Thèse n° 7978

Dynamic Personalized Ranking

Présentée le 26 août 2022

Faculté informatique et communications
Laboratoire de systèmes d'information répartis
Programme doctoral en informatique et communications

pour l'obtention du grade de Docteur ès Sciences

par

Jérémie RAPPAZ

Acceptée sur proposition du jury

Prof. P. Dillenbourg, président du jury
Prof. K. Aberer, directeur de thèse
Prof. L. Aiello, rapporteur
Prof. A. Anderson, rapporteur
Prof. P. Thiran, rapporteur

École
polytechnique
fédérale
de Lausanne

2022

To Maroussia, Hugo,
Claudine and Serge

Notation

p_k, q_k	Vector elements
$\mathbf{p}, \mathbf{q}$	Vectors
P, Q	Matrices
$\mathbf{M}_{r_1}$	Embedding vector of item i using embedding matrix M
θ	Model parameters
σ	Sigmoid function $\sigma(x) = \frac{1}{1+\exp^{-x}}$
K	Dimensionality of the model
R	Interaction matrix $\in \mathbb{R}^{\lvert U\rvert \times \lvert I\rvert}$
I	Item set
U	User set
$\lvert U\rvert$; $\lvert I\rvert$	Number of users and number of items
u_j	User $u_j \in U$
i_k	Item $i_k \in I$
$\hat{x}_{u_j i_k}$	Model prediction for user u_j and item i_k
$\mathrm{H}(\cdot)$	Heaviside step function

Abstract

Personalized ranking methods are at the core of many systems that learn to produce recommendations from user feedbacks. Their primary objective is to identify relevant items from very large vocabularies and to assist users in discovering new content. These techniques have proven successful in stationary regimes, but the transition to an interactive and social Web, and the rise of user-generated content, increasingly require learning from dynamic factors. Existing approaches, based on distributed vector representations, notoriously fail in fast-changing contexts and sparse regimes; their static representation of users and items prevents them from adapting to contextual changes. Given this limitation, this thesis focuses on introducing new methods to make predictions based on contextual variables, exploit rich (social, temporal) signals, and maintain model consistency over time.

First, we study a recommendation task on a live-streaming platform with a dynamically evolving set of available items. In this context, users have to choose from a limited subset of creators that are currently streaming content. To model this setting, we introduce a self-attentive approach that draws a dependency between available options and re-ranks the most promising candidates. We also show that repeat consumption, and the time at which it occurs, are predictive factors that we incorporate into our model.

Second, we propose a dynamic embedding scheme to maintain a latent space consistent over multiple temporal slices of a dataset, by penalizing unnecessary differences between successive solutions of the model. In order to highlight the potential for analysis of our approach, we apply our methodology to a dataset of news production. Thanks to its ability to propagate information over several time epochs, our model sheds light on important changes in news coverage induced by acquisitions of media companies.

Third, we propose an approach for modeling reciprocal interest on a bartering platform where users exchange goods, for which we design a trade recommendation system. We find that the social ties between members have a strong influence, as does the time at which they trade, therefore we extend our model to be socially- and temporally-aware.

Finally, we study the collaboration between users on Reddit Place, a social experiment where users drew together on a virtual canvas. We propose to use personalization methods to model user actions, capture the latent structure of this emergent collaborative effort and provide an interpretable representation of its social structure.

Keywords: personalized ranking, collaborative filtering, recommender systems, temporal dynamics, social dynamics, context-aware methods

Résumé

Les méthodes de classement personnalisées sont utilisées par de nombreux systèmes de recommandation. Ces systèmes identifient les intérêts des utilisateurs dans le but de leur présenter des éléments susceptibles de les intéresser (vidéos, musiques, produits, etc.) et de les aider à découvrir de nouveaux contenus. Si ces approches ont fait leurs preuves dans les régimes stationnaires, la transition vers un Web interactif et social nécessite de plus en plus de tenir compte de facteurs dynamiques. En effet, les approches existantes, basées sur des représentations vectorielles distribuées, sont inadaptées à des contextes en évolution rapide ; leur représentation statique des utilisateurs les empêche de s'adapter aux changements contextuels. Cette thèse se concentre sur l'introduction de nouvelles méthodes prédictives qui impliquent des variables contextuelles, exploitent des signaux riches (sociaux, temporels) et permettent de maintenir la cohérence d'un modèle dans le temps.

Tout d'abord, nous proposons un système de recommandation dans un contexte où les éléments à recommander ont une disponibilité dynamique. Nous étudions ce scénario sur une plateforme de streaming où les chaînes ne sont disponibles qu'aux moments où leurs créateurs diffusent du contenu. Nous introduisons un modèle auto-attentif qui établit une dépendance entre les éléments disponibles, afin de contextualiser les prédictions. Nous montrons également que la consommation récurrente de contenu joue un rôle important dans la prise de décision des utilisateurs.

Deuxièmement, nous proposons une méthode qui tend à maintenir un espace latent cohérent lors d'un apprentissage sur plusieurs tranches temporelles d'un jeu de données. Pour ce faire, nous pénalisons les différences inutiles entre les solutions successives du modèle. Nous appliquons cette méthodologie à un jeu de données relatif à la couverture de sujets d'actualité par différents médias. En particulier, notre modèle met en lumière les changements importants induits dans la ligne éditoriale de ces médias par les fusions et acquisitions des chaînes d'actualité.

Troisièmement, nous proposons de modéliser l'intérêt réciproque sur une plateforme en ligne où les utilisateurs troquent des biens. Nous constatons que les liens sociaux entre les membres ont une forte influence sur les échanges. Nous tenons également compte du moment auquel ces échanges s'effectuent. Nous proposons un modèle permettant de capturer ces dynamiques sociales et temporelles.

Enfin, nous étudions la collaboration des utilisateurs dans un environnement virtuel. Nous proposons d'utiliser des méthodes de personnalisation pour modéliser la collaboration entre les utilisateurs de Reddit. Notre méthode capture la structure latente de cet effort collaboratif émergent et fournit une représentation de sa structure sociale interprétable.

Keywords: classement personalisé, filtrage collaboratif, systèmes de recommandation, dynamiques temporelles, dynamiques sociales, modélisation du contexte

Contents

List of Figures

List of Tables

1 Introduction

Since its inception, the Web has evolved from a small set of static pages to a massive network of interconnected services that reflect most facets of human activities. As an increasing amount of content is available online, the rate at which humans can possibly process information is still limited. Therefore, methods to access, search and browse through large databases have been extensively researched and engineered through the years. Search engines, in particular, have become the primary mean of accessing the Web by letting users explicitly describe what they are looking for using keywords. However, there exist many scenarios in which query systems are unsuited for retrieving content. For example, identifying a book or a movie that a user might like is difficult to describe in words. Similarly, it is difficult to formulate queries for users that expect the system to provide some level of unexpectedness, typically because they are unaware of most items in a large database. Finally, there are many scenarios in which we cannot afford to ask users for feedback (e.g. there is little incentive for users to rate articles on a news feed) and therefore, must infer user interests from their natural interactions with the platform. Personalization methods represent a family of methods designed to solve these problems, by providing predictions with a *subjective* notion of correctness, i.e. predictions that are tailored from user history, user characteristics, or contextual factors.

Despite their pervasive use on the Web, personalization methods have notorious shortcomings. For example, they fail in “cold-start” settings, in which new users are introduced in the system, and have little to no historical data for the model to learn from. They also show their limitations in dynamic environments, for example when user decision processes evolve with time or when the items they interact with are not static. The high interactivity of the modern Web made this problem more apparent because of the fast pace at which content is created. Social media, for example, are built around user-generated content: new posts, images, or videos are constantly added to the platforms, which breaks the assumption of a static set of items made by traditional personalization methods. Moreover, items might not always be available at all times, such as channels on a video-streaming platform, that are only active for a few hours per day or per week. Items are also increasingly represented as dynamic objects that evolve over time, e.g. forum posts on a Q&A website that evolve with the quantity and the quality of user answers.

In this work, we argue that the transition to an interactive and social Web, and the advent of user-generated content, require modeling an increasing range of dynamic contextual factors since static embedding representations are unsuited to dynamic settings. To better position our work in the literature, we provide, in this section, an intuitive explanation of what the term dynamic refers to in our work. Specifically, we distinguish three cases of dynamic personalization.

1. **Items with dynamic characteristics:** We describe items as dynamic if their content, their characteristics or their availability evolves over time. Dynamic characteristics of items, such as their price, might affect user decisions if they are sensitive to these fluctuations [131]. Similarly, items with dynamic availability might not always be available for users to consume, for example on any real-time media (see Chapter 3). The content itself might also be dynamic and evolve over time. For example, under this definition, recommending a Wikipedia article to an editor is a dynamic task, since the content of the page is frequently edited.

2. **Items in evolving contexts:** Interactions occur in dynamic contexts if user decisions are modified by social or temporal factors. Social factors measure the influence of other users on one's decisions. In this work, we only consider direct relationships between pairs of users, e.g. friendship relationships predicting user interests [71], but our definition also applies to herding phenomena [66], social influence [89] or group dynamics [18]. Temporal factors generally take the form of decay of interest after the creation of an item [20]. For example, social media posts are frequently approximated by epidemic models that predict the propagation of a particular behavior which spikes during contagion (social factor) and decays over time (temporal factor).

3. **Drifting preferences:** We consider user preferences as dynamic if they drift over time. Traditionally, temporal dynamics are incorporated into preference models to capture evolving trends in item consumption [59], and are categorized into short- and long-term dynamics [45]. Evolving decision schemes might also affect content creation, for example on social media [54] and in news media (see Chapter 4).

In this thesis, we propose personalization models suited for dynamic settings. We build upon traditional methods that capture item semantics through static representations (*what* is the item?) and incorporate dynamic factors, such as temporal signals (*when* is the item created?) and social signals (*who* interacted with it?). We also propose novel model architectures that account for dynamically available content, i.e. items that aren't always available for users to consume. We finally introduce models that maintain temporally consistent embedding spaces over multiple retraining, which enables the analysis of long-term trends in the evolution of user decisions.

1.1 Potential Benefits and Applications

The advent of user-generated content increases the volume, frequency, and diversity of available content online. Web services and platforms exploiting this highly dynamic type of content are the main domain of application of our work.

First, our work finds applications in any two-sided platform that matches content creators with audiences, since the relevance and popularity of content heavily depend on dynamic factors. These platforms exist for various types of content, such as videos, podcasts, artworks, or newsletters. Methods in Chapter 3 are particularly suited for empowering real-time media, such as live-streaming platforms. A particular case of dynamic matching is discussed in Chapter 5, where social media users exchange second-hand goods on a social platform.

Second, our work finds applications in collaborative systems [82] where a multitude of users contributes locally to a global effort. For example, on Wikipedia, recommender systems are used to recommend pages that require a contribution from editors. The content of a Wikipedia page is a good example of dynamic personalization since it evolves over time, is often in cold start settings, and is affected by social factors (e.g. conflicts between editors). Similarly, recommending questions on a Q&A website (e.g. Stack Overflow) is often used as an incentive for domain experts to participate in the discussion.

Finally, our work also finds applications in the news domain. For example, we show in Chapter 4, a method that captures the decision process of news channels when filtering news events. Our method helps monitoring and mitigating a specific bias (e.g. news diversity). We believe that such a method could be applied to any news aggregation system to avoid misrepresenting the importance of news topics.

1.2 Contributions and Outline

Our main contribution lies in the design of personalized ranking models suited to dynamic settings, in order to make more accurate recommendations. Additionally, we aim to build bridges between recommender systems and social dynamics studies. On one hand, we borrow well-documented social phenomena from the literature, that we leverage in the development of personalization methods. On the other hand, we study those phenomena in novel scenarios and applications and investigate the use of personalization methods for analyzing user decisions in complex social systems. Finally, our work aims to study novel applications for personalization methods in dynamic contexts. Since the lack of datasets is frequently mentioned as a barrier to the development of novel recommendation methods, we commit to share multiple datasets with influential social and temporal factors.

Considering the above, we summarize our technical contribution into three high-level research questions.

- **RQ1:** *How to adapt model architectures when items are dynamic?* We seek methods to recommend items that evolve in their structure, content, or availability over time.
- **RQ2:** *How to exploit social, temporal signals to improve recommendations?* We investigate approaches to incorporate signals that go beyond pure item semantics into predictions. Social signals capture the social activity around an item, while temporal signals capture its temporal relevance.
- **RQ3:** *How to leverage personalization methods to analyze the evolution of social systems?* We exploit personalization methods to capture long-term trends in preferences schemes and user decisions.

The rest of this thesis if organized as follows. In Chapter 2, we introduce important concepts, methods and notations that are relevant to all other chapters. We also introduce more specific background work in chapters 3 to 6.

In Chapter 3, we propose a new model that computes a personalized ranking of a dynamically evolving set of available items. Additionally, this model accounts for repeat consumption of the same item by modeling time intervals between occurrences. We validate our approach on a large dataset collected from a live-streaming platform and show our method to outperform various strong baselines in ranking the currently available content. This chapter relates to the ranking of dynamically available items (RQ1) and to the modeling of temporal factors (RQ2).

In Chapter 4, we leverage personalized ranking methods to analyze content production. We characterize news channels by their editorial decisions, i.e. the world events they decide to cover. We introduce a dynamic embedding model that encourages temporal consistency and provided an interpretable representation. We demonstrate the monitoring potential of our approach by shedding light on important changes in programming induced by mergers and acquisitions, policy changes, or network-wide content diffusion. We also propose methods to re-rank news selection to reduce the... This chapter relates to the analysis of a content-production system (RQ3).

In Chapter 5, we study recommendations for exchange platforms, where users exchange second-hand goods. Since those goods represent scarce resources and since their availability on the platform is limited in time, we propose a ranking method that estimates cross-preferences between trade partners and recommend potential exchanges. Bartering requires us to understand not just users' preferences, but also the social dynamics of who trades with whom, and the temporal dynamics of when trades occur. Regarding the social aspect, we incorporate a directed social bias, that favors trades with peers that already traded together. Regarding temporal dynamics, we devise a method that captures the recent level of user activity and decreases the relevance of users that haven't traded in a long period. We evaluate our approach on trades covering books, video games, and beers, where we obtain promising empirical performance compared to existing techniques. (RQ2)

In Chapter 6, we use personalization methods to model emergent collaboration in a virtual sandbox. In particular, we perform a study on a dataset comprising more than 16M user actions, recorded on the online collaborative sandbox Reddit *r/place*. Participants had access to a drawing canvas where they could change the color of one pixel, every 5 minutes. We investigate models to predict future user actions and infer user relationships from their interactions in the environment. This chapter relates to the modelling of dynamic items (RQ1), influenced by social signals (RQ2).

1.2.1 Selected Publications

This thesis is based on the following research papers.

- **Jérémie Rappaz**, Maria-Luiza Vladarean, Julian McAuley, and Michele Catasta. "Bartering books to beers: a recommender system for exchange platforms". The International Conference on Web Search and Data Mining (WSDM), 2017.
- **Jérémie Rappaz***, Dylan Bourgeois*, and Karl Aberer. "Selection bias in news coverage: learning it, fighting it". The Web Conference (WWW), 2018.
- **Jérémie Rappaz**, Michele Catasta, Robert West, Karl Aberer. "Latent structure in collaboration: the case of Reddit R/place". The International AAAI Conference on Web and Social Media (ICWSM), 2018.
- **Jérémie Rappaz***, Dylan Bourgeois*, and Karl Aberer. "A dynamic embedding model of the media landscape". The Web Conference (WWW), 2019.
- **Jérémie Rappaz**, Julian McAuley, and Karl Aberer. "Recommendation on Live-Streaming Platforms: Dynamic Availability and Repeat Consumption" The ACM Conference on Recommender System (RecSys), 2021.

Authors with an asterisk contributed equally to the paper. The ideas for these research papers originated from the first author. Most experiments have been conducted by first and second authors. Other authors had advisory roles and helped with writing and proofreading.

2 Background

The idea of making personalized and automated recommendations traces back to the 1970s. For example, an early line of research used questionnaires to collect user characteristics, in order to automatically recommend books they might be interested in [102]. It if only from the 1990s that researchers explored the idea of learning user profiles from historical data [6], mostly because of the publication of large industrial datasets. Since then, a wide variety of approaches has been explored and most of these methods are all based on the same key principle: they exploit similarities between users, in order to filter relevant content and make personalized recommendations.

Recommender systems make the underlying assumption that, if multiple users made similar decisions in the past, they are likely to make similar decisions in the future. For example, two users having expressed strong interest for a specific type of action movies are likely to have tastes in common. One could then exploit this similarity in order to make predictions: if one of the two users has seen a movie and has a very positive feeling about it, this movie probably represents a good recommendation for the other user. This principle is often referred to as *collaborative filtering*, since large dataset are filtered, for a specific user, using knowledge from the entire user population. The role of personalization methods is to apply this principle at scale by learning from large historical dataset.

Early lines of research focused on heuristic-based approaches. Despite their simplicity, these method are still relevant today and provided surprisingly competitive results, especially in sparse data regimes. Since this work focus exclusively on parametrized approaches, we omit the formal definition of these techniques and focus on machine learning approaches. In this chapter, we introduce important personalization concepts and notations. First, we describe different family of supervised models, then we discuss various personalization tasks and loss functions.

2.1 Models

We first describe relevant machine-learning models without yet specifying the nature of their predictions.

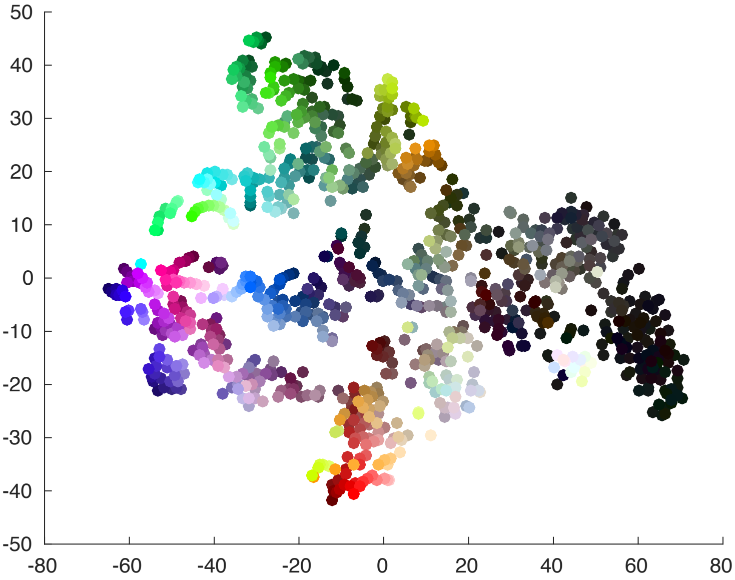


Figure 2.1 – Example visualization of items in embedding space (t-SNE projection): items are colors liked by users on https://www.colourlovers.com/. Even if the method does not model content (i.e. color values), similar colors are grouped together by user interactions: the embedding space structure is shaped by user preferences. Experiment performed on 3M user likes.

Matrix Factorization Methods: Matrix Factorization (MF) [62] is a supervised machine learning method that has been largely popularized by the Netflix Prize [14], a challenge proposed by the movie rental company to improve their recommendation algorithm. Earlier heuristic-based approaches were using similarity functions to either compare items (item-based approaches) or users (user-based approaches). MF-based models have the advantage to *learn* a representation instead of using similarity metrics over raw inputs, but also they *jointly* represent users and items by projecting them in a common latent space. In particular, each user u and each item i is represented as an embedding vector of k dimensions $p_u, q_i \in \mathbb{R}^k$. For any user-item pair, we generate a predicted score $\hat{x}_{u,i}$ using a dot product operation,

$$\hat{x}_{u,i} = \mathbf{p}_u \cdot \mathbf{q}_i^T = \sum_k p_{u,k} q_{i,k}. \tag{2.1}$$

Since the combination function is user-defined, the knowledge about users and items is fully delegated to the latent space representation. User and item embedding vectors can be visualized (see Fig. 2.1). One popular framework is proposed by t-SNE [125], for visualization in a 2-dimensional space.

