{2\overline{tp}}{2\overline{tp} + \overline{fn} + \overline{fp}}. \tag{3.7}$$

While this alternative abstracts the thresholding away, which is convenient for fine-tuning purposes, it does not fulfill the desirable properties of a binarization threshold surrogate (see Section 3.4.2). *unboundedF1* will be used to benchmark against our proposed *sigmoidF1* loss. Given the definitions of smooth confusion matrix metrics above, we can now write $\mathcal{L}_{\widetilde{F1}}$:

$$\mathcal{L}_{\widetilde{F1}} = 1 - \widetilde{F1}, \quad \text{where} \quad \widetilde{F1} = \frac{2\widetilde{tp}}{2\widetilde{tp} + \widetilde{fn} + \widetilde{fp}}. \tag{3.8}$$

sigmoidF1 is particularly suited for the multilabel setting because it is a proper hard thresholding surrogate as defined in the previous sections and because it contains a significant amount of information about label prediction accuracy: $\widetilde{tp}$, $\widetilde{fn}$ and $\widetilde{fp}$ are indicative of the number of predicted labels in each category of the confusion matrix but also contain a notion of certainty, given that they are rational numbers. The built in sigmoid function ensures that certainty increases along training epochs, as outlined by Property 2. Finally, as the harmonic mean of precision and recall (a property of F1 in general), it weighs in both relevance metrics.

In the next section, we implement Eq. 3.8 in PyTorch and TensorFlow as a custom loss as follows:

```
# with y the ground truth and z the outcome of the last layer
sig = 1 / (1 + exp(- β * (z + η)))
tp = sum(sig * y, dim=0)
fp = sum(sig * (1 - y), dim=0)
fn = sum((1 - sig) * y, dim=0)
sigmoid_f1 = 2*tp / (2*tp + fn + fp + 1e-16)
minimize(1 - sigmoid_f1)
```

The pseudocode above illustrates the elementwise multiplication of matrices $\mathbf{S}(\hat{\mathbf{y}})$ and $\hat{\mathbf{y}}$ over all examples in the batch and all possible classes.

3.5 Experimental Setup

We test multilabel learning using our proposed *sigmoidF1* loss function on four datasets across different modalities (image and text). For each modality we take a state-of-the-art model that generates an embedding layer and append a sigmoid activation and different losses. Multilabel deep learning is usually implemented with sigmoid binary cross-entropy directly on the last neural layer (a simplification of the OVA and PAL reductions). We follow this approach for our experiments (e.g., in (large) language models (Zaheer et al., 2020; Devlin et al., 2019)). Some baselines include multilabel reformulation choices: only keeping the top-n occurring classes (often 4–10) (e.g., Zhang et al., 2015; Cunha et al., 2021), multiclass classification on each entity within an example (objects

in an image, expressions in a text) (e.g., Lin et al., 2014; Wang et al., 2016; Wei et al., 2016; Zhu et al., 2017). We refrain from doing so.

Table 3.2: Descriptive statistics of our experimental datasets.

Dataset	Type	Classes	Average label count	Number of examples
moviePosters	image	28	2.2	37,632
arXiv2020	text	155	1.9	26,558
Pascal-VOC	image	20	1.6	9,963
MS-COCO	image	80	2.9	122,218

3.5.1 Datasets

Table 3.2 lists the datasets we use. Two of the datasets are multilabel in nature. moviePosters is related to movies (Neha, 2018) and arXiv2020 relates to arXiv paper abstracts (Cornell-University, 2021). We use the image segmentation datasets Pascal-VOC (Everingham et al., 2007) and MS-COCO (Lin et al., 2014), with bounding boxes and one label per box. By attributing all box labels to the image as a whole, it has been used as a reference benchmark for multilabel classification. We refer to Appendix B.4 for further descriptions of the datasets and references.

3.5.2 Learning framework

Our proposed learning framework consists of two parts: a pretrained deep neural network and a classification head (see Figure 3.2); different loss functions are computed in the classification head.

Neural network architecture. For the moviePoster image dataset, we use a MobileNetV2 (Sandler et al., 2018) architecture that was pretrained on ImageNet (Deng et al., 2009). This network architecture is typically used for inference on small computing devices (e.g., smartphones). We use a version of MobileNetV2 already stripped off of its original classification head (Google, 2021). For the three text datasets, we use DistilBERT (Sanh et al., 2019) as implemented in Hugging Face. This is a particularly efficient instance of the BERT model (Huggingface, 2021). For the Pascal-VOC and MS-COCO datasets, we use the recent state-of-the-art resnet TresNet (Ridnik et al., 2021) pretrained on ImageNet (Deng et al., 2009) and some of the best practices for Pascal-VOC and MS-COCO collected in a recent benchmark (Baruch et al., 2020). We use TresNet-m-21K; 21K stands for Imagenet21K, the larger ImageNet corpus. In all cases, we use the final pre-trained layer as an embedding of the input. To ensure that the results of different loss functions are comparable, we fix the model weights of the pretrained MobileNetV2, DistilBERT and TresNet and keep the hyperparameter values that were used to be trained from scratch. At training time, we optimize with Adam for all three architectures and use In-Place Activated BatchNorm (Inplace-ABN) for TresNet (Rota Bulò et al., 2018).

The **classification head** is a latent representation layer (the final pretrained layer mentioned above) connected with a RELU activation. This layer is linked to a final classification layer with a linear activation. The dimension of the final layer is equal to the number of classes in the dataset. The attached loss function is either BCE (Binary Cross-Entropy), focalLoss (Lin et al., 2017), ASL (Baruch et al., 2020), unboundedF1 or sigmoidF1 (ours). When computing the loss at training time, a sigmoid transforms the unbounded last layer to a $[-1, 1]$ bounded vector that contrasts positive and negative predictions. These values are then used as inputs to any of the loss functions above over all classes and the entire batch of examples. In the case of $\mathcal{L}_{\widetilde{F1}}$, this corresponds to a surrogate macro F1. Given the vectorized computation of $\mathcal{L}_{\widetilde{F1}}$ (see Section 3.4.3), the computational burden is only marginally affected. At inference time, the last layer is used for prediction and is bounded with a sigmoid function. A threshold must then be chosen at evaluation time to compute different metrics. Figure 3.2 depicts this learning framework.

Metrics. In our experiments, we report on microF1, macroF1, Precision, mAP (used in some recent multilabel benchmarks; see Appendix B.1) and (micro-)weightedF1 (where within-class scores are weighted by their representation in the dataset). We focus our discussion around weightedF1 as it is the most comprehensive F1 measure we could find on multilabel problems: it is a micro measure, thus accounts for differences between classes, and has a reweighing argument, thus accounting for class imbalance. Given limited resources we rerun each model on each loss with 5 random seeds. With only 5 runs per loss function, hypothesis testing results would have been particularly sensitive to the choice of distribution.[3] Instead, we show the distribution of results in Appendix B.5, which show robust statistics (median and interquartile range). Note that cross-validation cannot be performed as Pascal-VOC and MS-COCO have fixed train-validation-test sets. There is an interaction between our optimization on *sigmoidF1* and our evaluation using (weighted) F1 metrics. We expect higher values on F1-related metrics during evaluation and thus report on alternative metrics too.

3.5.3 Hyperparameters and reproducibility

We implemented all losses in Pytorch and Tensorflow. Batch size is set at a relatively high value of 256 to increase accuracy over traditional losses (Smith et al., 2017), but also allow heterogeneity in the examples within the batch, thus collecting enough values in each quadrant of the confusion matrix (see Section 3.4.4 for a discussion). Regarding the *sigmoidF1* hyperparameters β and η, we performed a grid search with the values in the range $[1, 30]$ for β and $[0, 2]$ for η. In our experiments, we evaluate the sensitivity of our method to these hyperparameters (see Figure 3.3 and Appendix B.4 for optimal values). We made sure to split the data in the same training, validation and test sets for each loss function. We trained for 60 (Pascal-VOC, MS-COCO) to 100 (arXiv2020,

[3] We found that, given some unstable results on unboundedF1, even a conservative student t distribution would imply that the 95% confidence interval covers metric values over 100%.

moviePosters) epochs, depending on convergence. Our code, dataset splits and other settings are shared to ensure reproducibility of our results.

3.6 Experimental Results

The goal of *sigmoidF1* ($\mathcal{L}_{\widetilde{F1}}$) is to optimize for the F1 score directly at training time in the context of multilabel classification. In this section, we test whether $\mathcal{L}_{\widetilde{F1}}$ can outperform existing loss functions on multiple classification metrics. We present multilabel classification results for $\mathcal{L}_{\widetilde{F1}}$ on four datasets, moviePosters, arXiv2020, Pascal-VOC and MS-COCO in Table 3.3.

We recall Table 3.1, in which we highlight that $\mathcal{L}_{\text{BCE}}$ is originally designed for binary classification, $\mathcal{L}_{\text{FL}}$ for imbalanced multiclass, $\mathcal{L}_{\text{ASL}}$ to optimize mAP for multilabel classification. They are computed over each class at training time, as opposed to per batch for our $\mathcal{L}_{\widetilde{F1}}$ and $\mathcal{L}_{\overline{F1}}$. The latter two explicitly account for label dependencies in the loss function.

In general, Table 3.3 shows that $\mathcal{L}_{\widetilde{F1}}$ outperforms other loss functions on three possible formulations of the F1 metric (weightedF1, microF1 and macroF1). We also confirm that the recent ASL loss outperforms other losses on the precision and mAP metrics. $\mathcal{L}_{\widetilde{F1}}$ is designed as an F1 surrogate, it is thus not surprising for it to perform best on F1 metrics and comes at no noticeable additional computational cost (see Appendix B.3). We first analyze the F1 metrics before interpreting the precision and mAP results in more detail.

Measured on the F1 metrics (weightedF1, microF1 and macroF1), $\mathcal{L}_{\widetilde{F1}}$ and $\mathcal{L}_{\text{BCE}}$ always share the top 2 in performance, oftentimes far ahead of other losses. This highlights that losses inspired by BCE are not yet tailored to optimize for the F1 score in multilabel classification, and also that BCE is a good default choice in general. However, in certain settings, and in particular with our standard datasets Pascal-VOC and MS-COCO, $\mathcal{L}_{\widetilde{F1}}$ can provide clear improvements over the original BCE. macroF1 on the moviePosters dataset is a counter-intuitive exception to that observation: BCE outperforms $\mathcal{L}_{\widetilde{F1}}$ only on the macro measure, although $\mathcal{L}_{\widetilde{F1}}$ is essentially a macro F1 loss function, as it is calculated across all classes and over each entire batch. Similarly focalLoss is dominant on MS-COCO macroF1, but not significantly (see Figure 3.5). There is room for improvement on MS-COCO because we did not finetune the sigmoidF1 hyperparameters (β and η) and instead reused the Pascal-VOC ones, due to resource constraints.

On precision and mAP, no top 2 losses emerge. Instead, results are dataset and modality dependent. Surprisingly, the traditional BCE loss outperforms other losses by far in precision on a thoroughly benchmarked dataset like Pascal-VOC. focalLoss delivers best results for MS-COCO on precision, probably because the original paper used MS-COCO as a benchmark to design their loss function (Lin et al., 2017). Precision performance gains are less clear on the two

smaller datasets (arXiv2020 and moviePosters); $\mathcal{L}_{\overline{F1}}$ performs reasonably well.[4] Regarding mAP, $\mathcal{L}_{\text{ASL}}$ expectedly outperforms other methods on Pascal-VOC, confirming their own benchmarks (Baruch et al., 2020) and their ability to beat focalLoss and BCE on MS-COCO and PASCAL-VOC. Notably, $\mathcal{L}_{\text{ASL}}$ is also first on mAP on text data. This is the first time that ASL is tested on text data to the best of our knowledge. Overall, these mitigated results for precision and mAP motivate further research in optimizing directly for precision and mAP at training time.

A note on thresholding and zero values. For the bigger and more standard datasets Pascal-VOC and MS-COCO,[5] our neutral metric threshold of 0.5 provides results in line with the literature. With our own fine-tuning regime on a smaller model (see Section 3.5.2), our mAP scores are 1–2% away from the current state of the art (Baruch et al., 2020). On smaller datasets like arXiv2020, moviePosters and others (see Appendix B.6), the sigmoid activation per class at inference time are closer to zero. To a certain extent, this can be interpreted as the model having less confidence in its predictions (Guo et al., 2017). As a result, a neutral 0.5 threshold resulted in zero values on almost all losses and metrics for small datasets. Given the range of values in these predictions, 0.05 seems like the next best neutral threshold. We refrain from further finetuning the threshold for each dataset, loss and metric.[6] As a result of the absence of finetuning, moviePosters display zero values for $\mathcal{L}_{\text{FL}}$ and $\mathcal{L}_{\text{ASL}}$ on most metrics. This can be explained by the higher average label count for moviePosters. This is in opposition to the propensity of $\mathcal{L}_{\text{FL}}$ and $\mathcal{L}_{\text{ASL}}$ to deal with sparser label representation.

The analysis above highlights that sigmoidF1 can indeed optimize for F1 metrics (weightedF1, microF1 and macroF1) reliably and consistently, over six datasets in total (see Appendix B.5). Given the more mitigated results for precision and mAP, it seems relevant to further explore opportunities of metrics-as-losses. Finally, BCE, which was designed with binary classification in mind, is a good first approximation.

Sensitivity analysis. In Figure 3.4, we show the sensitivity of *sigmoidF1* to different parametrizations of η and β. Within the chosen values (see Section 3.5.3), we chose to display a parameter space similar to the one illustrated in Figure 3.3. Moving the sigmoid to the left allows the learning algorithm to tend to a (local) optimum. In general and across datasets, when sampling for η, we noticed how the optimum tended towards positive values. Offsetting the sigmoid curve to the left has the effect of pushing more candidate predictions to the rank of positive

[4] $\mathcal{L}_{\overline{F1}}$ was found particularly unstable for Pascal-VOC over 5 different seeds (see the extended results in Appendix B.5). Provided it is unbounded, predictions can diverge towards (positive or negative) infinite values.

[5] The classes in Pascal-VOC and MS-COCO are a lot more concrete (e.g., car, person, bicycle) and are directly related to the original classes of ImageNet on which the TresNet and MobileNetV2 were trained, as opposed to movie genres for moviePosters or arXiv paper scientific domain.

[6] While optimizing the threshold at inference time is an interesting research topic, we refrain from doing so here, so as to disentangle the loss function benchmarking from the thresholding regime benchmarking.

Table 3.3: Multilabel classification mean performance in percent over 5 random seeds. The F1 metric variants are the focus here (weightedF1, microF1 and macroF1), since we aim to directly optimize for F1 at training time. precision and mAP are displayed for reference, as they are often used in the literature in that context. Metric are formally defined in Appendix B.1 and thresholds are indicated there for each dataset. We reused fine-tuned Pascal-VOC sigmoidF1 hyperparameters (β and η) for MS-COCO due to resource constraints.

Loss	weightedF1	microF1	macroF1	precision	mAP
TresNetm21K (Ridnik et al., 2021) on MS-COCO @0.5 (CNN)					
$\mathcal{L}_{\text{BCE}}$(Fisher, 1912)	79.02	75.81	79.55	82.52	81.21
$\mathcal{L}_{\text{FL}}$(Lin et al., 2017)	81.28	79.18	**81.76**	**85.73**	84.88
$\mathcal{L}_{\text{ASL}}$(Baruch et al., 2020)	73.48	70.36	70.81	60.16	**85.59**
$\mathcal{L}_{\overline{F1}}$[ours]	79.90	77.51	79.74	81.05	78.33
$\mathcal{L}_{\widetilde{F1}}$[ours]	**81.82**	**79.93**	81.67	80.62	81.98
TresNetm21K (Ridnik et al., 2021) on Pascal-VOC @0.5 (CNN)					
$\mathcal{L}_{\text{BCE}}$(Fisher, 1912)	87.52	85.85	87.76	**90.75**	91.54
$\mathcal{L}_{\text{FL}}$(Lin et al., 2017)	72.54	59.24	76.82	84.70	76.19
$\mathcal{L}_{\text{ASL}}$(Baruch et al., 2020)	77.85	76.53	75.98	65.36	**93.11**
$\mathcal{L}_{\overline{F1}}$[ours]	77.24	74.84	75.31	75.53	79.36
$\mathcal{L}_{\widetilde{F1}}$[ours]	**88.20**	**87.70**	**87.87**	85.36	92.36
DistilBert (Sanh et al., 2019) on arXiv2020 @0.05 (NLP)					
$\mathcal{L}_{\text{BCE}}$(Fisher, 1912)	20.59	18.19	18.42	10.15	10.50
$\mathcal{L}_{\text{FL}}$(Lin et al., 2017)	18.85	16.59	18.01	10.10	10.43
$\mathcal{L}_{\text{ASL}}$ (Baruch et al., 2020)	19.15	16.90	18.16	**10.32**	**10.53**
$\mathcal{L}_{\overline{F1}}$[ours]	15.23	13.74	14.50	10.27	10.49
$\mathcal{L}_{\widetilde{F1}}$[ours]	**20.60**	**18.20**	**18.43**	10.15	10.50
MobileNetV2 (Sandler et al., 2018) on moviePosters @0.05 (CNN)					
$\mathcal{L}_{\text{BCE}}$(Fisher, 1912)	13.79	9.47	**12.94**	5.51	5.78
$\mathcal{L}_{\text{FL}}$(Lin et al., 2017)	0.00	0.00	0.00	0.00	5.80
$\mathcal{L}_{\text{ASL}}$(Baruch et al., 2020)	0.00	0.00	0.00	0.00	5.80
$\mathcal{L}_{\overline{F1}}$[ours]	13.97	9.84	10.11	**5.59**	**5.90**
$\mathcal{L}_{\widetilde{F1}}$[ours]	**14.81**	**10.33**	10.57	5.58	5.81

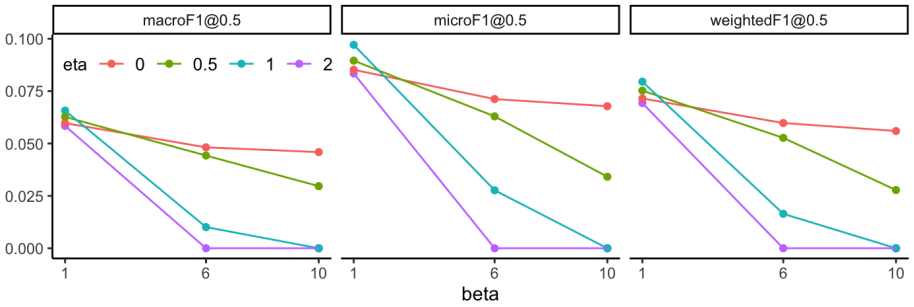


Figure 3.4: DistilBERT (NLP) on arXiv2020 – different weightedF1 scores at a 0.5 threshold for different values of η and β in a sampling region similar to Figure 3.3.

instance (or at least close to 1). We also note how β (which cannot be negative or otherwise the sigmoid function would flip around the horizontal axis) is at best close to a value close to 0 on this dataset (we show discrete values here for display purposes). The sigmoid is thus relatively smooth, which involves dynamic gradients over different batches. The idea is similar to a high learning rate. In our experiments, this rarely gave rise to divergent behavior in the loss function (learning curve). We learn that it is necessary to tune hyperparameters for each dataset, as it is for $\mathcal{L}_{\text{FL}}$, $\mathcal{L}_{\text{ASL}}$ and others in Table 3.1.

The results in this section show that, in general, multilabel classification results measured on F1 metrics can be improved using sigmoidF1 – independently of the dataset, its modality or the neural network architecture.

3.7 Discussion

In multilabel classification, and more generally in the context of deep neural networks, losses are formulated to be decomposable for gradient descent. At inference time, however, end-users tend to look for clear-cut actionable decisions from the model (e.g., to automize the arXiv keywords selection, one needs to obtain a clear-cut set of keywords given each abstract). This is probably why most evaluation metrics in the multilabel literature, with the notable exception of mAP, are also reliant on clear-cut counts (e.g., tp, fn, fp, tn). Although models are benchmarked on these values, we found little discussions on how to retrieve clear-cut counts from final softmax / sigmoid activations bounded by $[0, 1]$. Among our benchmarked losses, the authors of FocalLoss (Lin et al., 2017) use a global 0.5 threshold at inference-time. The authors of ASL (Wu et al., 2019a) do not mention thresholding in the paper but a GitHub issue hints at the fact that they used 0.8 as a global threshold for MS-COCO.[7] We feel that defining clear-cut counts deserves more attention.

[7] See `https://github.com/Alibaba-MIIL/ASL/issues/8` and also insightful learning tricks at `https://github.com/Alibaba-MIIL/ASL/issues/30`.

Such clear-cut counts are usually achieved via a *decision threshold*. Decubber et al. (2018) distinguish between *utility maximization* (at inference-time) and *decision-theoretic* (at training and at inference time) approaches.

Utility maximization methods. At inference time, a threshold can be set globally for all examples to optimize on the training data, before using it on the test data (Lipton et al., 2014; Decubber et al., 2018). This threshold can optimize for specificity, sensitivity (Chen et al., 2006) or directly for F1 (Decubber et al., 2018). Alternatively, different thresholds can be set per-class (Chu and Guo, 2017).

Decision-theoretic methods. Decision-theoretic methods operate both at training and at inference time. They stem from shallow learning fields and have multiple steps: (i) *encoding* the original item-label matrix to submatrices that each can be (ii) fit to a traditional loss function (cross-entropy variations), before (iii) *decoding* the submatrices back to the original item-label matrix format via an inference-phase optimization solver. This methodology can be found across the shallow learning fields of SVMs (Ye et al., 2012), logistic regession (Dembczynski et al., 2011; Zhang et al., 2020), multinomial regression (Dembczynski et al., 2013), and Bayesian networks (Gasse and Aussem, 2016). These methods were implemented in a deep learning setting, where they had more success than *utility maximization* or fixed thresholding methods (Decubber et al., 2018). Decision-theoretic methods have at least 3 moving parts mentioned above and are thus complicated to benchmark against each other, let alone against inference-time thresholding or fix thresholding.

As hinted before with ASL and focalLoss, modern deep learning models tend to not tune the decision threshold, either with *utility maximization* or *decision-theoretic* methods. We propose to take a first step in this direction in the following.

For the sake of this discussion, we focus on the simplest *utility maximization* (inference-time) thresholding implementation. *Threshold Averaging* (Decubber et al., 2018) is a method that uses the training set to tune a global threshold, before applying it to the test set. Take $\hat{\mathbf{y}}_i$, a set of label predictions for one example i. We select each $\hat{y}_{ij}$ as a possible thresholding candidate to binarise the vector $\hat{\mathbf{y}}_i$. We then calculate instance-wise F1 scores over $\hat{\mathbf{y}}_i$. The value $\hat{y}_{ij}$ that results in the highest F1 score for an instance is chosen as the instance's threshold. This process is repeated for each instance i in the training data. The average threshold over all instances in the training data is then chosen as the final global threshold for the test data.

In Table 3.4, we show results of *Threshold Averaging* (Decubber et al., 2018) on the arXiv dataset. It is notable here that ASL's mean results always outperform other losses. This time around, however, almost all boxplots IQRs intersect, thus results are very inconclusive (see Figure 3.9). We thus refrain from bold numbers like in Table 3.3. Most importantly, metrics are consistently below results from the original neutral fixed 0.05 threshold in Table 3.3. This is consistent with some of the results in (Decubber et al., 2018), showing that simple thresholding methods based on *utility maximization* are not sufficient to consistently beat fixed thresholds or *decision-theoretic* methods.

Table 3.4: Multilabel classification mean performance in percent over 5 random seeds. Global thresholds were found using a threshold-moving technique. Results are systematically lower than with a fixed threshold (see third row of Table 3.3). Metric are formally defined in Appendix B.1.

Loss	weightedF1	microF1	macroF1	precision	mAP
DistilBert (Sanh et al., 2019) on arXiv2020 (NLP) – Threshold moving					
$\mathcal{L}_{\text{BCE}}$(Fisher, 1912)	15.89	13.98	15.73	10.11	10.35
$\mathcal{L}_{\text{FL}}$(Lin et al., 2017)	16.40	14.14	17.22	9.83	10.42
$\mathcal{L}_{\text{ASL}}$ (Baruch et al., 2020)	17.49	14.86	17.77	10.33	10.51
$\mathcal{L}_{\overline{F1}}$[ours]	16.52	14.27	16.70	9.98	10.43
$\mathcal{L}_{\widetilde{F1}}$[ours]	15.11	13.19	15.20	10.05	10.41

Inference-time decisions can completely change the outcome of a prediction set, of its resulting evaluation metrics, and, thus, even of the winning model. We hope that thresholding will be more broadly discussed in the future or at least for the thresholding method to be openly stated in research papers; we chose fixed neutral thresholds, to focus on the benchmarking of losses at training-time.

Together, utility maximization (inference-time) thresholding methods and decision-theoretic methods (at training and at inference time) form an under-explored research domain, with several open questions: (i) Which data split should be used for thresholding? With the entire training dataset (Decubber et al., 2018), there is a risk of overfitting the threshold. Maybe it is worth introducing a holdout set that is only used for threshold tuning. (ii) Should we threshold globally for interpretability or have a per-class or even per-instance threshold? (iii) Are *decision theoretic* (a.k.a. at training and at inference time) approaches also not prone to overfitting and are they efficient on large neural networks for large datasets? (iv) Can other losses than the classical cross-entropy loss be used to train *decision theoretic* models?

3.8 Conclusions

To solve multilabel learning tasks, existing optimization frameworks are typically based on variations of the cross-entropy loss. Instead – inspired by the binary classification literature (see most recently (Gai et al., 2019) and their F1 surrogate loss functions on 3-layer neural networks) – we propose the *sigmoidF1* loss, as part of a general loss framework for confusion matrix metrics. *sigmoidF1* loss can achieve significantly better results for most metrics on four diverse datasets and outperforms other losses on the weightedF1 metric. We thereby provide evidence that *sigmoidF1* is robust to modality, model architecture and dataset size, when optimizing for F1 metrics. Generally, our smooth formulation of confusion

matrix metrics allows us to optimize directly for these metrics that are usually reserved for the evaluation phase. The proposed *unboundedF1* counterpart does not require hyperparameter tuning and delivered better results than existing multiclass losses on most metrics; it can act as a mathematically less robust approximation of *sigmoidF1*.

In future work and within the generic multilabel setting, a first incremental step could be to train on a bigger dataset like MS-COCO (Lin et al., 2014) (if provided with more resources) and use more robust transfer learning/finetuning procedures, for example with dynamic weight freezing for finetuning (Howard and Ruder, 2018). Alternatively, we could train a CNN or a BERT model for multilabel tasks with our smooth losses from scratch (cf., (Wu et al., 2019a) and (Lin et al., 2017)). If training from scratch, this can be combined with representation learning (Milbich et al., 2020; Wang et al., 2020) or self-supervised learning, in order to model abstract relationships.

Next, we could validate if F1 or another confusion-matrix-metric-as-a-loss can tackle other multilabel settings, such as hierarchical multilabel classification (Benites and Sapozhnikova, 2015), active learning (Nakano et al., 2020), multi-instance learning (e.g., Soleimani and Miller, 2017; Zhou et al., 2012), holistic label learning (see dataset *Large Scale Holistic Video Understanding* (Diba et al., 2019)), or extreme multilabel prediction (Chang et al., 2020; Liu et al., 2017; Babbar and Schölkopf, 2017; Yen et al., 2017; Prabhu et al., 2018) (with missing labels (Yu et al., 2014; Jain et al., 2016)), where the number of classes ranges in the tens of thousands. Beyond the multilabel setting, *sigmoidF1* could be tested on any model that uses F1 score as an evaluation metric such as AC-SUM-GAN (Apostolidis et al., 2020).

One limitation of *sigmoidF1* is that it is computed at a macro level over the whole batch and ignores (micro) per class F1 scores. Given our limited GPU memory, we could not load enough examples in each batch to represent each confusion matrix quadrant of each class reliably. If such a route is followed, we could eventually finetune or learn β_c and η_c – the parameters of the sigmoid function – per class c.

We believe that smooth metric surrogates should inform future research on multilabel classification tasks. There is evidence of a growing interest in the literature (Chang et al., 2019; Huang et al., 2015; Patel et al., 2022) for metrics as losses and the objective of this chapter is to further highlight their relevance, across modalities, architectures and dataset sizes. Based on the results presented in this chapter, we consider metrics-as-losses (e.g., Jaccard, confusion matrix metrics, ranking metrics) as the next step in the evolution of multilabel classification algorithms.

3.9 Upshots for the Personalization Flow

In this chapter we constructed a surrogate to the F1 metric, that is differentiable everywhere at training time. Thanks to that loss function, we are able to train image models that categorize thumbnails into an optimal set of categories

(as defined by the F1 score). We can then serve each user with personalized thumbnails, given their preference (action, romance, etc.). But this endeavor is only a small step towards two separate avenues: (i) If our goal is to optimize for certain non-differentiable metrics at inference-time, why don't we use more metric surrogates as losses at training-time? (ii) In the future, we could think of generating thumbnails or movie posters from scratch for each user, based on their preferences.

In the next chapter, we take a step back. Once we have served personalized recommendations with personalized thumbnails, we look at how the user behaves on the platform over time. More precisely, we combine the observable user behavior data (e.g., clicks) with unobservable intents (e.g., bookmark videos to watch later) to predict the satisfaction of that user.

Reproducibility

To facilitate the reproducibility of this chapter, our code is available at `https://github.com/gabriben/metrics-as-losses`.

B Appendices

B.1 Evaluation metrics

In our experimental evaluation in this chapter, we consider a suite of metrics that are commonly used in the evaluation of multilabel classification to measure the effectiveness of multilabel prediction. These metrics are based on the confusion matrix and for which we provided smoothed surrogates to optimize directly (see Table 3.5).

Table 3.5: Confusion matrix with our proposed smoothed confusion matrix entries, $\widetilde{tp}$, $\widetilde{fp}$, $\widetilde{fn}$ and $\widetilde{tn}$ and six derived loss functions that use these smoothed confusion matrix entries. $\mathcal{L}_{\widetilde{F1}}$ is used in our experiments.

	Condition positive	**Condition negative**	$\mathcal{L}_{\widetilde{Accuracy}} = \frac{\widetilde{tp}+\widetilde{tn}}{\widetilde{tp}+\widetilde{fp}+\widetilde{tn}+\widetilde{fn}}$
Predicted positive	True positive $\widetilde{tp} = \sum \mathbf{S}(\hat{\mathbf{y}}) \odot \mathbf{y}$	False positive $\widetilde{fp} = \sum \mathbf{S}(\hat{\mathbf{y}}) \odot (\mathbb{1} - \mathbf{y})$	$\mathcal{L}_{\widetilde{Precision}} = \frac{\widetilde{tp}}{\widetilde{tp}+\widetilde{fp}}$
Predicted negative	False negative $\widetilde{fn} = \sum(\mathbb{1} - \mathbf{S}(\hat{\mathbf{y}})) \odot \mathbf{y}$	True Negative $\widetilde{tn} = \sum(\mathbb{1} - \mathbf{S}(\hat{\mathbf{y}})) \odot (\mathbb{1} - \mathbf{y})$	$\mathcal{L}_{\widetilde{NPV}} = \frac{\widetilde{tn}}{\widetilde{tn}+\widetilde{fn}}$
	$\mathcal{L}_{\widetilde{Recall}} = \frac{\widetilde{tp}}{\widetilde{tp}+\widetilde{fn}}$	$\mathcal{L}_{\widetilde{Specificity}} = \frac{\widetilde{tn}}{\widetilde{fp}+\widetilde{tn}}$	$\mathcal{L}_{\widetilde{F1}} = \frac{2\widetilde{tp}}{2\widetilde{tp}+\widetilde{fn}+\widetilde{fp}}$

Table 3.6: Average training time over 5 seeds in minutes (60 epochs for MS-COCO and Pascal-VOC, 100 epochs for the reminder two).

	MS-COCO	Pascal-VOC	arXiv2020	moviePosters
$\mathcal{L}_{\text{BCE}}$(Fisher, 1912)	856	112	341	58
$\mathcal{L}_{\text{FL}}$(Lin et al., 2017)	851	108	428	59
$\mathcal{L}_{\text{ASL}}$(Baruch et al., 2020)	856	109	427	59
$\mathcal{L}_{\overline{F1}}$[ours]	858	116	381	58
$\mathcal{L}_{\widetilde{F1}}$[ours]	858	111	351	52

When true positives and false positives are used, recall that $tp = \mathbb{1}_{\hat{\mathbf{y}} \geq t} \odot \mathbf{y}$ and $fp = \mathbb{1}_{\hat{\mathbf{y}} \geq t} \odot (\mathbb{1} - \mathbf{y})$, and thus a threshold t must be set. For Pascal-VOC and MS-COCO, we set $t = 0.5$, as is commonly done in the early literature (Zhang and Zhou, 2014; Clare and King, 2001). In the recent literature, the chosen threshold at inference time can vary but was not found to be justified, we thus decide on neutral thresholds before training.

Extending F_1 to multiclass binary classification means deciding whether to pool classes. In a first pooled iteration, macro F_1 (Koyejo et al., 2015) equates to creating a single 2x2 confusion matrix for all classes:

$$F_1^{macro} = \frac{\sum^C 2tp_j}{2\sum^C tp_j + \sum^C fn_j + \sum^C fp_j}, \tag{3.9}$$

with $\sum^C(\cdot)$ as a short form of $\sum_{j=1}^C(\cdot)$, when summing over each class up to the C classes. Micro F_1 (Lipton et al., 2014; Koyejo et al., 2015) amounts to creating one confusion matrix per class or unpooling:

$$F_1^{micro} = \frac{1}{C}\sum_{j=1}^{C} \frac{2tp_j}{2tp_j + fn_j + fp_j} = \frac{1}{C}\sum_{j=1}^{C} F_1^j. \tag{3.10}$$

Weighted micro F_1 (Behera et al., 2019) is similar but includes weighing to account for class imbalance, i.e., weighing each class by the number of ground truth positives:

$$F_1^{weighted} = \frac{1}{C}\sum_{j=1}^{C} p_j F_1^j \quad \text{, where } p_j = \sum_i \mathbb{1}_{\mathbf{y_i^j}=1}. \tag{3.11}$$

We also define micro precision

$$P^{micro} = \frac{1}{C}\sum_{j=1}^{C} \frac{tp_j}{tp_j + fp_j}. \tag{3.12}$$

mean Average Precision (mAP) has different definitions. We use mAP as defined for the MS-COCO and Pascal-VOC datasets (Padilla et al., 2020). Traditionally, Precision and Recall is computed over the intersection of object detection boxes. We use a slightly modified mAP (e.g., in (Baruch et al., 2020)), where precision and recall are computed over the predictions of labels on the whole image. We first obtain the average precision over each class:

$$\begin{aligned} \mathrm{AP}_{\mathrm{all}} &= \sum_i \left(R_{i+1} - R_i\right) P_{\mathrm{interp}}\ (R_{i+1}) \\ P_{\mathrm{interp}}\ (R_{i+1}) &= \max_{\tilde{R}:\tilde{R} \geq R_{i+1}} P(\tilde{R}), \end{aligned} \tag{3.13}$$

and then compute mean Average Precision:

$$\mathrm{mAP}^{micro} = \frac{1}{C}\sum_{j=1}^{C} \mathrm{AP}_i. \tag{3.14}$$

We write micro here to be explicit, but it seems to be mostly computed at the micro level in the literature.

B.2 Focal loss definition

We write down the *focalLoss* (Lin et al., 2017), as it deals specifically with class imbalance and is used as a baseline due to its popularity in the multiclass domain.

$$\mathcal{L}_{FL} = -\alpha^{\mathrm{j}} \left(1 - \hat{y}^{\mathrm{j}}\right)^{\gamma} \log\left(\hat{y}^{\mathrm{j}}\right), \tag{3.15}$$

with α^j and γ hyperparameters. In the next section, we further specify the setup for focal loss and cross entropy as benchmarks for *unboundedF1* and *sigmoidF1*.