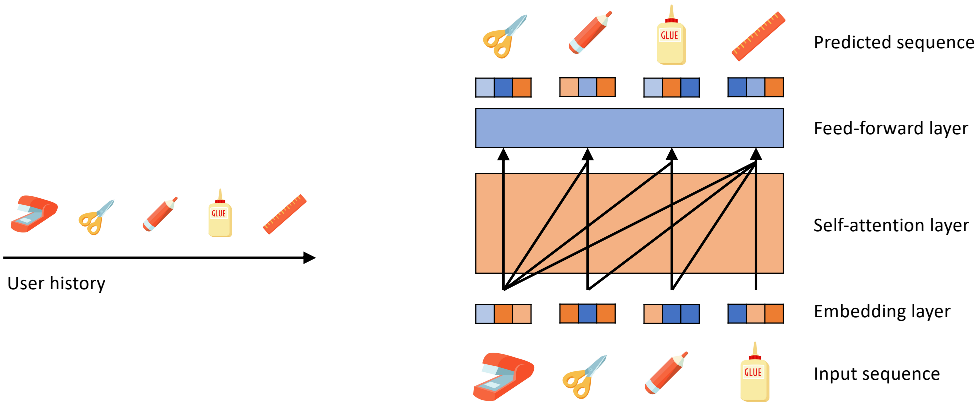


Figure 2.2 – Illustration of sequential recommender systems. The full sequence is fed into the network as a single forward-backward pass: each step i predicts the next step $i+1$ in the sequence. The self-attentive layer learns from all previous entries in the sequence, while future entries of the sequence are masked.

Neural Approaches As in many other fields, deep learning based approaches have become a popular modelling paradigm. One of the first successful attempts [48] replaces the dot product operation by a neural architecture to model arbitrary – potentially non-linear – interaction functions between users and items.

Recently, there has been a growing interest in sequential recommender systems. These models are inspired by the field of natural language processing; *sequences* are treated like *sentences*. Originally designed with recurrent neural networks [49], attention-based methods [55] and BERT-like approaches [120] have become popular because of their attention mechanisms that help the model focusing on specific parts of the sequence. Sequential models are interesting because they can learn complex inter-relationships between the elements of a sequence and can learn (potentially long-term) sequential dynamics. Sequence-based models have also efficient training procedures since they can learn from a full sequence in a single forward-backward pass (see Fig. 2.2). Most approaches rely on scaled dot-product attention [128] (see Eqn. 2.3), and use the same objects as queries, keys and values. In particular, each item in the sequence is embed using an embedding matrix E (we denote embedded items as $\hat{E}$). This representation is transformed through linear projections using three matrices W^Q, W^K and W^V.

$$Q = \hat{E}W^Q \qquad K = \hat{E}W^K \qquad V = \hat{E}W^V \tag{2.2}$$

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d}})V. \tag{2.3}$$

The product between Q and K represents an attention score between any pair of elements in the sequence. These scores are mapped to an output space using the values in V (each row represents an item). The term $\sqrt{d}$ prevents large values in the inner product QK^T.

Other forms of architecture have been explored, such as graph-based approaches [141, 138, 135]. The flexibility of deep learning architecture also provide natural was to model user interaction with images [46, 47] or texts [122, 51].

2.2 Feedbacks and Loss Functions

There exists different types of user feedbacks, that require different types of optimization schemes. We treat these two concepts jointly in this section.

Explicit Feedbacks: early lines of work were focused on learning from *explicit* feedbacks generated from a system asking users to provide their opinion about an item (e.g. on a five-stars scale). This family of methods is optimizing a reconstruction error. The model minimizes the difference between the observed score of a user towards an item r_{ui} and a model prediction $\hat{x}_{u,i}$.

$$\min_{\theta_\star} \sum_{u,i} (r_{ui} - \hat{x}_{u,i})^2 + \lambda ||\theta||^2 \tag{2.4}$$

A regularization term $\lambda||\theta||^2$ is generally added to discourage complex solution and, therefore, prevent overfitting. The original formulation use the ℓ^2-norm over θ, the full set of model parameters.

Implicit Feedbacks: ratings are a scarce resource. This observation motivates the modeling of other signal types, such as the number of times an item has been clicked on or the percentage of a video watched by a user. These types of signal are generally referred to as *implicit* feedbacks and could be seen as the data that a user generate while naturally interacting with a platform. This source of signal is, of course, much noisier than its explicit counterpart. Various ways to mitigate this uncertainty has been proposed, such as the addition of a confidence parameter in the loss function [52].

One-class Feedbacks: in this thesis, we focus on a setting called one-class collaborative filtering [86] where we learn from a specific type of implicit feedback made of binary observations. A positive interaction represents a user action, e.g. bookmarking an item or clicking on a link. Negative interactions represent user inactions that could either be interpreted as users having no interest in an item or simply being unaware of its existence. This uncertainty about negative signals, and the typical extreme sparsity of positive interactions, require adapted optimization schemes.

Bayesian Personalized Ranking: Until now, we only considered methods that optimize a reconstruction error by using a regression loss. In this setting, models are trained to best approximate a numerical score representing the interest of users towards specific items. This training procedure has proven to work well in practice but there exists a discrepancy between the optimization of a regression loss and the recommendation task: most real-world applications focus on *ranking* items, in order to provide users with a relevant item list. This observation motivates the direct optimization of a ranking criterion, as proposed by the Bayesian Personalized Ranking (BPR) [98] framework.

The overall objective of BPR is to provide users with a personalized ranking of all items in a dataset. They learn from one-class user feedbacks where only positive interactions are observed. Let $i \in \mathcal{I}_u^+$ be the set of positive items consumed by user u and $j \in \mathcal{I}_u^-$ be the set of remaining items with which u never interacted. Given an arbitrary model (we denote by Θ its set of parameters) and the preference scheme of user u denoted as $>_u$, we seek to maximize $P(i >_u j|\Theta)$, the likelihood of user u to prefer item i over item j. In an ideal scenario, the model should output a probability of 1 for $\Pr(i >_u j|\Theta)$ and a probability of 0 for $\Pr(j >_u i|\Theta)$. Defining $\hat{x}_{u,i}$ as the predicted score for user u and item i, this can be modeled as $H(\hat{x}_{uij})$ where $\hat{x}_{uij} = \hat{x}_{ui} - \hat{x}_{uj}$ and $H(\cdot)$ is the Heaviside step function. This function is not differentiable, and needs to be adapted to a gradient descent learning procedure. BPR approximates $H(\cdot)$ by using the logistic sigmoid function σ,

$$\Pr(i >_u j|\Theta) := \sigma(\hat{x}_{uij}(\Theta)) = \sigma(\hat{x}_{ui} - \hat{x}_{uj}) \tag{2.5}$$

Then, we could derive the log-likelihood maximum posterior estimator, BPR-OPT, that maximizes this probability.

$$\text{BPR-OPT} := \sum_{(u,i,j)\in D_s} \ln \sigma(\hat{x}_{uij}) - \lambda_\Theta ||\Theta||^2 \tag{2.6}$$

Here, Ds is defined as a training dataset set of size $|U| \times |I| \times |I|$, made of (user, positive, negative) triplets. The method maximizes the difference between the predicted score of a positive and a negative example that are sampled during training. Since BPR-OPT directly learn to rank all positive interactions higher than their negative counterpart, there is a direct analogy between this formulation and the Area Under the Curve (AUC) that is generally used as a ranking quality metric.

3 Dynamic Availability and Repeat Consumption

Live-streaming platforms broadcast user-generated video in real-time. Recommendation on these platforms shares similarities with traditional settings, such as a large volume of heterogeneous content and highly skewed interaction distributions. However, several challenges must be overcome to adapt recommendation algorithms to live-streaming platforms: first, content *availability* is dynamic which restricts users to choose from only a subset of items at any given time; during training and inference we must carefully handle this factor in order to properly account for such signals, where 'non-interactions' reflect availability as much as implicit preference. Streamers are also fundamentally different from 'items' in traditional settings: repeat consumption of specific channels plays a significant role, though the content itself is fundamentally ephemeral.

In this chapter, we study recommendation in this setting of a dynamically evolving set of available items. We propose *LiveRec*, a self-attentive model that personalizes item ranking based on both historical interactions and current availability. We also show that carefully modelling repeat consumption plays a significant role in model performance. To validate our approach, and to inspire further research on this setting, we release a dataset containing 475M user interactions on *Twitch* over a 43-day period. We evaluate our approach on a recommendation task and show our method to outperform various strong baselines in ranking the currently available content.

3.1 Introduction

Video streaming platforms, such as *Twitch* or *Youtube Live*, are increasingly becoming a major part of people's daily lives. As of February 2020, Twitch reported 3 million broadcasters monthly and 15 million daily active users.[I] The increasing volume of concurrent broadcasts, the growing audience, as well as the long-tail of niche content, suggest the need for systems designed specifically for such platforms.

On live-streaming platforms, content creators broadcast video in real-time on their

[I] https://en.wikipedia.org/wiki/Twitch_(service) respective channels. The broadcast of real-time content[II] implies that videos can only be consumed at specific points in time. This dynamically evolving availability of streams presents challenges for traditional methods. Implicit feedback methods, trained to capture preference signals from positive interactions, make an underlying assumption that positive observations outrank those items the user never interacted with; when availability is dynamic this assumption no longer holds. Therefore, the explicit modelling of availability signals is required to distinguish between 'non-interaction' resulting from implicit preferences and from unavailable items.

A live-streaming model should also account for repeat consumption that represents a large portion of observed interactions, since users repeatedly consume content produced by the same streamers. This contrasts with typical recommendation domains (e.g. movies, e-commerce) that generally assume one-off user-item interactions. At the same time, it contrasts with existing lines of work on repeat consumption [8], since the content of channels is dynamic and differs at each new interaction.

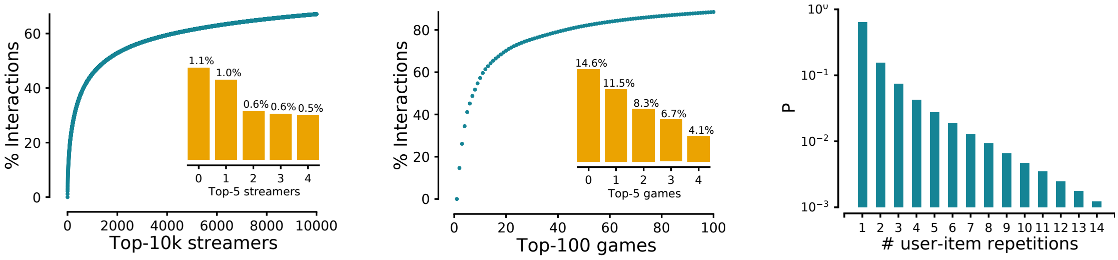
In this chapter, we introduce the task of live-streaming recommendation with temporally evolving availability. We first provide preliminary experiments that demonstrate why existing methods are unsuited to this setting and, in particular, why naive sampling strategies are insufficient to capture user preferences. In light of these experiments, we introduce a self-attentive model, *LiveRec*, that learns to recommend live-streaming channels under availability constraints. Our model first selects candidates from the pool of available items at a specific point in time. These candidates are fed into a self-attention block that parametrizes relationships among available items. Historical interactions are modelled using a sequence encoder that is both used to select candidates and compute the final predictions, thus making the approach fully end-to-end. In order to validate our approach, we introduce a large dataset of interactions from Twitch containing the consumption of logged-in users over a 43 day period. We identify key characteristics that differentiate our data from traditional settings, such as the high prevalence of recurring consumption, that we incorporate in our approach. To the best of our knowledge, this is the first publicly available dataset of content consumption detailing individual users' watching habits. We show that strong sequential baselines don't apply straightforwardly to this setting, and that our adaptations are necessary to achieve substantially better performance. We conclude with an in-depth analysis of the results and a discussion of the dynamics of live-streaming platforms.

3.2 Related Work

In this section, we first review relevant lines of work from the literature on streaming platforms. Then, we describe relevant lines of work in the field of recommendation with

[II] On some platforms, the content is also available on-demand, but we only consider the strict live-streaming setting in this work.

Figure 3.1 Figure 3.2 Figure 3.3



Left: CDF percentage of the top-1k streamers (items) in the dataset. The top-5 most popular streamers account for around 4% of the dataset. **Center:** CDF percentage of the top-100 games in the dataset. **Right:** Count of individual user-item pairs in the dataset.

an emphasis on sequential methods.

Streaming Platforms have been studied from a social dynamics perspective. The work from Hamilton et al. [44] investigates user motivations in joining live streaming channels. They conclude that, similar to on-demand video services, users are interested in a particular type of content, but also engage with the interactive characteristics of the service. Hilvert-Bruce et al. [50] report that compared to mass media, motivations of viewers on Twitch have a stronger social and community basis. They also suggest that viewers preferring small channels are more motivated by social engagement than users preferring large channels. Kaytoue et al. [56] characterize audience dynamics on the Twitch platform. Pires et al. [90] highlight the difficulty of identifying popular segments early on and show that there is no trivial solution to this problem. Nascimento et al. [81] identify several characteristic behaviors, such as the large audience drop at the end of a stream, investigate spectators assiduity and characterize the volume of comments in stream chats.

Temporal dynamics play a key role in the analysis of content consumption and have been studied in different scenarios. Early attempts focus on the modelling of long-term preference drifts [59, 140]. More recent approaches model evolving trends within the community of users in the context of fashion recommendation [46]. Another line of work focus on learning and inferring from temporally ordered data streams [16]. Several lines of work study the impact absolute position in the sequence and relative time intervals on performance [67, 143]. The CTA model [136] captures both temporal and contextual information by incorporating users' browsing activity. Wang et al. [133] incorporate various temporal patterns of repeat consumption using Hawkes processes. Wan et al. [132] model complementarity, compatibility and loyalty towards products in the grocery shopping domain. They propose AdaLoyal, a learning algorithm that explicitly account for users' must-buy purchases in addition to their overall preferences and needs. Anderson et al. [8]

	Bench.	**Full**
#Users	100k	15.5M
#Streamers	162.6k	465k
#Interactions	3M	474.7M
Watch time [h]	800k	124M
Density	9.2e-5	1.9e-5

Table 3.1 – Datasets statistics. Watch time is estimated by considering periods of 5 minutes (half the duration of a round).

analyze the repeat consumption patterns on different social platforms. They propose a hybrid model that predicts user choice based on a combination of recency and quality. In this study, we consider repeat consumption of the same channel broadcasting new content, which differs from what past work has considered.

Recommendation systems model relationships between users and items [61, 100, 52] for explicit [62] and implicit feedback settings [98]. Sequential approaches infer user preferences from sequences of interactions. Recently, neural approaches have become popular for they high expressivity. Various research lines have attempted linear [48], recurrent [49], convolutional and graph approaches [141, 138, 135]. More recently, self-attention mechanisms, inspired by the field of natural language processing [128], have been investigated in the context of recommendation [55, 120, 139, 17, 134]. They generally do not model users explicitly and only learn from sequences of items. In Section 3.5.2, we discuss ranking refinement techniques that have been explored in the context of entity linking [137, 22].

3.3 Data

In this section, we describe our data collection on Twitch in July 2019 over a 43-day period and give general statistics about the resulting dataset. In order to discover available channels, we queried the public Twitch API in rounds to list all available streams. The number of live streams ranged from around 20k to 75k for a single round during data collection. In each round, we also queried each available stream to get a list of connected users and the currently played game. In order to have sufficient time to query each stream in a single round, we set a 10 minute interval between each round. The final dataset was collected over 6,148 rounds. Our dataset will be released in two formats: the *full* version, that contains all collected data and is more suited to data analysis tasks; and a *benchmark* version, that contains all interactions from 100k uniformly sampled users and that is used to compare the performance of different methods. All performance evaluations reported in this work use the *benchmark* version. Statistics before and after pre-processing can

Figure 3.4

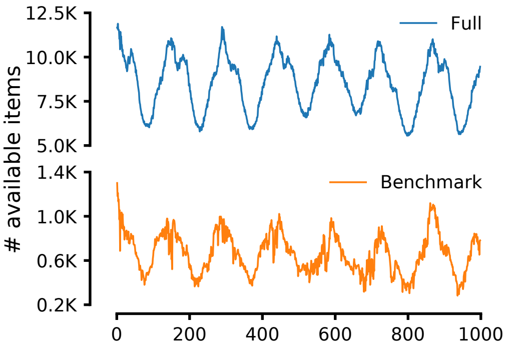
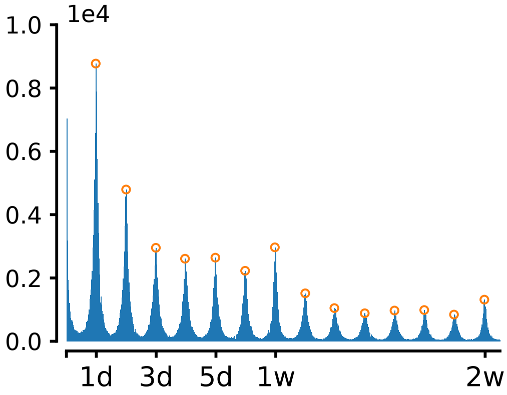


Figure 3.5



Left: Number of available items for the first 1000 time steps of 10 minutes. **Right:** Time interval distribution (normalized) of repeated user-item interactions.

be found in Table 3.1. Both anonymized versions shall be made available at publication time. Both datasets exhibit skewed interaction distributions over items. For example, in the *full* version of the dataset, the most popular streamer drives more than 1% of the total interactions (see Fig. 3.1) and the most popular game drives 14.6% of the total interactions (see Fig. 3.2). Datasets contain absolute timestamps but we only consider relative time interval in this work, thus mitigating time zones related side effects.

Repeat consumption is a common scenario in our dataset (see Fig. 3.3). Since Twitch is a social platform, content providers aim to grow and retain their respective audiences. We measure time intervals between any two interactions for the same user with the same streamer. We observe both daily and weekly dynamics, as seen in Fig. 3.5. We also observe short-term repeated interactions to be prevalent in our dataset.

3.4 What is different in Live-Streaming Recommendation?

Live-streaming differs from traditional scenarios in terms of the semantics of both positive and negative interactions. In this setting, negative examples can reflect either latent preferences, or simply the unavailability of a particular item (streamer) at interaction time. Additionally, several items appearing in a user history could become available simultaneously, which requires one to rank positive interactions among themselves. This differs from traditional settings where positive interactions included in the training set are discarded from the testing set. In this section, we present preliminary experiments that demonstrate the limitations of existing methods in this particular setting. Then, we propose an availability-aware sampling strategy that accounts for these observations.

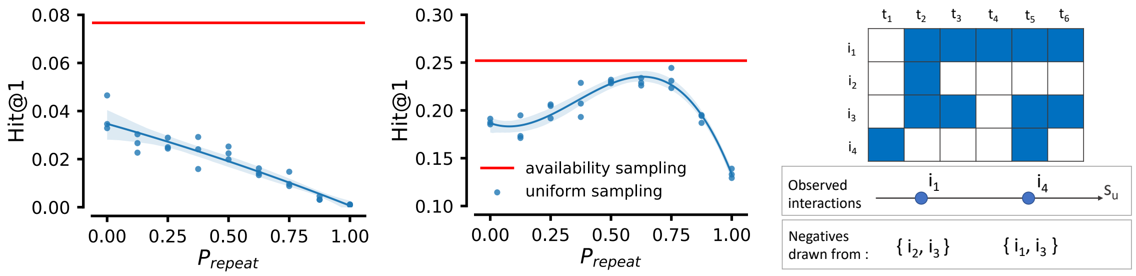


Figure 3.6 Figure 3.7 Figure 3.8

Left: *Hit@1* performance for novel interactions. **Center:** *Hit@1* performance for repeat interactions. Availability sampling performance is shown with a red line. **Right:** Illustration of the availability matrix from which availability sampling is performed. Negative items are sampled from the pool of available items for each positive interaction of the sequence s_u

3.4.1 Defining Items

On a streaming platform, items could have different definitions: one could define items as *streams*, that are unique segments during which a streamer broadcasts uninterruptedly. The main problem of this case is the fact that streams only happen at a single point in time which lets the model learn in extremely sparse regimes. Alternatively, one could define items as *streamers*, that typically broadcast content multiple times. The main problem of this case is the fact that the model has to learn from multiple occurrences of the same user-item interaction, even if a streamer constantly broadcasts new content and evolves over time. In this work, we consider items representing streamers and discuss on how to account for repeat consumption during training and evaluation.