B.3 Compute time

Table 3.6 shows compute times in minutes for different losses and different datasets on a single GPU *g4dn.12xlarge* AWS instance.[8] The run-time is not particularly long, given that we freeze model weights of the pretrained image / text model.

B.4 Experimental setup details

moviePosters consists of images of movie posters and their genres (e.g., *action*, *comedy*) (Chu and Guo, 2017).[9] The posters and labels have been extracted from IMDB and the dataset was previously used for per-class, post-training thresholding (see Section 4.2). The genre labels in this dataset are not mutually exclusive and of varying counts per movie.
arXiv2020 is a subset of the newly created *arXiv dataset*[10] with over 1.7 million open source articles and their metadata. Our experiments use the abstracts and categories that are suitably non-mutually exclusive and of varying counts per example. The limited number of labeled classes render the older dataset unsuitable for our experiments. We write arXiv2020 for the subset of the *arXiv dataset* that only contains documents published in 2020. This results in around 26k documents. There is a longer history of using arXiv to create research datasets; the dataset we use is not to be confused with an earlier long document dataset that only features 11 classes (He et al., 2019), and was used in a recent long transformer publication (Zaheer et al., 2020).
pascal-VOC and MS-COCO stand for Pascal Visual Object Classes Challenge (VOC 2007) (Everingham et al., 2007) and Microsoft Common Objects in Context (Lin et al., 2014), respectively. They are object recognition/segmentation datasets. The earlier Pascal-VOC dataset has 20 possible object classes and around 10K examples. The later MS-COCO dataset has 80 possible object classes and around 200K class-annotated examples. Some multilabel classification literature for the image domain use object detection / segmentation datasets to

[8] `https://aws.amazon.com/ec2/instance-types/g4/`
[9] Labels at `https://tinyurl.com/y7ydyedu` and images at `https://tinyurl.com/y7lfpvlx`.
[10] Available at `https://tinyurl.com/5kypspya`.

perform multilabel classification:[11] MS-COCO, Pascal-VOC, NUS-WIDE, etc. (note that transformer models, which effectively distinguish the original objects on the image while predicting labels, perform better on this task (Liu et al., 2021)). Regarding Tresnet-m-21k (Ridnik et al., 2021), while an L and an XL version of the model exist, the code available online did not allow for correct loading of the weights.

We choose to ignore classes that are underrepresented, in order to give the model a fair chance at learning from at least a few examples. We define underrepresentation as a global irrelevance threshold b for classes: any class c that is represented less than b times is considered irrelevant. We decided to set an irrelevance threshold b on all datasets prior to conducting experiments, so as to not fine-tune for that feature. It was set to 1000 for both *arXiv2020* (145 of the original 155 classes remaining) and *moviePosters* (14 of the 28 classes remaining) and at 10 for *chemicalExposure* (all 38 classes remaining) and *cancerHallmarks* (all 33 classes remaining), in proportion to the number of classes and labels in each dataset. We used all classes for Pascal-VOC and MS-COCO since we are comparing with benchmarks that also do so.

Hyperparameters. For Pascal-VOC, we found $\{\beta = -0.75; \eta = 10.25\}$ to work best on weightedF1. Given the similarity of the two datasets and the potentially resource-hungry hyperparameter tuning of MS-COCO, we used the same hyperparameters for MS-COCO. For arXiv2020 and moviePosters, $\{\beta = 1; \eta = 9\}$ works best on weightedF1. These hyperparameters where tuned on the validation set and we report on the held out test set. It would be hard to give a general recommendation of hyperparameters, but it seems that $\{\beta = -0.75; \eta = 10.25\}$ is a good basis for image and that $\{\beta = 1; \eta = 9\}$ is a good basis for text.

Setup. We performed our experiments on Amazon Web Services cloud machines with data parallelization on up to 4 GPUs *g4dn.12xlarge*[12], with TensorFlow 2 (Abadi et al., 2015) and PyTorch (Paszke et al., 2019) as a gradient-descent backend.

B.5 Extended results

Table 3.3 shows our results as point estimates over 5 training random seeds. This section contains the distributional counterpart of Table 3.3, namely boxplots (Figure 3.5, 3.6, 3.7 and 3.8) with median and inter quartile range in the blue box. Figure 3.9 is the distributional counterpart of Table 3.4 (threshold-moving technique on the arXiv dataset) and outlines less conclusive results than for fixed thresholds.

[11] See `https://paperswithcode.com/task/multi-label-classification`.
[12] `https://aws.amazon.com/ec2/instance-types/g4/`

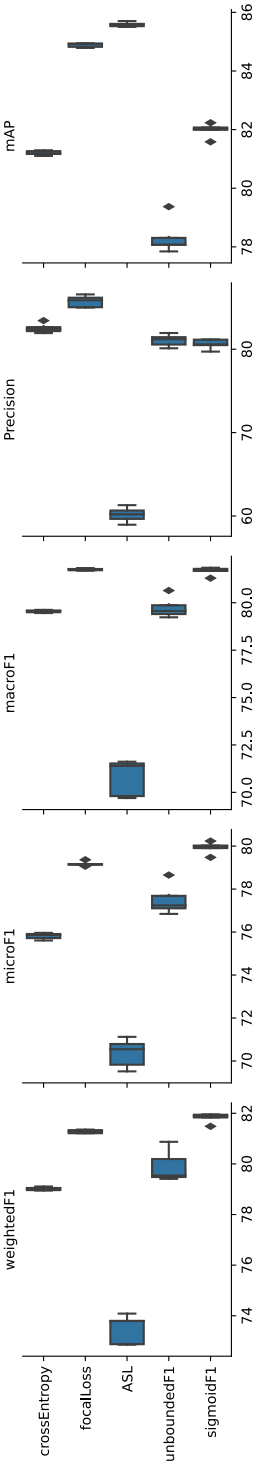


Figure 3.5: Tresnetm21K (CNN) on MS-COCO @0.5.

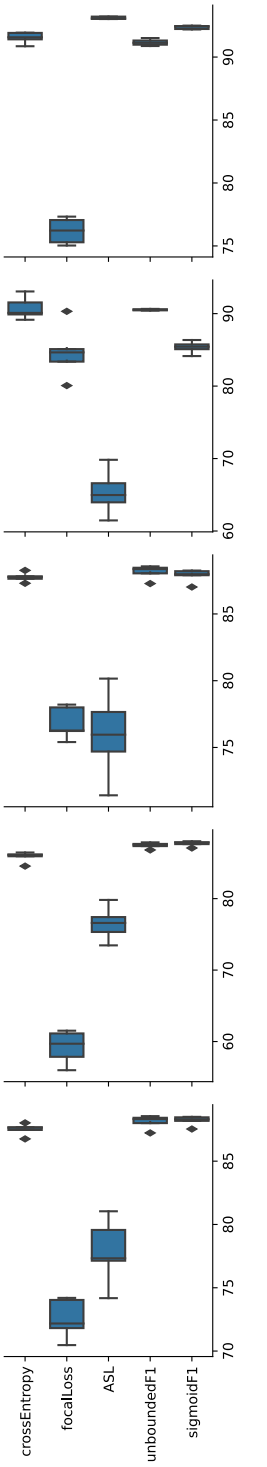


Figure 3.6: Tresnetm21K (CNN) on Pascal-VOC @0.5 (one outlier (<40) for unboundedF1 on each metric ignored for better visualization).

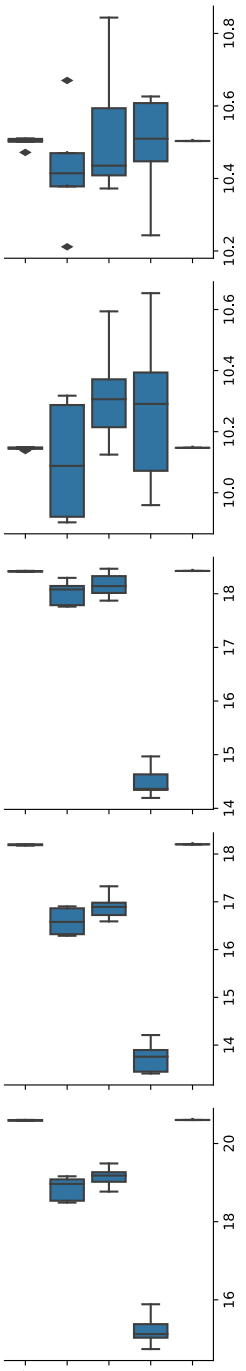


Figure 3.7: DistilBERT (NLP) on arXiv2020 @0.05.

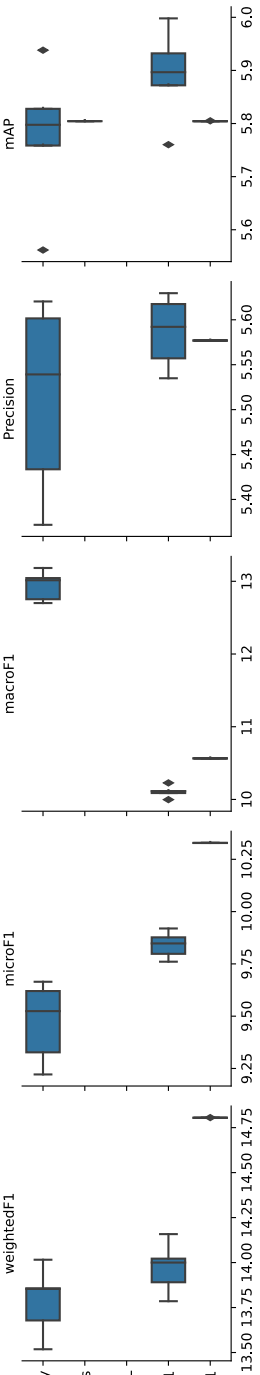


Figure 3.8: MobileNetV2 (CNN) on moviePosters @0.05 (zero values for focalLoss and ASL ignored for better visualization).

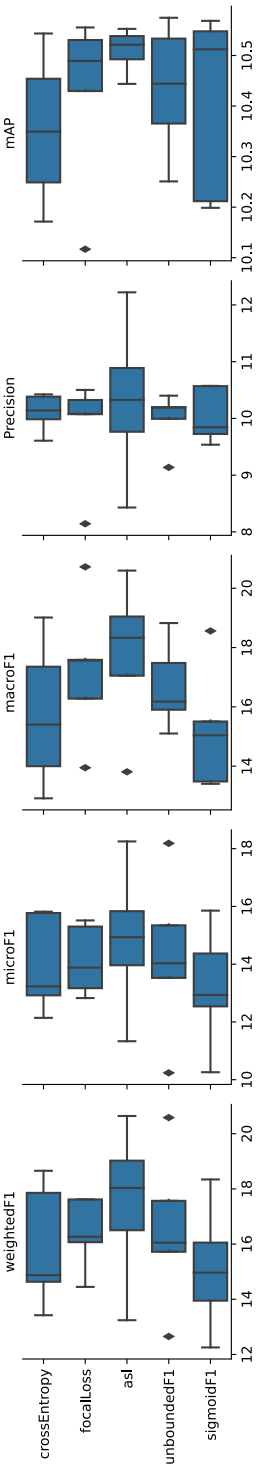


Figure 3.9: DistilBERT (NLP) on arXiv2020 – Threshold moving.

B.6 Additional experiments

This section details additional experiments on two further text datasets from the medical domain. Given that they are relatively small compared to our other benchmark datasets, we keep this discussion in the appendix of this chapter. Table 3.8 illustrates the difference between our 4 main paper datasets and the 2 appendix datasets. Results on the latter are displayed in Tables 3.7a and 3.7b.

ML-NET (Du et al., 2019) has an interesting multitask approach to *fit-algorithm-to-data* methods for multilabel learning with unknown label count on text. The cancerHallmark (Hanahan and Weinberg, 2011)[13] and chemicalExposure (Larsson et al., 2014)[14] datasets were used. The third dataset diagnosisCodes could not be obtained (neither from the authors of ML-NET nor those of the original paper (Perotte et al., 2014)). We aggregate sentence labels to the whole description for cancerHallmarks and chemicalExposure, as was done for ML-NET.

Table 3.7: Multilabel classification performance@0.05 on a single run.

(a) DistilBERT (NLP) + classification head on cancerHallmarks.

Loss	weightedF1	microF1	macroF1	Precision
$\mathcal{L}_{\text{BCE}}$	0.0	0.0	0.0	0.0
$\mathcal{L}_{\text{FL}}$	10.8	19.0	4.4	7.1
$\mathcal{L}_{\overline{F1}}$	17.0	17.6	**9.8**	**8.9**
$\mathcal{L}_{\widetilde{F1}}$	**20.2**	**31.3**	9.5	5.9

(b) DistilBERT (NLP) + classification head on chemicalExposure.

Loss	weightedF1	microF1	macroF1	Precision
$\mathcal{L}_{\text{BCE}}$	5.1	5.8	1.2	4.7
$\mathcal{L}_{\text{FL}}$	26.8	34.8	9.3	13.0
$\mathcal{L}_{\overline{F1}}$	21.8	19.4	**13.3**	**15.5**
$\mathcal{L}_{\widetilde{F1}}$	**31.9**	**43.2**	11.3	9.1

For arXiv2020, moviePosters, cancerHallmarks and chemicalExposure, we saw after a few preparatory training rounds that almost only *sigmoidF1* had non-zero results for $t = 0.5$. Class representation is a lot more sparse for these dataset, we thus set the evaluation metrics threshold to a reasonable value of 0.05 and train for 100 (arXiv2020, moviePosters) or 500 (cancerHallmarks and chemicalExposure) epochs until reaching convergence. Once thresholds were decided upon, no further threshold-hacking was performed. Note that a threshold of 0.8 on Pascal-VOC, as used by (Baruch et al., 2020), does not alter the results.

On the smaller chemicalExposure and cancerHallmarks datasets (see Ta-

[13] Available at `https://github.com/sb895/Hallmarks-of-Cancer`.
[14] Available at `https://github.com/sb895/chemical-exposure-information-corpus`.

bles 3.7a and 3.7b respectively), the *unboundedF1* loss delivers good results for macroF1 and Precision and the *sigmoidF1* loss leads to higher scores on the remainder of the metrics. We observe that *unboundedF1* scores higher than *sigmoidF1* on macroF1 on the two small text datasets (chemicalExposure and cancerHallmarks). Since *unboundedF1* forgoes thresholding altogether, we hypothesize that *unboundedF1* develops tolerance for sparse datasets with low number of class instances.

Table 3.8: Descriptive statistics of all datasets.

	Type	Classes	Avg label count	Num of examples
moviePosters	image	28	2.2	37,632
arXiv2020	text	155	1.9	26,558
chemExposure	text	38	6.1	3,661
cancerHallmarks	text	33	3.5	1,582
Pascal-VOC	image	20	1.6	9,963
MS-COCO	image	80	2.9	122,218

Notably for the cancerHallmarks dataset, predictions from a model trained with cross-entropy do not reach high enough values to surpass the threshold and thus all metrics return zero values. This was further observed during experimentation, thus cross-entropy loss might not be a good fit for solving small-dataset multilabel problems.

Chapter 4

Intent, Behavior and Satisfaction

We are now at a stage in the personalization flow where the platform has already nudged the user with specific content and their appearance on the page.

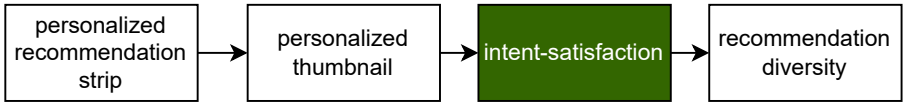


Figure 4.1: The third step of the personalization flow.

We would like to know whether the user is satisfied with the streaming platform over time. There is a lot of research based on observable behavioral data – oftentimes aggregated over user groups – such as the time spent on the platform (see Sections 4.2.1 and 4.2.1). But can we harness this data in a more personalized way and connect it to some latent aspects such as the users' intents and satisfaction on the platform? We reproduce a study performed in the music domain in the video domain at Videoland. We provide all resources to reproduce our study. More precisely, we are interested in modeling the following causal chain:

RQ3: Are users' intents together with their behavioral data useful signals to predict or explain satisfaction on a video streaming platform?

We answer that question for a second time, as a reproduction of a Spotify experiment (Mehrotra et al., 2019). While Mehrotra et al. (2019) proposed linear logistic regression models to predict satisfaction based on intent and behavioral data, we use random forests – for higher accuracy – and hierarchical Bayesian models – for better interpretability. We release simulated data, code and experimental design, to facilitate further research.

4.1 Introduction

Personalized content and experiences on music, video, and other types of content platforms, rely on user data as feedback (Ko et al., 2022). Such input often has the form of interaction data on a website or from a dedicated app and is then used as implicit feedback from the user (Morita and Shinoda, 1994). For

This chapter was published in the ACM Transactions on Recommender Systems (TORS) under the title "Intent-Satisfaction Modeling: From Music to Video Streaming" (Bénédict et al., 2023a)

paid-subscription platforms whose longer term goal is retention, this type of implicit feedback might not be enough (Duan and Zhai, 2015). In the short term, retention propensity translates to some form of satisfaction that is highly subjective, time-varying, and might form a signal hidden in the implicit feedback data. The literature lists two possible ways to approximate a measure of short-term satisfaction (Beheshti et al., 2020): (i) seek explicit feedback via surveys (e.g., in-person, in-app, in-email), or (ii) obtain implicit feedback from user behavior on the website or app (e.g., content consumption, time on site, time on homepage, etc.).

4.1.1 The importance of intent

Implicit and explicit feedback each have their own strengths and weaknesses (Guo and Agichtein, 2012; Dragone et al., 2019). Most weaknesses can be avoided through careful survey design for explicit feedback and through granular user tracking for implicit feedback. However, we identify one irreducible weakness: *missing context* from behavioral data. For example, someone might watch a few trailers during a session and never play a full movie/episode. This could be interpreted as an unsuccessful session. It could also be that the user did not have time to watch the full content and instead was selecting content for a family watching session later that evening.

One way to retrieve context is to explicitly ask users about their current intents, join that survey data to behavioral data for each session, and thus introduce context back into implicit behavioral data. Mehrotra et al. (2019) use a survey to retrieve users' current intent and satisfaction level, before collecting said user's interaction signals on a music streaming platform. They then show that satisfaction models are more accurate when intent is included as a variable. With visualizations and logistic regressions they show that intent together with behavioral data is more predictive of satisfaction than behavioral data alone.

4.1.2 From music to video streaming

We are interested in generalizing the lessons in (Mehrotra et al., 2019) from music to video streaming. There are important contextual differences between the two types of platforms that make this generalization far from obvious. See Table 4.1 (top) for a summary of key differences.

First, content length is a difference linked to content type and has important behavioral consequences (Jebara, 2019). Second, the music streaming domain has settled around half a dozen actors that each provide about the same deep catalog of music. But the opposite is happening in video streaming, where a plethora of platforms each have a few thousand movies and series available at any given time, with little to no content overlap between platforms (Hern, 2021). Third, the relative scarcity of content and plurality of paid subscription services encourage a strong return to piracy in 2019–2022 (Feldman, 2019; MUSO, 2022). This rise in fragmentation and piracy encourages video streaming actors to (i) quickly and accurately guide *decisive* users to the content they had in mind

Table 4.1: Contrasting music streaming and video streaming (top), and key differences in experimental setup (bottom).

	Music (Mehrotra et al., 2019)	Video [this chapter]
Content length	3–5 min	45 min–2 hrs
Catalog size[1]	> 70 million	> 5 thousand
Piracy[2]	1 pm	7.5 pm
In-app survey design		
Intent identification	One-on-one interviews with 12 users	User experience specialists
Platform	Mobile	Browser
Timing	Coming back to the homepage	On the homepage for 7 seconds
Intent	One per session	Multiple per session
Survey rate	NA[3]	20%
Response rate	4.5%	3%
Very Satisfied users	33%	44%

within a shallow catalog (compared to music), and (ii) provide a customized and seamless user experience for its *explorative* users looking for inspiration (via recommendations, personalized newsletters, etc.), in contrast with its illegal video streaming counterpart. To mirror this situation, we formulate the assumption that there exist two groups of intents, namely decisive and explorative, and show the essential role they play in video streaming platforms.

We follow (Mehrotra et al., 2019)'s methodology and adapt it for video streaming, in order to assess whether intent can indeed bring context back to explicit feedback. We adapt the original study to Videoland,[4] a video streaming platform in The Netherlands with over 1 million users. Two key differences in our experimental setup are that we use a browser (instead of a mobile app) and account for multiple intents per session (instead of only one); see Table 4.1.

This replicability study follows the ACM definition (different team, different experimental setup) (Ferro and Kelly, 2018). This study is an attempt at replicating and generalizing a large portion of the experimentation pipeline: we cover data collection, survey design, data preprocessing, data enrichment,

[1]Similar to the average for the EU competition in the video domain (Grece and Jiménez Pumares, 2020) and international competition in the music domain (Amazon, 2020; Apple, 2022; Deezer, 2022; KKBox, 2022).

[2]Average number of accesses to pirate sites per month and per internet user in the EU+UK in 2017–2020 for the respective video and music domains (Garcia-Valero et al., 2021).

[3]From the original study we know that 3 million US Spotify iPhone app users were sampled (Mehrotra et al., 2019). We could not find an official number on the US Spotify iPhone app users in 2019.

[4]`https://www.videoland.com/nl/`

modeling, and interpretation.

4.1.3 Insights

In this replicability study of (Mehrotra et al., 2019), we find that for the most part, the conclusions drawn for the music streaming domain also hold in the video streaming domain, both on the data analysis and modeling front. In particular, our contributions in terms of *generalization* are:

(1) A proposal of typical intents for a video streaming that we divide into explorative and decisive categories;

(2) An in-app survey design for a medium size streaming platform (∼1 million users), which involves some small sample adjustments; and

(3) In addition to (Mehrotra et al., 2019)'s frequentist logistic regression model, we test Bayesian multilevel models for visualization and explanations, along with random forests for improved accuracy.

In addition, our *technical* contributions to support replicability of work on intent-based satisfaction modeling are: (i) a detailed implementation of the in-app survey design; (ii) code for behavioral data retrieval from Google Analytics using BigQuery; and (iii) code for satisfaction modeling, all of which is shared at `https://github.com/rtlnl/streaming-intent-model`.

4.2 Related Work

Platforms are able to gather implicit feedback with highly granular logged data and explicit feedback via surveys. In-app surveys (Section 4.2.2) are only as granular as the number of questions asked to the user but are valuable to retrieve hidden signals that are unavailable in logged data (Section 4.2.1). Even more powerful is the fusion of explicit and implicit aspects (Section 4.2.3), in our case to assign intent and satisfaction levels to raw behavioral data.

4.2.1 Implicit feedback

In the context of interactive platforms, logged data (time on page, number of pages seen, etc.) has caught the attention of researchers early on (Morita and Shinoda, 1994). Recently, the use of implicit feedback such as click through rate (CTR) (Huang et al., 2012; Liu et al., 2015; Mehrotra et al., 2017) or dwell time (Yi et al., 2014; Kim et al., 2014) has been questioned, in favor of the concurrent use of other behavioral metrics (Mehrotra et al., 2018b; Guo and Agichtein, 2012; Dragone et al., 2019). Wen et al. (2019) highlight that, in the music domain, many users click a song but consume only a fraction of it, before skipping to the next. In the same domain, implicit feedback signals have been classified into four categories (Mehrotra et al., 2019): temporal (e.g., session length, seconds played), downstream (e.g., number of items played), surface level (e.g., number

of slates that were interacted with), and derivative (e.g., total clicks / number of items played). Derivative signals are combinations of the other three signals.

Implicit feedback signals are often used as input for, or for the evaluation of, a search or recommendation model. For example, comparing recommendation predictions with what users actually watched on different metrics and directly relating these metrics to satisfaction levels (Wang et al., 2021).

4.2.2 Explicit feedback

In the case of explicit feedback, the services of a representative sample of a user population are enlisted to obtain information on a task, such as recommendation accuracy (Beheshti et al., 2020). A survey can help reveal behavioral traits that are not apparent in the logged data. We argue there are two categories of higher order behavior on streaming platforms: *explorative* versus *decisive* (similar to *fetch*, *find* and *explore* in the domain of search for video streaming (Lamkhede and Das, 2019)). Decisive behavior refers to a session where the user already knows what she wants to stream and it is typically addressed in search (Lamkhede and Das, 2019). Exploration can be defined as the experience of finding and consuming content that was previously unknown to the user (Garcia-Gathright et al., 2018b). In the music streaming domain, surveys have shown that exploration is a complex time-varying personal need (Lee and Price, 2015), nurtures user retention (Caldwell Brown and Krause, 2016), and deeper social connection (Leong and Wright, 2013).

A major drawback of surveys is their inherent *response bias*: the response rate of satisfaction surveys is low because users have to deviate from their intent of consuming content in order to provide feedback (our response rate was 3%, compared to 4.5% in (Mehrotra et al., 2019), 4.6% at Spotify over emails (Garcia-Gathright et al., 2018b), and 2% at Google for individual item surveys (Christakopoulou et al., 2020)).

The willingness to participate in a survey is dependent on hidden factors such as time-on-hand, satisfaction with the platform in the first place (see the satisfaction distribution in Figure 4.3 and in (Christakopoulou et al., 2020; Garcia-Gathright et al., 2018b; Mehrotra et al., 2019)), etc. As a result, datasets collected through surveys have missing-not-at-random (MNAR) data (Steck, 2010). If data is available on who was shown the survey but did not respond, MNAR can be corrected for with inverse propensity scoring or multi-task neural networks (Christakopoulou et al., 2020).

Recently, a new type of item-satisfaction survey emerged, e.g., item recommendation satisfaction surveys on YouTube with a Likert scale (Youtube, 2019). Also notable is the trend of the *not interested* button on a recommended item, which is well entrenched in the search and recommendation domain (Chen et al., 2000), on platforms such as YouTube (Youtube, 2022), Twitch (Twitch, 2022), and TikTok (TikTok, 2020), with all three claiming it will help future recommendations. Such item-surveys suffer even more from response bias and thus motivate a new research field of sparse user-item pairs and debiasing (Christakopoulou et al., 2020).

A fruitful way to address the two major drawbacks of explicit feedback, response bias and sparsity, is to complement a user survey with logged interaction data from the same users, as we discuss next.

4.2.3 Connecting implicit and explicit feedback

Typically, evaluation of recommender systems is either done (i) in small-scale lab studies based on explicit feedback, (ii) in offline batch experiments with static test collections again based on explicit feedback, or (iii) through large-scale A/B tests based on implicit feedback. Garcia-Gathright et al. (2018a) argue for the use of qualitative research in user behavior to provide insight on implicit feedback metrics as a general methodological principle.

An important way of drawing links between implicit and explicit feedback is via the users' current intent (Chuklin et al., 2013). For example, Duan and Zhai (2015) study the problem of learning query intent representations for product retrieval. They propose a generative model to discover intent representations from entity search logs and show that the discovered intent representations can be directly used for improving the accuracy of product search and recommendation. Similarly, Bhattacharya et al. (2017) predict user intent from a user's task context and combine it with a frequency-based graphical model to recommend reports to users of a business analytics application.

Recent workshops provide a rich palette of examples of capturing and mining intent from user interactions (Bulathwela et al., 2020; Mehrotra et al., 2018a). Key domains where intent is an important feature for satisfaction prediction include: (i) e-commerce, where, for example, (Su et al., 2018) uncover different intents, find that different intents lead to different interaction behavior, and try to predict satisfaction from interaction signals, while Hendriksen et al. (2020) show that purchase intent prediction for identified (as opposed to anonymous) users can dramatically reduce friction; (ii) movie recommendation, where, for example, Chen et al. (2020) capture multiple intents from a (single) user's sequential behavior to guide the recommender to provide results that are diversified based on the intents discovered; (iii) news search and recommendation, where, for example, Lefortier et al. (2014a) discover that intents may shift dramatically based on real-world events and that user satisfaction may be hurt if the recommender does not shift with the shifting intents; (iv) search in video streaming platforms, where, for example, Lamkhede and Das (2019) show that search intents are markedly different from search intents behind web search queries and that new challenges arise from the unavailability of an item that a user is keen to watch; (v) point-of-interest recommendation on maps, where, for example, Omidvar-Tehrani et al. (2020) mine implicit intents by iteratively identifying groups of like-minded users and thereby increase user satisfaction; (vi) car GPS trajectories, where, for example, Snoswell et al. (2021) use reinforcement learning to discover unobserved behavior intents; and, finally, (vii) advertiser satisfaction prediction, where, for example, Guo et al. (2020) jointly model advertiser-side intent and advertiser satisfaction with attention mechanisms and recurrent neural networks. Other key aspects for which intent is an important predictor for user

satisfaction include search result page organization (Lefortier et al., 2014b) and ranking adjustments for different (inferred) needs for result diversity (Chuklin et al., 2013).

Identifying intents in search and recommendation can be a mix of supervised and unsupervised tasks that can involve users directly via interviews (Mehrotra et al., 2019) or research teams internally. In task-oriented dialogue systems, the task of intent is usually addressed as a supervised learning problem (Pei et al., 2021). Finally, Lin et al. (2020) discover new intents based on a catalog of pre-existing human-identified intents.

In the domain of entertainment, a seminal study at Pinterest found that not only intent was related to satisfaction, but that – using a simple logistic regression classifier – intent can be predicted quickly during a session (Cheng et al., 2017). On music streaming platforms, a study by Mehrotra et al. (2019) linked satisfaction with intent via a user survey and behavioral data on a music platform. This study is the most detailed one we found on the topic of intent-satisfaction modeling. This study's individual intents and behavioral data signals (such as *To play music in the background* or *songsPlayed*, respectively) raised questions about possible video domain counterparts.

To the best of our knowledge, there is no open dataset for intent-satisfaction modeling and no study of the effect of intent on satisfaction has been published yet for the video streaming domain. In this work we consider both implicit and explicit feedback to replicate and generalize (Mehrotra et al., 2019) from music to video streaming. We generalize to the video domain by proposing video-specific intents and a detailed implementation of the survey design. We replicate models with binarized satisfaction levels as outputs, behavioral data and optionally intent as input, thus testing whether intent can help to better predict satisfaction levels. We use (hierarchical) logistic regression as in the original study and further look at random forest models to optimize for accuracy and Bayesian models for interpretability.

4.3 Replication Setup for Video Streaming

Our aim is to verify if on a video streaming platform – like in the music streaming domain – behavioral data coupled with intent predicts satisfaction more accurately than behavioral data alone. To this end, we replicate the methodology of (Mehrotra et al., 2019) and adapt it to video streaming. We compare and contrast two specific music and video streaming settings, before explaining our replication design choices. We then describe our available data, acquired via in-app survey and behavioral data on the platform. Finally, we describe our satisfaction prediction model, with or without intent as input.

4.3.1 From music streaming to video streaming

For our replicability study we contrast a specific music streaming platform, Spotify, which provided the context for (Mehrotra et al., 2019), and a specific video

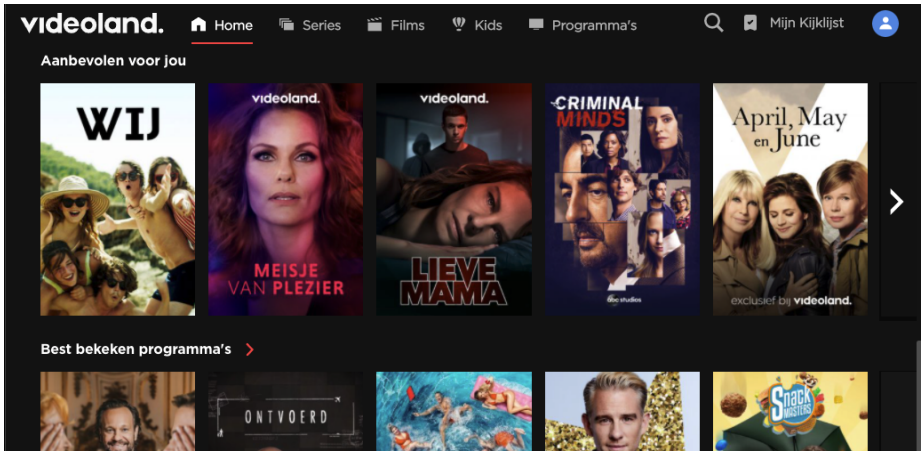


Figure 4.2: Videoland homepage with its (personalized) strips.

streaming platform, Videoland. Spotify is one of the largest music streaming platform with 180 million paid subscribers and over 70 million tracks. The most salient differences with Videoland, a streaming platform in The Netherlands with a little over 1 million users, are listed in Table 4.1. Videoland has a few thousand titles (movies, series, TV programs) with a mix of in-house productions, rotating external content, and live TV (RTL TV channels).

After a two weeks free trial, Videoland requires users to subscribe to one of three tiers. Both Spotify and Videoland require users to log in to use their platform on smart TVs, smartphones or computer browsers (and other devices for Spotify such as smart speakers). This guarantees access to identifiable behavioral data.

At Videoland, behavioral data varies greatly between device types (smart TVs, smartphones or computer browsers). Like in the replicated paper (Mehrotra et al., 2019), we focus on a single device type so as to reduce noise. TV is our most used device but is not suited for surveys, due to the laid-back context and difficulty of typing with a remote. We chose our second-most-used device: desktop browser (10% of Videoland sessions), instead of TV or smartphone (as in (Mehrotra et al., 2019)). We conduct in-app surveys with Usabilla and retrieve behavioral data via Google Analytics and BigQuery.

To manage both survey and behavioral data privacy, Videoland displays consent banners, uses a consent management system, and user preferences to allow individual user tracking limits, in accordance with GDPR regulations (European Commission, 2016).

Like in (Mehrotra et al., 2019), the homepage is the focus of our analysis. As detailed in (Semerci et al., 2019), at Spotify, each strip is either personalized or editorial and the order of strips is purely personalized for each session, at the time of the study replicated here. For Videoland, the homepage is where most people land (71% of users, during the survey period) and it is where the platform puts most effort on guiding the user to their desired content. It is populated

with recommended (Gutierrez Granada and Odijk, 2021) and editorial content. The homepage provides direct access to a search bar and a genre catalog at the top, a "continue watching" slate, a few live TV slates, and a mix of editorial and personalized slates (see Figure 4.2). The homepage layout (i.e., the strip order) is changed daily by human editors, aided with slate popularity models (corrected for position bias).

4.3.2 Survey and experimental design

Mehrotra et al. (2019) perform intent surveys in two stages: (i) intent identification, and (ii) a large-scale in-app survey. The first stage is intended as a way to discover intents of users. Mehrotra et al. (2019) held in-depth one-on-one interviews with twelve users on-site. To discover intents on Videoland, we collaborate with our user experience specialists, who have conducted numerous in-app, email, on-phone, and on-site interviews and surveys on topics surrounding intent. With them, we identified eight intents in two groups, described in the next section. In our in-app survey, we allow users to specify other intents that we might have missed in an "others" field (see Section 4.5.1, for the results).

The second step, the in-app survey, is the core of (Mehrotra et al., 2019) and of our replicability study. The major choice here is where and when to show the survey to the user. While replicating the work on a different platform, we need to reconsider this choice below.

When opening the Spotify mobile app, the user does not always land on the homepage. Thus, the reason for presence on the homepage must not be deliberate. This forced Mehrotra et al. (2019) to add an intent "Homepage is the first screen shown (i.e., default screen)". On the Videoland web app, most users land on the homepage (72% of users, during the survey period). Another fraction lands on the page of a content item. At Spotify, users switch back and forth between pages and tend to see the homepage in the middle of the session. On Videoland, most users start with the homepage, select and watch content, before closing the web app. This difference is strongly linked to the content type: listening to music can result in a lengthy session with dozens of music plays, whereas video streaming sessions tend to be dedicated to one movie or one series (thus little interest in returning to the homepage in the middle of a session).

Mehrotra et al. (2019) show the in-app survey whenever a user comes back to the homepage from another page. While it is desirable to survey users in the middle of a session in order to measure their satisfaction, this particular setup is not possible at Videoland. One possibility would have been to show the survey in between series episodes, but this was quickly discarded as being highly intrusive by our user experience researchers. We opt for the next best approach: showing the survey after having been on the homepage for seven seconds (the mean survival time of a user on the homepage, whether the user left the platform or clicked on an item). We look at the impact of that choice in Section 4.7.

Our survey, and thus the study as a whole, was conducted between November 18, 2021 and January 20, 2022. For every user logging in, there was a 20% chance of being surveyed. Each user is shown the survey at most once to avoid

Table 4.2: Behavioral variables obtained from traffic data.

	Behavioral metric	**Description**
Temporal	timeToFirstTrailer	Seconds to the first trailer played
	timeToFirstPlay	Seconds to first content play
	sessionLength	Session length in seconds
Down-stream	numTrailerPlays	Number of trailers played
	numPlays	Number of full content played
Surface level	nStrips	Number of strips seen
	nSearches	Number of content searches
	nSeriesDescr	Number of series description pages
	nMoviesDescr	Number of movies description pages
	nAccounts	Number of clicks on account icon
	nProfileClicks	Number of clicks on *manage profile*
	nBookmarks	Number of bookmarked items

pushing the survey several times to the same user (in line with (Mehrotra et al., 2019)).

4.3.3 Data collection

Next, we show the variables gathered at the session-level from two sources, namely interactions on the platform and an in-app survey.

Behavioral variables

Behavioral variables are obtained on the website at the session level (see Table 4.2) and can be grouped into temporal, downstream, and surface level signals (cf. (Mehrotra et al., 2019)). They refer to, respectively, time related events, streaming related events, and user interface interaction events. Our behavioral variables are similar to the replicated study, with the exception of *derivative signals* (Mehrotra et al., 2019), which are absent from our study. They are ratio combinations of other signals and therefore would exhibit high collinearity with some other variables in a regression model.

Note that we measure sessionLength as the difference between last and first user interaction. That last user interaction can be any surface level interaction, but we do not receive a log when a user closes her Videoland browser tab. Additionally, by default, Google Analytics creates a new session after 30 minutes of inactivity. The remainder of the implicit feedback signals are exact measures. We complement the behavioral variables with survey data to reveal user satisfaction and intent.

Table 4.3: Possible intents to be selected by survey respondents.

	Intent	Description
Explorative	new	I am looking for something new to watch
	genre	I am looking for a genre (e.g., action, comedy)
	watchlist	I want to look at my watchlist
	addwatchlist	I want to add something to my watchlist
Decisive	continuewatching	I want to continue watching a series/film where I left off
	livetv	I want to watch live TV
	catch-up	I want to catch-up on an episode I missed
	specifictitle	I am looking for a specific title

In-app survey variables

During the in-app survey (after seven seconds spent on the homepage), we ask two questions.[5] Namely,

(1) "How happy are you with your experience on the homepage today?" with satisfaction levels of 1 to 5 visualized using smiley faces (😎😟😐😊😍). In (Mehrotra et al., 2019), this question was answered on a numeric Likert scale from 1 to 5. We opted for emojis because our user experience specialists reported better results due to the more intuitive cues. We then ask

(2) "Why are you using the homepage today?" with eight multiple choice answers (see Table 4.3).

We divide intents into two main groups: *decisive* and *explorative*. Decisive users tend to arrive on the platform knowing what they want to watch. The exploration-seeking group indicates the opposite: the user is expecting the platform to help them decide what to watch. Mehrotra et al. (2019) allow users to choose only one intent. By letting the user choose one or more intents, we show that a user can have a mixture of intents for the same session (see Section 4.5.1). Additionally, we add an "others" field, to let users answer with their own words (as in (Mehrotra et al., 2019)). Mehrotra et al. (2019) analyzed the others field with a Bayesian non-parametric model (dd-CRP), in order to extract salient intents from free text. In the results section we report on the lack of signal in that data in our replicability study. We therefore did not algorithmically extract intents from the "others" field.

[5]See screenshots in Appendix C.2.

4.4 Replication of Satisfaction Models

In this section we describe our replications of the original satisfaction models with and without intent (Mehrotra et al., 2019), before describing our own models and the training setup.

4.4.1 A satisfaction model

Our satisfaction models are exactly aligned with (Mehrotra et al., 2019). We start with the simplest possible satisfaction model and iteratively add complexity. Each session on Videoland is linked to its corresponding survey data and a satisfaction level $y \in \{1, 2, 3, 4, 5\}$, in increasing order of satisfaction. Following (Mehrotra et al., 2019), we construct binarized satisfaction level vectors over all surveyed sessions:

$$\mathbf{y}_{overall} = \mathbb{1}_{\hat{y} \geq 4}, \quad \mathbf{y}_{satisfied} = \mathbb{1}_{\hat{y}=5}, \quad \mathbf{y}_{dissatisfied} = \mathbb{1}_{\hat{y}=1}, \tag{4.1}$$

with $\mathbb{1}_{(\cdot)}$ an indicator function, allowing for the use of binary satisfaction prediction models and to focus on different user groups.

A logistic regression model [w/o intent].[6] The most straightforward regression model can estimate satisfaction levels $\mathbf{y}$ via a logit link:

$$\text{logit}(\mathbf{y}) = ln\left(\frac{\mathbf{y}}{1-\mathbf{y}}\right) = \beta_0 + \sum_j \beta_j \mathbf{b}_j, \tag{4.2}$$

with β_0 the intercept, $\{\mathbf{b}_1; \ldots; \mathbf{b}_j; \ldots; \mathbf{b}_J\}$ the behavioral variables and $\{\beta_1; \ldots; \beta_j; \ldots; \beta_J\}$ their respective estimates.

Adding intent [w intent]. The model that we have just described does not include context: a user might be interested in adding elements to their watchlist for a later viewing session, but does not have time to watch content. In that case, a low number of minutes seen and a low number of video plays need not be bad indicators. As a next iteration, context and thus intents can be added as parameters,

$$\text{logit}(\mathbf{y}) = \beta_0 + \sum_j \beta_j \mathbf{b}_j + \sum_k \delta_k \mathbf{d}_k, \tag{4.3}$$

with $\{\mathbf{d}_1; \ldots; \mathbf{d}_k; \ldots; \mathbf{d}_K\}$ intents and $\{\delta_1; \ldots; \delta_k; \ldots; \delta_K\}$ their respective estimates.

One regression per intent [catch-up, ...]. Alternatively, one could consider fitting one model per intent d, reverting back to Eq. 4.2:

$$\text{logit}(\mathbf{y}^d) = \beta_0^d + \sum_j \beta_j^d \mathbf{b}_j^d. \tag{4.4}$$

This formulation is insightful to assess satisfaction levels of different session groups but ignores possible interaction effects between intents. It is also problematic in our small sample setting: some intents are only represented by a few

[6] In square brackets we include the labels that we use to refer to these models in Table 4.4.

hundred datapoints. This formulation does not measure the relative effect of a certain intent over another.

A global intent model [multiLevel]. We revert back to a single frequentist multilevel model (Krull and MacKinnon, 2001), that measures the effect of each intent as a group level effect, with a random intercept δ_k:

$$\begin{aligned} \text{logit}(\mathbf{y}) &= \delta_k + \sum_j \beta_j \mathbf{b}_j \\ \delta_k &\sim N\left(\mu_\delta, \sigma_\delta^2\right). \end{aligned} \tag{4.5}$$

This time, we clearly model a hierarchical structure in the data and can assess group-level (intent-level) marginal satisfaction effects.[7]

4.4.2 Further satisfaction models

To achieve higher accuracy, we use XGBoost, a common implementation of gradient boosting decision trees (Chen and Guestrin, 2016), with a logistic regression objective. XGBoost is a strong performer on tabular data, even when compared against recent transformer models adapted to tabular data (Gorishniy et al., 2021; Borisov et al., 2021).

For increased model interpretability, we opt for Bayesian satisfaction models with the same specifications as the frequentist versions above:

$$\begin{aligned} \text{logit}(\mathbf{y}^d) &= \beta_0^d + \sum_j \beta_j^d \mathbf{b}_j^d \\ \beta_j^d &\sim N\left(\mu_j, \sigma_j^2\right). \end{aligned} \tag{4.6}$$

They allow for the estimation of entire marginal posterior distributions and thus more granular interpretability. We keep to a simple Bayesian logistic regression per intent with with population-level effects only; the focus is on explanation, rather than building a holistic prediction model. We leave more sophisticated models (e.g. varying slope and / or intercept, temporal, neural models) for future work on predicting intent online or offline (see Section 4.7 on future work).

4.4.3 Training, evaluation and hyperparameter tuning

We recall the available data: behavioral data, user metadata, and survey data (intent and satisfaction level). The original study (Mehrotra et al., 2019) does not compute uncertainty intervals and we did not have access to their training regime, we thus opted for our own. The data is split into training and test sets in $k = 5$ folds, in order to provide out-of-sample estimates (Vehtari et al., 2016) and confidence intervals. The intent-specific models are trained on subsets of the data that contain each specific intent and, thus, each has its specific

[7] Given that this is a general linear mixed model, we have to approximate log-likelihood. We use the reliable adaptive Gauss-Hermite algorithm that takes the form of a Laplace approximation (Ju et al., 2020), by setting the integer scalar parameter to 1 (Bates et al., 2015).

5-fold split. For XGBoost we split each training set into a training and a validation set (with an 80/20% ratio) to tune the hyperparameters: *mphmax_depth* $[3; 10]$, *min_child_weight* $[1; 10]$, *subsample* $[0.5; 1]$, and *colsample_bytree* $[0.5; 1]$ (see documentation (Chen et al., 2021)). Regarding the Bayesian models, we checked for chain convergence in two ways: (i) visually with chain plots, and (ii) quantitatively with Rhat.[8] We assessed relative goodness-of-fit with leave-one-out cross-validation estimation with Pareto Smoothed Importance Sampling (PSIS) (Vehtari et al., 2015). We evaluate on the same metrics as in (Mehrotra et al., 2019): accuracy, precision, recall, and F1 score. To calculate these confusion matrix related metrics, predictions in the $[0; 1]$ range have to be binarized at a certain threshold. Given the imbalance in the data (see Figure 4.3), we refrain from using a heuristic 0.5 threshold, and instead use a threshold-moving technique at inference time, based on the F1 score, to balance precision and recall for each model and at each Likert-Scale binarization (*Overall*, *Satisfied* and *Unsatisfied*) Fernández, 2018, p. 53–55. This is an inference-time task and we distinguish it from hyperparameter tuning to be done on validation sets.

4.5 Data Analysis Replication

In this section we replicate the data analysis and visualizations from (Mehrotra et al., 2019) and assess whether the original conclusions generalize from the music to the video domain. We produce three plots in line with (Mehrotra et al., 2019), two of which are focused on survey results. The last plot mixes behavioral and survey data. For comparison purposes, the visualizations are kept similar to the original study.

4.5.1 Survey results

The response rate was 3%, with a survey rate of 20% from logged-in users after 7 seconds on the home page, we ended up with about $3,350$ sessions. 21% of these users responded to the first (satisfaction) but not to the second (intent) question and are thus not modelled in Section 4.6, leaving a total of $2,632$ survey responses in our datasets. The most selected intents were *continuewatching* (see Table 4.3). On average, users have 2.18 intents per session. Only 3.6% users added a remark in the "other" section. We thus decided to read them all. They were for a minor part bug reports, enunciating an existing intent in the list, some grateful or ungrateful comments, or asking for content to appear on Videoland. Given the lack of signal on intent in the "others" section, we decided to leave it out of this study.

Figure 4.3 displays the satisfaction levels across all sessions and reveals that most users who answered the survey are satisfied with the platform. This is in line with the setup in (Mehrotra et al., 2019), which let users rate their satisfaction with numbers from 1 to 5 instead of emojis in our case. Also note that quite satisfied users ($y \geq 4$) are overrepresented compared to their less

[8] Code and analysis available at `https://github.com/rtlnl/streaming-intent-model`.

satisfied neighbors ($y < 4$). This might be a sign of MNAR in our dataset (see Section 4.7 for a discussion on the topic).

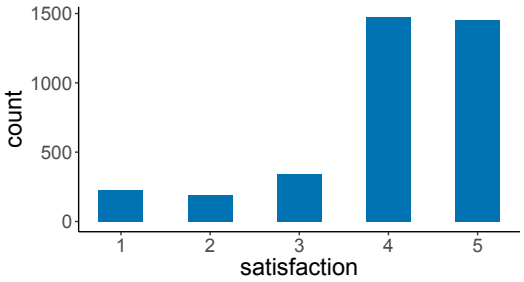


Figure 4.3: Our imbalanced dataset: distribution of Likert-scale satisfaction levels for all surveyed users and across intents.

Next, we look at relationships between satisfaction level and intent (Figure 4.4). We draw a violin plot as in (Mehrotra et al., 2019). From left to right, we notice that decisive users looking for live TV or a specific title have the most spread out satisfaction distribution; users who add content to their watchlists have the lowest representation of satisfaction levels 1 and 2; users who are looking for inspiration via new genres or new titles are the least satisfied (i.e., they have the highest concentration of levels 1, 2 and 3). Following our earlier discussions of rising fragmentation and piracy in the video streaming domain, it might be necessary to look closely at these unsatisfied decisive users and in particular those looking for a specific title, for which piracy or an alternative platform is the most natural substitute. In the following section we further investigate these intents in relation with the interaction data.

4.5.2 Correlation between survey and behavioral data

We recontextualize the raw behavioral data with users' revealed intents. The Pearson correlation plot in Figure 4.5a confirms a few intuitions. Users who intend to continue watching an episode interact the least with the platform, but it does not prevent them from watching a lot of content for long periods of time.

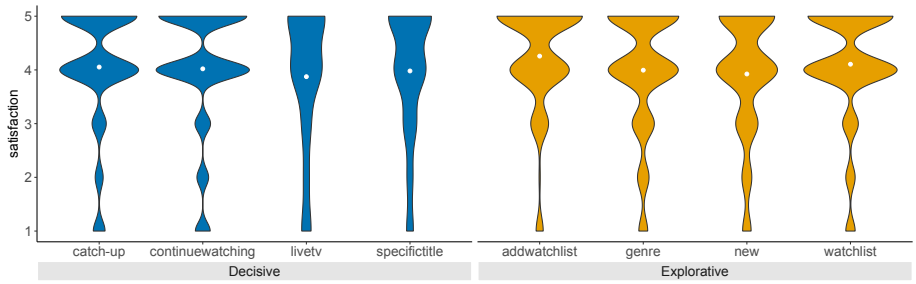


Figure 4.4: Satisfaction levels per intent and by intent group (dot indicates the mean).

Users who are looking for something new to watch interact with a number of features on the platform and watch a lot of trailers. They do not tend to find more content to watch than other users (as indicated by the lack of correlation with numPlays and sessionLength). For comparison, in music streaming, at Spotify, users even tend to play fewer songs for less time (negative correlations) when they desire "to discover new music to listen to now" (see Figure 4.5b).

We note one salient difference with the original interaction plot at Spotify: users whose intent is "to explore artists or albums more deeply" comparatively play songs for a longer time and do not have a particularly high number of interactions with the user interface. In other words, *in the music domain, users explore by playing. In the video domain, users explore by interacting with the platform.* The main reason is probably that a song listener can afford to listen and try out full 10–15 songs while a user watches a single movie or series episode.

Taking a step back, these disparities highlight the differences between the blind exploration phase in the music domain (limited interaction) and the more tedious, active exploration phase in the video domain. Thus, it seems that the video medium itself calls for exploratory user hand-holding. We emphasize the need to provide a thoroughly thought out and personalized user experience to a video streamer looking for inspiration, otherwise the video platform risks loosing the customer to piracy or a competing video streaming platform.

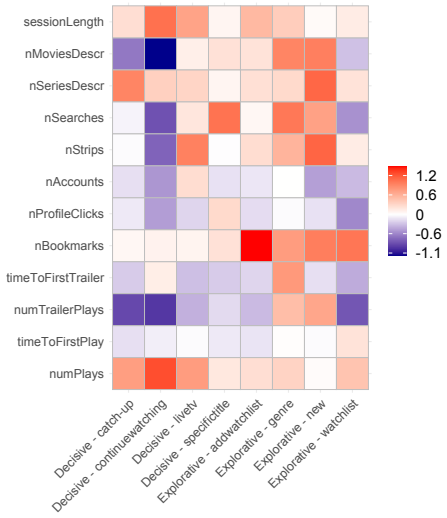
4.5.3 Upshot: Music versus video streaming

In replicating (Mehrotra et al., 2019), we collected data in a completely different streaming platform and we adapted the survey design to our context and needs (the main differences are recorded in Table 4.1). We found Mehrotra et al. (2019)'s data analysis to be replicable in several aspects. We observe the same imbalance in satisfaction levels, with levels 4 and 5 overly represented. Satisfaction by intent is less comparable, since we formulated video streaming intents. Unlike in (Mehrotra et al., 2019), we find that two intents clearly have a higher amount of dissatisfied users, namely the decisive users looking to watch *livetv* or a *specifictitle*. Overall, Figure 4.4 and 4.5a confirm the learnings from (Mehrotra et al., 2019), namely that users' satisfaction level and behavior are different depending on their intent.

Like in the original study, our conclusions might be influenced by response bias. For example, we typically observe little use of the bookmarking system on the platform. But our survey-behavioral dataset showed an unusually high number of users adding elements to their watchlist. We assume that users who use the watchlist are more likely to respond to the survey or maybe even that some users discovered the existence of the watchlist button after seeing it as an intent option in the survey: the average of 0.03 bookmarks per session for all sessions during the survey period jumps to 0.09 for our surveyed cohort who made it to the second question and saw the bookmarking intents.

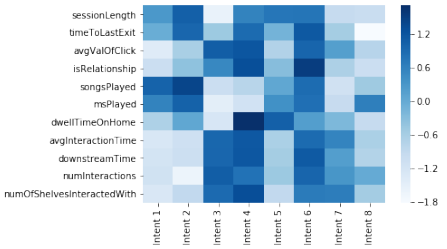
Table 4.4: Replication of (Mehrotra et al., 2019) with added mean and standard deviation over 5-fold cross-validation for the three binarizations of the $y \in \{1, 2, 3, 4, 5\}$ satisfaction score (outcome variable) and four metrics (accuracy, precision, recall, F1 score).

	Accuracy	**Precision**	**Recall**	**F1**
Method	Overall ($\mathbb{1}_{\hat{\mathbf{y}} \geq 4}$)			
w/o intent	0.81 ± 0.03	0.81 ± 0.03	1.00 ± 0.00	0.89 ± 0.02
w intent	0.81 ± 0.03	0.81 ± 0.03	1.00 ± 0.00	0.89 ± 0.02
multiLevel	0.81 ± 0.03	0.81 ± 0.03	1.00 ± 0.00	0.89 ± 0.02
XGB w/o intent	0.83 ± 0.03	0.83 ± 0.03	0.99 ± 0.01	0.90 ± 0.02
XGB w intent	0.82 ± 0.02	0.83 ± 0.02	0.98 ± 0.01	0.90 ± 0.01
catch-up	0.82 ± 0.04	0.82 ± 0.04	1.00 ± 0.01	0.90 ± 0.03
continuewatching	0.81 ± 0.03	0.81 ± 0.03	1.00 ± 0.01	0.89 ± 0.02
livetv	0.79 ± 0.16	0.80 ± 0.14	0.97 ± 0.08	0.87 ± 0.10
specifictitle	0.87 ± 0.22	0.87 ± 0.22	1.00 ± 0.00	0.91 ± 0.14
addwatchlist	0.95 ± 0.16	1.00 ± 0.00	0.95 ± 0.16	0.97 ± 0.11
genre	0.83 ± 0.32	0.83 ± 0.32	0.90 ± 0.32	0.86 ± 0.31
new	0.74 ± 0.07	0.74 ± 0.07	1.00 ± 0.00	0.85 ± 0.05
watchlist	0.84 ± 0.08	0.84 ± 0.08	1.00 ± 0.01	0.91 ± 0.05
	Satisfied ($\mathbb{1}_{\hat{\mathbf{y}}=5}$)			
w/o intent	0.46 ± 0.04	0.43 ± 0.04	0.95 ± 0.03	0.59 ± 0.04
w intent	0.47 ± 0.04	0.43 ± 0.04	0.94 ± 0.03	0.59 ± 0.04
multiLevel	0.45 ± 0.04	0.42 ± 0.04	0.96 ± 0.02	0.59 ± 0.04
XGB w/o intent	0.63 ± 0.04	0.53 ± 0.04	0.78 ± 0.07	0.63 ± 0.04
XGB w intent	0.57 ± 0.06	0.49 ± 0.05	0.83 ± 0.10	0.61 ± 0.06
catch-up	0.41 ± 0.07	0.40 ± 0.07	0.97 ± 0.04	0.56 ± 0.08
continuewatching	0.41 ± 0.04	0.40 ± 0.04	0.97 ± 0.02	0.56 ± 0.04
livetv	0.45 ± 0.17	0.44 ± 0.18	0.94 ± 0.12	0.58 ± 0.18
specifictitle	0.60 ± 0.22	0.60 ± 0.22	1.00 ± 0.00	0.73 ± 0.16
addwatchlist	0.55 ± 0.50	0.55 ± 0.50	0.60 ± 0.52	0.57 ± 0.50
genre	0.42 ± 0.29	0.38 ± 0.31	0.70 ± 0.48	0.48 ± 0.36
new	0.41 ± 0.04	0.37 ± 0.05	0.96 ± 0.04	0.53 ± 0.05
watchlist	0.42 ± 0.09	0.40 ± 0.10	0.94 ± 0.06	0.56 ± 0.11
	Unsatisfied ($\mathbb{1}_{\hat{\mathbf{y}}=1}$)			
w/o intent	0.87 ± 0.02	0.16 ± 0.08	0.26 ± 0.13	0.19 ± 0.10
w intent	0.92 ± 0.03	0.28 ± 0.17	0.20 ± 0.15	0.21 ± 0.12
multiLevel	0.87 ± 0.02	0.16 ± 0.08	0.26 ± 0.13	0.19 ± 0.10
XGB w/o intent	0.86 ± 0.03	0.23 ± 0.13	0.38 ± 0.19	0.28 ± 0.15
XGB w intent	0.91 ± 0.03	0.31 ± 0.15	0.40 ± 0.23	0.33 ± 0.16
catch-up	0.83 ± 0.05	0.10 ± 0.11	0.24 ± 0.25	0.13 ± 0.12
continuewatching	0.92 ± 0.03	0.42 ± 0.31	0.19 ± 0.15	0.26 ± 0.20
livetv	0.23 ± 0.11	0.11 ± 0.12	0.50 ± 0.53	0.18 ± 0.20
specifictitle	0.22 ± 0.33	0.03 ± 0.11	0.10 ± 0.32	0.05 ± 0.16
addwatchlist	0.70 ± 0.35	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
genre	0.33 ± 0.38	0.15 ± 0.34	0.20 ± 0.42	0.17 ± 0.36
new	0.86 ± 0.07	0.14 ± 0.16	0.28 ± 0.32	0.17 ± 0.17
watchlist	0.75 ± 0.07	0.06 ± 0.05	0.32 ± 0.33	0.10 ± 0.09



(a) On our video streaming platform. Red and blue indicate positive and negative correlation, respectively.

Intent	Definition
Intent 1	Homepage is the first screen shown (i.e. default screen)
Intent 2	To quickly access my playlists or saved music
Intent 3	To discover new music to listen to now
Intent 4	To play music that matches my mood or activity
Intent 5	To Find X
Intent 6	To find music to play in the background
Intent 7	To save new music or follow new playlists to listen later
Intent 8	To explore artists or albums more deeply.



(b) On a music streaming platform (table and visualization taken from (Mehrotra et al., 2019)).

Figure 4.5: Pearson Correlation (×10) plots between intents (x-axis) and behavioral data (y-axis).

4.6 Model Replication

We replicate multiple frequentist logistic regression satisfaction models: without intents, with intents, per intent, and with an intent as a hierarchical level, all as in (Mehrotra et al., 2019). Going beyond (Mehrotra et al., 2019), we additionally report on XGBoost predictions with and without intents; we then fit one Bayesian logistic regression per intent and report on marginal posterior distributions for each behavioral metric.

4.6.1 Satisfaction prediction results

Table 4.4 displays the prediction results with standard deviations using 5-fold test sets. The binarization of intent plays a predominant role in the results (*Overall*, *Satisfied*, *Unsatisfied*). For the *Overall* and *Satisfied* binarizations, the effect of adding intent to the model is not clear: *w/o intent* versus either *w intent* or its random-effects counterpart *multiLevel*. The per-intent models do not deliver satisfying results over the global model. We also find that, contrary to expectations, XGBoost does not always perform best; we believe that this is due to the linearity in the data, which is accurately modeled by logistic regression. Turning to classifiying *Unsatisfied* users, differences between results are more stark, especially for Accuracy (non-overlapping standard deviations).

This implies that dissatisfied users are the ones who deliver the most signals to researchers. Hence, we focus on dissatisfied users. Notably, *continuewatching* (when a user decisively continues watching a show she started) is the best performing per-intent model. That is, *continuewatching* users that are dissatisfied have very recognizable behavior. Finally, for predicting dissatisfied users, adding intents to either the plain logistic model (*w/o intent*) or the XGBoost model (*XGB w/o intent*) leads to performance increases. This confirms the important role of intent in user satisfaction across the music and video domains at least for dissatisfied users.

In the following, we analyze intent specific models in more detail, via their Bayesian counterparts.

4.6.2 Bayesian marginal posteriors

Figure 4.6 examines the role of implicit feedback in satisfaction prediction, with intent factored out (given one model per intent). This figure displays marginal posterior distributions of each behavioral metric, given each of eight intent models. Note, for example, that one unit increase in the nStrips coefficient corresponds to a one unit increase in log odds ratios for satisfaction. We kept the three variables with the highest absolute median posterior draws[9] (similarly to the frequentist variable importance analysis in (Mehrotra et al., 2019)).

Given the small-data context (around $3,000$ observations), we refrain from interpreting exact odds ratios. Instead, we focus on marginal posterior distributions whose IQR does not overlap with the zero effect line. Overall for decisive intents, the more a user dwells on different pages and interacts with them instead of playing full videos or trailers, the more their satisfaction is hurt: notably *nSearches*, *nProfileClicks*, and *nBookmarks* have negative coefficients in three out of four decisive intents (see the top row of Figure 4.6). The conclusions are more mixed for explorative users. We see that users who were looking for inspiration via genre pages are rather dissatisfied if they have to do searches instead, but are happy to spend time looking at series descriptions.

[9] We withdrew divergent draws (Rhat > 1.05) and confirmed they did not prevent other estimates to converge with chain plots. Distributional outliers shown in the descriptive statistics plots (`https://github.com/rtlnl/streaming-intent-model`).

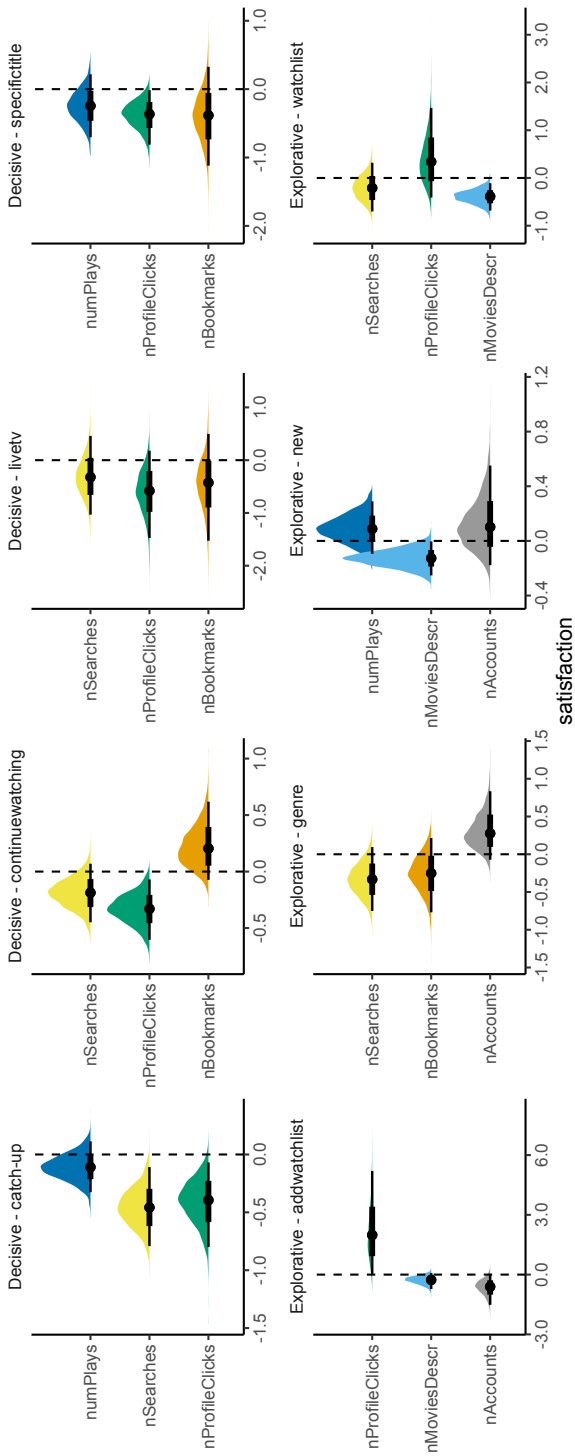


Figure 4.6: Marginal effect of behavioral variables on satisfaction; Table 4.2 provides descriptions of our behavioral variables. One Bayesian fit per intent (median and IQR in thicker marks, 99.8% of the probability density function in thinner line).

4.6.3 Upshot: Music versus video streaming

We fully re-implemented the predictive models used in (Mehrotra et al., 2019). We complemented the original study in three ways: (i) We dealt with imbalanced data by tuning inference-time thresholds (Fernández, 2018) instead of oversampling the dataset once with SMOTE (Chawla et al., 2002), thus refraining from duplicating datapoints. (ii) We computed uncertainty intervals by computing out-of-sample estimates on a rotation of five-fold different test sets (Vehtari et al., 2016). (iii) We ran XGBoost and Bayesian models, for prediction accuracy and interpretability.

The conservative measures (i) and (ii), together with a smaller dataset could be what lead to less noticeable differences across models than in the original study. It is also possible that our study expresses a reality, namely that in the video setting only dissatisfied users see their satisfaction vary with their intent. This speaks to the intuition that users responding with a 1/5 on the satisfaction scale are the ones sending the strongest signal. This motivates future research with a focus on dissatisfied users.

Overall, we could replicate the main finding of (Mehrotra et al., 2019), namely that at least for unsatisfied users intent seems to impact satisfaction levels.

4.7 Conclusions

We have replicated and generalized Mehrotra et al. (2019)'s work on intent-based satisfaction modeling, from music to video streaming. We have replicated the full experimental setup, from data collection – behavioral data and enrichment with an in-app survey – to computations. We provide our code for data preprocessing, visualization of the interactions between intents, satisfaction and behavioral data in line with the visualizations in (Mehrotra et al., 2019). Finally, we extended the modeling section with XGBoost models as standard tabular data benchmarks and per intent Bayesian models for interpretability.

4.7.1 Findings

Table 4.5 summarizes our findings in comparison to the replicated study (Mehrotra et al., 2019). We have found that in video streaming, as in music streaming, intent influences satisfaction levels together with behavioral data, although to a lesser degree than the original replicated study (Mehrotra et al., 2019). The video context also allowed us to draw new conclusions: (i) Unsatisfied users are more prone to reveal their intent via their behavior on the website (see Table 4.4). (ii) By introducing a differentiation between explorative and decisive intents, we highlight the tendency of video streamers to use the user interface for inspiration (Figure 4.5a and 4.6), whereas music streamers listen "blindly," without much interaction on the interface (Figure 4.5b), thus highlighting the higher relevance of behavioral data in the video context. (iii) Decisive users are not so keen on using the platform's personalized features and thus deserve special attention in the user experience design.

4.7.2 Broader impact

More broadly, this study reveals that it is possible to replicate a survey across different domains, device types and with smaller sample sizes. We hope this real-world small-sample replicable scenario further encourages human-scale studies in general and in the academic domain, where respondent recruitment is also prone to response bias. With regards to intents, two studies (this chapter and its replicated counterpart (Mehrotra et al., 2019)) now show that it is not enough to look at behavioral data alone to measure user satisfaction. Surveying and later predicting intents on each streaming platform help to better guide users to their goal or give users new perspectives.

4.7.3 Limitations

Our small-sample study also comes with its limitations. We surveyed respondents after seven seconds on the homepage. This means that there is a chance that the survey has influenced certain behaviors. Regarding response bias and MNAR, ideally we would have used the data on users who were shown the survey but did not answer. For future research we propose to track that data.

4.7.4 Further models

We focused on predictability (XGBoost) and interpretability (Bayesian intent model). For predictability, there is little evidence that improvement is possible with more sophisticated models, given the performance of XGBoost in the tabular data domain even in recent years (Gorishniy et al., 2021; Borisov et al., 2021). If we were to add a time aspect, such as sessions of the same user across time (i.e., longitudinal tabular data); we would consider a transformer neural network architecture (Lin and Luo, 2022). For interpretability, we could consider fitting a single Bayesian model with all intents and variables, given a bigger sample. If intents are modeled as latent hierarchical effects, the model can be useful for daily user data, where intent is not available (because no survey was shown). We could thus extend the model to all user data and predict satisfaction given behavioral data and unobserved intent.

4.7.5 Looking ahead

As to future work, we hope that this study and the materials that we share encourage researchers working in other domains to investigate, share insights on user intent and eventually try to predict them, given user behavior. We compared the setting of short songs versus long videos and revealed disparities related to the medium itself. This leaves open the effect of intent on platforms focused on longer audio content such as podcasts, short video content like TikTok, or emerging live streaming platforms like Twitch. Understanding intents and

[10] This is probably due to the sampling methodology. In (Mehrotra et al., 2019), the unsatisfied minority class is oversampled; while in the current study, the data is modelled as is.

their groupings (decisive, explorative, and maybe others) on different platforms could allow for experiences tailored to unobservable time-varying user needs as opposed to relying more on direct user feedback (clicks, scrolls, etc.). Finally, as we pointed out in our discussion of related work, a lot of previous work has highlighted explorative users; decisive users are somewhat neglected in the literature. Our study highlights the need for further research into algorithmically balancing the interests of decisive and explorative users.

4.8 Upshots for the Personalization Flow

In this third stage of the personalization flow, we observe users as opposed to nudging them. Our conclusions are more mitigated than in the reproduced study (Mehrotra et al., 2019) regarding the connection between intent, behavior and satisfaction data. To answer our research question: while we are not able to reliably predict satisfaction, we are able to assess that intent influences satisfaction greatly on our video streaming platform.

Answering our research question is only possible if a platform can perform user surveys over a long period of time, collect behavioral data and connect survey and behavioral data at user-level. This step is inherent to each platform's internal organization and is an essential stepping stone. Once an organization realizes that data collection process, we hope that we can at least facilitate the rest of the process, by providing simulated data, code, experimental and survey design. Further research in this area is needed to allow us to port this study to other domains like live streaming or podcasting.

In the next chapter of this thesis and the last step of our flow, we take a step back. This time, we use behavioral data to assess how diverse recommendations are, and how this fits to the platforms norms and values.

Reproducibility

To facilitate the reproducibility of the work in this chapter, our code and survey is available at `https://github.com/rtlnl/streaming-intent-model` and in Appendix C.