3.4.2 Preliminary Experiment: Repeat Consumption

When splitting our dataset in the temporal dimension,[III] around 65% of user-item interactions observed in the testing period are also present in the training set. In such a setting, a model shall not only learn to accurately predict novel interactions but also to balance between novel and repeated interactions. Without yet considering extra features for distinguishing among repeated interactions (e.g. time, content), a simple model could balance interactions by their frequency, by favoring interactions occurring more often during training. A natural way to incorporate repetition information into a model is to develop a sampling strategy accounting for item frequencies.

We demonstrate the challenge of balancing between novel and repeat interactions through

[III] Considering the same setting as the experiments in Section 3.6, the last 250 rounds of 10 minutes (4%) of the dataset are withheld for testing.

a preliminary experiment: we train a simple matrix factorization model with 20 latent dimensions using a ranking criterion [99] in a non-sequential setting. During training, for a user u and a positive item i, we sample negative examples from the pool of items consumed by u in the training set with a probability P_{repeat} and from a uniform distribution over all items with probability $(1 - P_{repeat})$. In other terms, the model learns to better rank positive examples among themselves by learning to rank a pair of positive items, e.g. two items appearing in the training sequence, with probability P_{repeat}. During testing, we evaluate the model on its capacity to rank interactions with new streamers (Fig. 3.6), as well as its capacity to rank streamers appearing in the training sequence of the considered user (Fig. 3.7). We observe that increasing the value of P_{repeat} leads to an increase in performance for repeat interactions and to a decrease in performance for novel interactions (as shown by blue lines in Fig. 3.6 and Fig. 3.7). The best overall performance (0.157) is obtained at $P_{repeat} = 0.5$. This score represents a relative improvement of **16.1%** over a uniform sampling strategy when $P_{repeat} = 0$ (0.135). This observation shows the importance of accounting for repeat consumption when positive instances require to be ranked among themselves and not only against negative samples. We also observe that our sampling strategy deteriorates predictions for novel items as P_{repeat} increases, which demonstrates the difficulty of balancing between novelty and repetition.

3.4.3 Preliminary Experiment: Availability

Availability signals are important to capture the meaning of 'non-interactions'; most recommendation systems relying on implicit feedback assume that the choice of item i over item j only reflects an implicit preference for i, since they also assume all items being available at all times. In a live-streaming setting, non-interaction with item j can reflect a preference for item i as much as item j simply being unavailable at interaction time.

In this experiment, we demonstrate the importance of accounting for availability during training. We first precompute an availability matrix of size $n \times t_{max}$ where the columns represent the set of available items at any point in time. Here, n is the total number of items and t_{max} is the total number of time steps of 10 minutes in our dataset (see in Fig. 3.8). Instead of sampling negative examples from all items, we sample a negative example j from the pool of available items at interaction time. As such, negative samples are more likely to result from user decisions instead of being the result of streamers being offline. We evaluate this strategy on temporally disjoint training and testing sets, in order to avoid the model learning the availability matrix directly. We observed this strategy to provide **21.3%** (0.190) of relative improvement over the best performing sampling strategy presented in the last section (as shown by red lines in Fig. 3.6 and Fig. 3.7). We select this sampling strategy in all further experiments.

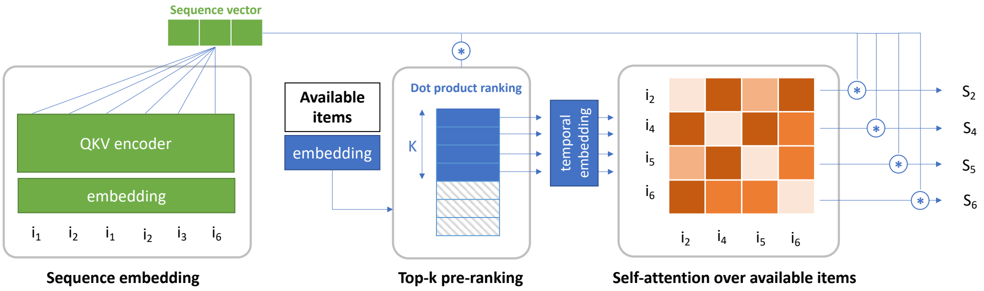


Figure 3.9 – Illustration of the *LiveRec* architecture that encodes an input sequence of user interactions (left) ranks all available items at interaction time (center) and draws a dependency between a top-k selection of available items using a self-attention mechanism (right).

3.5 Methods

Until now, we discussed ways to incorporate repeat consumption and availability information *implicitly*, through different sampling strategies. In this section, we discuss how to account for those signals by specific changes in model architecture.

3.5.1 Sequence Encoder

Let $s = (s_{p_1}, s_{p_2}, \ldots, s_{p_\ell})$ be a fixed-length user sequence, where s_p represents the p-th item entry. Sequences shorter than ℓ are padded with a null token and sequences longer than ℓ are cropped. The first key component of our model is a sequence encoder that converts each step s_p into a hidden representation vector $\mathbf{h_{s_p}}$. We use vector $\mathbf{h_{s_{p-1}}}$, that represents all previous entries in the sequence, for predicting the p-th item entry. We use an existing encoding technique, SASREC [55], even though our framework is not tied to this specific encoding scheme. First, we embed each item in s using an embedding matrix M of size $R^{n \times h}$ where n is the number of items and h is the dimensionality of our latent space. We also encode positions in the sequence with an embedding matrix P of size $R^{\ell \times h}$ that we learn during training. The resulting matrix $\hat{E}$ is computed as the sum of item embeddings and positional embeddings:

$$\hat{E} = \begin{bmatrix} \mathbf{M}_{s1} + \mathbf{P}_1 \\ \mathbf{M}_{s2} + \mathbf{P}_2 \\ \cdots \\ \mathbf{M}_{sl} + \mathbf{P}_2 \end{bmatrix} \tag{3.1}$$

Then, we pass the embeddings $\hat{E}$ into two *query-key-value* self-attention layers in order to learn a relationship between elements of the sequence. Similar to the original implementation [55], we use layer normalization and apply a causality mask that prevents the model from learning from future interactions.

3.5.2 Modelling Availability

We seek a method that learns from the set of available items at interaction time. Considering a user interacting with item i at time t, we learn from a set H_t containing all available items at t. Attention-based methods are typically of quadratic complexity and learning from the set H_t would generally be impractical, due to the large number of concurrently available items on a streaming platform. Instead, we learn from a limited set of candidate items. Candidates are dynamically retrieved by computing a relevance score $s(\cdot)$ for each available item in H_t. For predicting the p-th item entry, this score is obtained from a dot product operation between the sequence embedding and the item embedding vector.

$$s(i, p, t) = \mathbf{h_{p-1}} \cdot \mathbf{M_i}, \quad i \in H_t \tag{3.2}$$

Then, we sort all items in H_t by their score $s(\cdot)$, select the top-k highest scoring items and represent them as a sequence $(r_1, r_2, \ldots, r_k)$. We use matrix M to embed each of the top-k elements into a matrix $M_{\text{av}} \in \mathbb{R}^{k \times h}$. Our objective is to learn an attention function $f : \mathbb{R}^{k\times h} \mapsto \mathbb{R}^{k\times h}$ over M_{av}, in order to draw a dependency between each of the k-selected items. We adopt the widely used *query-key-value* form of self-attention and use the matrix M_{av} as input for queries, keys and values

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d}})V \tag{3.3}$$

$$Q = K = V = M_{\text{av}} = \begin{bmatrix} \mathbf{M}_{r_1} \\ \mathbf{M}_{r_2} \\ \cdots \\ \mathbf{M}_{r_k} \end{bmatrix} \tag{3.4}$$

The factor $\sqrt{d}$ is introduced to avoid overly large values of the inner product QK^T [128]. We use layer normalization, residual connections and dropout similar to SASREC [55]. We experimented with this approach with one and two layers of attention (see Section 3.6). Because the absolute position in the sequence $(r_1, r_2, \ldots, r_k)$ is irrelevant to the task, we do not encode positions and do not use any masking in this attention stage.

3.5.3 Modelling Repeat Consumption

We design an encoding scheme that distinguishes between novel and repeated interactions. In the repeat case, our scheme also encodes recency: for an item $i \in H_t$, we retrieve the time t_j of its last occurrence prior to t. We define $\mathbf{q_i} \in \{0, 1\}^{z+1}$, a one-hot encoded

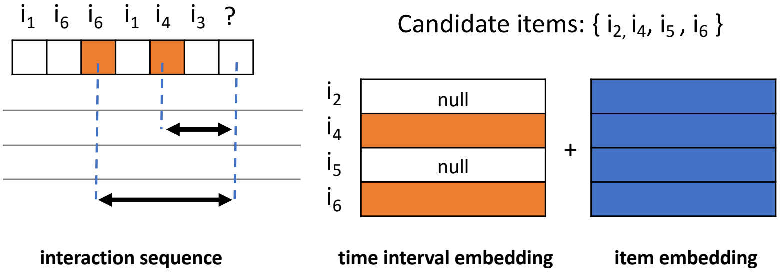


Figure 3.10 – Illustration of the time interval embedding module for candidate items $i_2, i_4, i_5, i_6 \in H_t$. Items i_2 and i_5 have no previous occurrences in the sequence and are considered as novel entries. Time interval embeddings for items i_4 and i_6 are computed from a time difference with their last respective occurrences.

vector that represents a mapping from the time interval $|t - t_j|$ to a bucket index in the range $[1, z]$. Based on our observations in Section 6.3, we divide time intervals into buckets of 24 hours and clip the maximum time interval to 20 days. We keep index zero of $\mathbf{q_i}$ for representing novel item entries with no previous occurrences in the sequence. We embed time intervals represented by $\mathbf{q_i}$ using an embedding matrix $T \in \mathbb{R}^{(z+1)\times h}$. Then, we combine the corresponding time interval embedding vector to the candidate item vector using element-wise addition (see Fig. 3.10). We use this new representation for the selection of candidates and the attention stage,

$$s(i, p, t) = \mathbf{h_{p-1}} \cdot (\mathbf{M_i} + \mathbf{q_i}T), \quad i \in H_t \tag{3.5}$$

$$Q = K = V = M_{av+rep} = \begin{bmatrix} \mathbf{M}_{r_1} + \mathbf{q_{r_1}}T \\ \mathbf{M}_{r_2} + \mathbf{q_{r_2}}T \\ \cdots \\ \mathbf{M}_{r_k} + \mathbf{q_{r_k}}T \end{bmatrix}. \tag{3.6}$$

Time interval embeddings are added prior to the self-attention stage for the model to learn from temporal dependencies among candidates. The self-attention module outputs, for each candidate item, a new representation that encodes a relationship with other candidates. We compute the final score $\hat{x}_{u,i,t}$ for each of the k candidates by computing a dot product between $\mathbf{h_{p-1}}$ and each of the k rows of matrix $O =$ Attention$(M_{\text{av+rep}}, M_{\text{av+rep}}, M_{\text{av+rep}}), O \in \mathbb{R}^{k\times h}$.

3.5.4 Training

Following existing sequential recommendation approaches, we compute predictions using mini-batches of sequences. During training, the model predicts the next item of each sequence step in a single forward pass. We train the model to maximize the difference in score between positive and negative instances through negative sampling. Instead of sampling items uniformly, and according to our observations in Section 3.4, we draw negative examples from a set $H_{i,t}$ for a positive item i at time t,

$$H_{i,t} = \{j \in A_t \wedge j \neq i\}. \tag{3.7}$$

Given a positive i and a negative items j that fulfill the constraint described above, we adopt the following cross entropy loss to train the model:

$$-\sum_{s} \sum_{(i,t)\in s} \log(\sigma(\hat{x}_{u,i,t})) + \sum_{j\in H_{i,t}} \log(1 - \sigma(\hat{x}_{u,j,t})). \tag{3.8}$$

During training, we use vector $\mathbf{h_{p-1}}$ for both selecting candidates and predicting the next entry in the sequence which makes the approach fully *end-to-end*.

3.6 Experiments

3.6.1 Evaluation

In order to avoid learning from future interactions, we split the dataset in three distinct time intervals. We withhold 250 time steps for validation and 250 time steps for testing, both by splitting from the end of the dataset. We evaluate all approaches using the metrics *hit@1*, *hit@10* and *NDCG@10* on the last interaction of each sequence s in the testing period.

Additionally, we break down this score into a *Hit-new* score and a *Hit-rep* score in Table 5.3. By doing so, we evaluate the model on its capacity to repeat an interaction from the input sequence, as well as its capacity to recommend serendipitous content. A testing interaction falls into *rep* if the item appears in the testing input sequence, and in *new* otherwise. For a sequence length of 16, the percentage of repeat consumption is equal to 51%.

Our experiments are conducted with different variants of our model.

LiveRec + rep is a variant that only uses time interval embeddings on top of a sequence encoder. *LiveRec + av* is a variant including self-attention over a top-k selection of items. *LiveRec + rep + av* is our final model, as described in Section 3.5, using both self-attention and time interval embedding. *LiveRec + rep* does not use any candidate selection strategy and ranks all available items H_t instead of a top-k selection.

3.6.2 Baselines

In this section, we compare various baselines with our approach in order to evaluate existing methods in a live-streaming setting (see Section 3.4) and demonstrate the benefits of the proposed architecture.

- **REP** is a simple model that predicts a score equal to the number of appearances of an item in the input sequence. *REP* is a strong predictor of repeated interactions but it is unable to recommend new items.
- **POP** this model gives a score equal to the popularity of an item in the training set. It does not consider the interaction sequence to compute predictions.
- **MF-BPR** is a matrix factorization model trained with a ranking criterion, as in [98]. This model does not account for sequential or temporal dynamics.
- **FPMC** is a sequential recommendation method that models transitions in terms of the last entry in the sequence [99]. The model is trained using a BPR loss [98].
- **SASREC** is a self-attentive recommendation method. Multiple query-key-value attention layers are stacked together to capture relevant information from the interaction sequences. Positional encoding is used to help the model encoding temporal information. We referred to as *SASREC - uniform* a model trained with negative samples drawn uniformly over all items.
- **BERT4REC** is a recent adaptation of the BERT language model. We modified a publicly available implementation[IV] in order to run the model using the same experimental setting. Similar to the original training environment, we define the masking probability ρ and predict masked items only. In order to be fair towards other approaches, we implement a *masked* cross-entropy loss that only backpropagates over available items.

3.6.3 Experimental Setting

We train all models until the score does not improve for 10 epochs on the validation set and store the model at each epoch. Then, we test with the checkpoint having the highest (validation) *hit@1* and evaluate performance on the testing set. We consider ℓ_2-regularization in the range $\ell_2 = \{0.0001, 0.001, 0.01, 0.1, 1.0\}$. All models are implemented in Pytorch and trained using the Adam optimizer with a learning rate of 0.0005. All attention-based approaches are trained with a fixed dimensionality of 128. Other methods are trained with a dimensionality in the range $\{16, 32, 64, 128\}$. Batch size is fixed to 100. For BERT4REC, we consider ρ in the range $\{0.25, 0.5, 0.75\}$ and results are reported with

[IV] https://github.com/jaywonchung/BERT4Rec-VAE-Pytorch

$\rho = 0.25$. We filter out users with fewer than 5 interactions and streamers with fewer than 3 interactions. If the input testing sequence is shorter than the maximum length l, we append interactions from the validation and the training set. *All code and data is publicly available* [V].

Model	H@1	H@1-new	H@1-rep.	H@10	H@10-new	H@10-rep.	NDCG@10
POP	0.0317	0.0237	0.0387	0.1350	0.1006	0.1650	0.0754
REP	0.3698	0.0166	0.6776	0.5347	0.0424	0.9637	0.4630
MF-BPR	0.0363	0.0279	0.0436	0.1537	0.1182	0.1848	0.0879
FPMC	0.0690	0.0372	0.0968	0.2515	0.1662	0.3258	0.1529
SASREC - uniform	0.1994	0.0721	0.3103	0.5827	0.3888	0.7517	0.3733
SASREC	0.3004	**0.1180**	0.4593	0.7221	0.5156	0.9021	0.5014
BERT4REC	0.3517	0.1018	0.5694	0.7089	0.4668	0.9199	0.5237
LiveRec + rep	0.3655	0.0686	0.6241	0.7581	0.4907	0.9912	0.5615
LiveRec + av	0.3357	0.1067	0.5352	0.7363	**0.5222**	0.9229	0.5303
LiveRec + rep + av	**0.4122**	0.0920	**0.6913**	**0.7655**	0.4998	**0.9970**	**0.5893**

Table 3.2 – Results for all considered approaches. The best performing method in each column is boldfaced.

3.6.4 Overall Performance Comparison

In this section, we discuss the results obtained by various architecture in Table 5.3. We first notice that *REP*, which only predicts future interactions by repeating elements from the input sequence, provides a reasonably competitive score. Since repeated interactions account for more than 50% of the testing data, *REP* represents an effective strategy for recommending content without a parametrized method, despite its inability to recommend new content. This result also shows the importance of measuring the two metrics *Hit-new* and *Hit-rep* individually.

LiveRec rep + av provides the best overall score. However, we notice that *SASREC* leads to a higher *Hit@1-new* and that *LiveRec + av* leads to a higher *Hit@10-new*. This result shows that there is still margin for improvement in balancing between novel and repeated interactions. Compared to the sequence encoder alone (*SASREC*), the additional modelling of repeat consumption (*rep*) and availability (*av*) lead to a relative improvement of **21.7%** and **11.8%**, respectively. The combination of the two leads to a relative improvement of **37.2%** over the sequence encoder alone (*SASREC*). The introduction of the time interval embedding (*rep*) encourages the model to favor repeated interactions which hurts novelty (*Hit@1-new*). However, incorporating self-attention (*rep + av*) over available items mitigates this effect. Compared to *REP*, *LiveRec* scores a higher *Hit@1-rep*, which suggests a better capacity to rank multiple repeat consumption options than a simple frequency-based ranking.

[V] https://github.com/JRappaz/liverec

BERT4REC provides higher performances than *SASREC*. This observation in accordance with the original paper [120]. We notice that the gain in performance comes primarily from a better capacity to model repeat consumption. We also emphasize that masking unavailable items in the loss function is critical to obtain this result. In order to compare results with our observations from Section 3.4, we trained *SASREC* with two different sampling strategies: uniform sampling and availability sampling. We observe a significant improvement by sampling negative interactions from the set of currently available items, in accordance with our preliminary results.

3.6.5 Analysis

In light of the results presented in Table 5.3, we seek to further elaborate on the influence of various factors on performance through the following questions.

Question 1: *What is the influence of the number of candidates k on performance?*

The number of candidates k is a critical parameter of our approach since only candidate items are considered in the final ranking. Therefore, this parameter should be sufficiently large to cover a broad spectrum of candidate items, the ranking of which will be refined by the self-attention module. We obtain the best results with $k = 128$ [VI] (see Fig. 3.12). We also experiment with 1 and 2 attention layers. A single layer module performs better for small values of k but fails to scale to a larger number of candidates. We hypothesize this phenomenon to be due to the self-attention module being only *distantly* personalized: it is trained on a selection of candidate items instead of raw interactions, a complex mapping that could be successfully captured with multiple layers of attention.

Question 2: *What is the influence of sequence length on performance?*

Increasing sequence lengths monotonically increases performance for our model (see Fig. 3.13). This gain diminishes, as we increase sequence length, since the average sequence length of the training dataset is equal to 28 and sequences shorter than this maximum length are padded. The results presented in Table 5.3 are consistent over all sequence lengths in Fig. 3.13. However, as we increase sequence length, we also increase the percentage of repeated interactions between users and streamers. For simplicity, we perform the full evaluation, presented in Table 5.3, at around 50% of repeated interactions and leave as future work a more in-depth analysis of the impact of this ratio on performance.

Question 3: *What is the popularity distribution of recommended items with LiveRec?*

Mitigating popularity bias has become a concern in the recommender systems litera-

[VI] In our case, the maximum value that can fit on a single GPU

Figure 3.11

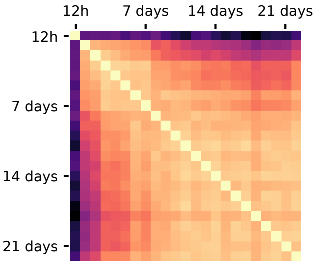


Figure 3.12

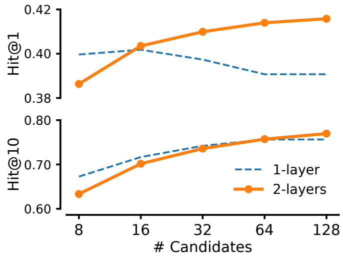
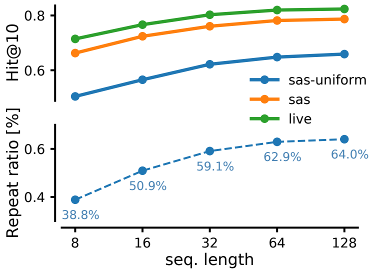


Figure 3.13



Left: Cosine similarity between time interval embeddings. **Center:** Performance with different numbers of pre-ranking candidates with 1 and 2 layers of attention. **Right:** Impact of sequence length on performance (top) and percentage of repeat interactions (bottom).

ture [41, 73, 118]; recommendation algorithms are known for recommending popular items frequently while ignoring items in the long tail. This situation could be problematic in a live-streaming setting, where the interaction distribution over items is already extremely skewed (see Fig. 3.1). Therefore, we investigate the relationship between the introduction of the self-attention mechanism presented in Section 3.5.2 and the popularity of recommended items.

In general, the introduction of the attention mechanism leads to a decrease in popularity of the recommended content. Specifically, we observe a reduction in the average popularity of recommended items of **17.1%** with the introduction of self-attention (av) and **20.3%** with time interval embedding ($av + rep$), compared to the sequence encoder alone ($SASREC$). As one can observe in Fig. **??**, the introduction of $rep + av$ leads to a popularity distribution of recommended items that matches the observed distribution of interactions more closely.

Question 4: *What types of dynamics are captured by time interval embeddings?*

In order to characterize temporal patterns captured by *LiveRec*, we compute the similarity matrix between embedding vectors after a full training phase (see Fig. 3.10). First, in accordance with our observations in Section 6.3, the first bucket, representing a time interval of less than 12 hours, is different from all other vectors. This observation suggests the dynamics of repeated interactions during the same day to be governed by distinct temporal dynamics (e.g. users returning to the same stream after a few minutes). Second, we observe a visible pattern for intervals of less than one week. This observation is in keeping with the weekly dynamics observed in Section 6.3. Finally, vectors representing time intervals of more than one week become increasingly similar as time intervals increase, which suggests that the model captures a unified representation of long-term repeat patterns.

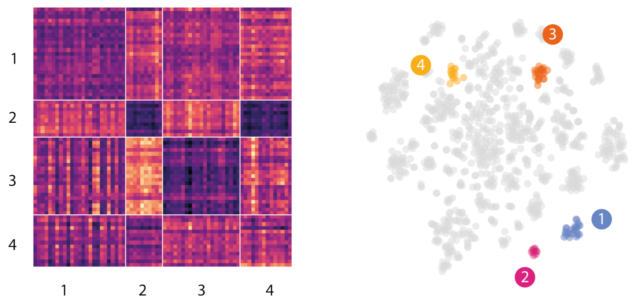


Figure 3.14 – Example of self-attention over available items: the top-1k most popular items are projected using t-SNE (right) and are used as an input for the attention module; the attention weights are displayed after the softmax operation (left). The four selected clusters shown in the projection correspond to the rows (queries) and columns (keys) of the weight matrix.

Question 5: *What types of dynamics are captured by the attention module over available items ?*

Specifically, we are looking for evidences that the attention module can capture dynamics beyond the semantic relatedness of items captured by the encoder. We compare item embeddings, learned from user sequences, and attention weights (av), learned from the context of available items. We embed 1k items using matrix M and provide those items as ($query$, key, $value$) inputs of the self-attention module. The attention weights between *query* and *key* items are shown in Fig. **??** (left). In order to observe the semantic relatedness of those items, we also show a 2D projection (t-SNE [124]) of their embedding vectors and select 4 clusters (center). We observe each cluster of items exhibiting a distinct patterns of attention: attention weights are consistent within clusters but strongly vary from one cluster to another. We also observe the model being able to give attention weight to *query* and *key* items with low content similarity (i.e. belonging to different clusters), which shows the ability of the model to learn this relationship between semantically unrelated items. Considering that the attention module (av) is only *distantly* supervised, since it only learns from a candidate selection of items, our observations suggest that it captures a different, and more global availability context compared to the sequence embedding module.

3.7 Discussion and Future Work

The growing audiences of live-streaming platforms emphasize the need for efficient retrieval and recommendation methods. In this work, we focused our modelling efforts on a dataset of limited size. Scaling our approach, especially to a production environment, would require extra considerations. First, the fast retrieval of available items represents a barrier to large scale studies. For the training on the *benchmark* dataset, we maintained

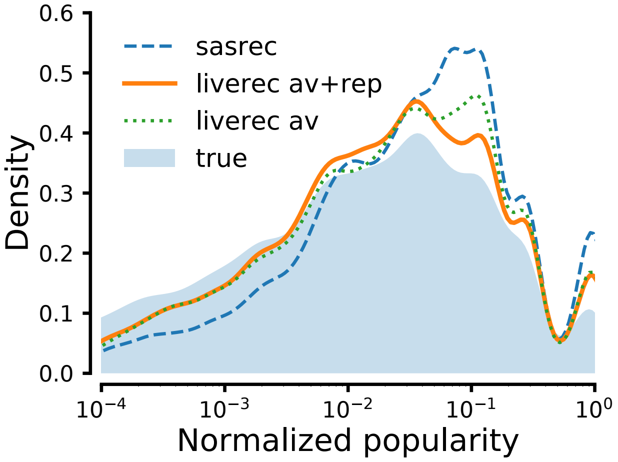


Figure 3.15 – Normalized popularity distribution of recommended items, alongside with the true popularity of consumed items.

availability sets in memory but this approach might reach its limits with larger applications and would require a more scalable design for retrieving and sampling available items. Second, the additional attention module (av) increases the complexity of the model, even though the module has the same architecture as the sequence encoder, making the additional complexity a constant factor. For example, introducing av in the model[VII] reduces the training speed from 8.5 it/s to 5 it/s. In order to account for this observation, one must carefully select sequence length, as well as the number of candidate items, in order to balance between accuracy and training time. Finally, scaling the approach to larger datasets might require more complex training strategies. For example, semi-supervised learning could be employed to first learn item representations on a large dataset, before fine-tuning a more costly approach on a smaller subset of data.

To the best of our knowledge, our work is the first attempt at modelling view dynamics on live-streaming platforms and many challenges remain. First, content-based methods have not yet been investigated. For example, visual features could potentially be exploited to characterize the various segments of a stream. Second, we believe that user dynamics could be further exploited. In this work, we focused on sequential methods that model item-to-item relationships. The dual scenario, the modelling of user-to-user interactions, remains unexplored. In particular, we believe that the modelling of users currently watching a stream could help improving performance. Since the number of users is, by an order of magnitude, higher than the number of streamers, future research should design efficient methods to learn from user interactions. Third, during our analysis, we noticed bursts of activity around specific channels. We believe that the traditionally static notion of item popularity could be adapted to a dynamic setting. For example, the number of concurrent users watching a stream could be exploited at inference time. Fourth, while we focus on *SASREC* for encoding historical data, our framework is not tied to any

[VII] For a sequence length of 16 and a 16 candidate items.

specific method for learning from interaction sequences. Future research could swap the encoder with a different encoding method (e.g. graph-based) to capture a richer context. Finally, many social aspects on live-streaming platforms remain to be explored. Our dataset could help to better understand the social dynamics taking place on a streaming platform. Going beyond the live-streaming setting, we believe that our approach could be leveraged for other types of applications, such as digital TV, and in settings where items only remain available for a limited period of time, such as the front page of a news website.

3.8 Summary

In this chapter, we introduced live-streaming recommendation, a scenario in which items are not always available for users to consume. We showed that a sampling strategy simulating this evolving availability is crucial to capture user preferences. Moreover, we described how to incorporate the notion of availability into our model architecture by performing an explicit comparison among available items. In order to account for the large number of concurrent broadcasts, we made this process efficient by comparing only on a subset of candidate streams. We also investigated repeated user interactions with streamers and proposed a way to model this phenomenon with time interval embeddings, which we show to improve performance. With the release of a large dataset of user interactions on Twitch, the various improvement over existing methods, our study paves the way for new research in live-streaming recommendation.

4 Dynamic Embedding

Online media select and filter the coverage they broadcast through their respective channels. The subjective nature of this filtering induces biases due to, among other things, resource constraints, ideological affinities, or even the fragmented nature of the information at the source's disposal. The magnitude and direction of these biases are difficult to characterize because of the lack of ground truth. In this chapter, we introduce a methodology, based on personalization methods, to analyze the filtering performed by online media.

First, we cast the problem to a personalized ranking task by following a collaborative filtering approach: we train a model to predict the selection of subjects from a media, knowing the coverage from other media. We evaluate our approach on a large set of events collected from the GDELT database. We show that a personalized approach exhibits higher accuracy in coverage prediction and provides an interpretable representation of the selection bias. We propose a method able to select a set of sources by leveraging the resulting latent representation. We show that selected sources provide more diverse and egalitarian coverage, all while retaining the most actively covered events.

Second, we present a dynamic embedding method that learns to capture the decision process of individual news sources over time. Specifically, our method maintains a consistent embedding space over successive time epochs, using a careful initialization of new vectors and introducing a regularization term penalizing large variations across time steps. Our approach enables the systematic detection of large-scale transformations in the media landscape over prolonged periods. We demonstrate the potential of the method for news monitoring applications and investigative journalism by shedding light on important changes in programming induced by mergers and acquisitions, policy changes, or network-wide content diffusion.

4.1 Introduction

World events are reported through an ever increasing number of information channels. These events happen on a variety of different scales, from global to highly local, and all across the planet. To get a grasp on the world's state, even avid readers must pre-process the event space, with such sampling inherently exposing them to a distorted perspective. This processing is a conscious *selection* which not only applies to news consumers, the reader, but also to news providers, the medias. News organizations are designed to be the initial filter of the event stream, pruning, condensing and categorizing it into manageable chunks of information. Unfortunately, it is difficult to guarantee the neutrality of this selection: the process is performed by the editorial team based on an arbitrary number of factors. Some of them are obvious, such as geographic considerations, editorial guidelines, thematic regards or even logistic capabilities. Others are not visible at a glance: ideological leanings or higher order structures such as broadcast syndications or corporate structures. Either one of these can compromise the representativity of the news sample presented: this is generally referred to as *gatekeeping* or *selection bias.*

Any attempt to measure the influence of these factors on news coverage in absolute terms is ill-fated: the factor space could never claim to be exhaustive, and a subset would be at best arbitrary. Additionally, these measures suffer from the absence of baselines: they are all relative estimates, having no ground truth to compare to. These issues are substantial barriers to the interpretability of biases in the coverage of news, which can have a very real impact on the readers' world views [24]. The concentration of media ownership also contributes to reinforcing these biases, since consolidating coverage mechanically weakens media pluralism. The lack of accountability in these issues is an obvious threat to broadcasting diversity and could jeopardize media integrity, aggravating the public's lack of confidence in news sources.[I]

As the selection made by each channel is partial by nature, the variety of outlets, each with a wide array of considerations in the choice of its reporting, is assumed to ensure the diversity of news to which the reader is exposed to. This principle is often referred to as the *external pluralism* assumption. It should ensure heterogeneity in the media space, encapsulating anything from the diversity of ownership to the independence of the editorial board. [II] *External pluralism* is also known as the "supplier" pluralism, since it should exclude the possibility of large broadcast groups exerting influence on downstream reporting. Yet in practice, this assumption does not always hold. News channels are often owned or operated by commercial, private entities, implying that the ecosystem as a whole is influenced by economically motivated forces, such as mergers, acquisitions, or regulatory actions. The increase in concentration of ownership has been observed to be the dominating force in the media landscape, as reported by the Pew Research Institute in a 2017 study on the acquisition of local television stations. [III]

While the literature, still debates the causal effects of market ownership structures on the

[I] http://news.gallup.com/poll/212852/confidence-newspapers-low-rising.aspx

[II] As opposed to *internal pluralism*, where sources are assumed to present a wide variety of ideological viewpoints, communicated through different mediums. [27]

[III] http://www.pewresearch.org/fact-tank/2017/05/11/buying-spree-brings-more-local-tv-stations-to-fewer-big-companies/

diversity of offerings, the Federal Communications Commission (FCC) [IV] has defended the idea that "there is a positive correlation between viewpoints expressed and ownership of an outlet". [V] This uncertainty strongly motivates the development of novel and interpretable methods, which would allow the observation of large-scale movements in the media landscape and allow correlating the observations with real-world factors.

In the first part of this chapter (Section 4.4), we establish a methodology to identify and characterize bias in the mainstream media landscape by recognizing it as a manifestation of the selection process performed by a news source. This paves the way for its treatment as a preference problem, well suited to approaches inspired by personalization methods. We first argue that capturing this bias would require comparing distributions of covered events *across* news sources, as a biased selection of stories from a news media cannot be observed by looking at the source alone. We thus intend to measure by how much a specific source's news selection deviates from another by learning a latent representation of this source's preferences from its observed selection of events. We hypothesize that this representation allows the study of relationships between sources, and sheds light on the factors that guide their decisions. We also propose a method to promote diversity and equality in the coverage of events by selecting a small representative subset of sources.

In the second part of this chapter (Section 4.8), we propose a dynamic embedding model of the media landscape. By predicting news sources' coverage, the model captures their similarities throughout the observed period. The embedding space is maintained consistent over time by augmenting the model with a knowledge of previous time steps, and by adding a temporal regularization on the model's parameters. This improves the model's predictive capabilities and provides a temporally coherent source-wise similarity metric, allowing the visualization and analysis of long-ranging fluctuations in the news ecosystem. We also propose a systematic method to detect abrupt transition patterns in this similarity space. This enables the analysis of the news ecosystem beyond domain-specific knowledge and hand-crafted analysis.

To illustrate, we provide some prototypical questions which could arise in a journalistic probe, and that could be elucidated by information derived from the proposed model:

- What effect does the ownership of a news channel have on its content diffusion?
- Which sources are most similar (resp. dissimilar) to a sample source, and how has this similarity evolved over time?
- Which are the most varying news channels in terms of broadcast content?
- Which broadcast groups exert a large influence on the content of their respective channels?

[IV] The FCC is the regulating body for multimedia communications in the United States.

[V] FCC, Biennial Media Ownership Order (2003)

4.2 Related Work

Media Bias: The presence of bias, as well as its formal definition, have been widely discussed in the literature. Early work in the domain, by Groseclose et al. [38], highlighted the left-right cleavage in the coverage of several of the major media outlets by computing an ideological score for each of them. Their approach relies on the observed number of citations of several policy groups in the news relative to the mentions of the same groups by several Congress members. More recently, Lin et al. [68] compared the coverage bias between mainstream media and social media, focusing on stories about the 111th US Congress (2009-2011). They reported a slant in terms of political leaning and a geographic bias.

Saez-Trumper et al. [108] analyze, at a large-scale, the bias in both traditional press and social media. Their study relies on an Principal Component Analysis (PCA), an unsupervised approach, to detect similarities across news channels. They considered three types of biases, namely: *gatekeeping bias*, that defines how stories are selected or ignored in the news, *coverage bias*, that measures how visible an issue is in the news and *statement bias*, that quantifies how the tone of an article is slanted toward or against a particular entity.

One of the consequences of a biased press is the formation of a figurative *echo-chamber*, an analogy to the acoustic echo-chamber in which sounds reverberate. The analogy sketches a press in which reputable sources go unquestioned and opposing views are censored. Moreover, the homogenization of views inside an *echo-chamber* artificially reinforces the perception of a universally accepted view. *Echo-chambers* have been studied in social media by Wallsten et al. [130], Flaxman et al. [31] and Bakshy et al. [10].

Steiner's seminal work [117] studies the interplay of consumer preferences and in-market competition on the diversity of radio broadcasting. This study omitted the role of external driving forces, which were only later modeled by Anderson & Coate. [9] They predicted that media consolidation, while economically beneficial for the market, would reduce competition and hence diversity for the viewer. This insight was later formulated in terms of ideological bias by Gentzkow & Shapiro [32] in the case of newspapers.

The tendency to integrate external driving factors has picked up steam in recent years, most notably with a theory of convergence in the media ecosystem. This convergence expresses itself through two seemingly contradictory features. On one hand, information is being delivered through an ever-increasing number of channels and means of diffusion. On the other hand, media ownership concentration has seen an upwards trend, with a large proportion of channels being owned by only a handful of media conglomerates. [92] This dichotomy has been studied by, among others, Jenkins. [53] The author proposed a sketch of the phenomenon that looks further than the sole technological influence, reaching for larger cultural factors. Vizcarrondo et al. [129] have more specifically investigated the concentration of media ownership. They reported on changes in the diversity of ownership within the media industry covering the 1976 through 2009 time period.

A large body of work has also been introduced to study the effect of an ideologically slanted press. For instance in a large-scale observational study, DellaVigna et al. [24] measured the effect of the introduction of a conservative-oriented channel (Fox News) led to gains of 0.4 to 0.6 percentage points in Republican voting in the towns where the channel was being broadcast. While specific to a particular orientation, this work is in line with studies showing the profound influence of the media in voters' political awareness [79] and their participation in the electoral process. [34, 85, 33]

The Federal Communications Commission (FCC) regularly issues studies regarding the state of the news ecosystem. Specifically, some of these studies focus on the effect of ownership on local news stations' content programming behaviors. However, by the authors' own admission [VI], these works often lack the breadth required by a large-scale empirical study. For example, Pritchard [93] conducted a study of the diversity of coverage for cross-owned media outlets during the 2000 presidential campaign but on a sample of only 10 newspapers. Groseclose & Milyo [39] proposed a measure of media bias which was evaluated on a set of 8 newspapers. Djankov et al. [26] did survey the news ecosystem on large scale, building a map of media ownership in 97 countries around the world, but this work dates back to 2003.

Methods: Temporally-aware methods have received increasing attention and many previous models have now been adapted to the temporal setting. The *dynamic embedding*, proposed by Rudolph et al. [106] as a variation of traditional embedding methods, is generally aimed toward temporal consistency. The method is introduced in the context of word embeddings, which are used to characterize the evolution of English language. The model is built upon the initial *exponential family embeddings* model. [107]

The field of personalization has many examples of temporally-aware models since human preferences tend to evolve over time. For example, influential work from Koren et al. [59] models the changing nature of preference through a linear drifting term. Another approach relies on the use of Tensor Factorization (TF), [28, 5, 140] in which the extra dimension models temporal patterns in the data. We do not consider TF-based methods as valid candidate approaches since we focus on the problem of grounding representations over time by penalizing unnecessary differences between successive solutions of the model. The temporal modelling capabilities of TF-based methods would predict the evolution of sources and introduce additional temporal variations, consequently degrading interpretability.