C Appendices

C.1 Implementation resources

To support replicability of our work, in the video or music streaming domain and beyond, we share[11] the following resources: (i) code for behavioral data retrieval (BigQuery); (ii) code for satisfaction modeling; and (iii) a detailed implementation of the in-app survey design. We cannot share individual user data, for GDPR compliance. However, to enable others to run our code, we

[11] `https://github.com/rtlnl/streaming-intent-model`

(a) Survey pop-up 1 on the bottom-right of the Videoland homepage, after 7 seconds.

(b) Survey pop-up 2 on the bottom-right of the Videoland homepage, after 7 seconds.

Figure 4.7: Pop-up 2 shows after "next" is clicked on pop-up 1. For a translation, see Section 4.3.3.

include simulated behavioral and survey data in the repository, replicating the distributions in our dataset.

Our repository contains the libraries we use, the data preparation steps, visualization code for the plots in this chapter and some additional distribution plots. Finally, the repository contains the modeling code to reproduce our cross-testing across different test sets and chain plots of marginal posterior distributions, to check for collinearity between sampling of different chains and variables.

C.2 Survey form

Figure 4.7 shows the survey pop-ups in the original language. See Section 4.3.3 for translations.

Table 4.5: Overview of conclusions from (Mehrotra et al., 2019) compared to the current work. A checkmark indicates that the conclusion holds in the replicability study

(Mehrotra et al., 2019)	Our work
8 key user intents for music	8 different intents for video
No particular grouping	Grouped in decisive and explorative
Imbalance in satisfaction levels	✓
For unsatisfied users intent impacts satisfaction	✓
2 intents with more dissatisfied users	✓
Intent influences satisfaction levels	✓(albeit to a lesser extent)
Level of satisfaction is not linked to amount of signal in behavioral data	Unsatisfied users are more prone to reveal their intent via behavioral data[10]
Intents important when predicting user satisfaction	✓
Different interaction signals important across intents	✓
Shared learning across intents improves satisfaction model	✓(albeit to a lesser extent)
Users explore by playing	Users explore by interacting with the platform.
Blind exploration phase	Active exploration phase
Call for using user-level idiosyncrasies	Calls for exploratory user handholding
Listen "blindly," without much interaction	Tendency to use the user interface for inspiration

Normative Diversity

Pushing content to the user, changing its appearance, and measuring the user satisfaction are essential business concerns for a streaming platform. Now that we have reached the last step of our personalization flow, we take a step back. Given a platform's norms and values, can the platform measure whether the users' behavior aligns with them?

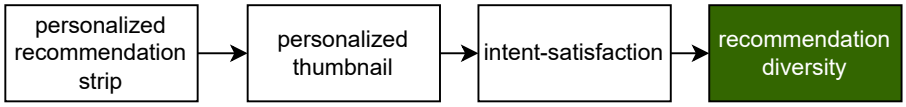


Figure 5.1: The final step of the personalization flow.

While there is emerging discussion on what a platform's norms and values can be (Helberger, 2019; Vrijenhoek et al., 2023; Zhang et al., 2023b), monitoring them quantitatively is less discussed. Instead of relating to normative concepts, existing metrics are related to objective concepts, like how item embeddings differ across a list (Liang and Qian, 2021). Moving away from existing metrics we ask,

RQ4: Can we formulate a divergence metric that measures the normative diversity of recommendations?

We propose a mathematical formulation of a diversity metric that is based on the Jensen-Shannon Divergence and that is rank-aware. We illustrate how this metric fits into an existing normative diversity paradigm. We use the MIND news dataset (Wu et al., 2020). Although it is a standard dataset, we found out we could make a worthwhile preliminary contribution: first analyze the quality of MIND and the distribution in news categories. We release code and a data preprocessing pipeline to extract concepts like text complexity, the presence of alternative voices, the polarity of the article etc. Altogether, this chapter forms a pipeline of its own, one that is dedicated to responsible recommendations.

This chapter was published at the ACM Conference on Recommender Systems (RecSys 2022) under the title "RADio – Rank-Aware Divergence Metrics to Measure Normative Diversity in News Recommendations" (Vrijenhoek et al., 2022), where it won a best paper runner up award.

5.1 Introduction

For centuries, the interplay between journalists and news editors has shaped how news items are created and how they are shown to their readers (Wu et al., 2019h). With the digitization of society, much has changed: while before, people would typically limit themselves to reading one type of newspaper, they now have a wealth of information available at the click of a button (Singer, 2014) – more than anyone could possibly be expected to read or make sense of. News recommender systems can filter the enormous amount of information available to just those news items that are in some way interesting or relevant to their users (Möller et al., 2018; Bodo, 2019). The use of news recommender systems is widespread, not just for *personalized* news recommendations, but also to automatically populate the front page of a news website (Møller, 2022), or present the reader of a particular news article with other articles about the same topic, but from a different perspective (Mulder et al., 2021). The use of news recommender systems has a wide range of benefits. They can increase engagement (Nic et al., 2018) and help raise informed citizens (Eskens et al., 2017). A news recommender system may broaden the horizons of their users through diverse recommendations, including items different from what they are used to or expect seeing. They could even foster tolerance and understanding (Ferrer-Conill and Jr., 2018; Strömbäck, 2005), and counter so-called filter bubbles or echo chambers (Pariser, 2011; Möller et al., 2018).

To realize the potential benefits of news recommender systems, much attention has been given to generating recommendations that reflect the user's interests and preferences (Karimi et al., 2018). However, with news recommenders taking over the role of human editors in news selection, they are becoming gatekeepers in what news is shown to audiences and have thus a democratic role to play in society (Lin and Lewis, 2022). As such, their evaluation has different requirements than those of other types of recommender systems (Beam, 2014; Wallace, 2018; Welbers et al., 2018; Bastian et al., 2021). Recent controversies have shown that merely optimizing for click-through rates and engagement may promote sensationalist content (Tenenboim and Cohen, 2015), and is particularly conducive to the spread of misinformation.[1] This observation is not limited to the academic literature – an increasing number of media organizations, both public service and commercial, have acknowledged the difficulties in translating their editorial norms into concrete metrics that can inform recommender system design (Grün and Neufeld, 2021; Boididou et al., 2021). News recommender systems exist in a complex space consisting of many different areas and disciplines, each with their own goals and challenges; think of balancing diversity and accuracy (Parapar and Radlinski, 2021), financial incentives (Braun and Eklund, 2019), nudging (Mattis et al., 2021) or even identifying user preferences (Lu et al., 2018; Bernheim et al., 2021) and biases (Wang and Chen, 2021). In this chapter, we focus on the process of translating normative theory (i.e., what it means for a

[1] See, for example, the alleged role Facebook played in the storming of the Capitol: `https://www.washingtonpost.com/technology/2021/10/22/jan-6-capitol-riot-facebook/`.

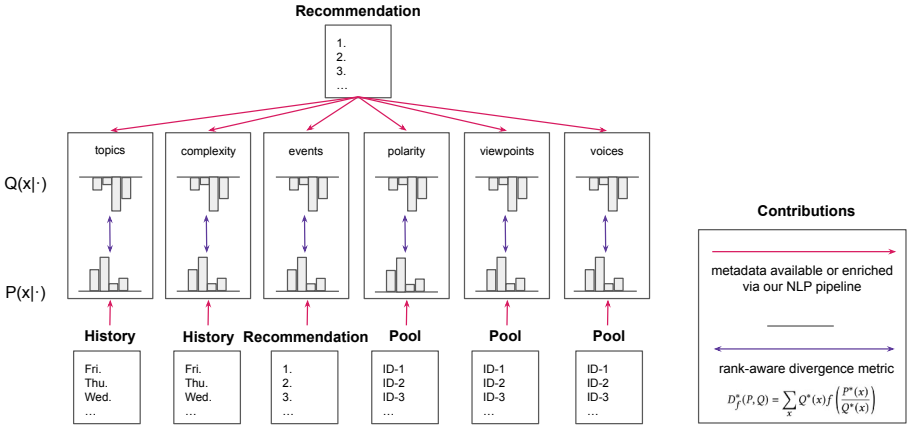


Figure 5.2: Comparing discrete diversity distributions in the context of news recommendations. First, metadata is collected in the news dataset or retrieved via our NLP pipeline (red). Discrete distributions of that metadata are then compared via a rank-aware divergence metric (purple). Recommendation set Q and the context articles P are compared with rank-aware f-Divergence.

recommendation to be diverse) into metrics that are usable and understandable for both technical and editorial purposes. We build on the work of Helberger (2019), who provides a theoretical foundation for conceptualizing diversity, and of Vrijenhoek et al. (2021), who propose a new set of metrics (DART) that reflect this theory. The DART metrics represent a first step towards a normative interpretation of diversity in news recommendations. We identify a number of improvements on these metrics: more consideration for the theory of metrics and distance functions, generalization to other normative concepts, unification under one framework, and rank-awareness. In this chapter, we focus on the mathematical aspects of a rank-aware metric, versatile to different normative concepts. We refer to our framework as the *Rank-Aware Divergence metrIcs to measure nOrmative diversity* (RADio).

Our contribution consists of a diversity metric that is (i) versatile to any normative concept and expressed as the divergence between two (discrete) distributions; (ii) rank-aware, taking into account the position of an item in a recommendation set; and (iii) mathematically grounded in distributional divergence statistics. We demonstrate the effectiveness of this formulation of the metrics by defining a natural language processing (NLP) metadata enrichment pipeline (e.g., sentiment analysis, named entity recognition) and running it against the MIND dataset (Wu et al., 2020).

Figure 5.2 illustrates the operationalization. The pipeline and the code produced for metadata enrichment and metric computation are available online.[2] The goal of RADio is not to serve as thresholds or strict guidelines for 'diverse recommendations,' but to provide developers of recommender systems with the

[2] `https://github.com/svrijenhoek/RADio`

tools to evaluate their systems on normative principles.

5.2 Related Work

We first highlight recent work on the formal mathematical work on diversity in news recommendation, before citing related work on the normative aspect of diversity. Finally we describe the gap that exists between descriptive and normative diversity.[3]

5.2.1 Descriptive (general-purpose) diversity

Diversity is a central concept in information retrieval literature (Clarke et al., 2008; Sakai and Zeng, 2019), albeit with a different interpretation than the normative diversity described in the previous section. During the development of news recommender systems, there is currently a large focus on the predictive power of an algorithm. However, this may unduly promote content similar to what a user has interacted with before, and lock them in loops of 'more of the same.' To tackle this, 'diversity' is introduced, which is typically defined as the 'opposite of similarity' (Bradley and Smyth, 2001). Its goal is to prevent users from being shown the same type of items in their recommendations list and is often expressed as intra-list-diversity (ILD) (Bradley and Smyth, 2001; Di Noia et al., 2014; Ekstrand et al., 2014; Jugovac et al., 2017; Du et al., 2021; Vargas and Castells, 2011; Castells et al., 2015; Lu et al., 2020): mean pairwise dissimilarity between recommended item lists. ILD requires the specification of a distance function between lists, and thus leaves it up to interpretation as to what it means for two lists to be distant. In theory, it could still be interpreted with a metric that accounts for the presence of different sources or viewpoints (Ekstrand et al., 2018). However, in practice, diversity is most often implemented as a descriptive distance metric such as cosine similarity between two bag-of-words models or word embeddings (Lu et al., 2020; Kunaver and Pozrl, 2017).

Other popular 'beyond-accuracy' metrics related to diversity are novelty (how different is this item from what the user has seen in the past), serendipity (is the user positively surprised by this item), and coverage (what percentage of articles are recommended to at least one user). These metrics can be taken into account at different points in the machine learning pipeline (Kunaver and Pozrl, 2017; Wu et al., 2019g). One can optimize for these descriptive notions of diversity (i) before training, by clustering users based on their profile diversity with JS divergence (Eskandanian et al., 2017), (ii) directly at training time (e.g., for learning-to-rank (Borodin et al., 2012; Vargas and Castells, 2011; Castells et al., 2015), collaborative filtering (Qin and Zhu, 2013), graphs (Gan et al., 2020; Puthiya Parambath et al., 2016) or bandits (Ding et al., 2021; Xie et

[3]This dichotomy is oftentimes referred to as normative (*what ought to be*) and positive (*what is*) statements (Hume, 1739) but can easily be confused with concepts such as positive / negative examples in machine learning. We thus opt for the more explicit normative / descriptive duo.

al., 2021)), (iii) by re-ranking a recommendation set and balance diversity vs. relevance (Chen et al., 2018) or popularity vs. relevance (Chakraborty et al., 2019), and (iv) by defining a post-recommendation metric to measure diversity for each recommendation set or at user-level (e.g., the generalist-specialist score (Waller and Anderson, 2019; Anderson et al., 2020)). With any of these four methods, a trade-off must be made between the relevance of a recommendation issued to users and the level of descriptive diversity, though there have also been studies indicating that increasing diversity does not necessarily need to negatively affect relevance (Lu et al., 2020). Nevertheless, this encouraged recent efforts in training neural-based recommenders that explicitly make a trade-off between accuracy and diversity (Raza and Ding, 2021). Also recently, there have been studies that differentiate between diversity needs of users (Wu et al., 2018).

5.2.2 Normative diversity

Diversity is extensively discussed as a normative concept in literature, and has a role in many different areas of science (Steel et al., 2018; Loecherbach et al., 2020), spanning from ecological diversity to diversity as a proxy for fairness in machine learning systems (Mitchell et al., 2020). While these interpretations of diversity are often related, they do not fully cover the nuances of a diverse *news* recommender system, the work on which stems from democratic theory and the role of media in society. Following (Helberger, 2019), we define a normative diverse news recommendation as one that succeeds in informing the user and supports them in fulfilling their role in democratic society. Out of the many theoretical models that exist in literature, Helberger (2019) describe four different models from the normative framework of democracy, each with a different view on what it means to properly inform citizen. The **Liberal** model states that eventually the users themselves know what is best for them, and primarily aims to enable personal development and autonomy. A recommender system following the Liberal model would have a strong focus on aligning with users' personal preferences. Most current recommender systems are, inadvertently, in line with this model. In comparison, the **Participatory** model takes on a more paternalistic role, as it values the common good over the individual and endeavors to know what is good for both. A recommender system following the Participatory model aims to enable users to fulfill their role as active citizens in a democratic society, informing them of important events in a way that is suitable to their needs and preferences. The **Deliberative** model fosters discussion and debate by equally presenting different viewpoints and opinions in a rational and neutral way, with the eventual goal of either reaching a consensus or agreeing on different values. Deliberative recommender systems have a strong focus on the representation of a plurality of sources, voices and opinions. Lastly the **Critical** model aims to challenge the status quo and to inspire the readers to take action against existing injustices in society. A Critical recommender system provides a platform to voices that would otherwise go unheard, and favors emotion-driven and activating content. However, it should *not* promote hate speech or conspiracy theories; those with different opinions should be seen

as opponents or adversaries, not enemies (Sax, 2022).

For more details regarding the different models, and what a recommender system following each of these models would look like, we refer to (Helberger, 2019). Since none of these models is inherently better or worse than the other, which model is followed is a decision that needs to be made by the media organization itself, and should be in line with their norms and values.

Conceptualized based on these models, the DART metrics (Vrijenhoek et al., 2021) take a first step towards normative diversity for recommender systems and reflect the nuances of the different democratic models described above. See Table 5.1 for an overview of the DART metrics and their interplay with democratic models. **Calibration** aims to reflect to what extent a recommendation is tailored to a user's personal preferences. This can be considered both in terms of an article's content, and express for example which topics a user is interested in, but also in terms of its complexity. This would then differentiate information needs for different users: a user can consume more complex material on topics they are an expert in, than on ones they are unfamiliar with. **Fragmentation** expresses to what extent there is a common public sphere, where users are frequently recommended the same items and therefore have a similar understanding of the content in the system. This metric is conceptually most related to the concept of the filter bubble. Next, **Activation** expresses the tone of articles: are they written in a neutral and rational tone, or one that is more emotional and activating? While neutrality is a core principle of quality journalism, a softer approach can be more affective and increase empathy. The last two metrics correspond to different aspects of viewpoint diversity. **Representation** aims to reflect the *content* of the different viewpoints in the recommendations. Here, it is important to think about how these viewpoints should be distributed across recommendations: (i) uniformly (a.k.a. equal representation), (ii) in proportion to the occurrence of viewpoints in the medium (a.k.a. reflective representation) or (iii) in inverse proportion to the occurrence of viewpoints in the medium (a.k.a. inverse representation). On the other hand, **Alternative Voices** considers the viewpoint *holder*, and more specifically whether this person belongs to a minority group or not. One could think of Alternative Voices in the context of algorithmic fairness, and aim to reflect an equal share of Alternative Voices compared to the overall supply of data.

When considered together, the metrics aim to reflect the characteristics of the different democratic models, summarized in Table 5.1. The **Liberal** model focuses on personal development and preferences. It is therefore most concerned with aspects of Calibration, and tolerant of a high degree of Fragmentation. The **Participatory** model, with its focus on a public sphere and informing audiences, aims for the opposite value in Fragmentation, but low divergence on Calibration in terms of article complexity. Viewpoints should reflect society, with a larger share for more dominant viewpoints. In comparison, with its focus on promoting rational discussion and debate, the **Deliberative** model favors a system with an equal representation of different viewpoints, presented in a rational tone. Lastly, the **Critical** model has a strong focus on promoting minority voices, and viewpoints recommended should therefore be the inverse of what is most

Table 5.1: Overview of the different models and expected value ranges for each metric. It should be noted that a high score should be interpreted as high *divergence*; As such, a high score does not necessarily mean a better score.

	Liberal	Participatory	Deliberative	Critical
Calibration (topic)	Low	High	–	–
Calibration (complexity)	Low	Low	–	–
Fragmentation	High	Low	Low	–
Activation	–	Medium	Low	High
Representation	–	Reflective	Equal	Inverse
Alternative voices	–	Medium	–	High

commonly heard.

There is a lot of discussion to be had on this topic. The four models described are the ones most frequently discussed (Helberger, 2019), and not exhaustive in the goals one could have with a news recommender system. The metrics following them may also differ, in their definition or operationalization, depending on the use case. The main take-away is that there is no single golden standard for diversity and what makes for a good news recommender system, but rather that this is very much dependent on what you wish to accomplish with the system. This is ultimately a discussion that needs to be had with the different stakeholders involved in recommender system design (Smets et al., 2022). For example, (Hada et al., 2022) did an implementation of Fragmentation and Representation in social media conversations, focusing on Daylight Savings Time and the more polarized topic of immigration. While Fragmentation and Representation are not directly tied in one of the democratic models, they found that the two metrics combined were capable of expressing the nuances of the discourse, and that the effects were stronger for the polarized discussion than for the control group.

5.2.3 The gap between normative and descriptive diversity

The descriptive diversity metrics described in Section 5.2.1 are general-purpose and meant to be applicable in all domains of recommendation. However, in their simplicity a large gap can be observed between this interpretation of diversity and the social sciences' perspective on media diversity that is detailed in Section 5.2.2. In their comprehensive work on the implementation of media diversity across different domains, Loecherbach et al. (2020) note that there is 'little to no overlap between concepts and operationalizations (of diversity) used in the different fields interested in media diversity.' As such, a recommendation that would score high on diversity according to traditional information retrieval-based metrics (Clarke et al., 2008; Sakai and Zeng, 2019), may not be considered to be diverse according to the criteria maintained by newsroom editors. This notion is also shared in industry: Boididou et al. (2021) (BBC) say that "*We need to devise appropriate metrics to capture the quality criteria and values expressed in our*

editorial guidelines',' whereas Grün and Neufeld (2021) (ZDF) note that "*the evaluation of KPIs and public values is more complex than the measurement of common machine learning metrics applied to recommendation algorithms.*" To move the issue forward, both Loecherbach et al. (2020) and Bernstein et al. (2020) call for truly interdisciplinary research in bridging this gap, where Bernstein et al. (2020) argue for close collaboration between academia and industry and the foundation of joint labs. This work is a step in that direction, as we provide a versatile and mathematically grounded rank-aware metric that can be used by practitioners to monitor their normative goals.

5.3 Operationalizing Normative Diversity for News Recommendation

With our RADio framework, we further refine the DART metrics that were defined by (Vrijenhoek et al., 2021) in order to resolve a number of the shortcomings of the metrics' initial formalizations. In their current form, each of the metrics has different value ranges; for example, *Activation* has a value range $[-1, 1]$, where a higher score indicates a higher degree of activating content, and *Calibration* has a range of $[0, \infty]$, where a lower score indicates a better Calibration. These different value ranges reduce the interpretability of the metrics, making them harder to explain and as such less likely to be adopted by news editors. Furthermore, the proposed metrics do not take the position of an article in a recommendation list into account. News recommendations are ranked lists of articles that are typically presented to users in such a way that the likelihood of a recommended article to be considered by the user decreases further down the ranking. As such, in the evaluation of the diversity of the recommender system we should also account for the position of an article in the recommendation ranking, rather than considering the set as a whole.

Thus, the two major challenges that we seek to address are that (i) scores should be comparable between the metrics and across recommendation systems, and (ii) scoring of both unranked and ranked sets of recommendations should be possible. In this section, we first detail these requirements (Section 5.3.1), then describe how we reformulate the metrics to each use the same divergence-based approach (Section 5.3.2). We then add the rank-aware aspect to the metrics (Section 5.3.3), before applying them to the five concrete DART metrics (Section 5.3.4).

5.3.1 Requirements

We first enunciate the classical definition of a distance metric, before specifying three desirable metric criteria for news recommendations. Take a set X of random variables and $x, y, z \in X$, then a metric D is a proper distance measure if $D(x, y) = 0 \Leftrightarrow x = y$, $D(x, y) = D(y, x)$ and $D(x, y) \leq D(x, z) + D(z, y)$. These are respectively the axioms of *identity*, *symmetry* and *triangle inequality*, that express intuitions about concepts of distance (O'Searcoid, 2007).

We add that our distance measure should (i) be bounded by $[0; 1]$, for comparisons of different recommendation algorithms, (ii) be unified, so as to fairly consider different diversity aspects (as opposed to e.g. using weighted averages or maxima in (Dhamala et al., 2021)), and (iii) allow for discrete rank-based distribution sets, to fit the ranked recommendation setting.

5.3.2 f-divergence

We model the task of measuring diversity as a comparison between probability distributions: *the difference in distribution between the recommendations* (Q) *and its context* (P). Each diversity metric prescribes its own Q and P. The elements in the distribution Q can be recommendation items (cf. Calibrated Recommendations (Steck, 2018)), but can also be higher-level concepts, such as distributions of topics and viewpoints. The context P may refer to either the overall supply of available items, the user profile, such as the reading history or explicitly stated preferences, or the recommendations that were issued to other users (see Figure 5.2). Intuitively, when P is linked to the same user as Q, we measure within user diversity (e.g., towards preventing getting locked in "filter bubbles"). When P is linked to another user than the one linked to Q, we measure diversity across users (e.g., monitoring diversity of viewpoints represented across personalized homepages). In the following, we formalize the role of P and Q in two different metric settings, starting with the simple and common KL divergence metric, before presenting its refinement, the Jensen-Shannon divergence, as our preferred metric.

Kullback-Leibler divergence

For every random variable there is an inherent level of 'uncertainty' in its possible outcomes. In Information Theory, the concept of entropy is a way to describe the average level of 'uncertainty' a random variable carries. The entropy of a discrete random variable X with possible outcomes $x_1, \ldots, x_n$ and corresponding probabilities $P(x_1), \ldots, P(x_n)$ is defined as $H(X) = -\sum_{i=1}^{n} P(x_i) \log_2 P(x_i)$. This value describes the average level of 'information' required to describe that random variable (Cover and Thomas, 1991). The concept of relative entropy or KL (Kullback–Leibler) divergence (Kullback and Leibler, 1951) between two probability mass functions P and Q (here, a recommendation and its context) is defined as:

$$D_{\mathrm{KL}}(P, Q) = -\sum_{x \in \mathcal{X}} P(x) \log_2 Q(x) + \sum_{x \in \mathcal{X}} P(x) \log_2 P(x). \tag{5.1}$$

This is often also expressed as $D_{\mathrm{KL}}(P, Q) = H(P, Q) - H(P)$, with $H(P, Q)$ the cross entropy of P and Q, and $H(P)$ the entropy of P. Both cross entropy and KL divergence can be thought of as measurements of how far the probability distribution Q is from the reference probability distribution P. Cross entropy however, does not guarantee the *identity* property. That is, when $P = Q$, it holds that $H(P, Q) = H(P, P) = H(P) > 0$: the cross entropy of P with itself

(the entropy of P) is never 0. KL divergence, though, does satisfy the *identity* property, bringing forward a good argument to prefer KL divergence over cross entropy. However, neither of them are truly proper metrics, as neither fulfil the requirements for *symmetry* and *triangle inequality*. This can be resolved by further refining KL divergence.

Jensen–Shannon divergence

A succession of steps from KL divergence lead to Jensen-Shannon (JS) divergence. KL divergence was first turned symmetric (Jeffreys, 1946) and then upper bounded (Lin, 1991), to lead to

$$\begin{aligned} D_{\mathrm{JS}}(P,Q) = &- \sum_{x\in\mathcal{X}} \frac{P(x)+Q(x)}{2} \log_2 \left(\frac{P(x)+Q(x)}{2}\right) \\ &+ \frac{1}{2}\sum_{x\in\mathcal{X}} P(x)\log_2 P(x) + \frac{1}{2}\sum_{x\in\mathcal{X}} Q(x)\log_2 Q(x). \end{aligned} \tag{5.2}$$

When the base 2 logarithm is used, the JS divergence bounds are $0 \leq D_{\mathrm{JS}}(P,Q) \leq 1$. Additionally, (Endres and Schindelin, 2003) show that $\sqrt{D_{\mathrm{JS}}}$ is a proper distance which fulfills the identity, symmetry and the triangle inequality properties. When we refer to D_{JS} or JS divergence below, we therefore implicitly refer to the square root of the JS formulation with log base 2.

f-divergence

Liese and Vajda (2006) defined *f-Divergence* (D_f): a generic formulation of several divergence metrics. Among them are the JS and KL divergences.[4] Further along the text, we use D_f as a shorthand notation for KL and JS divergences. D_f in discrete form is

$$D_f(P,Q) = \sum_x Q(x) f\left(\frac{P(x)}{Q(x)}\right), \tag{5.3}$$

where $f_{\mathrm{KL}}(t) = t\log t$ and $f_{\mathrm{JS}}(t) = \frac{1}{2}\left[(t+1)\log\left(\frac{2}{t+1}\right) + t\log t\right]$. To avoid misspecified metrics (i.e. division by zero) (Steck, 2018), we write $\overline{P}$ and $\overline{Q}$:

$$\overline{Q}(x) = (1-\alpha)Q(x) + \alpha P(x) \quad \overline{P}(x) = (1-\alpha)P(x) + \alpha Q(x), \tag{5.4}$$

where α is a small number close to zero. $\overline{P}$ prevents artificially setting D_f to zero when a category (e.g., a news topic) is represented in Q and not in P. In the opposite case (when a category is represented in P and not in Q), $\overline{Q}$ avoids zero divisions. In order for the entire probabilistic distributions $\overline{P}$ and $\overline{Q}$ to remain proper statistical distributions, we normalize them to ensure

[4] f-Divergence accommodates for other divergence metrics which are out of scope of this research, such as the Hellinger divergence, and the Pearson divergence (Liese and Vajda, 2006).

$\sum_x \overline{P}(x) = \sum_x \overline{Q}(x) = 1$. In the following sections P and Q will implicitly refer to $\overline{P}$ and $\overline{Q}$ in order to avoid notation congestion.

Below we verify by simple plug-in and factorization that both the Kullback-Leibler and the Jensen-Shannon divergence are f-divergences.

Theorem 5.3.1. *The Kullback-Leibler divergence is an f-divergence.*

Proof. Starting from Equation 5.1,

$$\begin{aligned} D_{\mathrm{KL}}(P,Q) &= -\sum_{x\in\mathcal{X}} P(x)\log_2 Q(x) + \sum_{x\in\mathcal{X}} P(x)\log_2 P(x) \\ &= \sum_{x\in\mathcal{X}} Q(x)\left[\frac{P(x)}{Q(x)}\left(\log_2 P(x) - \log_2 Q(x)\right)\right] \\ &= \sum_{x\in\mathcal{X}} Q(x) f_{KL}\left(\frac{P(x)}{Q(x)}\right). \end{aligned}$$

□

Theorem 5.3.2. *The Jensen-Shannon divergence is an f-divergence*

Proof. Starting from Equation 5.2,

$$\begin{aligned} &D_{\mathrm{JS}}(P,Q) \\ &= -\sum_{x\in\mathcal{X}} \frac{P(x)+Q(x)}{2}\log_2\left(\frac{P(x)+Q(x)}{2}\right) \\ &+ \frac{1}{2}\sum_{x\in\mathcal{X}} P(x)\log_2 P(x) + \frac{1}{2}\sum_{x\in\mathcal{X}} Q(x)\log_2 Q(x) \\ &= \sum_{x\in\mathcal{X}} Q(x)\frac{1}{2}\left[\left(\frac{P(x)}{Q(x)}+1\right)\log_2\left(\frac{2}{\left(\frac{P(x)}{Q(x)}+1\right)}\right) + \frac{P(x)}{Q(x)}\log_2\left(\frac{P(x)}{Q(x)}\right)\right] \\ &= \sum_{x\in\mathcal{X}} Q(x) f_{JS}\left(\frac{P(x)}{Q(x)}\right). \end{aligned}$$

□

5.3.3 Rank-aware f-divergence metrics

Our ranked recommendation setting (characteristic (iii) above) motivates a further reformulation of our f-divergence metric. It is well entrenched in the learning to rank (LTR) literature (Tax et al., 2015; Yilmaz and Robertson, 2010), and by extension in conventional descriptive diversity metrics (Castells et al., 2015) that a user is a lot less likely to see items further down a recommended ranked list (i.e., diminishing inspection probabilities). Note that the ranking oftentimes reflects relevance to the user, but it is not always the case for news (e.g., editorial layout of a news homepage).

We extend our metrics with an optional discount factor for P and Q to weigh down the importance of results lower in the ranked recommendation list.

The ranking relevancy metrics mean reciprocal rank (MRR) and normalized discounted cumulative gain (NDCG) are popular rank-aware metrics for LTR (Järvelin and Kekäläinen, 2002; Chakrabarti et al., 2008), in particular for news recommendation (Wu et al., 2020). In line with the LTR literature, we first define the discrete probability distribution of a ranked recommendation set Q^*, given each item i in the recommendation list R:

$$Q^*(x) = \frac{\sum_i w_{R_i} \mathbb{1}_{i \in x}}{\sum_i w_{R_i}}, \tag{5.5}$$

where w_{R_i}, the weight of a rank for item i, can be different depending on the discount form. For MMR, $w_{R_i} = 1/R_i$, for NDCG, $w_{R_i} = 1/\log_2(R_i + 1)$. When $w_{R_i} = 1$, Q^* is not discounted (i.e., $Q^* = Q$).

In news recommendation, the *sparsity bias* plays a predominant role: users will interact with a small fraction of a large item collection, such as scrollable news recommendation websites (Kille et al., 2013). We thus opt for weighing based on MRR rather than NDCG, because it applies a heavier discount along the ranking than NDCG. Note that the latter is said to be more suited for query-related rankings, where the user has a particular information need related to a query and thus higher propensity to scroll down a page (Chakrabarti et al., 2008).

The context distribution P is discounted in the same manner, when it is a ranked recommendation list. When P is a user's reading history (see Figure 5.2), the discount on P increases with time: articles read recently are weighted higher than articles read longer ago. There are situations when rank-awareness is not applicable, for example when P is the entire pool of available articles.[5] With rank-aware Q^* and optionally rank-aware P^*, we formulate RADio, our rank-aware f-divergence metric:

$$D^*_{f_{\mathrm{JS}}}(P, Q) = \sum_x Q^*(x) f_{\mathrm{JS}} \left(\frac{P^*(x)}{Q^*(x)} \right), \tag{5.6}$$

$Q^*(x)$ and $P^*(x)$ accommodate for multiple situations: for example, $Q^*(c|R)$ is the rank-aware distribution of news categories c over the recommendation set R. In the following, we specify $P^*(x|\cdot)$ and $Q^*(x|\cdot)$ in accordance to each normative concept of interest for our universal metric.

5.3.4 Normative diversity metrics as rank-aware f-divergences

In this section, we describe the RADio formalization of the general f-Divergence formulation above and apply it to the five DART metrics. We leave the exact

[5] There are several features along which such a pool of data could be ranked besides recency, such as the popularity during the last hour, day or week. As this is an editorial decision we remain agnostic as to the choice of that feature and refrain from ranking, though it remains possible in theory.

Table 5.2: Overview of the implementation approach for different methods. Numbers in bold correspond to the corresponding steps in the metadata enrichment pipeline presented above.

	Context	**Type**	**Distribution of**
Calibration (topics)	Reading history	Categorical	Article subcategories as provided in the MIND dataset.
Calibration (complexity)	Reading history	Continuous	Article complexity **(1)** as calculated with the Flesch-Kincaid reading ease test.
Fragmentation	Other users	Categorical	Recommended news story chains **(2)**.
Activation	Available articles	Continuous	Activation scores, which is approximated by the absolute value of a sentiment analysis score **(3)**.
Representation	Available articles	Categorical	The presence of political actors **(4)**.
Alternative voices	Available articles	Continuous	The presence of minority voices versus majority voices. We identify someone as a 'minority voice' when they are identified as a person through the NLP pipeline **(5)**, but cannot be linked to a Wikipedia page.[6]

implementation of the metrics in practice for a particular open news recommendation dataset to the next section. More formally, we define the following global parameters:

- S: The list of news articles the recommender system could make its selection from, also referred to as the "supply."
- R: The ranked list of articles in the recommendation set.
- H: The list of articles in a user's reading history, ranked by recency.

$R_i^u \in \{1, 2, 3, \ldots\}$ refers to the rank of an item i in a ranked list of recommendations for user u. In this work, metrics are defined for a specific user at a certain point in time, therefore R implicitly refers to R^u, unless stated otherwise. While this section contains some contextualization of the DART metrics (Vrijenhoek et al., 2021), the original paper contains further normative justifications.

Calibration (Equation 5.7) measures to what extent the recommendations are tailored to a user's preferences. The user's preferences are deduced from their reading history (H). Calibration can have two aspects: the divergence of the recommended articles' *categories* and *complexity*. The former is expected to be extracted from news metadata and thus categorical by nature, the latter is a binned (categorical) probabilistic measure extracted via a language model. As such, we compare $P^*(c|H)$, the rank-aware distribution of categories or complexity score bins c over the users' reading history, and $Q^*(c|R)$ the same in the recommendations issued to the user.

Fragmentation (Equation 5.8) reflects to what extent we can speak of a common public sphere, or whether the users exist in their own bubble. We measure Fragmentation as the divergence between every pair of users' recommendations. Here we consider $P^*(e|R^u)$ as the rank-aware distribution of news events e over the recommendations R for user u, and $Q^*(e|R^v)$ the same but for user v. KL Divergence is asymmetric (see Section 5.3.2), which means that its outcome differs depending on which user's recommendation is chosen as the target and which as the reference distribution. To avoid this, we compute the Fragmentation score as the average of KL Divergences with switched parameters. JS divergence is already symmetric and is thus implemented as for the other metrics. In theory, Fragmentation requires a user's recommendation to be compared to those of all other users. This is not feasible with a sizeable dataset and the requirement of a reasonable compute time. Instead we opt to randomly sample user pairs.