[VI] "[a] larger number of independent owners will tend to generate a wider array of viewpoints in the media than would a comparatively smaller number of owners. We believe this proposition, even without the benefit of conclusive empirical evidence." FCC, Biennial Media Ownership Order (2003)

He et al. [46] introduced a temporally-aware model of a recommender system in order to capture the evolution of fashion trends. Similar to Koren et al. [59], the authors also proposed the addition a drifting term to the model. The authors later proposed the use of a higher-order Markov chain that captures both short- and long-term dynamics [45]. Note that both models make use of Bayesian Personalized Ranking (BPR [96]) for their respective optimization procedure. In the context of networks, Yu et al. [144] proposed a temporal factorization for analyzing the evolution of network structures.

Maximal Marginal Relevance (MMR) [23] is an information retrieval technique that retrieves documents based on relevance, while enforcing diversity. It balances the two aspects through the use of a tunable parameter. We refer the reader to Section 4.6 for a more detailed description.

Research Questions: Given the work above, several research questions are of our interest and have remained unanswered:

RQ1: How to model news coverage as a collaborative filtering problem?

RQ2: Is the model interpretable? Are real-world factors visible through the resulting representation?

RQ3: How to exploit the learned bias representation to select a *diverse* set of sources?

RQ4: How can we maintain a consistent model over multiple time epochs?

RQ5: Is media consolidation highlighted by the resulting latent representation?

RQ6: How to systematically detect abrupt deviations in content diffusion from a source?

4.3 Data

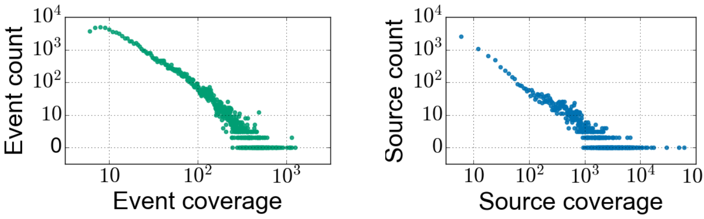


Table 4.1 – Distribution of events and sources for one week.

Date	Sources	Events
Week 1	9'501	76'966
Week 2	9'363	88'755
Week 3	9'741	88'082
Week 4	9'714	89'367
Week 5	9'961	87'574

Table 4.2 – Per-week data statistics

Event coding is the task of extracting, manually or automatically, interactions between (political) actors from large quantities of news text. For decades, event coding was performed manually. In the 1990s, the first automated systems started to gain traction in the academic community, with initiatives such as the KEDS system. [109] Its successor was proposed in the form of *Text Analysis By Augmented Replacement Instructions* (TABARI), which is the engine that runs event coding for GDELT. This framework is designed to process large amounts of text to extract the presence of pairs of *actors* and *verbs.* To do so it matches elements from user-provided dictionaries, which contain a massive collection of event protagonists (i.e. actors) ranging from recognizable named entities (e.g. *Barack Obama*) to functional placeholders (e.g. *a local woman*). These actors are able to interact with the world through *verbs* (i.e. actions), which can be self-contained (e.g. *announces their intent to*) or involve a second actor (e.g. *criticizes their opponent*). Several standards exist for these dictionaries. GDELT uses the *Conflict and Mediation Event Observations* (CAMEO [35]). [VII] Note that, as a remnant of previous hand-curated event annotation frameworks [65], TABARI also provides an interface for manual hand-off to domain experts if the sentences become too complex. This reinforces GDELT's ability to uniquely annotate even the most fine-grained events.

GDELT also augments every news event it tags by extracting meta-information about the article including, but not limited to, its location, its tone, its Goldstein Scale [37] and refences the URL the event was scanned at. It scours a wide array of sources, from television stations to blogs, news wires and papers. Thanks to the information provided by this augmented event coding framework, GDELT assigns, for each news event, a global identifier, which makes it possible to link the same event's coverage across different news sources. Beyond the rich annotations provided by GDELT, this tracking is central to our study given that we only work at the coverage level, without considering the content itself.

4.4 Static Modelling

In this section, we first introduce the use of Matrix Factorization methods in the context of news coverage modeling. Then, we exploit our model to study various factors influencing media decisions. Finally, we propose a method to select a diverse set of news sources by leveraging the latent representation learned by the model.

4.4.1 Model

We model the interrelationships between sources and events by relying on a Matrix Factorization (MF) method. MF methods represent a natural way of projecting two disjoint sets of items in a common latent space of K dimensions in order to model their interactions. Such personalized methods are commonly use by recommender systems,

[VII] An exhaustive list of the considered categories can be found at http://data.gdeltproject.org/documentation/CAMEO.Manual.1.1b3.pdf which routinely aim to model retail purchasing decisions. Used in our context, they model coverage decisions instead. We cast the problem to a One-Class learning setting [86] since we observe positive interactions only. The One-class formulation avoids making assumptions about negative examples: we do not distinguish between real negatives (i.e. the source purposely didn't cover an event) and unobserved interactions (i.e. the source wasn't aware of the event).

Let us consider a set of news sources S and a set of events E. Interactions between the two are represented by an interaction matrix $R \in \mathbb{R}^{|S| \times |E|}$. Observations take the form of dyadic interactions $(s_i, e_j), s_i \in S, e_j \in E$ which express source s_i's coverage of event e_j. Equivalently, we define in matrix form that $R_{i,j} = 1$ if source s_i has covered event e_j and $R_{i,j} = 0$ otherwise. Predicting the unobserved entries of matrix R is achieved by taking the dot-product of two low rank matrices, such that $R \approx P^T \cdot Q$, where $P \in \mathbb{R}^{K \times |S|}$ and $Q \in \mathbb{R}^{K \times |E|}$ with $K << |S|, |E|$. Every source s_i (resp. every event e_j) is represented by a column in P (resp. Q). We will refer to these columns as an embedding vector throughout the remainder of this work. We will refer to Θ as the set of parameters for our MF predictor, such that $\Theta = \{P, Q\}$.

Objectives: The model is trained with the objective of predicting the likelihood of a source covering a particular event. The predicted likelihood $\hat{x}_{s_i,e_j}$ of source s_i covering event e_j is computed as the dot-product between the two respective embedding vectors,

$$\hat{x}_{s_i,e_j} = p_{s_i}^T \cdot q_{e_j} \, . \tag{4.1}$$

Instead of best approximating the reconstruction of matrix R, this objective is stated as a ranking problem in which positive examples should obtain a higher rank than negative ones, i.e. to predict a higher score for an event that has been covered than for random negative samples. Optimizing a MF model with a ranking criterion is equivalent to maximizing the following probability,

$$\begin{aligned} \Pr(e_j >_{s_i} e_k | \Theta) &\coloneqq H(\hat{x}_{s_i,e_j} - \hat{x}_{s_i,e_k}) \\ &\approx \sigma(\hat{x}_{s_i,e_j} - \hat{x}_{s_i,e_k}) \end{aligned} \tag{4.2}$$

where e_j is an event covered by source s_i and e_k is a randomly sampled negative event; formally, $e_j \in E^+_{s_i}$ and $e_k \in E \setminus E^+_{s_i}$. We adopt the notation $>_{s_i}$ to denote a source s_i preferring to cover e_j over e_k and model the observation of this preference using $H(\cdot)$, the Heaviside step function: $H(\cdot)$ is equal to $\mathbf{1.0}$ for positive inputs and to $\mathbf{0.0}$ otherwise. Therefore, $H(\hat{x}_{s_i,e_j} - \hat{x}_{s_i,e_k})$ would always be equal to $\mathbf{1.0}$ for an ideal predictor. In practice, $H(\cdot)$ is approximated by the differentiable logistic sigmoid $\sigma(\cdot)$.

Finally, we maximize BPR, our log-likelihood criterion

$$\mathrm{BPR} := \sum_{(s_i,e_j,e_k)\in D} \ln \sigma(\hat{x}_{s_i,e_j} - \hat{x}_{s_i,e_k}) - \lambda_\Theta \, ||\Theta||^2 . \tag{4.3}$$

Note the inclusion of an ℓ_2-regularization term over the set of parameters Θ. We please refer the reader to the work of Rendle et al. [96] for more details about this optimization scheme.

4.4.2 Optimization

We aim to directly optimize the ranking structure of the problem rather than to provide an accurate reconstruction of the interaction matrix R. The BPR optimization scheme introduced by Rendle et al. [96] is particularly suited for this type of problem and could be applied to our problem using the following update step

$$\theta \leftarrow \theta + \alpha \cdot (\sigma(-\hat{x}_{s_i,e_j,e_k}) \, \frac{\partial \hat{x}_{s_i,e_j,e_k}}{\partial \theta} + \lambda_\theta \Omega'(\theta)), \tag{4.4}$$

where $\hat{x}_{s_i,e_j,e_k} = \hat{x}_{s_i e_j} - \hat{x}_{s_i e_k}$, and θ represents the set of parameters to be learned. $\Omega(\theta)$ denotes a regularizer. We opted for a ℓ_2 regularization $\Omega(\theta) = \|\Theta\|_2^2$.

4.4.3 Experimental Setting

The learning part of our analysis only required to build the interaction matrix between sources and events: we scrape the events and mentions tables to recover which events were covered by which sources in a given timespan. We filter low-count events and sources (sources that have covered less than 5 events, and conversely events covered by less than 5 sources) to limit the impact of the cold-start problem. Fig. 4.1 and Table 4.2 are computed from our dataset after this preprocessing step.

In order to abstract away temporal dynamics, we proceed to temporally split our data. We select five weeks of interest across 2 months (October and November 2016) in the dataset, which are described in Table 4.2. We select one-week chunks to get enough data, and replicate the experiment across the five weeks to measure temporal consistency.

As sources typically cover a highly variable number of events, we adopt a *leave-one-out* methodology to assess the accuracy of the model, with every source having the same weight in the evaluation. Specifically, we constitute our test set by sampling for each source, at random, one event that it covered during the last day of the week.

Reproducibility: We ran our experiment on a single computer, running a 2.3 GHz Intel Core i7 CPU, using Matlab R2014b. We trained our model with the following parameters: $\alpha = 0.1$, $\lambda_\theta = 0.01$, $K = 20$. We found that these were well-performing parameters for the proposed problem: the same parameters were used on all 5 weeks. We note that the number of latent factors K did not show significant information gain after $K = 20$ dimensions.

4.4.4 Evaluation

Prediction accuracy is not the primary goal of our approach but rather a mean to tune the predictor, in order to avoid under- or over-fitting, and to compare it to various approaches. Since the BPR optimization scheme directly optimizes a pairwise ranking criterion, we select the widely used metric *Area Under the Curve* (AUC) [113] as our measure of performance.

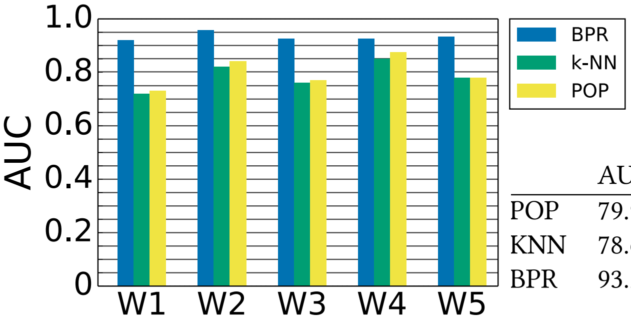
$$\text{AUC} = \frac{1}{|D|} \sum_{(s_i,e_j,e_k)\in D} H(\hat{y}_{s_i e_j} - \hat{y}_{s_i e_k}) = \frac{1}{|D|} \sum_{(s_i,e_j,e_k)\in D} H(\hat{x}_{s_i e_j e_k}), \tag{4.5}$$

where $H(\cdot)$ is the Heaviside step function (the latter formula uses the notation introduced in Section 4.4.1) and D is our evaluation set composed of one triplet (s_i, e_j, e_k) per source where s_i is a source, e_j is a randomly sampled event that has been covered by source s_i and e_k is a randomly sampled event that source s_i has not covered. This metric assesses the ability of the predictor to correctly rank a positive interaction withheld during training against a random negative example. An ideal predictor would obtain a score of $AUC = 1$, while a random selection would output a score around $AUC = 0.5$.

We compare our method to two common baselines used in recommendation problems: popularity and nearest-neighbor methods [101]. Popularity based methods simply rank the events based on the amount of coverage they receive. Nearest-neighbor methods infer a source's coverage from the coverage of its closest peers: the intuition is that congruent sources should exhibit similar coverage of the event space. We chose the k-Nearest Neighbors ($k = 10$) method for this baseline, using the Jaccard distance metric.

4.5 Results

Adopting a supervised learning approach presents the advantage of allowing the explicit evaluation of the quality of our model. We propose that the coverage prediction accuracy yields an adequate estimate of the learned embedding's quality. Indeed, reconstructing the interactions should only be possible if the latent factors captured sufficient information about how sources select the events they cover. This type of evaluation is not feasible with an unsupervised method (e.g. PCA, SVD [114]), which requires expert intervention



	AUC	STD
POP	79.9 %	5.8 %
KNN	78.6 %	5.1 %
BPR	93.2%	1.5 %

Figure 4.1 – Results with AUC as a performance metric. Results are shown per week. We show the averaged score as well as the standard deviation of the results obtained over the 5 weeks.

to judge the quality of results and interpret them.

We reference the results in Fig. 4.1, which shows higher prediction accuracies compared to the selected baselines.

4.6 Source Selection

In the following section, we describe how the apriori knowledge produced by our model can be exploited in the context of news selection, that is the problem of selecting N sources from a large and heterogeneous set. In this scenario, the selection of news sources should be done such that the resulting subset exhibits two desirable properties that makes it representative of the worlds' daily events distribution. First, the news sources should be picked in order to foster diversity. Intuitively, the resulting set should cover a large spectrum of the news while minimizing concentration around a small set of events, thus reducing the effect of the so-called *echo-chamber* [130]. Second, the resulting set of news covered by the selected sources should retain a large proportion of the most actively covered events, ensuring comprehensive coverage of the event space.

Without an accurate way of modeling the inter-relationship between sources, picking a representative subset of media can be difficult. Indeed, the main criterion of selection would have to come from side-information, e.g. the reputability of the source or its level of activity, etc. Therefore, we propose to exploit the knowledge gained from our model to guide this selection.

We adapt to our scenario a standard diversity-promoting retrieval method, Maximum Marginal Relevance (MMR) [23]. MMR is an iterative procedure that establishes a ranking of elements based on two criteria: a relevance score, that is application-specific and has to be defined, and a diversity measure of the retrieved set of elements. MMR balances the two aspects with a tunable parameter β. At each step, MMR selects the source to be added to the results set based on the relevance of the source, that we define as being comprised in the interval $[0, 1]$. This score is then weighted to include results with minimal similarity to the current retrieved set, thus ensuring its diversity. The procedure ranks the sources iteratively based on the following score function

$$\mathrm{MMR}(s_i) := \beta * \mathrm{relevance}(s_i) - (1 - \beta) * \max_{s_j \in B} \Big[\mathrm{sim}(s_i, s_j) \Big], \tag{4.6}$$

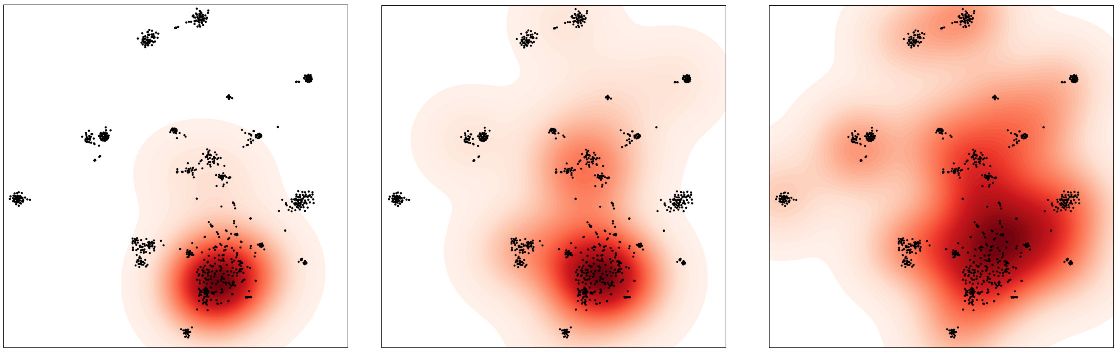
where β is a parameter that controls the strength of the diversification and B is the set of elements already selected (the first pick is thus based on relevance only). With a β value of 1, the ranking is based on relevance only, while with a β value of 0 the ranking is the most diverse set of items possible achievable in a greedy fashion. The formulation of Eqn. 4.6 requires a measure of similarity between sources. After experimenting with different options, we obtained satisfactory results by using $\mathrm{sim}(s_i, s_j) = 1/dist(p_i, p_j)$ as our measure of similarity, with $dist(\cdot)$ being the Euclidean distance between p_i and p_j, two sources' latent representation vectors. We use as a relevance function the activity level of the source, i.e. the number of articles published by the source (see Section 4.7.3).

4.7 Discussion

In the following section, we consider the results of our experiments. We first discuss the method's predictive performance. Then, we analyze the resulting representations yielded by our approach, providing ways of explaining the observed variance. Last, we describe the results of leveraging this representation with our method to promote diversity in a news source selection problem.

4.7.1 Coverage prediction accuracy

As mentioned in Section 4.9, our method of choice presents the advantage of supervised learning procedures, in that it provides a measure of the accuracy of the predicted coverage. Therefore, it allows the comparison with other types of personalization techniques. We select two baselines: the raw popularity of the events and k-Nearest Neighbor (k-NN). Popularity based methods are not personalized: they simply rank the events based on the amount of coverage they received. We show that we can outperform this method as a result of the personalization of the coverage prediction. We also compare to a personalized method, k-NN, and observe that our method achieves better accuracy, due to the fact that it is also parameterized. We report a score (AUC) greater than 90% for the 5 selected weeks. We also observe less variability across the weeks in the results obtained from our method of choice.



original $\beta = 0.75$ $\beta = 0.5$

Figure 4.2 – We illustrate the effect of the β parameter on week 5 of our dataset on a query of size N=100 . Sources' positions in latent space are displayed as single dots. We overlay the density (gaussian KDE) around the sources contained in the selected subset. The original selection picks sources solely based on their level of activity (β=1). The center and right figures have nonzero values of β which diversifies the selection of sources.

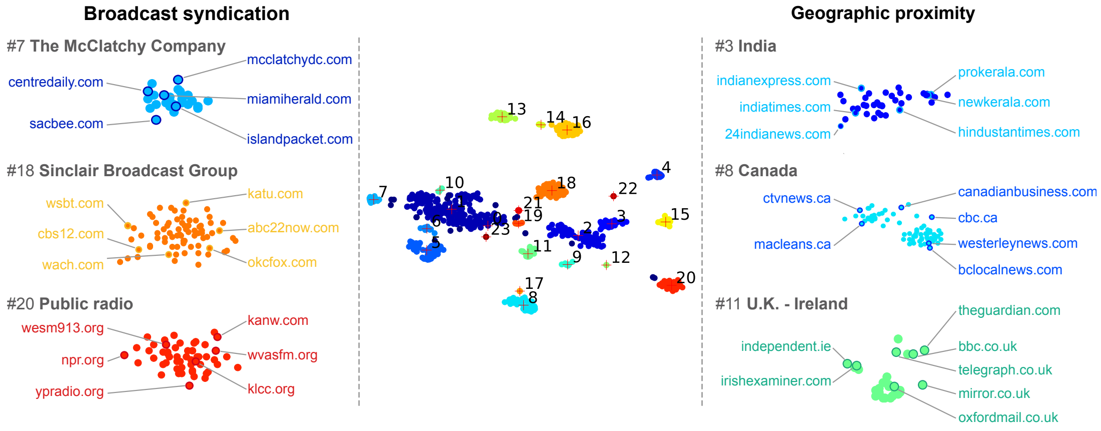


Figure 4.3 – Source agglomerations in latent space (best seen in color)
Left: After investigation, we observe clusters explainable by the publishing structure of sources in the cluster: all are part of a publishing network, such as the public radio network (#20: *left, bottom*) or are all owned by a larger commercial entity (#7: *left, top*, #18: *left, center*). **Center:** Position of the sources in latent space, reduced in dimensionality with t-SNE [127]. An unsupervised cluster learning method (DBSCAN [30]) is applied to show agglomerates of sources that are similar in the latent space. 24 clusters are extracted in this example (Week 1). Visual inspection allows interpretation through the discovery of the biases detailed in 4.7.2. **Right:** We notice several geographical clusters, three of which are detailed here: a cluster of Indian news sources (#3: *right, top*), a cluster of Canadian news sources (#8: *right, center*) and a cluster of sources from Great-Britain and Ireland (#11: *right, bottom*).

4.7.2 Leveraging representations to uncover biases

The methodology described in Section 4.4.1 yields latent-space representations of the source preferences, i.e. a low-dimensional description of the selection bias. By investigating the distances between sources in this preference space, we uncover interesting correlations between them, indicating the presence of a common bias. We also apply standard unsupervised clustering methods to explicitly group sources together. While the measures are done in the latent space, we project these vectors down to 2 dimensions for visual inspection.

Since the structure arises directly from the coverage we can extract factors of the bias, such as those that we mentioned in Section 4.1 (geographic relationships, thematic regards, higher-order structures, ...) despite them not always being evident to the inexperienced eye (for example broadcast affiliates owned by larger structures which are not reflected in branding).

Geographic proximity: The simplest similarity between sources comes from their geographic proximity: local or national sources orient their coverage to their respective scales. Hence sources with similar geographic dependencies should present similarities in their coverage, and be close together in the latent space. This effect is indeed captured by our method, as shown in Fig. 4.3, *right.* This geographic relationship between sources is confirmed by the proximity of regional sources, such as `prokerala.com` and `newkerala.com`, two sources from the region of Kerala in India: they are in a cluster of Indian news sources, but are also close together in the latent space as they cover national and regional news. The same effect is visible in a portion of cluster #8, with sources from British Columbia, Canada, being close together (`westerleynews.com` and `bclocalnews.com` are shown here).

Affiliation and ownership: Local news sources are an essential part of the news coverage network, most notably in rural areas where they represent one of the only sources of information with a granularity level fine enough to cover very local events. While it is to their advantage to also provide general news coverage to their readers (national or international news), they usually lack the resources to be involved in the treatment of events at that scale. Hence a common method has long been to agglomerate into larger organizations: groups of local news sources dedicating a fraction of their budgets to pool the coverage between them, forming a broadcast syndication network [70].

Note that these groupings are not necessarily horizontal: they can also be the fruit of consolidations through mergers or acquisitions by larger organizations (the Pew Research Center estimates that the five largest broadcast companies now own 37% of local television stations in the United States[VIII]). In cluster #18, we show a group of sources all owned

[VIII] http://www.pewresearch.org/fact-tank/2017/05/11/buying-spree-brings-more-local-tv-stations-to-by the same corporate structures, formed by a wave of acquisitions in the local news space.

These larger structures are not always obvious at a glance. Many familiar networks are present in the list of sources shown in Fig. 4.3, such as the American Broadcasting Company (ABC[IX]) (`abc22now.com`), Columbia Broadcasting System (CBS[X]) (`cbs12.com`) or even Fox[XI] (`okcfox.com`) but none of these are actually operated by the network their name suggests: they are all operated by the same broadcast entity.

Medium: Some of the larger structures that form are driven by platforms based on similar media. Cluster #20 brings together a network of public radio stations. They are usually affiliated with one or several organizations such as NPR[XII], Public Radio International[XIII] or American Public Media[XIV], all of which are non-profit entities exchanging content to form a radio syndicate.

A few observations are left as side-notes. First, we report the clusters discussed in this section to be largely consistent throughout the 5 selected weeks. We report an average Pearson correlation of **0.82** between the pairwise distances in embedding space of the top-1000 most active sources across the 5 weeks. Second, we did not observe any clear left-right cleavage, and, therefore, do not report on it.

4.7.3 Application to source selection

In this section, we develop the results obtained by the proposed method in the context of source selection. In particular, the properties of their combined coverage of the event space is of our interest. A skewed selection of news sources could induce side-effects. The selected sources could cover a too-small or non-representative portion of the event space by focusing on a few highly discussed topics. As a consequence, those events would be overrepresented while other topics of importance would be drowned. Therefore, we discuss the results of the news selection problem with respect to two aspects. First, we report a metric of coverage equality received by the events. An *egalitarian coverage* should give a similar importance to all events treated by our selected sources. Second, we report the ability of the method to retain the most actively covered events in the set.

We first select a subset of N news sources based on a ranking criterion that does not require any side-information: their respective levels of activity. This naive approach ensures the resulting selection to include the largest possible number of articles. We

fewer-big-companies/

[IX] abcnews.go.com

[X] cbs.com

[XI] foxnews.com

[XII] npr.org

[XIII] pri.org

[XIV] americanpublicmedia.org

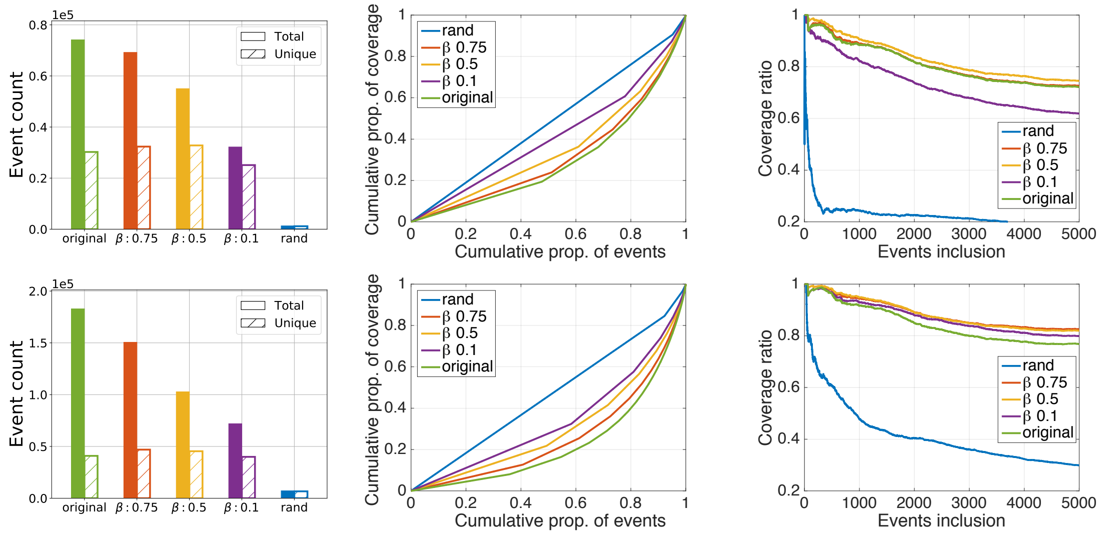


Figure 4.4 – top-25 sources selection (first row) and top-100 sources selection (second row). We report the coverage produced by the original ranking (ranked by the number of articles published), the same ranking with the diversity constraint for different values of β, and the coverage of a random selection of sources. Left: Number of articles covered (total and unique) by the selected subset of news sources. Center: Lorenz curves of the coverage received by individual events in the selected subset of news sources. Right: Proportion of the top-5000 most discussed events of the week included in the coverage of the selected subset of news sources. For example, a top-100 on the x-axis represents the percentage of the 100 most covered events in the entire set that have been covered at least once by the selected subset of news sources.

therefore expect it to contain a wide spectrum of events. We then compare this coverage with the one produced by a ranking with the additional diversity constraint presented in Section 4.6. We report that a skewed attention in the original ranking of sources provides a ratio of #events/#articles of **0.41** for top-25 and **0.22** for top-100. This ratio suggests a lot of repetitions around the same subset of events. However, we observe this effect being mitigated by the re-ranking procedure. For example, we obtain a ratio of **0.60** for top-25 and **0.44** for top-100, by fixing the value of the β parameter to 0.5. A more detailed view of this discrepancy, and its mitigation, is shown in Fig. 4.4 (left).

This ratio gives an indication of the overall novelty provided by a set of sources. However, it does not show the unequal treatment of the event, which we have hypothesized. If we consider the coverage of news sources as a budget of attention, we observe the attention income that every event receives. In fact, the Lorenz curves (Fig. 4.4 center) indeed reveal the attention budget of the press being spent unequally for a selection of the most active sources. We report that this effect is also mitigated by the proposed approach. We also estimate the imbalance in the resulting coverage in statistical terms using the GINI coefficient which measures the inequality of a distribution. A perfectly egalitarian coverage would have a GINI coefficient of 0, meaning all events receive equal attention.

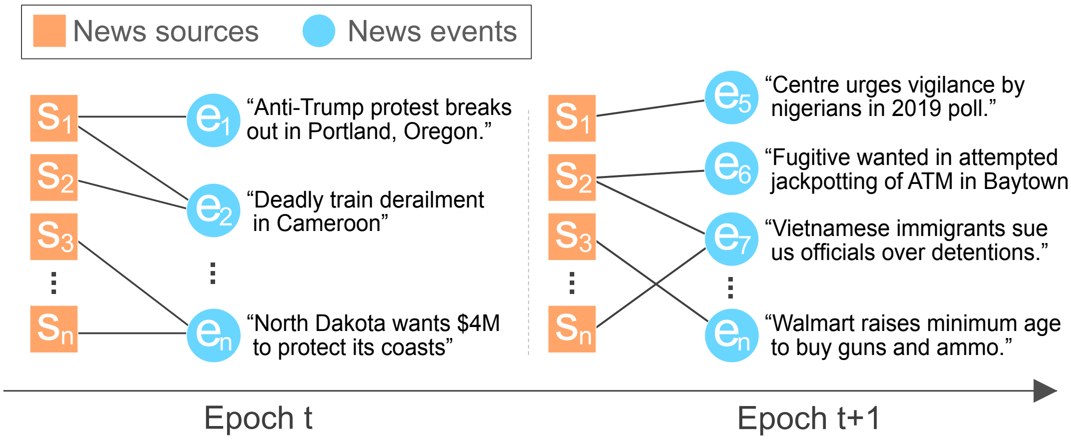

News sources
News events
S1
S2
S3
Sn
e1
"Anti-Trump protest breaks out in Portland, Oregon."
e2
"Deadly train derailment in Cameroon"
en
"North Dakota wants $4M to protect its coasts"
S1
S2
S3
Sn
e5
"Centre urges vigilance by nigerians in 2019 poll."
e6
"Fugitive wanted in attempted jackpotting of ATM in Baytown."
e7
"Vietnamese immigrants sue us officials over detentions."
en
"Walmart raises minimum age to buy guns and ammo."
Epoch t
Epoch t+1

Figure 4.5 – Illustration of the setting: we observe the coverage of news events from a fixed set of sources over several time epochs. Example events are extracted from the GDELT database.

For a selection of 25 sources, we obtain a GINI coefficient of **0.79** that reduces to **0.74** after re-ranking ($\beta = 0.5$). Similarly, for a selection of 100 sources, we obtain a GINI coefficient of **0.78** that reduces to **0.68** after re-ranking ($\beta = 0.5$).

Although equality in the coverage is a desirable property, we cannot sacrifice the total coverage to achieve an egalitarian distribution: this would mean discarding too many important events for the coverage to be meaningful. Hence we also report the propensity of our selected subset to retain events of importance, as shown in Fig. 4.4 (*right*). We ranked the event by importance, the top events being the ones that have been covered by a larger number of sources during the week. We show the resulting selection of news sources includes a larger proportion of the most discussed topics despite covering a smaller set of unique events.

The last point of our discussion treats of the balance between coverage equality and top-event retention. The choice of the β parameter is a trade-off between the two aspects that could be fixed through numerical analysis or include human judgment. However, a value of $\beta = 0.5$ allows to substantially reduce the imbalance, while still including a larger proportion of top events in the resulting coverage.

4.8 Dynamic Modelling

In this section, we propose two strategies to adapt the model to a temporal setting. This change enables the visualization of the news ecosystem over time and to detect abrupt variations in coverage of individual channels.

4.8.1 Data Preprocessing

From the massive resource maintained by GDELT we can gather a dataset of interactions between sources and events, recording which sources covered which uniquely identifiable events. We focus our data collection campaign on the publicly available dumps of GDELT 2.0, released every 15 minutes since February 2015. General statistics about the dataset are presented in Table 4.3 as well as Fig. 4.6 and Fig. 4.7.

When considering the full time-span, GDELT references more than 105K different news sources. This represents a considerable increase compared to the 63K sources reported by Kwak et al. [7] in 2016. However, as is shown in Fig. 4.7, most of the sources have only published a few articles over the relevant stretch. To maintain a consistent number of channels over time, we discard all channels inactive in any one of the time slices from our dataset. This retains around 7 278 news sources in our dataset. The filtering step does remove a large fraction of available channels, but it mostly affects sources with a very low publishing rate: despite preserving only 7% of the total channels, the selection still accounts for more than 76% of interactions in the dataset (see Fig. 4.7).

Figure 4.6

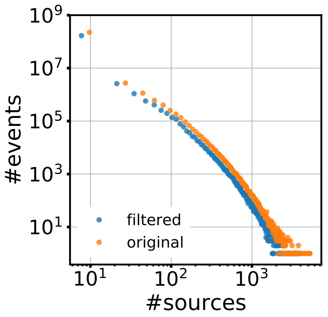
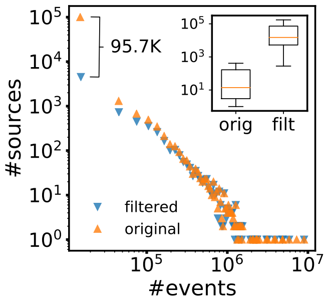


Figure 4.7



GDelt Dataset

# sources	7 278
# unique events	174M
# interactions	588M
time span	2015 - 2018

Table 4.3

Left: Number of sources covering an event. **Center:** Number of events covered per source. **Right:** Dataset statistics.

The dataset is split into slices with a duration of one month. This allows for a decent trade-off between having a significant amount of events covered in the training set, while also providing enough samples to observe time-dependent changes over the considered period. In principle, this scale could be modified to study the media landscape at different granularity levels. For example, a more fine-grained split might allow the observation of changes correlated with specific events. However, we choose to leave this analysis as future work given the significant amount of computation that the model requires.

4.8.2 Temporal Setting

In this section, we describe the adoption of the dynamic embedding scheme proposed by Rudolph et al. [106] in the context of news coverage modeling. In particular, we adopt two strategies to maintain temporal consistency across time slices, respectively based (i) on a Gaussian random walk (RW) and (ii) on the addition of a temporal regularization term (RG). We adopt the notation $p_{s_i}^{(t)}$ to denote the embedding vector of news source s_i at the t-th time step.

Prior on the embedding vectors: Without information about former time slices, existing methods typically initialize embedding vectors to small, randomly distributed values. However, such approaches do not take advantage of any prior knowledge acquired during anterior training steps. The addition of a prior on embedding vectors represents a simple, yet powerful strategy to leverage previously acquired knowledge about sources. In particular, embedding vectors at the t-th time step are initialized using a Gaussian random walk around their final values at time step $(t-1)$. The Gaussian random walk is expressed as follows:

$$p_{s_i}^{(t)} \sim \begin{cases} \mathcal{N}(0, \gamma^{-1} I), & \text{if t=0}\,. \\ \mathcal{N}(p_{s_i}^{(t-1)}, \gamma^{-1} I), & \text{otherwise}\,. \end{cases} \tag{4.7}$$

This initialization scheme ensures a smooth transition of the parameter set learned in two consecutive time slices. This yields a more stable embedding space, offering a coherent expression of divergence across time-steps. Since events are inherently much more volatile than sources, we initialized their embedding vectors at random at each new time slice.

Optimization using temporal regularization: The second part of the dynamic scheme takes the form of a temporal regularization term. The newly introduced term penalizes large variations across time steps by minimizing the distance of an embedding vector at the t-th step to its final value at step $(t-1)$.

The final log-likelihood criterion, BPR-T, can then be formulated as follows for the t-th time split

$$\begin{aligned} \text{BPR-T}^{\,(t)} := \sum_{(s_i,e_j,e_k)\in D} & \ln \sigma(\hat{x}_{s_i,e_j} - \hat{x}_{s_i,e_k}) \\ & - \lambda_\Theta \,||\Theta||^2 \\ & - \underbrace{\lambda_T \| p_{s_i}^{(t)} - p_{s_i}^{(t-1)} \|^2}_{\text{temporal regularization}} \end{aligned} \tag{4.8}$$

The model is optimized using stochastic gradient ascent and is fitted once for every time split. Update steps are defined as follows:

$$\begin{aligned} q_{e_j}^{(t)} &\leftarrow q_{e_j}^{(t)} + \alpha(\sigma(-\hat{x}_{s_i,e_j,e_k}) \cdot \quad p_{s_i}^{(t)} \quad - \lambda_\Theta \, q_{e_j}^{(t)}) \\ q_{e_k}^{(t)} &\leftarrow q_{e_k}^{(t)} + \alpha(\sigma(-\hat{x}_{s_i,e_j,e_k}) \cdot (-p_{s_i}^{(t)}) - \lambda_\Theta \, q_{e_k}^{(t)}) \\ p_{s_i}^{(t)} &\leftarrow p_{s_i}^{(t)} + \alpha(\sigma(-\hat{x}_{s_i,e_j,e_k}) \cdot (q_{e_j}^{(t)} - q_{e_k}^{(t)}) \\ &\qquad - \lambda_\Theta \, p_{s_i}^{(t)} \\ &\qquad - \lambda_T(p_{s_i}^{(t)} - p_{s_i}^{(t-1)})) \end{aligned} \tag{4.9}$$

where α is our learning rate. We use the notation $\hat{x}_{s_i,e_j,e_k}$ to denote the quantity $(\hat{x}_{s_i,e_j} - \hat{x}_{s_i,e_k})$. Note that triplets $(s_i, e_j, e_k), e_j \in E_{s_i}^+$ and $e_k \in E \setminus E_{s_i}^+$ forming the training dataset D are randomly sampled during the optimization. [XV]

4.8.3 Evaluation

To assess the performance of the different methods, we adopt a *leave-one-out* methodology, in which a single event per source is withheld at random from the training set D to constitute the test set D_s. This approach ensures that all sources have similar weights in the evaluation. We adopt the widely used *Area Under the Curve* (AUC) as a measure of performance. In the context of this work, the evaluation procedure is formally defined as follows

$$\text{AUC} = \frac{1}{|D_s|} \sum_{(s_i,e_j,e_k) \in D_s} H(\hat{x}_{s_i,e_j} - \hat{x}_{s_i,e_k}) \,. \tag{4.10}$$

where e_j is an event covered by s_i and e_k is an event that s_i hasn't covered, randomly sampled at testing time. Negative samples are drawn uniformly at random across all unique event of the current time slice (we omitted the time indices for the sake of brevity).

4.8.4 Experimental Setting

The code for experiment and analysis will be made available at publication time under an open-source license. All experiment-related code was run on a 6-core machine, equipped with an `Intel(R) Xeon(R) CPU E5-2630 @ 2.30GHz`, for a total training time of approximately 3 days. [XVI] We restricted the tuning of hyper-parameter λ_T to a subset of values $\in \{0.001, 0.01, 0.1, 1.0\}$. All scores and figures are reported using $\lambda_T = 0.1$, which we

[XV] We sampled both positive and negative examples uniformly. More complex sampling approaches exist [95] but are outside of the scope of this work.

[XVI] The training time is reported for the full 3-year period; a production-ready application would typically be optimized and use incremental updates instead.

Model	AUC
POP	0.6509
BPR	0.8959
BPR + RG	0.9089
BPR + RW	0.9318
BPR + RW + RG	**0.9337**

Table 4.4 – Method contribution to performance

found to provide the highest cross-validated accuracy. For computational reasons, the other parameters were coarse-tuned on a static snapshot and were set to $K = 20$ and $\lambda_\Theta = 0.1$, $\gamma = 0.01$ and $\alpha = 0.1$.