Activation (Equation 5.9) Most off-the-shelf sentiment analysis tools analyze a text, and return a value $(0, 1]$ when the text expresses a positive emotion, a value $[-1, 0)$ when the expressed sentiment is negative, and 0 if it is completely neutral. The more extreme the value, the stronger the expressed sentiment is. As proposed in (Vrijenhoek et al., 2021), we use an article's absolute sentiment score as an approximation to determine the height of the emotion and therefore the level of Activation expressed in a single article. This then yields a continuous value between 0 and 1. $P(k|S)$ denotes the distribution of (binned) article Activation score k within the pool of items that were available at that point (S). $Q^*(k|R)$ expresses the same, but for the binned Activation scores in the rank-aware recommendation distribution.

Representation (Equation 5.10) aims to approximate a notion of viewpoint diversity (e.g., mentions of political topics or political parties), where the viewpoints are expressed categorically. Here p refers to the presence of a particular viewpoint, and $P(p|S)$ is the distribution of these viewpoints within the overall pool of articles, while $Q^*(p|R)$ expresses the rank-aware distribution of viewpoints within the recommendation set.

Alternative Voices (Equation 5.11) is related to the Representation metric in the sense that it also aims to reflect an aspect of viewpoint diversity. Rather than focusing on the content of the viewpoint, it focuses on the viewpoint holder, and specifically whether they belong to a "protected group" or not. Examples of such protected/unprotected groups could be non-male/male, non-white/white, etc.[7] This approach is based on the implementation of balanced neighbourhoods in recommender systems (Burke et al., 2018). With m we refer to the distribution of protected vs. non-protected groups, with $m \in \{Minority, Majority\}$. $P(m|S)$

[6] We acknowledge that this is a largely oversimplified approach towards identifying minority voices. This is a complex question that cannot be resolved to satisfaction within the scope of this paper.

[7] For more examples, see the UK 2010 Equality Act: `https://www.legislation.gov.uk/ukpga/2010/15/part/2/chapter/1`.

and $Q^*(m|R)$ refer to the distribution of these groups in the pool of available articles and rank-aware recommendation distribution respectively.

Below is a summary of the formalization of DART with the RADio framework, the notation of which is defined in this section. In the next section, we show how to retrieve the necessary features from an example news dataset:

$$\begin{aligned} \textit{Calibration} &= Cal\big(P^*(c|H), Q^*(c|R)\big) \\ &= \sum_c Q^*(c|R) f_{\text{JS}}\left(\frac{P^*(c|H)}{Q^*(c|R)}\right) \end{aligned} \tag{5.7}$$

$$\begin{aligned} \textit{Fragmentation} &= Frag\big(P^*(e|R^u), Q^*(e|R^v)\big) \\ &= \sum_e Q^*(e|R^v) f_{\text{JS}}\left(\frac{P^*(e|R^u)}{Q^*(e|R^v)}\right) \end{aligned} \tag{5.8}$$

$$\begin{aligned} \textit{Activation} &= Act\big(P(k|S), Q^*(k|R)\big) \\ &= \sum_k Q^*(k|R) f_{\text{JS}}\left(\frac{P(k|S)}{Q^*(k|R)}\right) \end{aligned} \tag{5.9}$$

$$\begin{aligned} \textit{Representation} &= Rep\big(P(p|S), Q^*(p|R)\big) \\ &= \sum_p Q^*(p|R) f_{\text{JS}}\left(\frac{P(p|S)}{Q^*(p|R)}\right) \end{aligned} \tag{5.10}$$

$$\begin{aligned} \textit{AlternativeVoices} &= AltV\big(P(m|S), Q^*(m|R)\big) \\ &= \sum_m Q^*(m|R) f_{\text{JS}}\left(\frac{P(m|S)}{Q^*(m|R)}\right) \end{aligned} \tag{5.11}$$

5.4 Experimental Setup

In order to demonstrate RADio's potential effectiveness, we developed an NLP pipeline to retrieve input features to the metrics in Section 5.3.4 and ran them on the MIND dataset, an open source dataset made available by Microsoft News. The MIND dataset (Wu et al., 2020) contains the interactions of 1 million randomly sampled and anonymized users with the news items on MSN News between October 12 and November 22, 2019. Each interaction contains an impression log, listing which articles were presented to the user, which were clicked on and the user's reading history. The MIND dataset was published accompanied by a performance comparison on news recommender algorithms trained on this dataset,[8] including news-specific neural recommendation methods NPA (Wu et al., 2019c), NAML (Wu et al., 2019b), LSTUR (An et al., 2019) and NRMS (Wu et al., 2019d). It was shown that these algorithms outperform general-purpose ones (Wu et al., 2020) or common collaborative filtering models (such as alternating least squares (ALS)), in particular due to the short lifespan of news items (Garcin et al., 2013). These algorithms are trained on the

[8] Code available at `https://github.com/microsoft/recommenders/tree/main/examples/00_quick_start`.

impression logs in order to predict which items the users are most likely to click on. For the purpose of this chapter we will evaluate these neural recommendation methods with the RADio framework (on the DART metrics) and compare their performance with two naive baseline methods, based on a reasonable set of candidates (the original impression log): a random selection, and a selection of the most popular items, where the popularity of the item is approximated by the number of recorded clicks in the dataset. There are a number of limitations to the usage of the MIND dataset (see Section 5.8 and (Vrijenhoek, 2023)). However, at the moment of writing, it is the only large open source news dataset with sufficiently granular logs for such experiments. Future work would benefit from a dataset that is explicitly prepared for normative diversity, bypassing the need for an additional preprocessing pipeline. It is not the goal of this chapter to improve on the identification and extraction of the relevant parameters proposed in (Vrijenhoek et al., 2021), but rather to express their outcomes in a meaningful way. We scrape articles via the URLs provided in the MIND dataset. Each article's metadata is enriched in five steps:

1. **Complexity analysis** – Each item is assigned a complexity score based on the Flesch-Kincaid reading ease test (Kincaid et al., 1975), implemented in the Python module py-readability-metrics (DiMascio, 2020). Complexity is then discretized into bins, to accommodate for the discrete form of D_f^*.

2. **Story clustering** – The individual news items are clustered into so-called news story chains, which means that stories about the same event will be grouped together. This way, we add a level of analysis between individual news items and higher level categories (see Section 5.3.4). We use a TF-IDF based unsupervised clustering algorithm based on cosine similarity and a three days moving window, following the setup of (Trilling and van Hoof, 2020).

3. **Sentiment analysis** – Using the textBlob open source NLP library we assign each article a sentiment polarity score (Loria, 2021). Our focus is on the relative neutrality of articles, we thus take the absolute value of the negative / positive polarity score.

4. **Named entity recognition** – Using spaCy, we identify the people, organizations and locations mentioned in the text (Honnibal et al., 2022), and count their frequency.

5. **Named entity augmentation** – For the entities identified in the text in the previous step, we attempt to link them to their Wikidata[9] entry through fuzzy name matching, to figure out if they are politicians, or in the case of organizations, political parties.[10]

[9] `https://www.wikidata.org/`

[10] In the future one could also use additional data available on Wikidata for further refinement of the metrics, such as gender or place of birth / ethnicity for persons, industry type for organizations or country code for locations.

Linking back to the DART metrics, this means that for Calibration we use the article category that is supplied in the dataset, and the complexity score calculated with the Flesch-Kincaid reading ease test (Kincaid et al., 1975). We compare those to what was in the users' reading history. For Fragmentation, we compare the different news stories, identified through story clustering, that are recommended to different users. The Activation score of an article is determined by the absolute sentiment polarity score, and for Representation we look at the distribution of political actors as identified through Named Entity Linking. For Activation, Representation and Alternative Voices we compare the recommendations to the available pool of data, here interpreted as the set of items the recommender system could make its selection from (see also Table 5.2). Since RADio computes the average of all $\{P, Q\}$ pairs, we retrieve confidence intervals over paired distances too, as illustrated in the sensitivity analyses in the next section. In a traditional model evaluation setting, it would be desirable to generate confidence intervals via different model seeds or cross-validation splits. We refrain from doing this for our metric evaluation as this would introduce a multidimensional confidence interval (e.g., over $\{P, Q\}$ pairs and over model seeds).

It should be noted that this pipeline is an imperfect approximation, and that each metric individually would benefit from more sophisticated methods. Table 5.2 links the numbered list above with the DART metrics. It provides an overview of the different metrics and their respective context distribution P over normative concepts. The code for this implementation is available online.[11] We evaluate the outcome of our RADio framework for different recommender strategies (LSTUR, NAML, NPA, NRMS, most popular and random), with both KL Divergence and Jensen-Shannon as divergence metrics, with and without discounting for the position in the recommendation and at different ranking cutoffs. Additionally, we calculate the intra-list diversity (ILD) as a typical descriptive diversity metric. We used cosine similarity as a distance metric (Bradley and Smyth, 2001) and TF-IDF (Bun and Ishizuka, 2001) of the article's body (text) as article representations.[12]

5.5 Experimental Results

Having described our methodology and experimental setup around the operationalization of DART metrics, we analyze the results of the experiments on MIND. We separate descriptive analysis of the results in this section from the normative interpretation of the metrics in Section 5.8. As the default method, we choose to implement RADio with rank-awareness and JS divergence with a rank cutoff @N, which corresponds to the entire ranking list. After commenting

[11] `https://github.com/svrijenhoek/RADio/`

[12] To make calculating ILD on all pairs of articles computationally feasible we used Spark (Zaharia et al., 2016) to load the entire matrix of all intra-list article pairs and parallelize the distance calculation. As this approach was incompatible with the way we did our random selection, which happened later in the pipeline, we have at this time no ILD score to report for the random recommender.

Table 5.3: Results in percentage for our RADio framework for recommendation algorithms on the MIND dataset. We use our preferred setup: JS divergence with rank-awareness @10. For interpretation of the results it should be noted that though a *higher* score does imply higher divergence, this does not necessarily mean this is a *better* score. Rather what it means to be better is dependent on the metric and the model chosen, for which we refer to Table 5.1.[13]

Algorithm	Calibration (topic)	Calibration (complexity)	Fragmentation	Activation	Representation	Alternative voices	NDCG	ILD
LSTUR	58.47	36.32	90.46	18.19	12.61	4.09	41.34	98.37
NAML	57.09	35.93	88.36	18.42	12.30	3.84	40.91	98.95
NPA	58.38	36.19	89.79	18.41	13.59	3.90	40.68	98.68
NRMS	56.62	35.48	88.72	17.94	12.78	3.62	41.63	98.97
Most pop.	65.26	34.77	89.23	19.49	12.68	3.42	27.50	91.79
Random	66.36	39.81	94.39	27.15	25.78	6.98	29.49	-

on the overall results, we further motivate this choice with a sensitivity analysis to different hyperparameters. We alter the divergence metric (KL or JS), rank-awareness (with and without a discount) and ranking cutoffs (@n, with $n = 1, 2, 5, 10, 20, N$) for the different recommender models.

Table 5.3 displays results for RADio with rank-aware JS divergence, NDCG and ILD. NDCG scores are comparable to the results obtained in (Wu et al., 2020). Scores for ILD are generally high, and show little difference between the neural recommenders. This is in line with the nature of natural language: without accounting for synonyms or using word embeddings it is not surprising that little similarity would be found between texts. The most popular recommender does differ significantly in terms of ILD compared to the neural recommenders. Explaining this behavior requires a more in-depth look at the articles that have a lot of clicks recorded in the dataset. Given the nature of MSN News, it is likely that these are often sports or lifestyle articles. This would imply a similar topic and thus a common vocabulary and lower diversity. However, the only conclusion that can be drawn from this is that a most popular recommender tends to recommend more items of the same category, and is as such not very useful in terms of the requirements for normative diversity as described in Section 5.2.2.

For the normative diversity metrics, higher values imply higher divergence scores. Whether high or low divergence is desired depends on the goal of the recommender system, which we will further elaborate in Section 5.8. The random recommender scores highest on divergence for all metrics and is also one of the

[13] We computed NDCG for popular and random, and report on the original NDCG of the MIND publication for the neural recommenders, as it is more informative to the reader. We obtained similar results but no exact match.

Table 5.4: Results for our RADio framework for recommendation algorithms on the MIND dataset. KL divergence with rank-awareness @10. For interpretation of the results it should be noted that though a *higher* score does imply higher divergence, this does not necessarily mean this is a *better* score. Rather what it means to be better is dependent on the metric and the model chosen, for which we refer to Table 5.1. These metrics are executed on a random sample of 35.000 users.

Algorithm	Calibration (topic)	Calibration (complexity)	Fragmentation	Activation	Representation	Alternative voices
LSTUR	2.6038	1.1432	7.7201	0.1481	0.1078	0.0142
NAML	2.5333	1.1287	7.3926	0.1531	0.1047	0.0127
NPA	2.5945	1.1390	7.6202	0.1521	0.1237	0.0134
NRMS	2.5013	1.1204	7.4519	0.1442	0.1114	0.0113
Most pop.	2.9384	1.1082	7.6377	0.1605	0.1028	0.0102
Random	3.6038	1.5985	8.6295	0.8079	1.1248	0.0420

least relevant by definition (see NDCG score). *Most popular* and *random* have comparable NDCG results. Popularity scores for the articles are derived from the clicks recorded in the MIND interaction logs, and many articles have zero or only one click recorded. When the candidate list contains exclusively articles with a similar number of clicks this forces the *most popular* recommender to a random choice, which explains the artificial similarity between *most popular* and *random* in terms of the NDCG score.

Figure 5.3 is the pendant of Table 5.3. It displays robust estimates based on the median and quantiles. Outliers are predominant for the Alternative Voices metric as most articles do not contain any Alternative Voices. As detailed in Section 5.4, Alternative Voices are tagged as such if a named entity could not be found on Wikipedia.

Between the neural recommenders, most scores for LSTUR, NPA, NRMS and NAML are close to zero, indicating a low divergence between the recommendations and the context distribution. Note that they produce similar recommendations (see NDCG values and (Wu et al., 2020)). Some notable differences can be observed when comparing these neural methods to the baselines. For example, we see that the neural recommenders are more Calibrated to the items present in people's reading history, though the most popular baseline performs marginally better in terms of Calibration of complexity. In the following, we further analyse the entire distribution of individual recommendation list divergences and test the sensitivity of RADio to different settings. Boxplots for all metrics and all recommender strategies are available in the online repository, where we highlight the importance of rank-awareness.

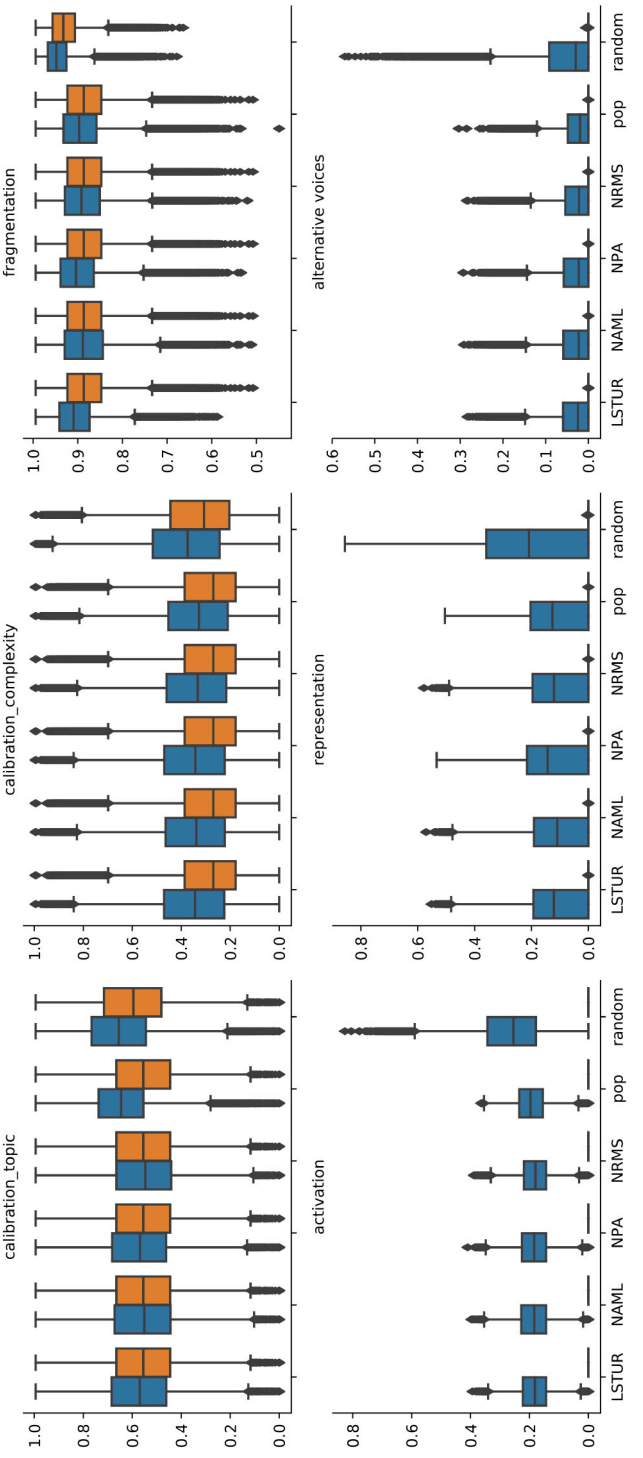


Figure 5.3: Boxplots over each P, Q pair for all algorithms and all metrics for the JS divergence with rank-awareness. This is the pendant of Table 5.3. The horizontal line and the box boundaries show median and inter quartile range (IQR) respectively. Single dots indicate outliers, defined as being further outside the 0.05 to 0.95 quantiles.

5.6 Sensitivity Analysis

This section contains a detailed analysis of the sensitivity of RADio to different settings: the divergence metric (JS or KL), rank-awareness (yes or no) and rank cut-off point (@n).

5.6.1 Sensitivity to the divergence metric

JS divergence is our preferred implementation of universal diversity metrics. It is a proper distance metric and bounded between 0 and 1 (see Section 5.3.2). Figure 5.6 substantiates that claim empirically, visualizing the sensitivity of RADio to the two described f-divergence metrics: KL and JS divergence. Clear differences can be observed in the distributions; KL divergence is skewed towards lower divergence, while JS divergence yields a more centered distribution of values. Additionally, JS divergence applies more contrast between the neural recommender systems and the naive recommendation methods and especially the random baseline. In certain cases KL introduces consequential skew in the distribution of individual P, Q comparison pairs across recommendation models; this does not occur to that extent with JS.

Nevertheless, for demonstration purposes we show results for the RADio framework coupled with KL divergence in Table 5.4. This time, metrics are not bounded by $[0, 1]$. If one were to rank the values for each metric, Table 5.3 and 5.4 would score the algorithms in the same order (Liese and Vajda, 2006). In conclusion, although KL divergence is a well-known metric that can be found in many applications and is simpler to express mathematically, we find JS divergence to be a better fit both theoretically and empirically.

5.6.2 Sensitivity to rank-awareness

In the original formulation of DART metrics (Vrijenhoek et al., 2021), rank-awareness was not considered for most of the defined metrics. In our formalization, rank-awareness is the default. In Figure 5.7, we visualize the effect of removing the rank-awareness (in blue) on Fragmentation and compare to the original rank-aware Fragmentation (in orange). Rank-awareness allows for better differentiation between methods: LSTUR and "most popular" seem to be similarly distributed without a rank discount. Introducing rank-awareness shifts LSTUR towards a larger divergence, whereas "most popular" remains largely the same. Figures 5.4 and 5.5 show a detailed analysis of rank-awareness of robust estimates (median and quantiles) for KL and the JS divergence.

5.6.3 Sensitivity @n

One could also consider adding a cut-off point where only the top n recommendations are considered for evaluation, the results of which are shown in Figure 5.8. The figure shows that the effect of rank-awareness becomes stronger with a higher cut-off point, and causes the divergence score to stabilize after roughly 10

recommendations. This is because our MMR rank-awareness strongly discounts values further down the ranking. @20 and @N (no cutoff) are similar for all metrics because MIND rarely contains more than 20 recommendation candidates. Note that when calculating the divergence score for Activation, Representation or Alternative Voices without rank-awareness and without cutoff point, there is no divergence to be reported as recommendation and target distribution are identical in these cases.

5.6.4 Normative evaluation

By comparing divergence scores across different recommender strategies, we can draw conclusions on the way they influence exposure of news to users. This is especially the case when comparing neural methods to the random recommender, which should reflect the characteristics of the overall supply of data. Combining this with DART's different theoretical models of democracy (summarized in Table 5.1), one can make informed decisions on which recommender system is better suited to one's normative stance than others. Imagine, for example, a media organization that aims to help their users find content on topics they are interested in. In this case they are looking for a low divergence on Calibration, which is shown in the scores of the neural recommenders. This would indicate that those models are more suitable for this organization's goals. In comparison, imagine a large media organization that wants to dedicate a small section of their website to Critical principles, consisting of one element with recommendations called "A different perspective." This calls for a high divergence score in both Representation and Alternative Voices. Given that the random recommender scores best according to these principles, the neural recommenders would not be very suitable for this goal. Nevertheless, the conclusion that a random recommender is suitable for Critical norms and values is moot. Additional steps should be taken to further improve upon these scores: recommendation algorithm developers could tweak the trade-off between different target values in the recommendation, or even explicitly optimize on these metrics.

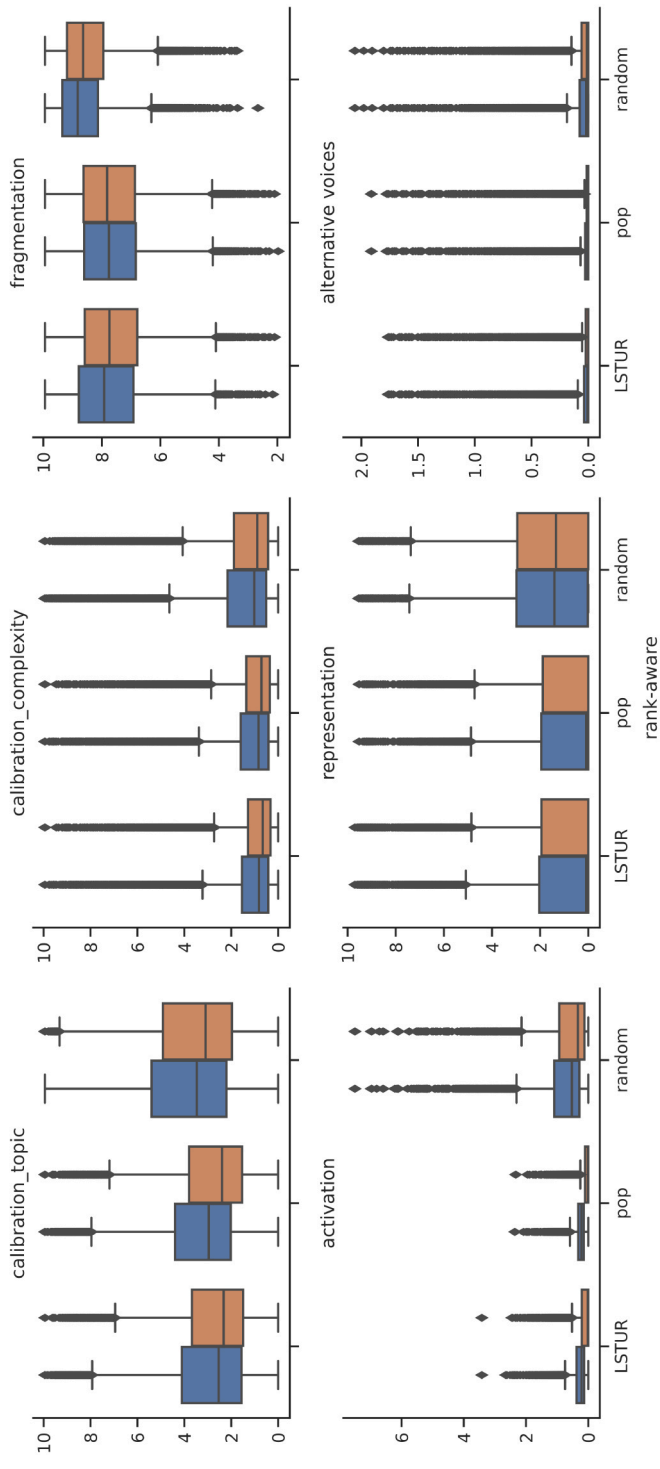


Figure 5.4: Boxplots over each P, Q pair for all metrics for the KL divergence with and without rank-awareness. The horizontal line and the box boundaries show median and inter quartile range (IQR), respectively. Single dots indicate outliers, defined as being further outside the 0.5 to 0.95 quantiles.

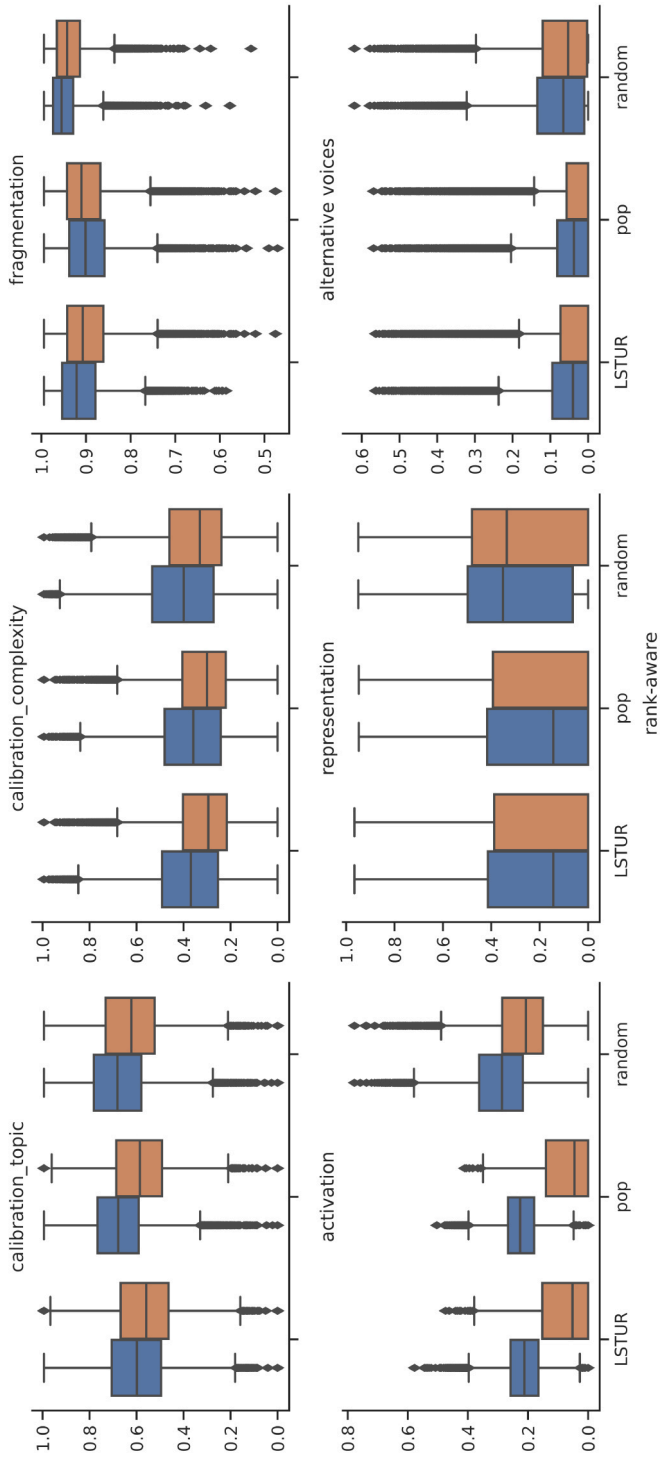


Figure 5.5: Boxplots over each P, Q pair for all metrics for the JS divergence with and without rank-awareness. The horizontal line and the box boundaries show median and inter quartile range (IQR), respectively. Single dots indicate outliers, defined as being further outside the 0.5 to 0.95 quantiles.

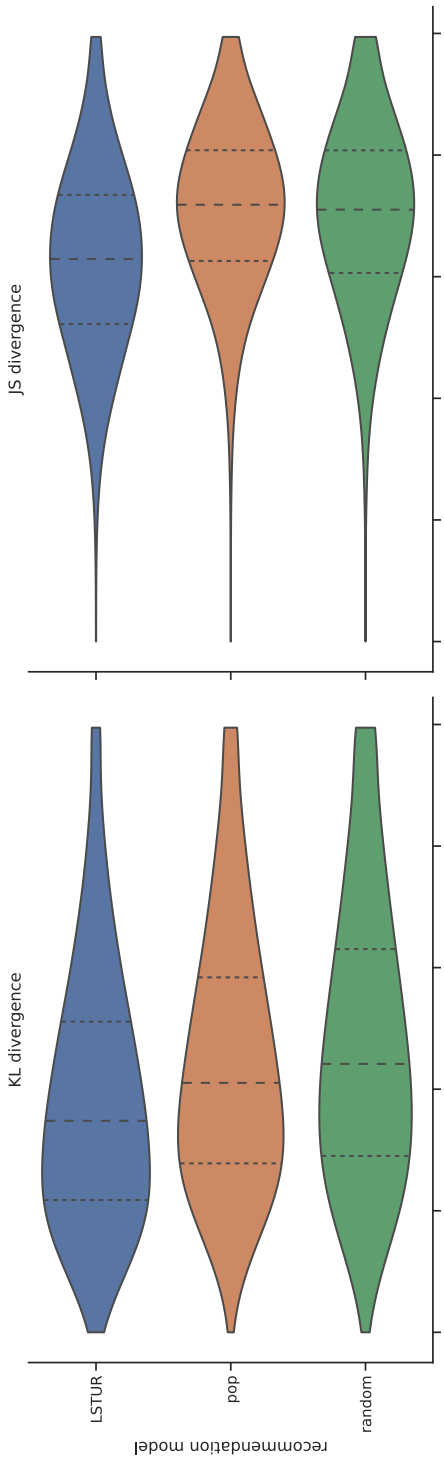


Figure 5.6: Violin kernel density functions (Wickham, 2016) over each P, Q pair, for the Calibration (topic) rank-aware metric, rank cutoff @N (no cuttof), using KL (left) and the JS divergence (right). Thick and thin dashes show median and inter quartile range (IQR), respectively.

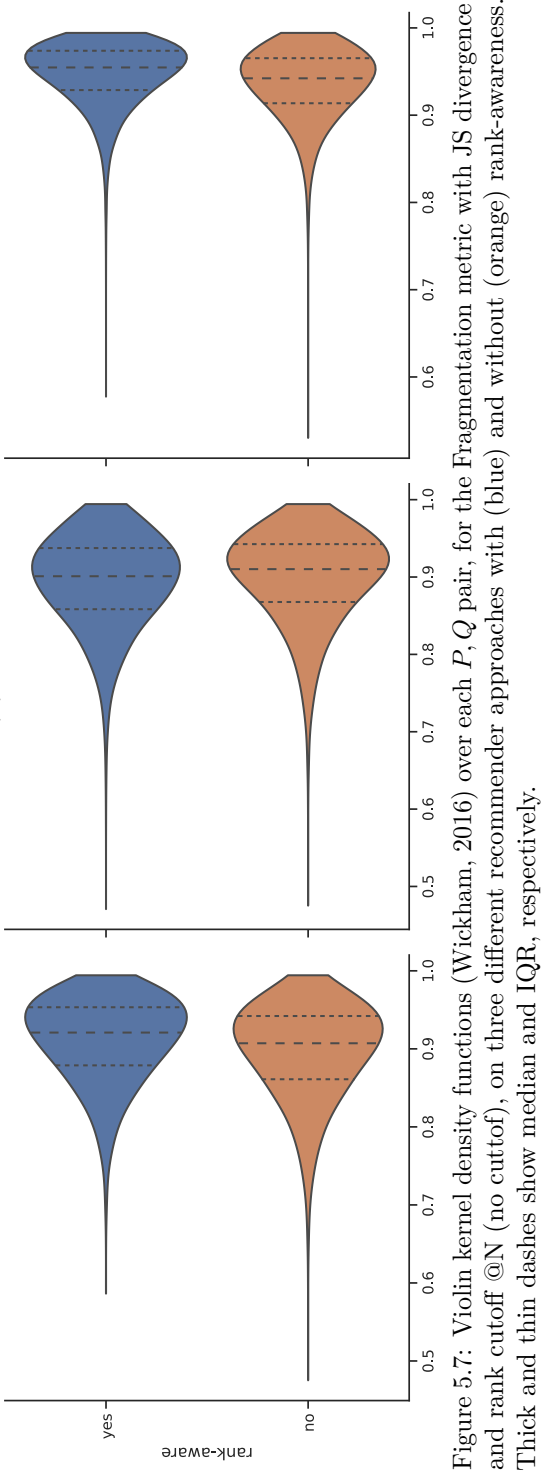


Figure 5.7: Violin kernel density functions (Wickham, 2016) over each P, Q pair, for the Fragmentation metric with JS divergence and rank cutoff @N (no cuttof), on three different recommender approaches with (blue) and without (orange) rank-awareness. Thick and thin dashes show median and IQR, respectively.

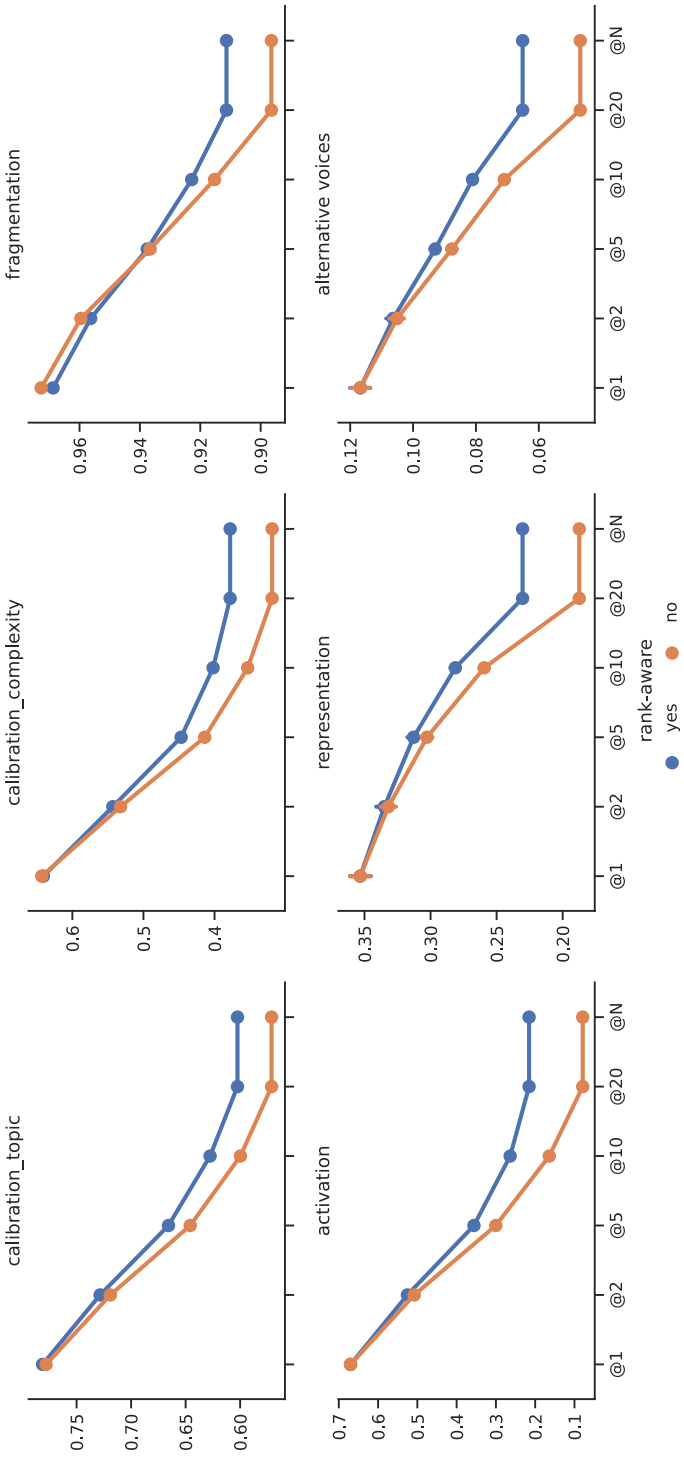


Figure 5.8: Mean and 95% confidence interval for each DART metric implemented with JS divergence for the LSTUR recommender. Sensitivity analysis of RADio on rank-awareness (blue and orange) and rank cutoff.

5.7 The Effects of Metric Design Choices

In the following section, we will take a more detailed look at the effects of different design choices for a normative diversity metric, using the Calibration metric as an example. Calibration, which aims to measure the degree in which a recommendation reflects a user's personal preferences, is in this experimental setting the most suitable for this type of analysis, as it relies on categorical data that is directly available in MIND. Differences in behavior and performance will as such directly be caused by differences in the behavior of the recommender system and the design choices of the metric, rather than as a result of the data extraction pipeline.

Table 5.5 displays the distribution of article categories in the different steps of the recommendation pipeline. For both the training and validation sets, we analyze the article categories present in the full dataset ('all'), the items that were shown to the user and which the recommender system ranks ('impr'), what was present in the users' reading history ('hist'), and which items they eventually clicked on ('click'). We compare this to the categories present in the top 5 items recommended by the neural recommenders LSTUR and NRMS, and the random baseline. The categories in the full datasets are very similarly distributed between the training and validation sets, with both the news and sports categories comprising about 30% of the data, and the other categories ranging between 3-6%. The items in the user histories are also decidedly similar. In both sets, the random recommender reflects the distribution of the impressions, which is expected behavior. Discrepancies between the two sets can be observed in both the candidate list and the content that users eventually clicked. The training data contains many more articles of the category 'news', whereas the validation set has a much higher share of sports articles, especially among the clicked content. This is also reflected in the recommendations generated by the neural recommenders. The difference can be explained by the dynamic nature of news itself, which also underlines a caveat of using a dataset that only consists of a single day: if a major world event happened on that day, it stands to reason that there is both a lot more content in that category, and a higher interaction rate of users with it. This availability of content will then also influence the rate in which a recommender system can generate highly Calibrated recommendations. To account for this variability, we carry out the analysis on the predictions generated on the training set, which comprises of six days. We emphasize that this analysis is not conducted to make judgments about the *performance* of the algorithms, in which case it would be imprudent to include training data in analysis, but rather to exemplify the multiple design choices that are relevant when constructing a normative diversity metric.

Table 5.5: Distribution of article categories in both the training and validation set of MIND. This includes the full dataset ('all'), the content people have seen and which the recommender ranks ('impr'), what was in the user's history ('hist'), the items the users clicked ('click'), and the top 5 items ranked by the neural recommenders and random baseline.

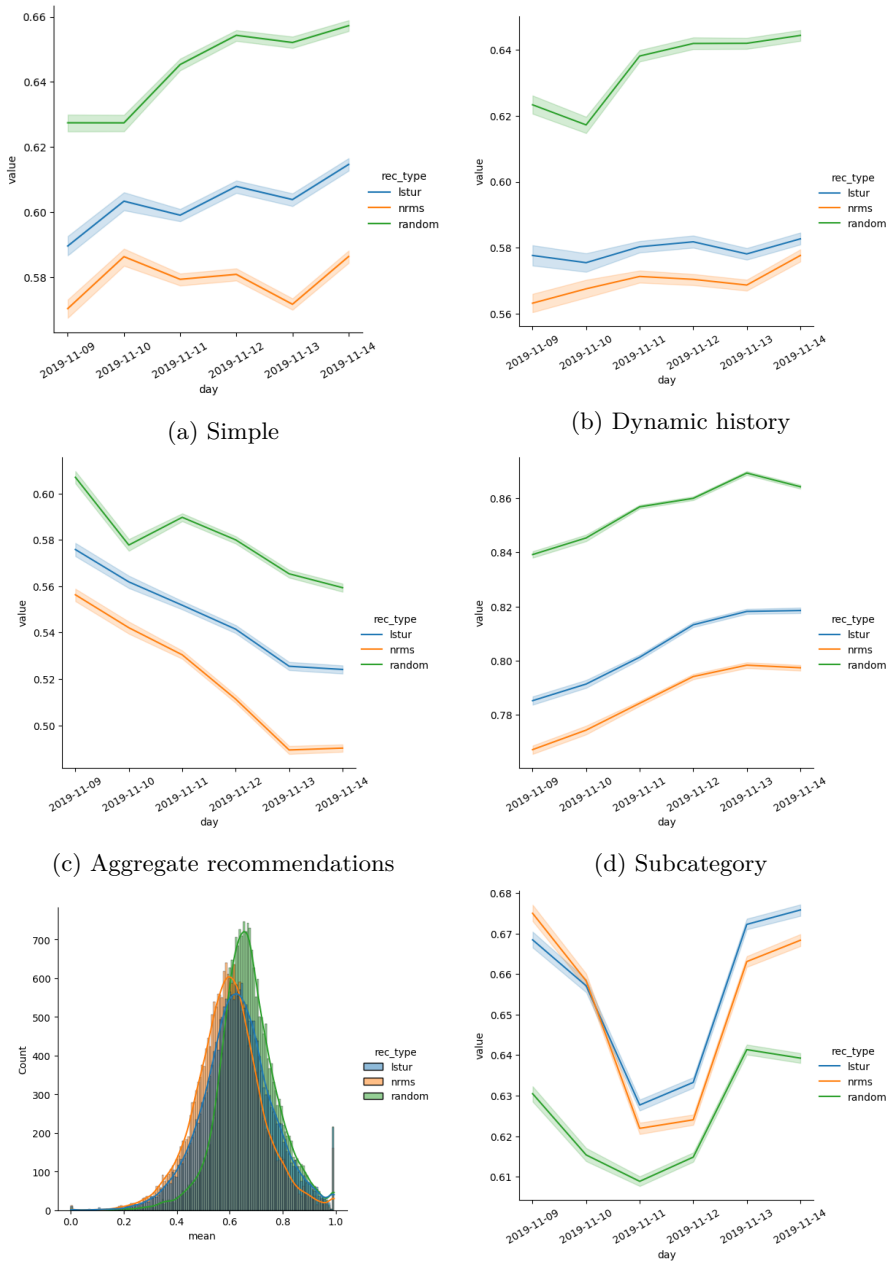
	Training							Validation						
	all	impr	hist	click	lstur	nrms	rand	all	impr	hist	click	lstur	nrms	rand
news	0.3	0.31	0.293	0.299	0.313	0.311	0.31	0.303	0.233	0.279	0.233	0.213	0.198	0.233
sports	0.315	0.116	0.138	0.125	0.129	0.136	0.116	0.3	0.163	0.143	0.246	0.275	0.254	0.163
finance	0.058	0.096	0.071	0.087	0.085	0.078	0.096	0.06	0.069	0.069	0.035	0.024	0.015	0.069
foodanddrink	0.044	0.042	0.049	0.04	0.033	0.038	0.042	0.048	0.068	0.05	0.056	0.09	0.092	0.068
lifestyle	0.045	0.104	0.104	0.106	0.114	0.114	0.105	0.048	0.171	0.105	0.178	0.161	0.174	0.171
travel	0.049	0.039	0.03	0.032	0.025	0.022	0.039	0.047	0.027	0.03	0.02	0.027	0.026	0.027
video	0.045	0.016	0.02	0.018	0.019	0.019	0.016	0.041	0.021	0.019	0.022	0.017	0.017	0.021
weather	0.042	0.027	0.013	0.023	0.021	0.022	0.027	0.039	0.028	0.012	0.015	0.017	0.013	0.027
autos	0.03	0.032	0.036	0.03	0.027	0.027	0.031	0.034	0.03	0.036	0.025	0.015	0.016	0.03
health	0.029	0.037	0.046	0.041	0.04	0.044	0.037	0.033	0.034	0.047	0.034	0.044	0.045	0.034

In Figure 5.9a, we visualize Calibration in the same way as in Section 5.4: as the average divergence between the categories in individual recommendations and the user's reading history, though here it is plotted over time. This graph shows clear differences between the different recommender strategies. Furthermore, we find a statistically significant negative correlation between the divergence scores and the number of items in a user's history ($p < 0.001$): the more items a user has consumed in the past, the lower the Calibration divergence is. In MIND, the user's history is static. Even if a user has accessed the website more than once, there will be no change recorded in the dataset. This makes it impossible to say how the recommender systems would adapt the recommendations over time, as more information about the user becomes available. We simulate a dynamic history in the diversity calculations by including the registered clicks of past interactions in the current history. This is visualized in Figure 5.9b. Compared to the simple approach, we see a slightly lower and mostly more consistent divergence score. Some conflation with the fact that this is training data may happen here. The recommender systems are trained to predict the clicked items, which are also added to the history, resulting in a lower divergence score.

With the dynamic availability of content, and the user's relative interest in major world events, it may not be possible for every single recommendation to achieve the desired normative diversity score, though they may do so over time. In Figure 5.9c, we conceptualize our metric to not only consider the current recommendation, but also the top 5 items of each recommendation in the past, weighted by recency. Here we see that the divergence indeed decreases over time. However, f-Divergence penalizes the score comparatively much if smaller categories are not present in the recommendation (Steck, 2018). Having more individual items in the calculations thus increases the chance that smaller categories are also represented, leading to a lower divergence score. This is reflected by the fact that the random recommender also decreases in divergence over time, though to a lesser extent than the neural recommenders (on average, a 0.03 versus a 0.06 decrease).

Besides the top-level category, MIND also supplies article subcategories, which splits up 'news' into among others 'newsus' and 'newspolitics', and sports into 'football_nfl' and 'baseball_mlb'. The more fine-grained categorization could potentially be much more expressive of a user's interest. Figure 5.9d displays the Calibration score when considering subcategory. It shows roughly the same pattern as in Figure 5.9a, although in a higher range: the average value here is centered around 0.8, whereas in Figure 5.9a it is around 0.6. When deploying Calibration in production, an informed decision needs to be made about the level of detail that is meaningful for the metric.

One other thing that might be relevant to consider is the number of users for whom the system does not work: those who, even on average, receive recommendations that are very divergent from their preferences. This is visualized in Figure 5.9e, which plots the frequency of mean divergence scores per user. The information obtained from such an analysis could be used to investigate who the users are that consistently receive sub-par recommendations, and whether they



(a) Simple

(b) Dynamic history

(c) Aggregate recommendations

(d) Subcategory

(e) Mean values

(f) Supply

Figure 5.9: The Calibration metric with different approaches to metric design.

share characteristics. This could then steer future development and tweaking of the algorithm. In a similar vein, one could investigate whether there are users who exist in their own 'bubble' and receive recommendations that are very different from the 'brand' of the organization. This is conceptualized in Figure 5.9f by calculating the divergence between the recommendations and the general distribution of articles. This type of analysis could be especially relevant when measuring organizational values and metrics such as Representation or Alternative Voices.

The plots in Figure 5.9 convey very different types of information, even though they are based on the same feature and on the same set of recommendations. This shows that the design choices made when implementing the metric greatly influence the patterns it shows. Again, how the metric should behave is dependent on what the organization deploying the recommender system aims to measure and wants to achieve with it. As such, it is necessary that each of the choices are explainable to all stakeholders involved and that the process is well-documented.

5.8 Discussion

Choosing an f-divergence score as the base for our metrics allows us to construct a single base formalization with a clear interpretation amongst all metrics; when the value is 0, the distribution between the recommendations and the chosen context is identical. The larger the measurements, the larger the divergence is. It is also flexible to the use of different target distributions and relevant features, and has as such a strong advantage over ILD, which requires a static model of what it means for two articles to be different. Formalizing the diversity of the recommendation is as such brought down to three questions: what feature are we interested in, which distribution do we want to compare to, and should the divergence between these distributions be low, high, or something else? The answers to these questions could be informed by normative diversity metrics described in this chapter, but could also be interpreted differently. However, f-divergence also comes with a number of limitations. For one, it does not take the relations between categorical values into account, and the ordering of the categorical values in the input vector is arbitrary. For example, two left-wing political parties in the Representation metric may be more similar than an extremely left-wing and an extremely right-wing party, but this is currently not taken into account. Related to this, in order to make continuous values suitable for our general discrete definition of f-divergence, they need to be discretized into arbitrarily defined bins. This means that two very similar values may end up in different bins. Future work may propose a different approach for calculating divergence between continuous variables. Lastly, even design choices within the metric itself, such as the level of detail expressed in the relevant metric or the number of recommendations included in the calculation of the divergence score, may have large impacts on its behavior. Furthermore, the content that is available to a recommender also logically bounds to which extent it can generate normatively diverse recommendations. Careful justification and documentation

of design choices when constructing the metric are thus a necessity.

Regarding the data enrichment pipeline, we identify a number of enhancement points. While some metrics, such as topic Calibration, work with simple data on news topics that is often directly available in a dataset, other metrics require a more sophisticated data enrichment pipeline. The differences in these approaches appear in the results: the metrics with more trivial metadata retrieval setups show clear and distinct patterns for different recommender algorithms, but this is not the case for the more complicated ones. Furthermore, it is not possible to determine the quality of the pipeline, as we do not have a ground truth for evaluation. For future work, we suggest to take the base formalizations as constructed in this chapter as a starting point, and work to improve the extraction of the relevant parameters for metrics such as Representation, Alternative Voices and Activation. Especially for the first two, there is already a large body of work that can facilitate this process (Baden and Springer, 2017; Draws et al., 2021). Human evaluation, including the input from editorial teams, would then be a promising way to evaluate these three normative metrics, similar to the work in the context of language generation bias (Dhamala et al., 2021).

At the same time, more insight needs to be gained on the influence of the choice of dataset. The MIND dataset contains a significant amount of so-called soft news, including articles on lifestyle, sport and entertainment, whereas the DART metrics are mostly applicable to hard news. The influence of the chosen dataset needs to be investigated in more detail, which can then lead to more informed decision-making on the trade-off between diversity and click-through rate, and what can reasonably be expected of a news recommender system. Similarly, while the RADio metrics already provide an improvement over ILD by taking a target distribution into account instead of only what is present in the recommendations, they cannot look 'beyond'. A dataset or target distribution that is already skewed will not be identified as such. A possible solution could be to externalize the target distribution. For example, one could compare to the content produced by other organizations, such as (a number of) competitors or state-funded media. Alternatively one could consider official statistics, such as the distribution of political parties in government. Possibly this is an issue that cannot be solved purely through algorithmic means, and should also be approached in a procedural way. Many news organizations already have organization-wide protocols and procedures in place to guarantee the quality of content produced, and perhaps these protocols should be extended to also account for algorithmic recommendation.

Steps for adoption

Adopting the normative diversity metrics in practice is not a task that can be undertaken by an individual. We envision this process in the following steps:

1. **Identify the relevant stakeholders**. As a first step, all the relevant stakeholders should be involved in the process (Smets et al., 2022). Besides technical and editorial teams, this would also include management and

business teams that eventually have to make an informed decision on the acceptable trade-off between generating clicks and diversity.

2. **Determine the purpose of the recommender systems**. Together, the stakeholders should determine what the purpose of the envisioned recommender system is. This can be inspired by one of the democratic models described in Section 5.2, but should most of all stem from open discussion. A recommender system can have multiple envisioned purposes, and these purposes can be at odds with each other (for example, finding relevant content and encountering different perspectives). In such cases, the dynamic between the different purposes must be made as explicit and concrete as possible.

3. **Determine what type of diversity reflects this purpose**. With step 2 in mind, the stakeholders should define the type of diversity they are looking for using the three questions mentioned earlier in this section: what feature to measure, which distribution to compare to, and the expected divergence range. When the purpose is to help a user find relevant content, this means a low divergence of recommended topics between the recommendation and the user's history. For new perspectives, this means a high divergence between the perspectives in the recommendation versus the user's reading history. Collaboration between the different stakeholders is still instrumental: while the editorial and business teams may have the clearest vision on what a recommendation needs to accomplish, while the technical teams need to communicate what is technically feasible. Many of the aspects related to diversity are technically hard to measure and often even conceptually ambiguous (what is a 'perspective'?). Eventually, all stakeholders should agree on chosen simplifications, be aware of what these metrics can and cannot measure, and know where additional work and research is still necessary.

In its current form RADio is an evaluation framework. We do not believe that the metrics should *replace* existing evaluation metrics, but should rather be used concurrently. Ideally, however, the normative diversity metrics would be incorporated as target for recommender system optimization, rather than post-hoc evaluation. For example, they could be used to inform a reranking algorithm that ensures that over time, the recommendations issued to a user reflect the chosen target distribution. Current diversification methods often rely on a distance measure between individual items. This is not suitable for the normative diversity metrics: as they are based on divergence scores, the recommendation list can only be considered as a whole. As such, additional theoretical work is necessary in order to use the metrics for optimization.

Lastly, if metrics such as these normative diversity metrics are to be incorporated in a production system, they need to be carefully constructed and monitored. Neglecting to do so might lead to so-called 'ethics-washing', where empty metrics are in place that convey no meaning but merely serve as a checkbox, and gamification of the metrics, where external parties learn how to utilize

the metrics to boost the content that fits their purposes. Once more, this is not an effort that can be accomplished by technical teams alone, and is one more reason to close the gap between the technical teams and the newsroom.

5.9 Conclusion

In this chapter we have made a first attempt at constructing and implementing new evaluation criteria for news recommender systems, with a foundation in normative theory. Based on the DART metrics, first theoretically conceptualized in earlier work, we propose to look at diversity as a divergence score, observing differences between the issued recommendations and a metric-specific target distribution. We proposed RADio, a unified rank-aware f-divergence metric framework that is mathematically grounded and that fits several possible use cases within the original DART metrics and we hope beyond in future work. We showed that JS divergence was preferred over other divergence metrics. At first mathematically, as JS is a proper distance metric, and empirically, via a sensitivity analysis to different cutoff, rank-awareness and divergence metric regimes. When our approach is adopted in practice, it enables the evaluation of news recommender systems on normative principles beyond user relevance. Finally, we wish to emphasize that the metrics proposed are meant to supplement standard recommender system evaluation metrics, in the same way that current beyond-accuracy metrics do. Most importantly, they are meant to bridge the gap between different disciplines involved in the process of news recommendation and to support more informed discussion between them. We hope for future research to foster interdisciplinary teams, leveraging each fields' unique skills and specialties.

5.10 Upshots for the Personalization Flow

In this last chapter we look at aspects related to the platforms role in society. We found that it is possible to formulate a metric that adapts to sanity checks on any norms and values, as long as they can be expressed in terms of distributions and are measurable. Oftentimes downplayed or ignored, and although we will never be able to measure its actual downstream effect, we try to put the platforms' immense role in society into perspective.

For the future, we hope to first motivate and possibly force platforms to monitor their level of diversity on different normative levels, especially for news and video platforms that are now part of most of the connected world's daily life. But we hope it does not stop here. Rather, why not use these diversity metrics as losses (see Chapter 3) for our next recommender system? And thus close the loop between passive observations via metrics and active nudges via recommendations. In other terms, make our personalization flow a personalization loop.

Reproducibility

To facilitate the reproducibility of the work in this chapter, our code is available at `https://github.com/svrijenhoek/RADio`.

Chapter 6

Conclusions

In this chapter, we describe our main findings across the four chapters of the thesis. This thesis focuses on a generative personalization flow throughout the user journey on a video streaming platform. Along this flow, we first present RecFusion, a system that uses diffusion models to generate novel and relevant recommendations for users, as part of the emerging field of generative information retrieval. To make these recommendations more appealing, we then propose a method to generate personalized stills from movies using sigmoidF1, a multilabel classification technique that adapts the still to the user's taste. Then, using our intent-satisfaction framework, we analyze how the user interactions on streaming platforms are influenced by the explicit data that is collected by web analytics, but also the implicit data that is hidden from them. Finally, we ensure that the recommendations we generate respect the normative diversity defined by the content providers, using RADio, a framework that measures and optimizes fairness and diversity of recommendations.

6.1 Main Findings

Below, we describe how each research question was handled by a chapter.

RQ1 Can we use diffusion to do recommendation in the classical user-item matrix setting?

The answer to **RQ1** is yes: in Chapter 2 (Bénédict et al., 2023), we propose RecFusion, which comprise a set of diffusion models for recommendation. Unlike image data which contain spatial correlations, a user-item interaction matrix, commonly utilized in recommendation, lacks spatial relationships between users and items. We formulate diffusion on a 1D vector and propose *binomial diffusion*, which explicitly models binary user-item interactions with a Bernoulli process. We show that RecFusion approaches the performance of complex VAE baselines on the core recommendation setting (top-n recommendation for binary non-sequential feedback) and the most common datasets (MovieLens and Netflix).

RQ2 Is there a way we can generate personalized thumbnails for each item on a streaming platform?

The answer to **RQ2** is yes: in Chapter 3 (Bénédict et al., 2022), we propose a solution to classify thumbnails (i.e., images) into one or more movie genres.

More generally, we propose a loss function for multilabel classification. Multilabel classification is the task of attributing multiple labels to examples via predictions. Current models formulate a reduction of the multilabel setting into either multiple binary classifications or multiclass classification, allowing for the use of existing loss functions (sigmoid, cross-entropy, logistic, etc.). These multilabel classification reductions do not accommodate for the prediction of varying numbers of labels per example. Moreover, the loss functions are distant estimates of the performance metrics. We propose *sigmoidF1*, a loss function that is an approximation of the macro F1 score that (i) is smooth and tractable for stochastic gradient descent at training time, (ii) naturally approximates a multilabel metric, and (iii) estimates both label suitability and label counts. We show that any confusion matrix metric can be formulated with a smooth surrogate. We evaluate the proposed loss function on text and image datasets, and with a variety of metrics, to account for the complexity of multilabel classification evaluation. sigmoidF1 outperforms other loss functions on one text and three image datasets over several metrics. These results show the effectiveness of using inference-time metrics as loss functions for non-trivial classification problems like multilabel classification.

RQ3 Are users' intents together with their behavioral data useful signals to predict or explain satisfaction on a video streaming platform?

In Chapter 4 (Bénédict et al., 2023a), we took a study on that topic in the music domain and reproduced it for the video domain on Videoland, the video streaming platform of RTL NL. Logged behavioral data is a common resource for enhancing the user experience on streaming platforms. In music streaming, Mehrotra et al. (2019) have shown how complementing behavioral data with user intent can help predict and explain user satisfaction. Do their findings extend to video streaming? Compared to music streaming, video streaming platforms provide relatively shallow catalogs. Finding the right content demands more active and conscious commitment from users than in the music streaming setting. Video streaming platforms, in particular, could thus benefit from a better understanding of user intents and satisfaction level. We replicate Mehrotra et al. (2019)'s study from music to video streaming and extend their modeling framework on two fronts: (i) improved modeling accuracy (random forests), and (ii) interpretability (Bayesian models). Thus, the answer to **RQ3** is yes: like the original study, we find that user intent affects behavior and satisfaction itself, even if to a lesser degree, based on data analysis and modeling. By proposing a grouping of intents into decisive and explorative categories we highlight a tension: decisive video streamers are not as keen to interact with the user interface as exploration-seeking ones. Meanwhile, music streamers explore by listening. In this study, we find that in video streaming, unsatisfied users provide the main signal: intent influences satisfaction levels together with behavioral data, depending on our decisive vs. explorative grouping.

RQ4 Can we formulate a divergence metric that measures the normative diversity of recommendations?

The answer to **RQ4** is yes: in Chapter 5 (Vrijenhoek et al., 2022), we propose the RADio framework, Rank-Aware Divergence metrIcs to measure nOrmative diversity. In the traditional recommender system literature, diversity is often seen as the opposite of similarity, and typically defined as the distance between identified topics, categories or word models. However, this is not expressive of the social science's interpretation of diversity, which accounts for a news organization's norms and values and which we here refer to as *normative* diversity. We introduce RADio, a versatile metrics framework to evaluate recommendations according to these normative goals. RADio introduces a rank-aware Jensen Shannon (JS) divergence. This combination accounts for (i) a user's decreasing propensity to observe items further down a list, and (ii) full distributional shifts as opposed to point estimates. We evaluate RADio's ability to reflect five normative concepts in news recommendations on the Microsoft News Dataset and six (neural) recommendation algorithms, with the help of our metadata enrichment flow. We find that RADio provides insightful estimates that can potentially be used to inform news recommender system design.

6.2 Future Directions

Our RADio framework to monitor diversity in recommendations was the last step of our personalization flow (see Figure 1.1). The personalization flow can be extended to an entire *pipeline*, if each thesis chapter was linked with the necessary data architecture (e.g., collection of user data, deployment of recommendation models). We could push the analogy of *flow* and *pipeline* a little further, as a way to introduce future research directions.

The first half of the personalization flow, namely Chapter 2 (Bénédict et al., 2023) and Chapter 3 (Bénédict et al., 2022), presents modeling methods that match user and content or *nudge* the user towards certain behaviors. The second half, namely Chapter 4 (Bénédict et al., 2023a) and Chapter 5 (Vrijenhoek et al., 2022) propose user monitoring methods. In practice, we can extend this to a *loop*: taking the bottom of Figure 1.1, the nudges influence the monitoring and the monitoring the nudges, and so on. In other words, the model influences the metrics and the metrics the model.

The loop analogy seems straightforward, but it actually points to a more complex problem that needs to be addressed. Models are often trained on some differentiable constraints that are not direct proxies of common non-differentiable metrics. These metrics are usually used to test model performance after training and reflect the investigator's true measure of interest. For example, cross-entropy loss is commonly used for multiclass or multilabel classification, but the metrics are distant cousins, such as F1 score, precision, recall or ROC AUC (Bénédict et al., 2022). In recommendation, VAEs and diffusion models use the ELBO loss which is agnostic to ranking positions, but ranking metrics tend to be used at testing time, like NDCG, MRR, etc. (Bénédict et al., 2023).

Overall, we think that nudging via deploying models and monitoring via metrics at testing time should be brought closer together. We discuss future

work based on our model-centric and metric-centric chapters below.

Models

sigmoidF1 (Chapter 3) is most related to the personalization loop initiative (see previous section), where monitoring metrics are used as losses for our models. In the multilabel setup, we are first interested in experimenting with further metrics-as-losses associated with the confusion matrix, such as precision, recall, accuracy, etc. Further we would like to experiment with metrics-as-losses on other problems than multilabel classification, such as image captioning and CLIP models (Radford et al., 2021). Our metrics-as-losses repository aims to tackle these issues in that order:[1] beyond F1 for multilabel classification, can we build further losses that are surrogates to non-differentiable metrics and optimize directly for the metric of choice via stochastic gradient descent?

Like with sigmoidF1, RecFusion (Chapter 2) opens the door to include user monitoring as input at training time. Beyond our RecFusion method, which is mostly about fitting diffusion models to yet a new problem, we hope to open the door to guided diffusion, to condition recommendations on user and item history/metadata (a.k.a. critiquing / controllable recommendation (Luo et al., 2020; Yang et al., 2021; Li et al., 2020; Wu et al., 2019e)). This would allow to fit more of the monitoring data into the recommendation nudging mechanisms: in other words to allow for more personalized recommendations. Our proposed diffusion models, that are specialized for 1D and/or binary setups, have implications beyond recommendation systems, such as in the medical domain with MRI and CT scans. Future work on both RecFusion and sigmoidF1 should bring nudging closer to monitoring of users.

Metrics

Chapter 4 focused on intent. Intent – representing unobservable time-varying user needs – could be inferred at training time to understand the user and change the user interface in real-time. Predicting the intent could be done with fast methods like XGBoost. Changing the user interface on the fly would then be more of an engineering than a modeling challenge. Intent and other unobserved metrics should always remain hidden from the platform for privacy reasons. Guessing these unobserved metrics, is a way for a platform to increase personalization and respect users' privacy. Subsequent work could bring further unobserved monitoring metrics as model input (e.g., is the user alone or with friends? Is the user in a binge watching mood?).

The same can be said about RADio (Chapter 5): why not use our proposed diversity metrics as input at training time? We see four options: (i) train a reinforcement learning algorithm that rewards both accuracy of a recommendation list and diversity; (ii) make the RADio metrics differentiable, for example via a sigmoid trick like in sigmoidF1; (iii) at inference time, create a recommendation list, balanced according to diversity clusters (e.g., news topics); and (iv) rerank

[1] `https://github.com/gabriben/metrics-as-losses`

recommendation lists via a sampling technique like Plackett-Luce (Luce, 1959). We think that our framework RADio and the study of hidden intents should both inspire further work on using monitoring metrics as input for training nudging mechanisms on streaming platforms.

Upshot

We have described how metrics (user monitoring) can be used directly as feedback signal to our models (nudging mechanisms). This is part of a broader initiative of end-to-end machine learning solutions. Generative information retrieval (GenIR) is such an initiative (Metzler et al., 2021), that we pursue with a series of workshops that started at the SIGIR conference in summer 2023. GenIR is end-to-end because it relies on a single autoregressive system to predict the next token (word or document ID) to perform all retrieval tasks: retrieval, reranking, query-reformulation, natural language answer, conversation, etc. By maintaining a list of generative retrieval endeavors, we hope to encourage further work in this direction.[2]

[2] `https://github.com/gabriben/awesome-generative-information-retrieval`

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, 2015. Software available from tensorflow.org.

[2] Rahul Agrawal, Archit Gupta, Yashoteja Prabhu, and Manik Varma. Multi-Label Learning with Millions of Labels: Recommending Advertiser Bid Phrases for Web Pages. *Proceedings of the 22nd International Conference on World Wide Web*, pages 13–24, 2013.

[3] Amazon. Amazon Music Has More Than 55 Million Customers Worldwide. `https://www.aboutamazon.com/news/entertainment/amazon-music-has-more-than-55-million-customers-worldwide`, 2020.

[4] Hesam Amoualian, Parantapa Goswami, Laurent Ach, Pradipto Das, and Pablo Montalvo. SIGIR 2020 E-Commerce Workshop Data Challenge Overview. *Proceedings of ACM SIGIR Workshop on eCommerce*, 2020.

[5] Mingxiao An, Fangzhao Wu, Chuhan Wu, Kun Zhang, Zheng Liu, and Xing Xie. Neural News Recommendation with Long- and Short-term User Representations. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 336–345, 2019.

[6] Ashton Anderson, Lucas Maystre, Ian Anderson, Rishabh Mehrotra, and Mounia Lalmas. Algorithmic Effects on the Diversity of Consumption on Spotify. *Proceedings of The Web Conference*, 2020.

[7] Evlampios Apostolidis, Eleni Adamantidou, Alexandros I. Metsai, Vasileios Mezaris, and Ioannis Patras. AC-SUM-GAN: Connecting Actor-Critic and Generative Adversarial Networks for Unsupervised Video Summarization. *IEEE Transactions on Circuits and Systems for Video Technology*, 2020.

[8] Apple. Apple Music. https://www.apple.com/apple-music/, 2022. Accessed on 03.02.2022.

[9] Rohit Babbar and Bernhard Schölkopf. DiSMEC: Distributed Sparse Machines for Extreme Multi-Label Classification. *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 721–729, 2017.

[10] Christian Baden and Nina Springer. Conceptualizing Viewpoint Diversity in News Discourse. *Journalism*, 18(2):176–194, 2017.

[11] Emanuel Ben Baruch, Tal Ridnik, Nadav Zamir, Asaf Noy, Itamar Friedman, Matan Protter, and Lihi Zelnik-Manor. Asymmetric Loss For Multi-Label Classification, 2020. arXiv: 2009.14119.

[12] Mariella Bastian, Natali Helberger, and Mykola Makhortykh. Safeguarding the Journalistic DNA: Attitudes towards the Role of Professional Values in Algorithmic News Recommender Designs. *Digital Journalism*, 0(0):1–29, 2021.

[13] Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. Fitting Linear Mixed-Effects Models Using lme4. *Journal of Statistical Software*, 67(1):1–48, 2015.

[14] Michael A Beam. Automating the News: How Personalized News Recommender System Design Choices Impact News Reception. *Communication Research*, 41(8):1019–1041, 2014.

[15] Joeran Beel and Victor Brunel. Data Pruning in Recommender Systems Research: Best-Practice or Malpractice? *13th ACM Conference on Recommender Systems (RecSys)*, 2019.

[16] Joeran Beel and Stefan Langer. A Comparison of Offline Evaluations, Online Evaluations, and User Studies in the Context of Research-Paper Recommender Systems. *Lecture Notes in Computer Science*:153–168, 2015.

[17] Bichitrananda Behera, G. Kumaravelan, and P. Kumar B. Performance Evaluation of Deep Learning Algorithms in Biomedical Document Classification. *2019 11th International Conference on Advanced Computing (ICoAC)*, pages 220–224, 2019.

[18] Amin Beheshti, Shahpar Yakhchi, Salman Mousaeirad, Seyed Mohssen Ghafari, Srinivasa Reddy Goluguri, and Mohammad Amin Edrisi. Towards Cognitive Recommender Systems. *Algorithms*, 13(8), 2020.

[19] Gabriel Bénédict. Generative Adversarial Networks. *Spectra ML Review Paper Competition*, 2021. eprint: 09.00009.

[20] Gabriel Bénédict, Olivier Jeunen, Samuele Papa, Samarth Bhargav, Daan Odijk, and Maarten de Rijke. RecFusion: A Binomial Diffusion Process for 1D Data for Recommendation. *The 1st Workshop on Recommendation With Generative Models on the 32nd ACM International Conference on Information and Knowledge Management*, 2023.

[21] Gabriel Bénédict, Hendrik Vincent Koops, Daan Odijk, and Maarten de Rijke. sigmoidF1: A Smooth F1 Score Surrogate Loss for Multilabel Classification. *Transactions on Machine Learning Research*, 2022.

[22] Gabriel Bénédict, Daan Odijk, and Maarten de Rijke. Intent-Satisfaction Modeling: From Music to Video Streaming. *ACM Transactions on Recommender Systems*, 1(3), 2023.

[23] Garbiel Bénédict, Ruqing Zhang, and Donald Metzler. Gen-IR@SIGIR 2023: The First Workshop on Generative Information Retrieval. *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3460–3463, 2023.

[24] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

[25] Fernando Benites and Elena Sapozhnikova. HARAM: A Hierarchical ARAM Neural Network for Large-Scale Text Classification. *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, pages 847–854, 2015.

[26] James Bennett and Stan Lanning. The Netflix Prize. *Proceedings of KDD Cup Workshop 2007*, pages 3–6. ACM, 2007.

[27] Paul N. Bennett, Ryen W. White, Wei Chu, Susan T. Dumais, Peter Bailey, Fedor Borisyuk, and Xiaoyuan Cui. Modeling the Impact of Short- and Long-Term Behavior on Search Personalization. *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 185–194, 2012.

[28] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for Hyper-Parameter Optimization. *Advances in Neural Information Processing Systems*, volume 24, 2011.

[29] James Bergstra, Daniel Yamins, and David Cox. Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of number 1, pages 115–123, 2013.

[30] B Douglas Bernheim, Luca Braghieri, Alejandro Martínez-Marquina, and David Zuckerman. A Theory of Chosen Preferences. *American Economic Review*, 111(2):720–54, 2021.

[31] Abraham Bernstein, Claes de Vreese, Natali Helberger, Wolfgang Schulz, Katharina Zweig, Christian Baden, Michael A Beam, Marc P Hauer, Lucien Heitz, Pascal Jürgens, et al. Diversity in News Recommendations, 2020. arXiv: `2005.09495`.

[32] Samarth Bhargav and Evangelos Kanoulas. Controllable Recommenders using Deep Generative Models and Disentanglement, 2021. arXiv: `2110.05056`.

[33] Kush Bhatia, Himanshu Jain, Purushottam Kar, Manik Varma, and Prateek Jain. Sparse Local Embeddings for Extreme Multi-label Classification. *Advances in Neural Information Processing Systems*, volume 28, 2015.

[34] Biswarup Bhattacharya, Iftikhar Burhanuddin, Abhilasha Sancheti, and Kushal Satya. Intent-Aware Contextual Recommendation System. *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, 2017.