Figure 4.8

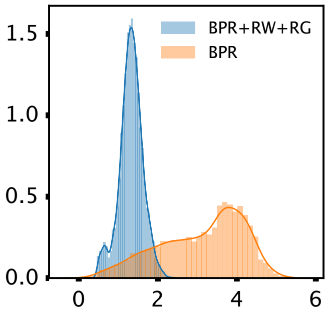
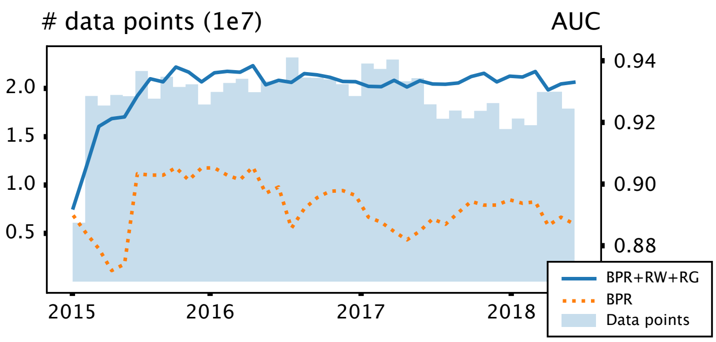


Figure 4.9



Left: Average displacement of embeddings. **Right:** Performances (AUC) and dataset size per month.

4.9 Results

In this section, we compare the existing static embedding model with the dynamic method presented in Section 6.5. The effect of both the prior on the embedding vectors and the temporal regularization are then measured in isolation.

Table 4.4 summarizes the performances of the various approaches by taking the mean AUC scores obtained, for each month, over the considered period. *BPR* denotes the core of the algorithm without any temporal component. *RG* denotes the use of temporal regularization and *RW* denotes the use of a Gaussian random walk for embedding initialization. To compare with a non-parametrized approach, we also include *POP*, a popularity-based baseline: the score for a given event is a function of its frequency in the training set.

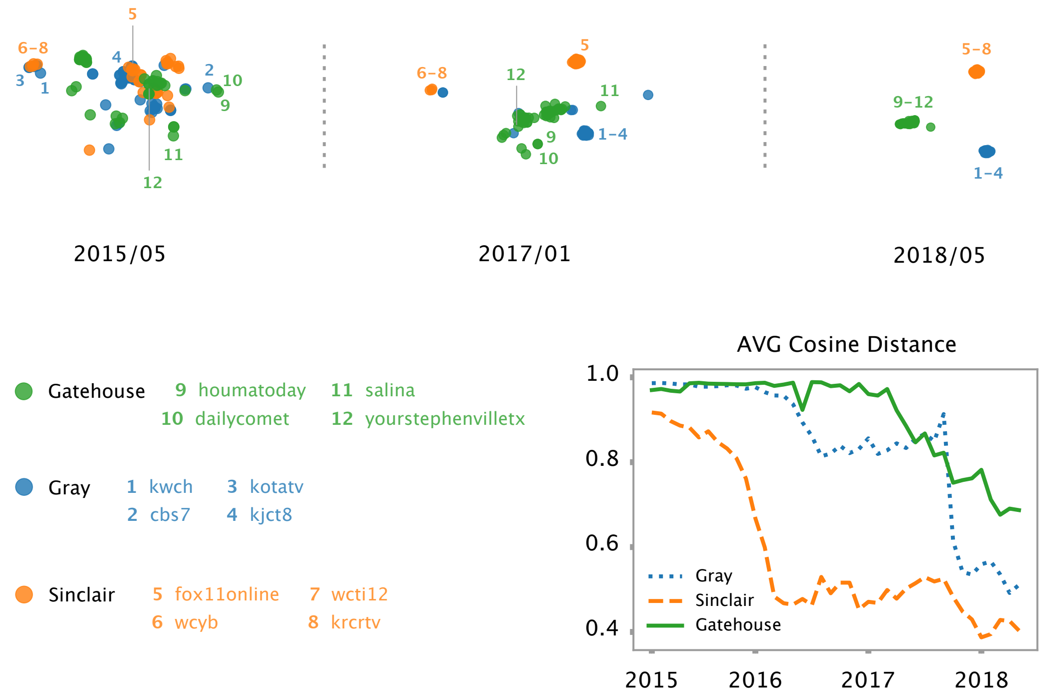


Figure 4.10 – Case-study of the influence of ownership on the selection of covered events (best seen in color)

Left: Sample of the temporal evolution of the media landscape, learned with dynamic embeddings, illustrated with t-SNE. [125] To test the effect of ownership on diversity of coverage, we monitor the evolution of a set of sources all owned by the same broadcast group in 2018, backtracking their evolution in embedding space. Starting with an initial seed source known to be operated by - or affiliated to - one of three large American media conglomerates (Gray Television Inc., Sinclair Broadcasting Group and GateHouse Media), we build a set of 50 sources per group so as to have a comparably sized sample for each. These sources are also all verified to be owned - or operated by - one of these groups at the last observed time-frame of our dataset (May 2018) by cross-checking publicly available information. We project their positions from the embedding space onto a display-friendly plane, showcasing snapshots of their movement over time, as they collapse into highly similar and cohesive clusters in the similarity space.

Right: Average inter-source cosine distance between sources of each group in the dynamic embedding space, over time.

Several observations can be made in light of these results. Firstly, the combination of the two strategies *RG* and *RW* is shown to provide the best observed predictive performances. Secondly, the individual strategies do not provide the same performance improvements. Results suggest that a proper initialization contributes much more to an accurate prediction than strong temporal regularization. In parallel experiments, we even observed that an increase of λ_T decreases performances. We hypothesize that this is due to the model's inability to handle abrupt changes in source behavior, since a strong regularizer would penalize a large difference with respect to the previous time step.

The use of the proposed dynamic strategies (*RG* and *RW*) provides better overall consistency of the latent space across time slices. This is visible by measuring the average displacement of sources in embedding space. As shown in Fig. 4.8, sources are much more stable in the dynamic setting compared to the static embedding procedure. The added stability of the embedding space provides a usable expression of divergence across time-steps. This means source similarity can be coherently compared across the entire observed period, while also providing an overall improvement of the AUC scores (see Fig. 4.9).

4.10 Analysis

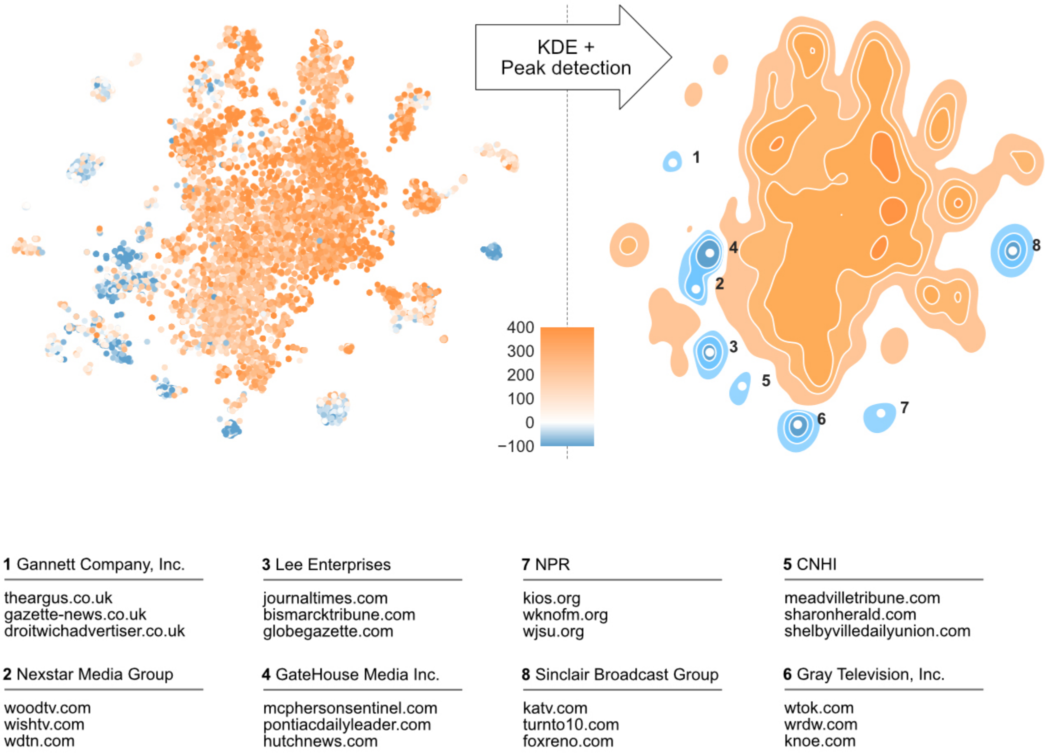
In the following section, we discuss the interpretation of the model introduced in Section 6.5. We first describe an approach to visualize the evolution of the news ecosystem. Then, we propose a systematic, unsupervised way to detect abrupt deviations in this space.

4.10.1 Visualizing the Media Landscape

We start by introducing an example case-study, which should illustrate the usefulness of the presented model in facilitating the understanding of the news ecosystem. The study is centered on three representative media conglomerates that we track throughout the 40-month period covered by our dataset: *Gray Television Inc.* (over 100 television stations), *Sinclair Broadcasting Group* (over 190 television stations) and *GateHouse Media* (over 140 newspapers). [XVII]

As a starting point, the model described in Section 6.5 is optimized in its most successful setting (*RW+RG*). Dimensionality reduction is performed in order to have a more interpretable view into the embedding vectors. In Fig. 4.10 - (*left*) we use a t-distributed Stochastic Neighbor Embedding (t-SNE [125]), a popular method for the visualization of high-dimensional data. In order to maintain consistency across time steps in the projection, we initialize the parameters of the t-SNE optimization procedure at time slice

[XVII] as of August 2018.



1 Gannett Company, Inc.

theargus.co.uk
gazette-news.co.uk
droitwichadvertiser.co.uk

3 Lee Enterprises

journaltimes.com
bismarcktribune.com
globegazette.com

7 NPR

kios.org
wknofm.org
wjsu.org

5 CNHI

meadvilletribune.com
sharonherald.com
shelbyvilledailyunion.com

2 Nexstar Media Group

woodtv.com
wishtv.com
wdtn.com

4 GateHouse Media Inc.

mcphersonsentinel.com
pontiacdailyleader.com
hutchnews.com

8 Sinclair Broadcast Group

katv.com
turnto10.com
foxreno.com

6 Gray Television, Inc.

wtok.com
wrdw.com
knoe.com

Figure 4.11 – **Top-left:** t-SNE [125] projection of the embedding space for May 2018, colors represent the weight w_{s_i} of each source s_i. **Top-right:** Identified attractor map built from the attraction potential map ($\sigma = 1.9$, $k = 500$). **Bottom:** Detail and affiliation of attractors identified in (*center*), with the set of 3 sources closest to the uncovered poles. Best seen in color.

t with the final parameters of time step $t - 1$. This seeds the projected points' positions instead of assigning them to a random initial position, allowing for easier tracking between time steps.

Additionally, in order to avoid interpreting from model parameters only, which might be misleading due to optimization artifacts, we correlate sources' trajectories with the average pairwise cosine distance between sources of each group. These distances are computed using sources' respective sets of covered events for each month. Overall, this procedure allows us to coherently visualize the evolution of the media landscape over time, uncovering non-obvious dynamics at several scales, from the ecosystem as whole (e.g. convergence phenomenons) down to individual sources (e.g. shift toward a group).

The designed visualization of the media landscape is presented in Fig. 4.10. Please refer to Section 5.5.4 for a more detailed discussion.

4.10.2 Detecting Fluctuations in the Media Landscape

Even with extensive domain knowledge, tracking the evolution of a group of sources belonging to specific entities remains a tedious task, since it requires the manual identification of sources of interest and the validation of their common factors. Therefore, in the following section, we propose an unsupervised method which leverages the models' a priori knowledge to identify abrupt changes in sources' content diffusion patterns. In particular, the proposed framework aims to identify attractors, e.g. sources that tend of attract others in latent space, suggesting an alignment of coverage.

Attractors: News channels involved in a consolidation of resources typically tend to have increasingly similar coverage patterns. As seen in Fig. 4.10 - (*right*), the phenomenon manifests itself as the convergence of a subset of sources toward a common position in embedding space. Systematically detecting such gatherings around a common location, that we will loosely refer to as attractors, would allow to interpret each of these patterns in isolation. We propose a method in two steps. Firstly, we identify sources whose distances to other channels are abruptly reduced at any point in time. Secondly, we identify the absolute position towards which those sources tend to converge.

We first define the matrix $Z \in \mathbb{R}^{|S| \times |S|}$ that represents the difference of distance between the first and the last considered time step, for any two news sources i and j. More formally, we define Z as follows.

$$Z_{i,j} = \quad \|p_{s_i}^{(1)} - p_{s_j}^{(1)}\| - \|p_{s_i}^{(t)} - p_{s_j}^{(t)}\| \tag{4.11}$$

where $\|\cdot\|$ is the Euclidean norm. We rely on the matrix Z to identify channels having undergone large reductions of distance with other sources during the considered period $[1 \ldots t]$. In more details, we retrieve from Z the k-sources having the largest negative difference with any other source, i.e. the minimal value of each row in $Z_{i,j}$. By taking into consideration only the top-k, we can capture large shifts only and avoid considering small movements due to random factors. Once identified, the relative displacement of these k sources can be visualized in latent space. In particular, each source s_i in our dataset will be qualified by a single weight w_{s_i}, computed as the sum of the difference in Z with respect to the k considered sources. As shown in Fig. 4.11 - (*left*), negative values reveal sources that tend to exhibit agglomerative behaviors.

Until now, we observed strong fluctuations in terms of inter-source distances. The next step is to define a systematic way of identifying the centers around which these shifts occur, in absolute terms and at any given point in time. On a 2D projection of sources,

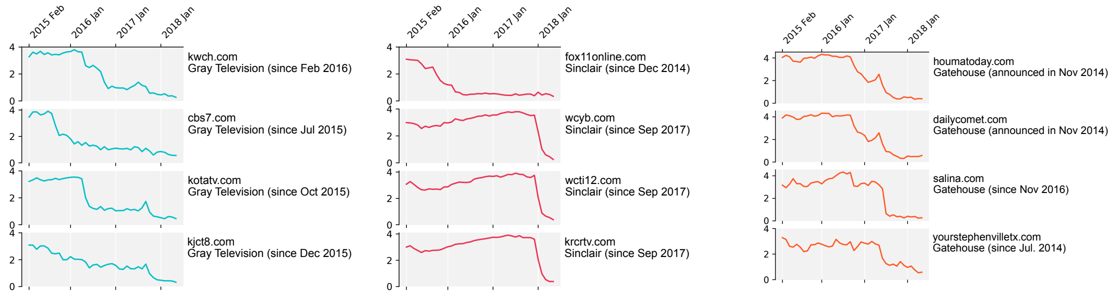


Figure 4.12 – Euclidean distance from channels to their respective affiliations, in latent space, over time. Affiliation positions are computed as the centroid of 3 seed sources (taken from Fig. 4.11 - (*right*)). Those patterns are extracted without any supervision: retrieved channels are the ones having undertaken the largest shift of content over the considered period (2015-2018). Channels acquisition dates are given for reference.

we apply a weighted Kernel Density Estimator (KDE), a non-parametric method for estimating the Probability Density Function from a set of samples, under weak smoothness assumptions. The objective is to detect areas containing a high density of sources with negative weights and locate their density peaks. We use the weights w_{s_i} as input of the estimator. The bandwidth selection is a function of the data's covariance, multiplied by a constant factor σ. [115] An example of the resulting density is presented in Fig. 4.11 - (*center*). Finally, local extrema are collected using a local minimum filter, a simple method routinely used in computer vision. The set of identified poles and the top-3 closest sources surrounding them is shown in Fig. 4.11 - (*right*).

Attractees: Having identified a set of attractors in latent space, the dual observation can be made: the identification of sources that experienced large movements in latent space toward any of the previously identified attraction poles. These are sources that have been strongly influenced by external forces, for example in the content consolidation phase after an acquisition. The detection of these phenomenon is done relative to a specific pole of attraction. In order to track the distance to a pole over time, we start with a set of seed sources. An obvious choice is to study the top-3 closest sources from the poles, detailed in Fig. 4.11 - (*right*). The ranking of sources having undergone a large shift can once again be made systematic. In particular, we rank sources according to the largest difference in distance to the pole between two consecutive time steps. The distance to the centroid of these sources yields the distance maps shown in Fig. 4.12 for the top-4 sources with the largest shifts.

4.11 Discussion

The structure of the news landscape is in a constant state of flux. It is often difficult to follow the evolution of its organizational structure and even more so to determine what influenced these changes. In this section, we discuss how the fluctuations of broadcast patterns can be informative about channels' organizational structure. We report that important changes in this structure are identifiable through abrupt shifts in content diffusion, and showcase the models' ability to systematically highlight this variability in the coverage space.

4.11.1 Observing the effects of ownership on the media landscape

The selection and diffusion of events by individual news channels could be influenced by a large number of factors, from obfuscated economic drivers to convoluted distribution schemes. Theoretically, the external pluralism assumption would prevent large-scale organizational changes in news outlets from inducing significant shifts in coverage. Our findings question the validity of this assumption.

We provide evidence that ownership can indeed exert its influence on the content being distributed downstream. The most distinct and recurring pattern pointing to this conclusion is the subsequent alignment of coverage patterns after an outlet's acquisition. Some examples are clearly observable in Fig. 4.10, such as the acquisition of 14 stations from the Bonten Media Group by Sinclair Broadcasting Group (SBG) in a deal completed on September 1st, 2017. [XVIII] Visible through the lens of a decrease in the average inter-source distance of the Sinclair stations, this consolidation of coverage can also be tracked in the embedding spaces' visualization in Fig. 4.10 - (*left*). This can be observed in Fig. 4.12 as well, albeit with a slight delay with a sharp decrease in the distance from channels like `wcyb.com`, `wcti12.com` or `krcrtv.com` to the center of the Sinclair attraction pole (#8 in Fig. 4.11).

We observe similar behaviors during others large-scale acquisitions, for example in the purchase of a group of assets from the Morris Publishing Group by GateHouse in August 2017. [XIX] Other observations of this phenomenon include the sudden increase in coverage similarity of Gatehouse-owned stations around April 2017 (see Fig. 4.12 - *right*). While not directly correlated to a specific merger or acquisition, these movements could hint at a company-wide content alignment campaign.

Such observations also support the convergence hypothesis. The visualization in Fig. 4.10 exemplifies this effect: many of the sources that are present in one of these group's media

[XVIII] https://tvnewscheck.com/article/103465/sinclair-buying-bonten-stations-for-240m/

[XIX] https://www.poynter.org/news/gatehouse-acquires-morris-publishings-11-daily-newspapers portfolios start out from vastly different regions in embedding space. This is visible in their high initial average cosine distance in Fig. 4.10 - (*right*) and their dispersed placement in Fig. 4.10 - (*left*). In the last frame however, these same sources form highly coherent, tight groups in embedding space - and in the visualization. Despite the fact that this case-study back-tracks the evolution of sources across time, explaining the density of the last frame, their convergence points to a unification of coverage patterns over time.

4.11.2 Detecting highly influential broadcast groups

News outlets present complex content distribution schemes, as is particularly visible in television broadcasting: the on-air content is produced by a wide range of affiliates, from well-known household names to in-house teams, [XX] being distributed through channels with another, often different set of owners. While the consolidation of broadcast material for economies of scale or investigative resources for economies of scope can be economically beneficial for the broadcaster, it is also potentially deceitful for a news consumer, as the exact origin of the broadcast content is not always known. By extension, the unique slant it carries in its selection of news is not clearly obvious. Not only does it carry is unique biases in terms of the way in which it covers the content, a topic not discussed in this work, but it also has the ability to over-emphasize or under-report certain events with little accountability.

This influence on coverage can be observed when interpreting the agglomeration dynamics highlighted by Fig. 4.11. Information sinks can be highlighted through the discovery of attraction poles in the embedding space.