[35] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. 5th edition, 2007.

[36] Marie-Joëlle Blosseville, Georges Hébrail, Marie-Gaëlle Monteil, and Nadine Pénot. Automatic Document Classification: Natural Language Processing, Statistical Analysis, and Expert System Techniques Used Together. *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 51–58, 1992.

[37] Balazs Bodo. Selling News to Audiences – A Qualitative Inquiry into the Emerging Logics of Algorithmic News Personalization in European Quality News Media. *Digital Journalism*, 0(0):1–22, 2019.

[38] Christina Boididou, Di Sheng, Felix J Mercer Moss, and Alessandro Piscopo. Building Public Service Recommenders: Logbook of a Journey. *Fifteenth ACM Conference on Recommender Systems*, pages 538–540, 2021.

[39] Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. Deep Neural Networks and Tabular Data: A Survey, 2021. arXiv: `2110.01889`.

[40] Allan Borodin, Hyun Chul Lee, and Yuli Ye. Max-Sum Diversification, Monotone Submodular Functions and Dynamic Updates. *Proceedings of the 31st ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 155–166, 2012.

[41] Matthew R. Boutell, Jiebo Luo, Xipeng Shen, and Christopher M. Brown. Learning Multi-label Scene Classification. *Pattern Recognition*, 37(9):1757–1771, 2004.

[42] Keith Bradley and Barry Smyth. Improving Recommendation Diversity. *Proceedings of the 12th Irish Conference on Artificial Intelligence and Cognitive Science*, pages 85–94, 2001.

[43] Joshua A Braun and Jessica L Eklund. Fake News, Real Money: Ad Tech Platforms, Profit-Driven Hoaxes, and the Business of Journalism. *Digital Journalism*, 7(1):1–21, 2019.

[44] Klaus Brinker, Johannes Fürnkranz, and Eyke Hüllermeier. A Unified Model for Multilabel Classification and Ranking. *Proceedings of the 2006 Conference on ECAI 2006: 17th European Conference on Artificial Intelligence August 29 – September 1, 2006, Riva Del Garda, Italy*, pages 489–493, 2006.

[45] Trstenjak Bruno, Mikac Sasa, and Dzenana Donko. KNN with TF-IDF based Framework for Text Categorization. *Procedia Engineering*, 69:1356–1364, November 2013.

[46] Sahan Bulathwela, María Pérez-Ortiz, Rishabh Mehrotra, Davor Orlic, Colin de la Higuera, John Shawe-Taylor, and Emine Yilmaz. SUM'20: State-Based User Modelling. *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 899–900, 2020.

[47] Khoo Khyou Bun and M. Ishizuka. Emerging Topic Tracking System. *Proceedings Third International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems. WECWIS 2001*, pages 2–11, 2001.

[48] Robin Burke, Nasim Sonboli, and Aldo Ordonez-Gauger. Balanced Neighborhoods for Multi-sided Fairness in Recommendation. *Conference on Fairness, Accountability and Transparency*, pages 202–214, 2018.

[49] Steven Caldwell Brown and Amanda Krause. A Psychological Approach to Understanding the Varied Functions that Different Music Formats Serve. English. *Proceedings of the 14th International Conference on Music Perception and Cognition*, pages 849–851, July 2016. 14th Biennial International Conference on Music Perception and Cognition, ICMPC14.

[50] Pablo Castells, Neil J Hurley, and Saul Vargas. Novelty and Diversity in Recommender Systems, *Recommender Systems Handbook*, pages 881–918. 2015.

[51] Soumen Chakrabarti, Rajiv Khanna, Uma Sawant, and Chiru Bhattacharyya. Structured Learning for Non-Smooth Ranking Losses. *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 88–96, 2008.

[52] Abhijnan Chakraborty, Gourab K. Patro, Niloy Ganguly, Krishna P. Gummadi, and Patrick Loiseau. Equality of Voice: Towards Fair Representation in Crowdsourced Top-K Recommendations. *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 129–138, 2019.

[53] Wei-Cheng Chang, Hsiang-Fu Yu, Kai Zhong, Yiming Yang, and Inderjit S. Dhillon. Taming Pretrained Transformers for Extreme Multi-label Text Classification. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020.

[54] Wei-Cheng Chang, Hsiang-Fu Yu, Kai Zhong, Yiming Yang, and Inderjit S. Dhillon. The Unknown Benefits of using a Soft-F1 Loss in Classification Systems. *Towards Data Science*, 2019.

[55] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[56] Hansheng Chen, Jiatao Gu, Anpei Chen, Wei Tian, Zhuowen Tu, Lingjie Liu, and Hao Su. Single-Stage Diffusion NeRF: A Unified Approach to 3D Generation and Reconstruction, 2023. arXiv: 2304.06714v2.

[57] James J. Chen, Chen-An Tsai, Hojin Moon, Hongshik Ahn, John J. Young, and Chun-Houh Chen. Decision Threshold Adjustment in Class Prediction. *SAR and QSAR in Environmental Research*, 17(3):337–352, 2006.

[58] Laming Chen, Guoxin Zhang, and Hanning Zhou. Fast Greedy MAP Inference for Determinantal Point Process to Improve Recommendation Diversity. *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 5627–5638, 2018.

[59] Tao Chen, Chenhui Wang, and Hongming Shan. BerDiff: Conditional Bernoulli Diffusion Model for Medical Image Segmentation, 2023. arXiv: 2304.04429.

[60] Tianqi Chen and Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016.

[61] Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen, Rory Mitchell, Ignacio Cano, Tianyi Zhou, Mu Li, Junyuan Xie, Min Lin, Yifeng Geng, and Yutian Li. *xgboost: Extreme Gradient Boosting*. R package version 1.5.0.2. 2021.

[62] Wanyu Chen, Pengjie Ren, Fei Cai, Fei Sun, and Maarten de Rijke. Improving End-to-End Sequential Recommendations with Intent-aware Diversification. *CIKM 2020: 29th ACM International Conference on Information and Knowledge Management*, pages 175–184, 2020.

[63] Wei Chen, Tie-yan Liu, Yanyan Lan, Zhi-ming Ma, and Hang Li. Ranking Measures and Loss Functions in Learning to Rank. *Advances in Neural Information Processing Systems*, volume 22, 2009.

[64] Yifan Chen and Maarten de Rijke. A Collective Variational Autoencoder for Top-N Recommendation with Side Information. *3rd Workshop on Deep Learning for Recommender Systems*, 2018.

[65] Zhixiang Chen, Xiannong Meng, Binhai Zhu, and R.H. Fowler. WebSail: From On-line Learning to Web Search. *Proceedings of the First International Conference on Web Information Systems Engineering*, volume 1, 206–213 vol.1, 2000.

[66] Justin Cheng, Caroline Lo, and Jure Leskovec. Predicting Intent Using Activity Logs. *Proceedings of the 26th International Conference on World Wide Web Companion - WWW '17 Companion*, 2017.

[67] Konstantina Christakopoulou, Madeleine Traverse, Trevor Potter, Emma Marriott, Daniel Li, Chris Haulk, Ed H. Chi, and Minmin Chen. Deconfounding User Satisfaction Estimation from Response Rate Bias. *Fourteenth ACM Conference on Recommender Systems*, 2020.

[68] Wei-Ta Chu and Hung-Jui Guo. Movie Genre Classification based on Poster Images with Deep Neural Networks. *Proceedings of the Workshop on Multimodal Understanding of Social, Affective and Subjective Attributes*, 2017.

[69] Aleksandr Chuklin, Pavel Serdyukov, and Maarten de Rijke. Using Intent Information to Model User Behavior in Diversified Search. *34th European Conference on Information Retrieval (ECIR'13)*, 2013.

[70] Amanda Clare and Ross D. King. Knowledge Discovery in Multi-label Phenotype Data. *Lecture Notes in Computer Science*:42–53, 2001.

[71] Charles L.A. Clarke, Maheedhar Kolla, Gordon V. Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. Novelty and Diversity in Information Retrieval Evaluation. *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 659–666, 2008.

[72] Cornell-University. arXiv dataset and metadata of 1.7M+ scholarly papers across STEM. `kaggle.com/Cornell-University/arxiv`, 2021.

[73] Corinna Cortes and Vladimir Vapnik. Support-Vector Networks. *Machine Learning*, pages 273–297, 1995.

[74] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. 1991.

[75] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion Models in Vision: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*:1–20, 2023.

[76] Washington Cunha, Vítor Mangaravite, Christian Gomes, Sérgio Canuto, Elaine Resende, Cecilia Nascimento, Felipe Viegas, Celso França, Wellington Santos Martins, Jussara M. Almeida, Thierson Rosa, Leonardo Rocha, and Marcos André Gonçalves. On the cost-effectiveness of neural and non-neural approaches and representations for text classification: A comprehensive comparative study. *Information Processing & Management*, 58(3):102481, 2021.

[77] Maurizio Ferrari Dacrema, Simone Boglio, Paolo Cremonesi, and Dietmar Jannach. A Troubling Analysis of Reproducibility and Progress in Recommender Systems Research. *ACM Transactions on Information Systems*, 39(2):20:1–20:49, 2021.

[78] Stijn Decubber, Thomas Mortier, Krzysztof Dembczynski, and Willem Waegeman. Deep F-Measure Maximization in Multi-label Classification: A Comparative Study. *ECML/PKDD*, pages 290–305, 2018.

[79] Deezer. Deezer About Page. https://www.deezer.com/en/company, 2022. Accessed on 03.02.2022.

[80] Florence Dehart. The Application of Special Library Services and Techniques to the College Library. *College & Research Libraries*, 27(2):130–152, 1966.

[81] Krzysztof Dembczyński, Weiwei Cheng, and Eyke Hüllermeier. Bayes Optimal Multilabel Classification via Probabilistic Classifier Chains. *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pages 279–286, 2010.

[82] Krzysztof Dembczynski, Arkadiusz Jachnik, Wojciech Kotlowski, Willem Waegeman, and Eyke Huellermeier. Optimizing the F-Measure in Multi-Label Classification: Plug-in Rule Approach versus Structured Loss Minimization. *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of number 3, pages 1130–1138, 2013.

[83] Krzysztof Dembczynski, Willem Waegeman, Weiwei Cheng, and Eyke Hüllermeier. An Exact Algorithm for F-Measure Maximization. *Advances in Neural Information Processing Systems*, volume 24, 2011.

[84] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[85] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, June 2019.

[86] Jwala Dhamala, Tony Sun, Varun Kumar, Satyapriya Krishna, Yada Pruksachatkun, Kai-Wei Chang, and Rahul Gupta. BOLD: Dataset and Metrics for Measuring Biases in Open-Ended Language Generation. *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 862–872, 2021.

[87] Tommaso Di Noia, Vito Claudio Ostuni, Jessica Rosati, Paolo Tomeo, and Eugenio Di Sciascio. An Analysis of Users' Propensity toward Diversity in Recommendations. *Proceedings of the 8th ACM Conference on Recommender Systems*, pages 285–288, 2014.

[88] Ali Diba, Mohsen Fayyaz, Vivek Sharma, Manohar Paluri, Jurgen Gall, Rainer Stiefelhagen, and Luc Van Gool. Large Scale Holistic Video Understanding, 2019. arXiv: 1904.11451v3.

[89] Carmine DiMascio. py-readability-metrics. https://github.com/cdimascio/py-readability-metrics, version 1.4.5, 2020.

[90] Qinxu Ding, Yong Liu, Chunyan Miao, Fei Cheng, and Haihong Tang. A Hybrid Bandit Framework for Diversified Recommendation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(5):4036–4044, 2021.

[91] Paolo Dragone, Rishabh Mehrotra, and Mounia Lalmas. Deriving User- and Content-specific Rewards for Contextual Bandits. *The World Wide Web Conference on - WWW '19*, 2019.

[92] Tim Draws, Nava Tintarev, and Ujwal Gadiraju. Assessing Viewpoint Diversity in Search Results Using Ranking Fairness Metrics. *ACM SIGKDD Explorations Newsletter*, 23(1):50–58, 2021.

[93] Jingcheng Du, Qingyu Chen, Yifan Peng, Yang Xiang, Cui Tao, and Zhiyong Lu. ML-Net: Multi-label Classification of Biomedical Texts with Deep Neural Networks. *Journal of the American Medical Informatics Association*, 26(11):1279–1285, 2019.

[94] Yu Du, Sylvie Ranwez, Nicolas Sutton-Charani, and Vincent Ranwez. Is Diversity Optimization Always Suitable? Toward a Better Understanding of Diversity within Recommendation Approaches. *Information Processing & Management*, 58(6):102721, 2021.

[95] Huizhong Duan and ChengXiang Zhai. Mining Coordinated Intent Representation for Entity Search and Recommendation. *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 2015.

[96] Elad Eban, Mariano Schain, Alan Mackey, Ariel Gordon, Ryan Rifkin, and Gal Elidan. Scalable Learning of Non-Decomposable Objectives. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54, pages 832–840, 2017.

[97] Michael D. Ekstrand, F. Maxwell Harper, Martijn C. Willemsen, and Joseph A. Konstan. User Perception of Differences in Recommender Algorithms. *Proceedings of the 8th ACM Conference on Recommender Systems*, pages 161–168, 2014.

[98] Michael D. Ekstrand, Mucun Tian, Ion Madrazo Azpiazu, Jennifer D. Ekstrand, Oghenemaro Anuyah, David McNeill, and Maria Soledad Pera. All The Cool Kids, How Do They Fit In?: Popularity and Demographic Biases in Recommender Evaluation and Effectiveness. *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, volume 81, pages 172–186, 2018.

[99] André Elisseeff and Jason Weston. A Kernel Method for Multi-Labelled Classification. *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, pages 681–687, 2001.

[100] Dominik Maria Endres and Johannes E Schindelin. A New Metric for Probability Distributions. *IEEE Transactions on Information theory*, 49(7):1858–1860, 2003.

[101] Farzad Eskandanian, Bamshad Mobasher, and Robin Burke. A Clustering Approach for Personalizing Diversity in Collaborative Recommender Systems. *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*, pages 280–284, 2017.

[102] Sarah Eskens, Natali Helberger, and Judith Moeller. Challenged by News Personalisation: Five Perspectives on the Right to Receive Information. *Journal of Media Law*, 9(2):259–284, 2017.

[103] European Commission. General Data Protection Regulation. `https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679`, April 26, 2016.

[104] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The PASCAL visual object classes challenge 2007 (VOC2007) results. `http://www.pascal-network.org/challenges/VOC/voc2007/index.html`, 2007.

[105] Brian Feldman. Piracy Is Back. *New York Magazine*, June 26, 2019.

[106] William Feller. On the Theory of Stochastic Processes, with Particular Reference to Applications. *Proceedings of the First Berkeley Symposium on Mathematical Statistics and Probability*, 1949.

[107] Alberto Fernández. *Learning from Imbalanced Data Sets*. eng. 1st ed. 2018. Edition, 2018.

[108] Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. Are We Really Making Much Progress? A Worrying Analysis of Recent Neural Recommendation Approaches. *Proceedings of the 13th ACM Conference on Recommender Systems*, 2019.

[109] Raul Ferrer-Conill and Edson C. Tandoc Jr. The Audience-Oriented Editor. *Digital Journalism*, 6(4):436–453, 2018.

[110] Nicola Ferro and Diane Kelly. SIGIR Initiative to Implement ACM Artifact Review and Badging. *SIGIR Forum*, 52(1), 2018.

[111] Ronald A. Fisher. On an Absolute Criterion for Fitting Frequency Curves. *Messenger of Mathematics*, 41:155–160, 1912.

[112] K. Fukushima. Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position. *Biological Cybernetics*, 36:193 –202, 1980.

[113] Yu Gai, Zheng Zhang, and Kyunghyun Cho. Gradient-based learning for F-measure and other performance metrics, 2019. OpenReview: `H1zxjsCqKQ`.

[114] Lu Gan, Diana Nurbakova, Léa Laporte, and Sylvie Calabretto. Enhancing Recommendation Diversity Using Determinantal Point Processes on Knowledge Graphs. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2001–2004, 2020.

[115] Zhujin Gao, Junliang Guo, Xu Tan, Yongxin Zhu, Fang Zhang, Jiang Bian, and Linli Xu. Difformer: Empowering Diffusion Models on the Embedding Space for Text Generation, 2023. arXiv: 2212.09412.

[116] Jean Garcia-Gathright, Christine Hosey, Brian St. Thomas, Ben Carterette, and Fernando Diaz. Mixed Methods for Evaluating User Satisfaction. *Proceedings of the 12th ACM Conference on Recommender Systems*, 2018.

[117] Jean Garcia-Gathright, Brian St. Thomas, Christine Hosey, Zahra Nazari, and Fernando Diaz. Understanding and Evaluating User Satisfaction with Music Discovery. *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018.

[118] Francisco Garcia-Valero, Michal Kazimierczak, Carolina Arias-Burgos, and Nathan Wajsman. Online Copyright Infringement in the European Union. *European Union Intellectual Property Office*, 2021.

[119] Florent Garcin, Christos Dimitrakakis, and Boi Faltings. Personalized News Recommendation with Context Trees. *Proceedings of the 7th ACM Conference on Recommender Systems*, pages 105–112, 2013.

[120] Maxime Gasse and Alex Aussem. F-measure Maximization in Multi-Label Classification with Conditionally Independent Label Subsets, 2016. arXiv: 1604.07759.

[121] Carlos A. Gomez-Uribe and Neil Hunt. The Netflix Recommender System. *ACM Transactions on Management Information Systems*, 6(4):1–19, 2015.

[122] Irving J. Good. Rational Decisions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 14(1):107–114, 1952.

[123] Google. Feature Vectors of Images with MobileNet V2 (depth multiplier 1.00) trained on ImageNet (ILSVRC-2012-CLS). https://tfhub.dev/google/imagenet/mobilenet_v2_100_224/feature_vector/4, 2021.

[124] Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting Deep Learning Models for Tabular Data. *Advances in Neural Information Processing Systems*, 2021.

[125] Josif Grabocka, Randolf Scholz, and Lars Schmidt-Thieme. Learning Surrogate Losses, 2019. arXiv: 1905.10108.

[126] Christian Grece and Marta Jiménez Pumares. Film and TV Content in VOD Catalogues – 2020 edition. *European Audiovisual Observatory*, 2020.

[127] Andreas Grün and Xenija Neufeld. Challenges Experienced in Public Service Media Recommendation Systems. *Fifteenth ACM Conference on Recommender Systems*, pages 541–544, 2021.

[128] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On Calibration of Modern Neural Networks. *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 1321–1330, 2017.

[129] Liyi Guo, Rui Lu, Haoqi Zhang, Junqi Jin, Zhenzhe Zheng, Fan Wu, Jin Li, Haiyang Xu, Han Li, Wenkai Lu, and et al. A Deep Prediction Network for Understanding Advertiser Intent and Satisfaction. *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020.

[130] Qi Guo and Eugene Agichtein. Beyond Dwell Time. *Proceedings of the 21st international conference on World Wide Web - WWW '12*, 2012.

[131] Mateo Gutierrez Granada and Daan Odijk. Recommendations at Videoland. *Fifteenth ACM Conference on Recommender Systems*, pages 580–582, 2021.

[132] Rishav Hada, Amir Ebrahimi Fard, Sarah Shugars, Federico Bianchi, Patricia Rossini, Dirk Hovy, Rebekah Tromble, and Nava Tintarev. Beyond Digital "Echo Chamber": The Role of Viewpoint Diversity in Political Discussion, 2022. arXiv: 2212.09056.

[133] Douglas Hanahan and Robert A. Weinberg. Hallmarks of Cancer: The Next Generation. *Cell*, 144(5):646–674, 2011.

[134] F. Maxwell Harper and Joseph A. Konstan. The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems*, 5:Article 19, 2015.

[135] Jun He, Liqun Wang, Liu Liu, Jiao Feng, and Hao Wu. Long Document Classification From Local Word Glimpses via Recurrent Attention Learning. *IEEE Access*, 7:40707–40718, 2019.

[136] Natali Helberger. On the Democratic Role of News Recommenders. *Digital Journalism*, 7(8):993–1012, 2019.

[137] Mariya Hendriksen, Ernst Kuiper, Pim Nauts, Sebastian Schelter, and Maarten de Rijke. Analyzing and Predicting Purchase Intent in E-commerce: Anonymous vs. Identified Customers. *eCOM 2020: The 2020 SIGIIR Workshop on eCommerce*, 2020.

[138] Alex Hern. Streaming Was Supposed to Stop Piracy. Now It Is Easier Than Ever. *The Guardian*, October 2, 2021.

[139] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning Basic Visual Concepts With a Constrained Variational Framework. *International Conference on Learning Representations*, 2017.

[140] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851, 2020.

[141] Jonathan Ho and Tim Salimans. Classifier-Free Diffusion Guidance. *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

[142] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video Diffusion Models, 2022. arXiv: 2204.03458.

[143] Henning Hohnhold, Deirdre O'Brien, and Diane Tang. Focusing on the Long-term (It's Good for Users and Business). *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015.

[144] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength Natural Language Processing in Python. *Release 3.2.1*, 2022.

[145] Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax Flows and Multinomial Diffusion: Learning Categorical Distributions. *Advances in Neural Information Processing Systems*, volume 34, pages 12454–12465, 2021.

[146] Jeremy Howard and Sebastian Ruder. Universal Language Model Fine-tuning for Text Classification. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018.

[147] Haoji Hu, Xiangnan He, Jinyang Gao, and Zhi-Li Zhang. Modeling Personalized Item Frequency Information for Next-basket Recommendation. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020.

[148] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative Filtering for Implicit Feedback Datasets. *2008 Eighth IEEE International Conference on Data Mining*, pages 263–272, 2008.

[149] Hao Huang, Haihua Xu, Xianhui Wang, and Wushour Silamu. Maximum F1-Score Discriminative Training Criterion for Automatic Mispronunciation Detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(4):787–797, 2015.

[150] Jeff Huang, Ryen W. White, Georg Buscher, and Kuansan Wang. Improving Searcher Models Using Mouse Cursor Activity. *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval - SIGIR '12*, 2012.

[151] Jin Huang, Harrie Oosterhuis, and Maarten de Rijke. It Is Different When Items Are Older: Debiasing Recommendations When Selection Bias and User Preferences are Dynamic. *WSDM 2022: The Fifteenth International Conference on Web Search and Data Mining*, pages 381–289, 2022.

[152] Huggingface. DistilBERT. `https://huggingface.co/transformers/model_doc/distilbert.html`, 2021.

[153] David A Hull. *Information Retrieval Using Statistical Classification*. PhD thesis, Stanford University, 1994.

[154] Eyke Hüllermeier, Johannes Fürnkranz, Weiwei Cheng, and Klaus Brinker. Label ranking by learning pairwise preferences. *Artificial Intelligence*, 172(16):1897–1916, 2008.

[155] David Hume. *A Treatise of Human Nature*. 1739.

[156] Himanshu Jain, Venkatesh Balasubramanian, Bhanu Chunduri, and Manik Varma. Slice: Scalable Linear Extreme Classifiers trained on 100 Million Labels for Related Searches. *WSDM '19, February 11–15, 2019, Melbourne, VIC, Australia*, 2019. Best Paper Award at WSDM '19.

[157] Himanshu Jain, Yashoteja Prabhu, and Manik Varma. Extreme Multi-Label Loss Functions for Recommendation, Tagging, Ranking; Other Missing Label Applications. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 935–944, 2016.

[158] Dietmar Jannach, Gabriel de Souza P. Moreira, and Even Oldridge. Why Are Deep Learning Models Not Consistently Winning Recommender Systems Competitions Yet? *Proceedings of the Recommender Systems Challenge 2020*, 2020.

[159] Kalervo Järvelin and Jaana Kekäläinen. Cumulated Gain-Based Evaluation of IR Techniques. *ACM Transactions on Information Systems*, 20(4):422–446, October 2002.

[160] Tony Jebara. Personalization of Spotify Home and TensorFlow. `https://www.oreilly.com/radar/personalization-of-spotify-home-and-tensorflow/`, 2019. Accessed on 13.02.2022.

[161] Harold Jeffreys. An Invariant Form for the Prior Probability in Estimation Problems. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 186(1007):453–461, 1946.

[162] Choi Jeongwhan, Hong Seoyoung, Noseong Park, and Sung-Bae Cho. Blurring-Sharpening Process Models for Collaborative Filtering. *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2023.

[163] Yacine Jernite, Anna Choromanska, and David Sontag. Simultaneous Learning of Trees and Representations for Extreme Classification and Density Estimation. *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, pages 1665–1674, 2017.

[164] Bo Jin, Ke Cheng, Liang Zhang, Yanjie Fu, Minghao Yin, and Lu Jiang. Partial Relationship Aware Influence Diffusion via a Multi-Channel Encoding Scheme for Social Recommendation. *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, 585–594, 2020.

[165] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of Tricks for Efficient Text Classification. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, April 2017.

[166] Ke Ju, Lifeng Lin, Haitao Chu, Liangliang Cheng, and Chang Xu. Laplace Approximation, Penalized Quasi-likelihood, and Adaptive Gauss-Hermite Quadrature for Generalized Linear Mixed Models: Towards meta-analysis of binary outcome with sparse data. *BMC Medical Research Methodology*, 20, June 2020.

[167] Michael Jugovac, Dietmar Jannach, and Lukas Lerche. Efficient Optimization of Multiple Recommendation Quality Factors According to Individual User Tendencies. *Expert Systems with Applications*, 81:321–331, 2017.

[168] In-Ho Kang and GilChang Kim. Query Type Classification for Web Document Retrieval. *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, pages 64–71, 2003.

[169] Mozhgan Karimi, Dietmar Jannach, and Michael Jugovac. News Recommender Systems–Survey and Roads Ahead. *Information Processing & Management*, 54(6):1203–1227, 2018.

[170] Firas Khader, Gustav Müller-Franzes, Soroosh Tayebi Arasteh, Tianyu Han, Christoph Haarburger, Maximilian Schulze-Hagen, Philipp Schad, Sandy Engelhardt, Bettina Baeßler, Sebastian Foersch, Johannes Stegmaier, Christiane Kuhl, Sven Nebelung, Jakob Nikolas Kather, and Daniel Truhn. Denoising diffusion probabilistic models for 3D medical image generation. *Scientific Reports*, 13(1), 2023.

[171] J. Kiefer and J. Wolfowitz. Stochastic Estimation of the Maximum of a Regression Function. *The Annals of Mathematical Statistics*, 23(3):462 –466, 1952.

[172] Benjamin Kille, Frank Hopfgartner, Torben Brodt, and Tobias Heintz. The Plista Dataset. *Proceedings of the 2013 International News Recommender Systems Workshop and Challenge*, pages 16–23, 2013.

[173] Youngho Kim, Ahmed Hassan, Ryen W. White, and Imed Zitouni. Comparing Client and Server Dwell Time Estimates for Click-level Satisfaction Prediction. *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, 2014.

[174] J. Peter Kincaid, Robert P. Fishburne Jr., Richard L. Rogers, and Brad S. Chissom. Derivation of New Readability Formulas (Automated Readability Index, Fog count and Flesch Reading Ease Formula) for Navy Enlisted Personnel, 1975.

[175] Diederik P. Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. On Density Estimation with Diffusion Models. *Advances in Neural Information Processing Systems*, 2021.

[176] Diederik P. Kingma and Max Welling. Auto-encoding Variational Bayes, 2013. arXiv: `1312.6114`.

[177] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment Anything, 2023. arXiv: 2304.02643.

[178] KKBox. KKBox About Page. https://www.kkbox.com/about/en/about, 2022. Accessed on 03.02.2022.

[179] Yasamin Klingler, Claude Lehmann, João Pedro Monteiro, Carlo Saladin, Abraham Bernstein, and Kurt Stockinger. Evaluation of Algorithms for Interaction-Sparse Recommendations: Neural Networks don't Always Win. *25th International Conference on Extending Database Technology*, 2022.

[180] Hyeyoung Ko, Suyeon Lee, Yoonseo Park, and Anna Choi. A Survey of Recommendation Systems: Recommendation Models, Techniques, and Application Fields. *Electronics*, 11(1):141, 2022.

[181] Ron Kohavi and Foster Provost. Applications of Data Mining to Electronic Commerce. *Applications of Data Mining to Electronic Commerce*:5–10, 2001.

[182] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix Factorization Techniques for Recommender Systems. *Computer*, 42(8):30–37, 2009.

[183] Oluwasanmi O. Koyejo, Nagarajan Natarajan, Pradeep K. Ravikumar, and Inderjit S. Dhillon. Consistent Multilabel Classification. *Advances in Neural Information Processing Systems*, volume 28, pages 3321–3329, 2015.

[184] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[185] Jennifer L. Krull and David P. MacKinnon. Multilevel Modeling of Individual and Group Level Mediated Effects. *Multivariate Behavioral Research*, 36(2):249–277, 2001.

[186] Solomon Kullback and Richard A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79 –86, 1951.

[187] Matevz Kunaver and Tomaz Pozrl. Diversity in Recommender Systems – A Survey. *Knowledge-Based Systems*, 123:154 –162, 2017.

[188] Hsiangchu Lai and Tzyy-Ching Yang. A system architecture for intelligent browsing on the Web. *Decision Support Systems*, 28(3):219–239, 2000.

[189] Sudarshan Lamkhede and Sudeep Das. Challenges in Search on Streaming Services. *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019.

[190] Maksim Lapin, Matthias Hein, and Bernt Schiele. Loss Functions for Top-k Error: Analysis and Insights. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[191] Maksim Lapin, Matthias Hein, and Bernt Schiele. Top-k Multiclass SVM. *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, pages 325–333, 2015.

[192] Kristin Larsson, Ilona Silins, Yufan Guo, Anna Korhonen, Ulla Stenius, and Marika Berglund. Text Mining for Improved Human Exposure Assessment. *Toxicology Letters*, 229:S119, 2014.

[193] Sara Latifi, Noemi Mauro, and Dietmar Jannach. Session-aware Recommendation: A Surprising Quest for the State-of-the-art. *Information Sciences*, 573:291–315, 2021.

[194] Jae Sik Lee and Jin Chun Lee. Customer Churn Prediction by Hybrid Model. *Advanced Data Mining and Applications*, pages 959–966, 2006.

[195] Jin Ha Lee and Rachel Price. Understanding Users of Commercial Music Services through Personas: Design Implications. *Proceedings of the 16th International Society for Music Information Retrieval Conference, ISMIR 2015, Málaga, Spain, October 26-30, 2015*, pages 476–482, 2015.

[196] Damien Lefortier, Pavel Serdyukov, and Maarten de Rijke. Online Exploration for Detecting Shifts in Fresh Intent. *CIKM 2014: 23rd ACM Conference on Information and Knowledge Management*, pages 589–598, 2014.

[197] Damien Lefortier, Pavel Serdyukov, Fedor Romanenko, and Maarten de Rijke. Blending Vertical and Web Results: A Case Study Using Video Intent. *36th European Conference on Information Retrieval (ECIR 2014)*, 2014.

[198] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, and Christian Bizer. DBpedia – A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web*, 6(2):167–195, 2015.

[199] Zhaoqi Leng, Mingxing Tan, Chenxi Liu, Ekin Dogus Cubuk, Jay Shi, Shuyang Cheng, and Dragomir Anguelov. PolyLoss: A Polynomial Expansion Perspective of Classification Loss Functions. *International Conference on Learning Representations*, 2022.

[200] Tuck W. Leong and Peter C. Wright. Revisiting Social Practices Surrounding Music. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2013.

[201] Hanze Li, Scott Sanner, Kai Luo, and Ga Wu. A Ranking Optimization Approach to Latent Linear Critiquing for Conversational Recommender Systems. *Fourteenth ACM Conference on Recommender Systems*, 13–22, 2020.

[202] Ming Li, Sami Jullien, Mozhdeh Ariannezhad, and Maarten de Rijke. A Next Basket Recommendation Reality Check, 2021. arXiv: `2109.14233`.

[203] Yuncheng Li, Yale Song, and Jiebo Luo. Improving Pairwise Ranking for Multi-label Image Classification. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[204] Zihao Li, Aixin Sun, and Chenliang Li. DiffuRec: A Diffusion Model for Sequential Recommendation, 2023. arXiv: 2304.00686.

[205] Dawen Liang, Laurent Charlin, James McInerney, and David M. Blei. Modeling User Exposure in Recommendation. *Proceedings of the 25th International Conference on World Wide Web*, 951–961, 2016.

[206] Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. Variational Autoencoders for Collaborative Filtering. *Proceedings of the 2018 World Wide Web Conference*, 689–698, 2018.

[207] Yile Liang and Tieyun Qian. Recommending Accurate and Diverse Items Using Bilateral Branch Network, 2021. arXiv: 2101.00781.

[208] Friedrich Liese and Igor Vajda. On Divergences and Informations in Statistics and Information Theory. *IEEE Transactions on Information Theory*, 52(10):4394–4412, 2006.

[209] Bibo Lin and Seth C. Lewis. The One Thing Journalistic AI Just Might Do for Democracy. *Digital Journalism*, 10(10):1627–1649, 2022.

[210] Jeffrey Lin and Sheng Luo. Deep Learning for the Dynamic Prediction of Multivariate Longitudinal and Survival Data. *Statistics in Medicine*, 41(15):2894–2907, 2022.

[211] Jianhua Lin. Divergence Measures Based on the Shannon Entropy. *IEEE Transactions on Information theory*, 37(1):145–151, 1991.

[212] Ting-En Lin, Hua Xu, and Hanlei Zhang. Discovering New Intents via Constrained Deep Adaptive Clustering with Cluster Refinement. *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34 of number 05, pages 8360–8367, 2020.

[213] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal Loss for Dense Object Detection. *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.

[214] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. *Lecture Notes in Computer Science*:740–755, 2014.

[215] Zachary C. Lipton, Charles Elkan, and Balakrishnan Naryanaswamy. Optimal Thresholding of Classifiers to Maximize F1 Measure. *Lecture Notes in Computer Science*:225–239, 2014.

[216] Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang. Deep Learning for Extreme Multi-label Text Classification. *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017.

[217] Shilong Liu, Lei Zhang, Xiao Yang, Hang Su, and Jun Zhu. Query2Label: A Simple Transformer Way to Multi-Label Classification, 2021. arXiv: 2107.10834.

[218] Yiqun Liu, Ye Chen, Jinhui Tang, Jiashen Sun, Min Zhang, Shaoping Ma, and Xuan Zhu. Different Users, Different Opinions. *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2015.

[219] Felicia Loecherbach, Judith Moeller, Damian Trilling, and Wouter van Atteveldt. The Unified Framework of Media Diversity: A Systematic Literature Review. *Digital Journalism*:1–38, 2020.

[220] Steven Loria. textblob Documentation. `https://textblob.readthedocs.io/en/dev/quickstart.html`, version 0.7.0, 2021.

[221] Eneldo Loza Mencia and Johannes Furnkranz. Pairwise learning of multilabel classifications with perceptrons. *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 2899–2906, 2008.

[222] Feng Lu, Anca Dumitrache, and David Graus. Beyond Optimizing for Clicks: Incorporating Editorial Values in News Recommendation, 2020. arXiv: 2004.09980.

[223] Hongyu Lu, Min Zhang, and Shaoping Ma. Between Clicks and Satisfaction: Study on Multi-Phase User Preferences and Satisfaction for Online News Reading. *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 435–444, 2018.

[224] Robert Duncan Luce. *Individual Choice Behavior: A Theoretical Analysis*. 1959.

[225] Malte Ludewig and Dietmar Jannach. Evaluation of session-based recommendation algorithms. *User Modeling and User-Adapted Interaction*, 28(4–5):331–390, 2018.

[226] Kai Luo, Hojin Yang, Ga Wu, and Scott Sanner. Deep Critiquing for VAE-Based Recommender Systems. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1269–1278, 2020.

[227] Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. Learning disentangled representations for recommendation. *Advances in Neural Information Processing Systems*, volume 32, 2019.

[228] Jun Ma and Bo Wang. Segment Anything in Medical Images, 2023. arXiv: 2304.12306.

[229] Gjorgji Madjarov, Dragi Kocev, Dejan Gjorgjevikj, and Sašo Džeroski. An extensive experimental comparison of methods for multi-label learning. eng. *Pattern recognition*, 45(9):3084–3104, 2012.

[230] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. 2008.

[231] Benjamin Marlin. *Collaborative Filtering: A Machine Learning Perspective*. 2004.

[232] Nicolas Michael Mattis, Philipp K Masur, Judith Moeller, and Wouter van Atteveldt. Nudging towards Diversity: A Theoretical Framework for Facilitating Diverse News Consumption through Recommender Design, 2021.

[233] Rishabh Mehrotra, Ahmed Hassan Awadallah, Milad Shokouhi, Emine Yilmaz, Imed Zitouni, Ahmed El Kholy, and Madian Khabsa. Deep Sequential Models for Task Satisfaction Prediction. *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017.

[234] Rishabh Mehrotra, Ahmed Hassan Awadallah, and Emine Yilmaz. LearnIR: WSDM 2018 Workshop on Learning from User Interactions. *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 2018.

[235] Rishabh Mehrotra, Mounia Lalmas, Doug Kenney, Thomas Lim-Meng, and Golli Hashemian. Jointly Leveraging Intent and Interaction Signals to Predict User Satisfaction with Slate Recommendations. *The World Wide Web Conference on - WWW '19*, 2019.

[236] Rishabh Mehrotra, James McInerney, Hugues Bouchard, Mounia Lalmas, and Fernando Diaz. Towards a Fair Marketplace. *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018.

[237] Aditya K Menon, Ankit Singh Rawat, Sashank Reddi, and Sanjiv Kumar. Multilabel reductions: what is my loss optimising? *Advances in Neural Information Processing Systems*, volume 32, 2019.

[238] Clara Menzen, Manon Kok, and Kim Batselier. Alternating linear scheme in a Bayesian framework for low-rank tensor approximation, 2021. arXiv: `2012.11228`.

[239] Donald Metzler, Yi Tay, Dara Bahri, and Marc Najork. Rethinking Search: Making Domain Experts out of Dilettantes. *SIGIR Forum*, 55(1), 2021.

[240] Lien Michiels, Robin Verachtert, and Bart Goethals. RecPack: An (other) Experimentation Toolkit for Top-N Recommendation using Implicit Feedback Data. *Proceedings of the 16th ACM Conference on Recommender Systems*, pages 648–651, 2022.

[241] Timo Milbich, Omair Ghori, Ferran Diego, and Björn Ommer. Unsupervised Representation Learning by Discovering Reliable Image Relations. *Pattern Recognition*, 102:107107, 2020.

[242] Margaret Mitchell, Dylan Baker, Nyalleng Moorosi, Emily Denton, Ben Hutchinson, Alex Hanna, Timnit Gebru, and Jamie Morgenstern. Diversity and Inclusion Metrics in Subset Selection. *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 117–123, 2020.

[243] Judith Möller, Damian Trilling, Natali Helberger, and Bram van Es. Do Not Blame It on the Algorithm: An Empirical Assessment of Multiple Recommender Systems and their Impact on Content Diversity. *Information, Communication & Society*, 21(7):959–977, 2018.

[244] Lynge Asbjørn Møller. Recommended for You: How Newspapers Normalise Algorithmic News Recommendation to Fit Their Gatekeeping Role. *Journalism Studies*, 23(7):800–817, 2022.

[245] Masahiro Morita and Yoichi Shinoda. Information Filtering Based on User Behavior Analysis and Best Match Text Retrieval. *SIGIR '94*, pages 272–281, 1994.

[246] Mats Mulder, Oana Inel, Jasper Oosterman, and Nava Tintarev. Operationalizing Framing to Support Multiperspective Recommendations of Opinion Pieces. *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 478–488, 2021.

[247] MUSO. Muso Discover Q1 2022 Digital Piracy Data Insights. *MUSO*, 2022.

[248] Felipe Kenji Nakano, Ricardo Cerri, and Celine Vens. Active Learning for Hierarchical Multi-label Classification. *Data Mining and Knowledge Discovery*, 34(5):1496–1530, 2020.

[249] Harikrishna Narasimhan, Purushottam Kar, and Prateek Jain. Optimizing Non-Decomposable Performance Measures: A Tale of Two Classes. *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, pages 199–208, 2015.

[250] Neha. Movie Genre from its Poster. `https://www.kaggle.com/neha1703/movie-genre-from-its-poster`, 2018.

[251] Preksha Nema, Alexandros Karatzoglou, and Filip Radlinski. Disentangling Preference Representations for Recommendation Critiquing with ß-VAE. *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 1356–1365, 2021.

[252] Newman Nic, Richard Fletcher, Antonis Kalogeropoulos, David AL Levy, and Rasmus Kleis Nielsen. Reuters Institute Digital News Report 2018. *Reuters Institute for the Study of Journalism*:39, 2018.

[253] Xia Ning and George Karypis. SLIM: Sparse Linear Methods for Top-N Recommender Systems. *2011 IEEE 11th International Conference on Data Mining*, pages 497–506, 2011.

[254] Behrooz Omidvar-Tehrani, Sruthi Viswanathan, Frederic Roulland, and Jean-Michel Renders. Sage: Interactive State-aware Point-of-interest Recommendation. *Workshop on State-Based User Modelling (SUM '20)*, 2020.

[255] Mícheál. O'Searcoid. *Metric Spaces*. eng. 1st ed. 2007. Edition, 2007.

[256] Rafael Padilla, Sergio L. Netto, and Eduardo A. B. da Silva. A Survey on Performance Metrics for Object-Detection Algorithms. *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 237–242, 2020.

[257] Rong Pan, Yunhong Zhou, Bin Cao, Nathan N. Liu, Rajan Lukose, Martin Scholz, and Qiang Yang. One-Class Collaborative Filtering. *2008 Eighth IEEE International Conference on Data Mining*, pages 502–511, 2008.

[258] Javier Parapar and Filip Radlinski. Towards Unified Metrics for Accuracy and Diversity for Recommender Systems. *Fifteenth ACM Conference on Recommender Systems*, pages 75–84, 2021.

[259] Eli Pariser. *The Filter Bubble: How the New Personalized Web is Changing What We Read and How We Think*. 2011.

[260] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. 2019.

[261] Yash Patel, Giorgos Tolias, and Jiri Matas. Recall@k Surrogate Loss with Large Batches and Similarity Mixup. *CVPR*, 2022.

[262] Jiahuan Pei, Pengjie Ren, and Maarten de Rijke. A Cooperative Memory Network for Personalized Task-oriented Dialogue Systems with Incomplete User Profiles. *The Web Conference 2021*, 2021.

[263] Adler Perotte, Rimma Pivovarov, Karthik Natarajan, Nicole Weiskopf, Frank Wood, and Noémie Elhadad. Diagnosis Code Assignment: Models and Evaluation Metrics. *Journal of the American Medical Informatics Association*, 21(2):231–237, 2014.

[264] Yashoteja Prabhu, Anil Kag, Shrutendra Harsola, Rahul Agrawal, and Manik Varma. Parabel: Partitioned Label Trees for Extreme Classification with Application to Dynamic Search Advertising. *Proceedings of the 2018 World Wide Web Conference*, pages 993–1002, 2018.

[265] Shameem A. Puthiya Parambath, Nicolas Usunier, and Yves Grandvalet. A Coverage-Based Approach to Recommendation Diversity On Similarity Graph. *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 15–22, 2016.

[266] Lijing Qin and Xiaoyan Zhu. Promoting Diversity in Recommendation by Entropy Regularizer. *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, pages 2698–2704, 2013.

[267] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139, pages 8748–8763, 2021.

[268] Harish G. Ramaswamy, Balaji Srinivasan Babu, Shivani Agarwal, and Robert C. Williamson. On the Consistency of Output Code Based Learning Algorithms for Multiclass Learning Problems. *Proceedings of The 27th Conference on Learning Theory*, volume 35, pages 885–902, 2014.

[269] Shaina Raza and Chen Ding. Deep Dynamic Neural Network to trade-off between Accuracy and Diversity in a News Recommender System, 2021. arXiv: `2103.08458`.

[270] Sashank J. Reddi, Satyen Kale, Felix Yu, Daniel Holtmann-Rice, Jiecao Chen, and Sanjiv Kumar. Stochastic Negative Mining for Learning with Large Output Spaces. *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89, pages 1940–1949, 2019.

[271] Steffen Rendle, Walid Krichene, Li Zhang, and Yehuda Koren. Revisiting the Performance of iALS on Item Recommendation Benchmarks. *Sixteenth ACM Conference on Recommender Systems*, 2022.

[272] Steffen Rendle, Li Zhang, and Yehuda Koren. On the Difficulty of Evaluating Baselines: A Study on Recommender Systems, 2019. arXiv: `1905.01395v1`.

[273] Tal Ridnik, Hussam Lawen, Asaf Noy, Emanuel Ben Baruch, Gilad Sharir, and Itamar Friedman. TResNet: High Performance GPU-Dedicated Architecture. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1400–1409, 2021.

[274] Herbert Robbins and Sutton Monro. A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3):400 –407, 1951.

[275] Royi Ronen, Elad Yom-Tov, and Gal Lavee. Recommendations meet web browsing: enhancing collaborative filtering using internet browsing logs. *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, 2016.

[276] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation, pages 234–241, 2015.

[277] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation, 2015. arXiv: `1505.04597`.

[278] Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. In-Place Activated BatchNorm for Memory-Optimized Training of DNNs. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[279] James Rucker and Marcos J. Polanco. Siteseer: Personalized Navigation for the Web. *Commun. ACM*, 40(3):73–76, 1997.

[280] Noveen Sachdeva, Mehak Preet Dhaliwal, Carole-Jean Wu, and Julian McAuley. Infinite Recommendation Networks: A Data-Centric Approach. *Advances in Neural Information Processing Systems*, 2022.

[281] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo-Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. *Advances in Neural Information Processing Systems*, 2022.

[282] Tetsuya Sakai and Zhaohao Zeng. Which Diversity Evaluation Measures Are "Good"? *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 595–604, 2019.

[283] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted Residuals and Linear Bottlenecks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4510–4520, June 2018.

[284] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter, 2019. arXiv: `1910.01108`.

[285] Amartya Sanyal, Pawan Kumar, Purushottam Kar, Sanjay Chawla, and Fabrizio Sebastiani. Optimizing Non-decomposable Measures with Deep Networks. *Machine Learning*, 107, September 2018.

[286] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. *Proceedings of the 10th international conference on World Wide Web*, pages 285–295, 2001.

[287] Marijn Sax. Algorithmic news diversity and democratic theory: adding agonism to the mix. *Digital Journalism*:1–21, 2022.

[288] Jonathan Scarlett and Volkan Cevher. *An Introductory Guide to Fano's Inequality with Applications in Statistical Estimation. Information-Theoretic Methods in Data Science*. 2021, 487–528.

[289] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. AutoRec: Autoencoders Meet Collaborative Filtering. *Proceedings of the 24th International Conference on World Wide Web*, 111–112, 2015.

[290] Oguz Semerci, Aloïs Gruson, Clay Gibson, Ben Lacker, Catherine Edwards, and Vladan Radosavlijevic. Homepage Personalization at Spotify. *RecSys*, 2019.

[291] Fumin Shen, Yadong Mu, Yang Yang, Wei Liu, Li Liu, Jingkuan Song, and Heng Tao Shen. Classification by Retrieval: Binarizing Data and Classifiers. *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 595–604, 2017.

[292] Ilya Shenbin, Anton Alekseev, Elena Tutubalina, Valentin Malykh, and Sergey I. Nikolenko. RecVAE: A New Variational Autoencoder for Top-N Recommendations with Implicit Feedback. *Proceedings of the 13th International Conference on Web Search and Data Mining*, 2020.

[293] Jane B Singer. User-generated Visibility: Secondary Gatekeeping in a Shared Media Space. *New Media & Society*, 16(1):55–73, 2014.

[294] Annelien Smets, Jonathan Hendrickx, and Pieter Ballon. We're in This Together: A Multi-Stakeholder Approach for News Recommenders. *Digital Journalism*:1–19, 2022.

[295] Samuel L. Smith, Pieter-Jan Kindermans, and Quoc V. Le. Don't Decay the Learning Rate, Increase the Batch Size, 2017. arXiv: 1711.00489.

[296] Aaron J. Snoswell, Surya P. N. Singh, and Nan Ye. LiMIIRL: Lightweight Multiple-Intent Inverse Reinforcement Learning, 2021. arXiv: 2106.01777.

[297] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pages 2256–2265, 2015.

[298] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pages 2256–2265, 2015.

[299] Hossein Soleimani and David J. Miller. Semisupervised, Multilabel, Multi-Instance Learning for Structured Data. *Neural Computation*, 29(4):1053–1102, 2017.

[300] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising Diffusion Implicit Models. *International Conference on Learning Representations*, 2021.

[301] Harald Steck. Calibrated Recommendations. *Proceedings of the 12th ACM Conference on Recommender Systems*, pages 154–162, 2018.

[302] Harald Steck. Embarrassingly Shallow Autoencoders for Sparse Data. *The World Wide Web Conference*, 2019.

[303] Harald Steck. Evaluation of Recommendations: Rating-Prediction and Ranking. *Proceedings of the 7th ACM Conference on Recommender Systems*, 213–220, 2013.

[304] Harald Steck. Training and Testing of Recommender Systems on Data Missing Not At Random. *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '10*, 2010.

[305] Daniel Steel, Sina Fazelpour, Kinley Gillette, Bianca Crewe, and Michael Burgess. Multiple Diversity Concepts and their Ethical-epistemic Implications. *European Journal for Philosophy of Science*, 8(3):761–780, 2018.

[306] Stephen V. Stehman. Selecting and interpreting measures of thematic classification accuracy. *Remote Sensing of Environment*, 62(1):77–89, 1997.

[307] Jesper Strömbäck. In Search of a Standard: Four Models of Democracy and their Normative Implications for Journalism. *Journalism Studies*, 6(3):331–345, 2005.

[308] Ning Su, Jiyin He, Yiqun Liu, Min Zhang, and Shaoping Ma. User Intent, Behaviour, and Perceived Satisfaction in Product Search. *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 2018.

[309] Niek Tax, Sander Bockting, and Djoerd Hiemstra. A Cross-benchmark Comparison of 87 Learning to Rank Methods. *Information Processing & Management*, 51(6):757–772, 2015.

[310] Jaime Teevan, Susan T. Dumais, and Eric Horvitz. Personalizing Search via Automated Analysis of Interests and Activities. *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 449–456, 2005.

[311] Ori Tenenboim and Akiba A Cohen. What Prompts Users to Click and Comment: A Longitudinal Study of Online News. *Journalism*, 16(2):198–217, 2015.

[312] Ambuj Tewari and Peter L. Bartlett. On the Consistency of Multiclass Classification Methods. *Learning Theory*, pages 143–157, 2005.

[313] TikTok. How TikTok Recommends Videos #ForYou. `https://newsroom.tiktok.com/en-us/how-tiktok-recommends-videos-for-you`, 2020. Accessed on 11.02.2022.

[314] Damian Trilling and Marieke van Hoof. Between Article and Topic: News Events as Level of Analysis and Their Computational Identification. *Digital Journalism*, 8(10):1317–1337, 2020.

[315] Grigorios Tsoumakas and Ioannis Katakis. Multi-Label Classification: An Overview. *International Journal of Data Warehousing and Mining (IJDWM)*, 3(3):1–13, 2007.

[316] John Wilder Tukey. *Exploratory data analysis*. eng. 1977.

[317] Twitch. Removing Recommendations You Are Not Interested In. `https://help.twitch.tv/s/article/Removing-recommendations-you-are-not-interested-in?language=en_US`, 2022. Accessed on 11.02.2022.

[318] Ali Vardasbi, Gabriel Bénédict, Shashank Gupta, Maria Heuss, Pooya Khandel, Ming Li, and Fatemeh Sarvi. The University of Amsterdam at the TREC 2021 Fair Ranking Track. *TREC Fair Ranking*, 2021.

[319] Saúl Vargas and Pablo Castells. Rank and Relevance in Novelty and Diversity Metrics for Recommender Systems. *Proceedings of the fifth ACM conference on Recommender systems*, pages 109–116, 2011.

[320] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. *Advances in Neural Information Processing Systems*, volume 30, 2017.

[321] Aki Vehtari, Andrew Gelman, and Jonah Gabry. Practical Bayesian Model Evaluation Using Leave-one-out Cross-validation and WAIC. *Statistics and Computing*, 27(5):1413–1432, 2016.

[322] Aki Vehtari, Daniel Simpson, Andrew Gelman, Yuling Yao, and Jonah Gabry. Pareto Smoothed Importance Sampling. 2015. arXiv: `1507.02646v8`.

[323] Koen Verstrepen, Kanishka Bhaduriy, Boris Cule, and Bart Goethals. Collaborative Filtering for Binary, Positiveonly Data. *SIGKDD Explor. Newsl.*, 19(1):1–21, 2017.

[324] Sanne Vrijenhoek. Do You MIND? Reflections on the MIND Dataset for Research on Diversity in News Recommendations. *Advances in Bias and Fairness in Information Retrieval*, pages 147–154, 2023.

[325] Sanne Vrijenhoek, Gabriel Bénédict, Mateo Gutierrez Granada, Daan Odijk, and Maarten de Rijke. RADio – Rank-Aware Divergence Metrics to Measure Normative Diversity in News Recommendations. *Proceedings of the 16th ACM Conference on Recommender Systems*, pages 208–219, 2022.

[326] Sanne Vrijenhoek, Mesut Kaya, Nadia Metoui, Judith Möller, Daan Odijk, and Natali Helberger. Recommenders with a Mission: Assessing Diversity in News Recommendations. *Proceedings of the 2021 Conference on Human Information Interaction and Retrieval*, pages 173–183, 2021.

[327] Sanne Vrijenhoek, Lien Michiels, Johannes Kruse, Jordi Viader Guerrero, Alain Starke, and Nava Tintarev. NORMalize: The First Workshop on Normative Design and Evaluation of Recommender Systems. *17th ACM Conference on Recommender Systems*, 2023.

[328] Willem Waegeman, Krzysztof Dembczyński, Arkadiusz Jachnik, Weiwei Cheng, and Eyke Hüllermeier. On the Bayes-Optimality of F-Measure Maximizers. *J. Mach. Learn. Res.*, 15(1):3333–3388, January 2014.

[329] Joojo Walker, Ting Zhong, Fengli Zhang, Qiang Gao, and Fan Zhou. Recommendation Via Collaborative Diffusion Generative Model. *Knowledge Science, Engineering and Management: 15th International Conference, KSEM 2022, Singapore, August 6–8, 2022, Proceedings, Part III*, 593–605, 2022.

[330] Julian Wallace. Modelling Contemporary Gatekeeping: The Rise of Individuals, Algorithms and Platforms in Digital News Dissemination. *Digital Journalism*, 6(3):274–293, 2018.

[331] Isaac Waller and Ashton Anderson. Generalists and Specialists: Using Community Embeddings to Quantify Activity Diversity in Online Platforms. *The World Wide Web Conference*, pages 1954–1964, 2019.

[332] Jiang Wang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, and Wei Xu. CNN-RNN: A Unified Framework for Multi-Label Image Classification. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[333] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, and et al. Deep High-Resolution Representation Learning for Visual Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*:1–1, 2020.

[334] Ningxia Wang and Li Chen. User Bias in Beyond-Accuracy Measurement of Recommendation Algorithms. *Fifteenth ACM Conference on Recommender Systems*, pages 133–142, 2021.

[335] Shoujin Wang, Liang Hu, Yan Wang, Longbing Cao, Quan Z Sheng, and Mehmet Orgun. Sequential Recommender Systems: Challenges, Progress and Prospects, 2019. arXiv: `2001.04830`.

[336] Wenjie Wang, Fuli Feng, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. Clicks can be Cheating: Counterfactual Recommendation for Mitigating Clickbait Issue. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021.

[337] Wenjie Wang, Xinyu Lin, Fuli Feng, Xiangnan He, and Tat-Seng Chua. Generative Recommendation: Towards Next-generation Recommender Paradigm, 2023. arXiv: `2304.03516v1`.

[338] Xuanhui Wang, Cheng Li, Nadav Golbandi, Mike Bendersky, and Marc Najork. The LambdaLoss Framework for Ranking Metric Optimization. *Proceedings of The 27th ACM International Conference on Information and Knowledge Management (CIKM '18)*, pages 1313–1322, 2018.

[339] Yunchao Wei, Wei Xia, Min Lin, Junshi Huang, Bingbing Ni, Jian Dong, Yao Zhao, and Shuicheng Yan. HCP: A Flexible CNN Framework for Multi-Label Image Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(9):1901–1907, 2016.

[340] Kasper Welbers, Wouter van Atteveldt, Jan Kleinnijenhuis, and Nel Ruigrok. A Gatekeeper among Gatekeepers: News Agency Influence in Print and Online Newspapers in the Netherlands. *Journalism Studies*, 19(3):315–333, 2018.

[341] Hongyi Wen, Longqi Yang, and Deborah Estrin. Leveraging Post-click Feedback for Content Recommendations. *Proceedings of the 13th ACM Conference on Recommender Systems*, 2019.

[342] Wang Wenjie, Xu Yiyan, Fuli Feng, Xinyu Lin, Xiangnan He, and Tat-Seng Chua. Diffusion Recommender Model. *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2023.

[343] Hadley Wickham. ggplot2: Elegant Graphics for Data Analysis – Violin plot. `https://ggplot2.tidyverse.org/reference/geom_violin.html`, 2016.

[344] Baoyuan Wu, Weidong Chen, Yanbo Fan, Yong Zhang, Jinlong Hou, Jie Liu, and Tong Zhang. Tencent ML-Images: A Large-Scale Multi-Label Image Database for Visual Representation Learning. *IEEE Access*, 7:172683–172693, 2019.

[345] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. Neural News Recommendation with Attentive Multi-view Learning, 2019. arXiv: `1907.05576`.

[346] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. NPA: Neural News Recommendation with Personalized Attention. *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2576–2584, 2019.

[347] Chuhan Wu, Fangzhao Wu, Suyu Ge, Tao Qi, Yongfeng Huang, and Xing Xie. Neural News Recommendation with Multi-head Self-attention. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6389–6394, 2019.

[348] Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, and Ming Zhou. MIND: A Large-scale Dataset for News Recommendation. *ACL*, 2020.

[349] Ga Wu, Kai Luo, Scott Sanner, and Harold Soh. Deep Language-Based Critiquing for Recommender Systems. *Proceedings of the 13th ACM Conference on Recommender Systems*, 137–145, 2019.

[350] Le Wu, Peijie Sun, Yanjie Fu, Richang Hong, Xiting Wang, and Meng Wang. A Neural Influence Diffusion Model for Social Recommendation. *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 235–244, 2019.

[351] Qiong Wu, Yong Liu, Chunyan Miao, Yin Zhao, Lu Guan, and Haihong Tang. Recent Advances in Diversified Recommendation, 2019. arXiv: `1905.06589`.

[352] Shangyuan Wu, Edson C. Tandoc, and Charles T. Salmon. Journalism Reconfigured: Assessing Human-machine Relations and the Autonomous Power of Automation in News Production. *Journalism Studies*, 20(10):1440 –1457, 2019.

[353] Wen Wu, Li Chen, and Yu Zhao. Personalizing Recommendation Diversity based on User Personality. *User Modeling and User-Adapted Interaction*, 28(3):237–276, 2018.

[354] Xi-Zhu Wu and Zhi-Hua Zhou. A Unified View of Multi-Label Performance Measures. *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 3780–3788, 2017.

[355] Yao Wu, Christopher DuBois, Alice X. Zheng, and Martin Ester. Collaborative Denoising Auto-Encoders for Top-N Recommender Systems. *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, 153–162, 2016.

[356] Marek Wydmuch, Kalina Jasinska, Mikhail Kuznetsov, Róbert Busa-Fekete, and Krzysztof Dembczyński. A No-Regret Generalization of Hierarchical Softmax to Extreme Multi-Label Classification. *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 6358–6368, 2018.

[357] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. SUN database: Large-scale scene recognition from abbey to zoo. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3485–3492, 2010.

[358] Ruobing Xie, Qi Liu, Shukai Liu, Ziwei Zhang, Peng Cui, Bo Zhang, and Leyu Lin. Improving Accuracy and Diversity in Matching of Recommendation with Diversified Preference Network, 2021. arXiv: 2102.03787.

[359] Guandao Yang, Abhijit Kundu, Leonidas J. Guibas, Jonathan T. Barron, and Ben Poole. Learning a Diffusion Prior for NeRFs, 2023. arXiv: 2304.14473v1.

[360] Hojin Yang, Tianshu Shen, and Scott Sanner. Bayesian Critiquing with Keyphrase Activation Vectors for VAE-Based Recommender Systems. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2111–2115, 2021.

[361] Yiming Yang. An Evaluation of Statistical Approaches to Text Categorization. *Information Retrieval*, 1:69–90, 2004.

[362] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. XLNet: Generalized Autoregressive Pretraining for Language Understanding. *Advances in Neural Information Processing Systems*, volume 32, pages 5753–5763, 2019.

[363] Nan Ye, Kian Ming Adam Chai, Wee Sun Lee, and Hai Leong Chieu. Optimizing F-measure: A Tale of Two Approaches. *ICML*, 2012.

[364] Ian E.H. Yen, Xiangru Huang, Wei Dai, Pradeep Ravikumar, Inderjit Dhillon, and Eric Xing. PPDsparse: A Parallel Primal-Dual Sparse Method for Extreme Classification. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 545–553, 2017.

[365] Xing Yi, Liangjie Hong, Erheng Zhong, Nanthan Nan Liu, and Suju Rajan. Beyond Clicks. *Proceedings of the 8th ACM Conference on Recommender systems - RecSys '14*, 2014.

[366] Emine Yilmaz and Stephen Robertson. On the Choice of Effectiveness Measures for Learning to Rank. English. *Information Retrieval*, 13(3):271–290, 2010.

[367] Youtube. Manage Your Recommendations and Search Results. `https://support.google.com/youtube/answer/6342839?hl=en&co=GENIE.Platform%3DAndroid`, 2022. Accessed on 11.02.2022.

[368] Youtube. YouTube Survey FAQs. `https://support.google.com/youtube/thread/1920627/youtube-survey-faqs?hl=en`, 2019. Accessed on 11.02.2022.

[369] Hsiang-Fu Yu, Prateek Jain, Purushottam Kar, and Inderjit S. Dhillon. Large-Scale Multi-Label Learning with Missing Labels. *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, pages I–593–I–601, 2014.

[370] Matei Zaharia, Reynold S. Xin, Patrick Wendell, Tathagata Das, Michael Armbrust, Ankur Dave, Xiangrui Meng, Josh Rosen, Shivaram Venkataraman, Michael J. Franklin, Ali Ghodsi, Joseph Gonzalez, Scott Shenker, and Ion Stoica. Apache Spark: A Unified Engine for Big Data Processing. *Commun. ACM*, 59(11):56–65, 2016.

[371] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big Bird: Transformers for Longer Sequences. *Advances in Neural Information Processing Systems*, 2020.

[372] Chenshuang Zhang, Chaoning Zhang, Mengchun Zhang, and In So Kweon. Text-to-image Diffusion Models in Generative AI: A Survey, 2023. arXiv: `2303.07909v2`.

[373] Jizhi Zhang, Keqin Bao, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. Is ChatGPT Fair for Recommendation? Evaluating Fairness in Large Language Model Recommendation. *17th ACM Conference on Recommender Systems*, 2023.

[374] Min-Ling Zhang and Zhi-Hua Zhou. A Review on Multi-Label Learning Algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1819–1837, 2014.

[375] Min-Ling Zhang and Zhi-Hua Zhou. ML-KNN: A Lazy Learning Approach to Multi-label Learning. *Pattern Recognition*, 40(7):2038–2048, 2007.

[376] Min-Ling Zhang and Zhi-Hua Zhou. Multilabel Neural Networks with Applications to Functional Genomics and Text Categorization. *IEEE Transactions on Knowledge and Data Engineering*, 18(10):1338–1351, 2006.

[377] Mingyuan Zhang, Harish G. Ramaswamy, and Shivani Agarwal. Convex Calibrated Surrogates for the Multi-Label F-Measure. *Proceedings of the 37th International Conference on Machine Learning*, 2020.

[378] Tong Zhang. Statistical Behavior and Consistency of Classification Methods Based on Convex Risk Minimization. *The Annals of Statistics*, 32(1):56 –85, 2004.

[379] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-Level Convolutional Networks for Text Classification. *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, pages 649–657, 2015.

[380] Zhi-Hua Zhou, Min-Ling Zhang, Sheng-Jun Huang, and Yu-Feng Li. Multi-instance Multi-label Learning. *Artificial Intelligence*, 176(1):2291 –2320, 2012.

[381] Feng Zhu, Hongsheng Li, Wanli Ouyang, Nenghai Yu, and Xiaogang Wang. Learning Spatial Regularization With Image-Level Supervisions for Multi-Label Image Classification. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Summary

Streaming platforms are the prime medium for consuming audio and video. They took over DVDs, CDs, and even ended piracy more than 10 years ago but their main asset – bringing the right content to the right user, through personalization – is still largely under-researched today. We provide a first academic attempt at a machine learning-based personalization flow to match users and content, and monitor the results. As we proceed, we find that there are still many open questions in personalization and especially in recommendation. When recommending an item to a user, how do we use unobservable data, e.g., intent, user and content metadata as input? Can we optimize directly for non-differentiable metrics? What about diversity? To answer these questions, this thesis proposes data, experimental design, loss functions, and metrics.

In Chapter 2, we make sure that content is matched with users via recommendations. With RecFusion, we explore the potential of diffusion models in recommendation contexts. We introduce a binary diffusion model tailored to 1D data and suitable for recommendation scenarios. Beyond the attempt to find novel applications for an existing method, we open the door to guided diffusion models for recommendation that integrate user and item metadata as input to the model.

In Chapter 3, we personalize the recommended content to the user by selecting a single image of the video for recommendation strips, based on users' categorical preferences, such as a favorite movie genre. The task then reduces to classifying candidate movie shots into one or multiple movie genres: multilabel classification. We propose sigmoidF1, a new multilabel classification loss that is a differentiable surrogate of the F1 score. This way, we explicitly model correlation between class labels – e.g., romance and comedy.

In Chapter 4, we take a step back and observe the impact of recommendations on users. We measure how intent, e.g., binge-watching, and behavioral data, e.g., clicks, influence user satisfaction. This chapter is a reproducibility study from the music streaming to the video streaming domain. We provide our experimental design, simulated data, and model design to encourage further research into modeling unobservable user data.

In Chapter 5, we propose another monitoring contribution: a way to measure content diversity on an online platform, adaptable to any kind of definition of diversity the platform may prioritize (e.g., diversity in political positions of the content). With RADio, we introduce a framework for evaluating the diversity of news recommendations by comparing categorical distributions over recommendation lists. We wrote this chapter to inspire researchers to optimize

directly for diversity in recommendation models and to apply these metrics to further domains, such as video and podcast streaming.

Chapter 2 and 3 focus on models to nudge users to content, Chapter 4 and 5 monitor the effects via metrics. In the future we hope to bring these concepts closer together via end-to-end solutions, where personalization models are directly optimized for the desired metric.

Samenvatting

Streamingplatformen zijn de belangrijkste media voor het consumeren van audio en video. Meer dan 10 jaar geleden namen zij DVD's, CD's en zelfs illegale downloads over, maar hun belangrijkste troef – het leveren van de juiste content aan de juiste gebruiker via personalisatie – is tot op heden nog grotendeels onderbelicht in academisch onderzoek. In dit proefschrift doen we een eerste academische poging tot het inrichten van een machine leren *personalization flow* om gebruikers en content te matchen, en de resultaten te monitoren. We ontdekken dat er nog veel open vragen zijn op het gebied van personalisatie, met name op het gebied van aanbevelingen. Hoe gebruiken we niet-waarneembare gegevens, bijvoorbeeld de intentie van gebruikers, en de metadata van gebruikers en de content als input? Kunnen we direct optimaliseren voor niet-differentieerbare metrieken? En hoe zit het met diversiteit optimaliseren? Om deze vragen te beantwoorden, introduceren we in dit proefschrift data, experimenteel ontwerp, verliesfuncties en metrieken.

In Hoofdstuk 2 matchen we content en gebruikers door middel van aanbevelingen. Met RecFusion verkennen we de potentie van diffusiemodellen in de aanbevelingscontext. We introduceren een binair diffusiemodel dat aangepast is op 1D-gegevens, en geschikt is voor aanbevelingsscenario's. Naast onze poging om nieuwe toepassingen te vinden voor een bestaande methode, moedigen we toekomstige onderzoekers aan *guided* diffussiemodellen voor aanbevelingen te gebruiken, met de metadata van zowel gebruikers als items als input.

In Hoofdstuk 3 personaliseren we de aanbevolen content voor de gebruiker door een enkele afbeelding uit de film of serie in kwestie te selecteren voor de rij aanbevelingen, op basis van de categorische voorkeuren van gebruikers, zoals een favoriet filmgenre. De taak wordt dan gereduceerd tot het classificeren van mogelijke afbeeldingen in één of meerdere filmgenres: multilabelclassificatie. We stellen sigmoidF1 voor, een nieuwe verliesfunctie voor multilabelclassificatie die een differentieerbare surrogaat is van de F1-score. Op deze manier modelleren we expliciet correlatie tussen labels van verschillende categorieën (bijvoorbeeld romantiek en komedie).

In Hoofdstuk 4 doen we een stap terug en observeren we de impact van aanbevelingen op gebruikers. We meten hoe de intentie – bijvoorbeeld *bingewatchen* – en gedragsgegevens – bijvoorbeeld muisklikken – invloed hebben op de tevredenheid van gebruikers. Dit hoofdstuk is een reproduceerbaarheidsstudie van het muziek- naar het videostreamingsdomein. We introduceren experimenteel ontwerp, gesimuleerde data, en modelontwerp om verder onderzoek naar het modelleren van niet-waarneembare gebruikersdata aan te moedigen.

In Hoofdstuk 5 leveren we een andere bijdrage aan monitoring: een manier om de diversiteit van content op een online platform te meten, aanpasbaar aan elke definitie van diversiteit die het platform mogelijkerwijs prioriteert (bijvoorbeeld diversiteit in politieke kleur van de content). Met RADio introduceren we een kader voor het evalueren van de diversiteit van nieuwsaanbevelingen door categorische distributies over aanbevelingslijsten te vergelijken. We hebben dit hoofdstuk geschreven om onderzoekers te inspireren om het aanbevelingsmodel direct te optimaliseren voor diversiteit en om deze metriek toe te passen op verdere domeinen, zoals video- en podcaststreaming.

Hoofdstuk 2 en 3 richten zich op modellen om gebruikers naar inhoud te sturen, Hoofdstuk 4 en 5 monitoren de effecten via metrieken. In de toekomst hopen we deze concepten dichter bij elkaar te brengen via *end-to-end* oplossingen, waarbij personalisatiemodellen direct worden geoptimaliseerd voor de gewenste metriek.

Abstract

This thesis describes a machine learning-based personalization flow for streaming platforms: we match users and content like video or music, and monitor the results. We find that there are still many open questions in personalization and especially in recommendation. When recommending an item to a user, how do we use unobservable data, e.g., intent, user and content metadata as input? Can we optimize directly for non-differentiable metrics? What about diversity in recommendations? To answer these questions, this thesis proposes data, experimental design, loss functions, and metrics. In the future, we hope these concepts are brought closer together via end-to-end solutions, where personalization models are directly optimized for the desired metric.