Such sets of highly accretive sources, i.e. sources that draw other sources to align with their coverage, cluster neatly into large broadcast entities, some of which have been mentioned before. Fig. 4.11-(right) presents these groupings more exhaustively. The three large media groups chosen for analysis in Fig. 4.10 are present (*Gray Television Inc.*, *Sinclair Broadcasting Group* and *GateHouse Media Inc.*), along with several other large players in the American media landscape.

Previous studies [112] and [11] have studied differences in terms of the types of content covered in television and newspapers outlets, finding that TV stations cover proportionally more "global" news than newspapers. Television is traditionally thought to be more impacted by media consolidation for this reason: content is costly to produce, hence it makes sense for large entities to share their footage at scale. In our model, this should intuitively lead to the flagging of television conglomerates as the strongest attractors,

[XX] A study conducted by Pew Research in 2014 already demonstrated a steady decline in locally produced content, with 1 in 4 local news stations not producing any of their own content [92].

with high content similarities. However, we observe in Fig. 4.11 that all mediums are represented and impacted by the convergence phenomenon. This could hint to the effect a convergence of mediums can have on the media landscape, with the efficacy co-ownership regulations being jeopardized by the all-encompassing nature of online content delivery.

4.11.3 Interpretation of the temporal consistency

None of this qualitative analysis would be possible without a temporal consistency constraint on the embedding space. Without such stability, the model could take advantage of an unnecessarily large number of degrees of freedom to align sources. In consequence, it would converge to very different solutions from one epoch to another. Due to the stochastic nature of the procedure, coverage changes would be rendered indistinguishable from optimization artifacts (see Fig. 4.4). By penalizing sources that deviate from their previous positions, only significant coverage changes can force a source to migrate to a different region in space. In other words, in order to provoke a displacement, the channels' coverage should differ enough from the previous time step to outweigh the temporal constraint. If this condition is met, the source will converge towards a different neighborhood that better fits its coverage patterns, typically getting closer to similar channels.

This variability in time can be tuned through the regularization parameter λ_T, as detailed in Section 6.5, providing a way to highlight more global dynamics - in the case of strong regularization - or more individual variations - with weak regularization. We also observe that the constraint provides predictive gains. This can be explained by the accumulation of knowledge about sources over time. This last hypothesis is corroborated by the pattern observed in Fig. 4.9, in which the accuracy reaches its maximum after the first few epochs before stabilizing until the end of the considered period.

4.12 Conclusion

In the first part of this chapter, we studied the presence and nature of selection biases in static news datasets. By treating the event selection performed by news channels as a collaborative filtering task, we reported distinct and interpretable communities of news sources by learning from their coverage alone. The learned representations shed light on many real-world relationships between news entities, which in turn influence the coverage of the news by these sources. Notably, we detected geographic dependencies, conserving even regional links, as well as same-medium sources without the use of side-information. We also report the ability to extract non-trivial relationships, such as affiliations, broadcast syndications, and even the inclusion of sources in a corporate network. The identification of these non-obvious structures is an important step towards the transparency needed to restore public trust in the reporting process. We further leveraged the learned representations to propose a re-ranking procedure producing more diverse and egalitarian coverage of the news which preserves a larger proportion of the most discussed events compared to a simple selection of highly active sources.

In the second part of this chapter, we proposed an approach to keep the embedding space of the model consistent over successive time slices of a news dataset. We show that this dynamic embedding model can outperform static approaches thanks to its ability to propagate knowledge obtained from former time slices to the current prediction step. We present this model as a framework in which to reason about the evolution of the media landscape, and that enables the analysis and the visualization of important shifts in the media ecosystem, at a large scale but also the individual source level. We demonstrate the potential of the method on several channel acquisition campaigns. We show drastic post-acquisition content alignment in channels belonging to large, well-known broadcast conglomerates. This corroborates the hypothesis of deep consolidation of broadcast material inside news networks. Finally, we automate this investigative process and explore several strategies to systemically identify abrupt variations in the news ecosystem, fingerprints of sharp changes in media programming.

5 Personalized Matching

5.1 Introduction

At its inception, the economy is assumed to have been barter-based [21]. Money later appeared as a medium of exchange and a measure of value, making the pricing of assets an easier task, and shaping the economic practices of today. With the advent of widespread digital communication, barter has re-emerged into the lives of 21st century consumers [43]. The idea on which this revived economic model rests is that of extending the lifetime of goods, in order for them to serve the purposes of multiple owners, or to give users access to obscure or difficult-to-obtain items. Numerous platforms are dedicated to swapping items of various categories, such as **swapacd.com**, **swapadvd.com**, **readitswapit.co.uk**, **bookmooch.com**, etc.

However, the aforementioned platforms are remarkably ad-hoc and lack mechanisms to recommend trades,[I] requiring that users manually search for compatible trading partners. Since the main prerequisite for barter is a double coincidence of 'wants' (i.e., that both parties desire each other's goods at the same time) such an endeavor becomes challenging. Given the recent shift towards green practices, a category in which bartering naturally fits, this problem presents high potential in terms of improving consumer experience. However, little research has been done on methods for recommending trades within an online bartering platform.

In order to build a recommender system for bartering platforms, eligible trading partners need to be found within the user base. Each platform user has a public 'wish list' comprising items they wish to acquire, and a public 'give-away list,' containing items to be given away in exchange for the desired ones. Initial work done on the problem [2, 1, 119] proposes 'strict' matching criteria between explicit user 'wants' and 'haves,' rendering a pair of users trade-compatible only if their reciprocal wish list/give-away list intersections are simultaneously non-empty. Surprisingly, we find that such an approach is highly ineffective on real-world datasets collected from online bartering platforms, as the double

[I] There exist some other platforms that employ a trade matching method, like **barterquest.com** [15], but their data was inaccessible. Their matching method, however, does not involve user preference modeling.

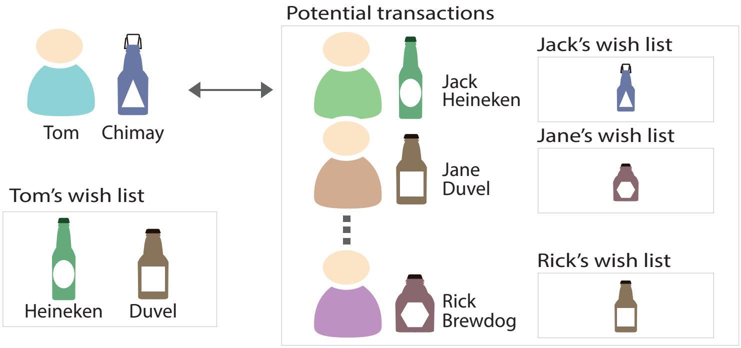


Figure 5.1 – Illustration of the problem setting in which a user (Tom) can exchange an item with owners of other items available on the platform (assuming the recipient has a reciprocal interest in the item being given away).

coincidence of 'wants' and 'haves' is very low, with fewer than 5% of users being eligible to receive recommendations. Moreover, real data reveals that the items being transacted are not always listed in users' wish lists prior to the transaction, suggesting the need for a system that can offer 'serendipity.' Such a system would be able to recommended items that a user likes, but which are not explicitly mentioned among their preferences, either because the user omitted them when creating the wish list, or because they are unaware of their existence. In summary, we find that existing approaches generally do not yield recommendations that are consistent with observed transactions, possibly suggesting that users are guided by criteria other than those revealed by wish list analysis.

In this chapter, we propose a model based on Matrix Factorization [62] that estimates cross-preferences between potential trade partners, or more precisely the strength of the reciprocal interest that two users have for each other's items. The end goal of our system is to discover, for each pair of candidate users, a pair of items that are most likely to be exchanged between them; swap recommendations are then made by computing the sorted list of partner-item combinations in order of reciprocal interest.

We build an initial model following traditional matrix factorization approaches, which we then extend by incorporating social and temporal dynamics, as we find that users develop trust in trading partners through repeated transactions, and they tend to trade in bursts of repeated activity. In order to capture these effects, we propose a model that is both socially- and temporally- aware, showing substantial improvements over previous matching-based approaches and 'vanilla' matrix factorization.

Another contribution of our work is the introduction of three large scale real-world datasets, composed not only of wish lists and give-away lists, but also of actual transaction histories. This allows us to qualitatively evaluate our approaches, by testing how well they rank transactions which have actually taken place against others that have not. This contribution is very important, as the user behavior revealed by the data is quite different from what has previously been assumed about bartering platforms.

We test the quality of the produced recommendations against the ground truth of the collected bartering histories, a form of evaluation that has surprisingly been missing in previous works on bartering [2, 1, 119, 88, 69]. We compare against a state-of-the-art item exchange method [119], and discuss its shortcomings on concrete examples of real-world bartering datasets. Our approach deals with several drawbacks of previous methods, by tackling the problem in a more flexible way through the use of user preference modeling, rather than relying on the incomplete truth provided in users' wish lists. This technique allows us to rank all the swap opportunities that a user has in the system, thus providing more choice, as well as serendipity.

5.2 Related work

The most closely related topics of related work to ours are (a) those that study bartering and exchange in general, and (b) those that model the latent preferences of users toward items. We discuss each in detail below.

Early works on optimal barter exchange strategies. The study of algorithms for exchange markets [4] was, at its inception, inspired by the kidney exchange problem [105, 104]. In order to improve the number of patients receiving kidneys despite having incompatible living donors, algorithms have been developed to determine cross compatible patient-donor pairs from the regional pool of transplant cases. The problem is solved by Roth *et al.* [105], using the The Top Trading Cycles and Chains mechanism. Another relevant work is that of Haddawy *et al.* [42], as it solves the problem of determining a balanced matching of buyers and sellers in the context of barter trade exchanges. The trades are managed by an intermediary, and parties are matched based on supply and demand information, as well as their credit in terms of a private label currency. The problem is modeled as a minimum cost circulation on a network. Lastly, the work of Mathieu [77] tackles the problem of finding bartering rings in an e-marketplace, based on similarity matching of *seek* and *offer* queries, expressed as weighted trees.

The Circular Single-item Exchange Model (CSEM). Determining exchange cycles in a bartering network is a more general problem than that of pairwise kidney exchange. As opposed to people receiving and donating one item (a kidney), in a traditional exchange market users have multiple items to give away, and potentially multiple incoming items. Abassi *et al.* [2, 3] model this setting as a directed graph, where nodes represent users and edges are labeled with item identifiers. The edge labels are determined by the wish lists and give-away lists of the users. A directed cycle in this graph represents a potential transaction (where each user gives away the item to the subsequent node, while receiving another item from the preceding node).

The Binary Value Exchange Model (BVEM). A different perspective is taken by Su *et al.* [119], who solve the item exchange problem for cycles of length two (i.e., swaps). The system is designed to be used in competitive online environments such as online games with a hefty real-time update schedule. Moreover, each item is associated with a user-defined price, and the value to be optimized is the maximum gain of each user.

Matrix Factorization (MF) is a popular technique in recommender systems. **MF** estimates unobserved user preferences from a sparse interaction matrix $R \in \mathbb{R}^{|U| \times |I|}$, where U is the user set and I is the item set, which is low-rank approximated [52]. **MF** techniques project every user and every item into a common low-dimensional space, such that their dot product approximates the observed interactions, i.e., the 'compatibility' between a user and an item.

Later, when considering social relationships and temporal dynamics, we mainly build upon established ideas that extend **MF** to incorporate social regularization [71], and temporal dynamics in recommender systems [60].

Bayesian Personalized Ranking (BPR). Bayesian Personalized Ranking is a *pairwise* optimization procedure proposed by Rendle *et al.* [97], that directly optimizes a ranking measure (AUC). This technique naturally deals with implicit feedback, as it only considers 'positive' user-item interactions, while not differentiating between negative observations and missing values. The intuition here is essentially that users prefer items they have observed over the ones they have not. This pairwise optimization technique can be used in conjunction with various model classes, such as Matrix Factorization, or Adaptive k-Nearest-Neighbors [97].

5.2.1 Key Differences

Our work is related to **BPR** in the sense that we also aim to discover latent factors in order to optimize ranked lists of recommendations in terms of the AUC. In terms of exchange models, **BVEM** comes the closest to the present problem formulation. Firstly, **BVEM** is also concerned with recommending swaps, as opposed to **CSEM** where exchange cycles are longer. Secondly, **BVEM** recommends a list of swaps for each user, sorted decreasingly according to user gain in terms of item price. Or approach also produces recommendations as sorted lists, but uses a different scoring function, which is based on the estimated user preference. The key difference, however, is that **BVEM**'s recommendations are still based on exact matches between the wish lists and the give-away lists of the two partners, which is too restrictive on real datasets, as we show in the following section.

Platform	user count	item count	transaction count	% of users w/ at least one swapping partner
Bookmooch	84,989	2,098,699	148,755	0.2%
Ratebeer	2,215	35,815	125,665	65.9%
/r/gameswap	9,888	3,470	2,008	-
Swapacd	4,516	244,893	-	0.5%
Swapadvd	7,562	91,241	-	0%
ReaditSwapit	33,151	94,399	-	4.2%

Table 5.1 – Statistics for our collected platform data. The rightmost column shows the percentage of users that have at least one trading opportunity, according to their public lists. On most platforms, users have very few eligible trading partners.

5.3 Data Analysis

To evaluate our approach, we first conducted an empirical study by collecting the following datasets:

1. **Swapacd**[II] is a CD exchange platform.
2. **Swapadvd**[III] is a DVD exchange platform.
3. **ReaditSwapit**[IV] is a book exchange platform.
4. **Bookmooch**[V] is a book exchange platform.
5. **Ratebeer**[VI] is a beer exchange (and rating) platform.
6. **/r/gameswap**[VII] is a self-organized subreddit made for users to exchange video games.

Basic statistics of the datasets are shown in Table 5.1. Of the six datasets, 4, 5 and 6 also have transaction histories, and are thus our main focus throughout the paper.

The platforms have object type specificity. It is worth noting that each platform is oriented towards only one kind of item (books, games, beers). This suggests that the value of items does not vary significantly, as particularly valuable objects (e.g. Leonardo da Vinci's Codex Hammer) would be very unlikely to be listed for exchange. Therefore, a coarse approximation could say that, with few exceptions, most objects on such platforms

[II] http://www.swapacd.com

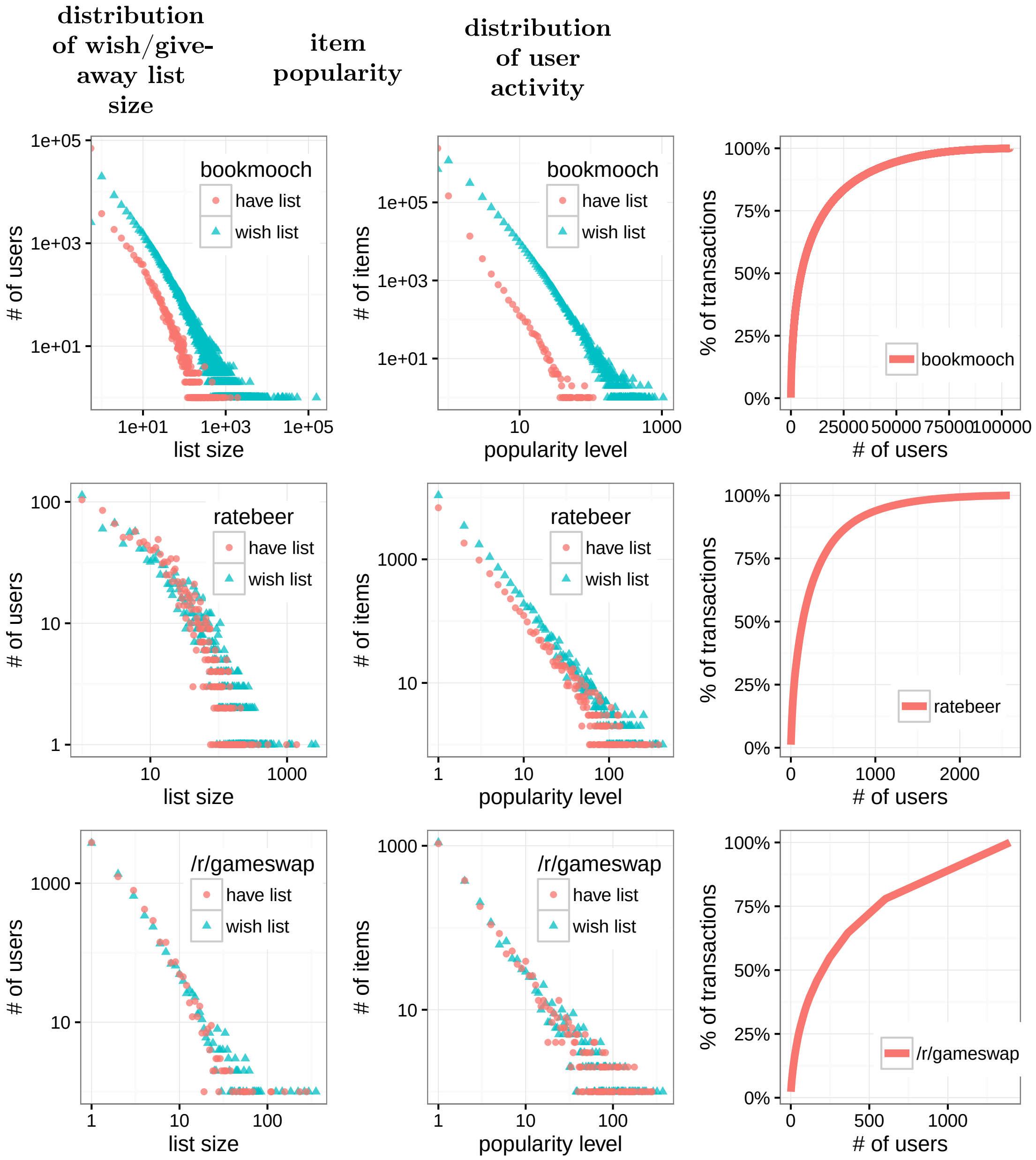
[III] http://www.swapadvd.com

[IV] http://www.readitswapit.co.uk

[V] http://www.bookmooch.com

[VI] http://www.ratebeer.com

[VII] http://www.reddit.com/r/gameswap



distribution of wish/give-away list size

item popularity

distribution of user activity

Figure 5.2 – The distribution of item list sizes and item popularity approximately follows a power-law for the three considered platforms (top: **Bookmooch**; middle: **Ratebeer**; bottom: **/r/gameswap**). The CDF plots on the right show user activity in terms of the number of swap transactions each user participates in. The presence of power users, who account for a majority of the transactions, is apparent on all three platforms.

are of comparable value. There exist, however, bartering platforms where exchange is possible between items of different categories (e.g., a book for a microwave). We have come across a number of such platforms during our research, for example **www.swapz.co.uk** and **www.barteronly.com**, but their data was not accessible.

All the datasets were obtained through crawling the corresponding websites, except for **Bookmooch**, which constantly exposes an updated snapshot of its database. For **Ratebeer** and **/r/gameswap**, the transactional information was extracted from textual submissions of users, containing information about their completed transactions (i.e., the transacting parties, along with the beers and games they exchanged, respectively).[VIII]

The datasets are long-tailed. The distribution of the size of users' wish lists and give-away lists, as well as the popularity of each item (both in terms of how many users own it and of how many users desire it), are depicted in Fig. 5.2, as are the Cumulative Distribution Functions for the number of transactions that each user has taken part in (right column). These quantities appear to approximately follow power-laws, and suggest the presence of 'power users' [80] on the platforms. **Swapacd**, **Swapadvd** and **ReaditSwapit**, yield similar results, but are omitted due to space considerations.

The fact that these quantities follow power-laws is not surprising, but it partly explains our following observation that there is little coincidence between 'haves' and 'wants' among real trades—while the platforms have many users, there are long tails of rare items among small wish and give-away lists.

Pairs of 'eligible' swapping partners are very scarce. Two users are eligible swapping partners if each of them desires one or more items in the other's give-away list. The percentage of users having at least one eligible swapping partner is summarized in Table 5.1, for the considered snapshots of the swapping platforms. The table contains no entry for **/r/gameswap**, because the organization of the information on the threads rendered us unable to extract an exact snapshot of *all* the users' 'wants' and 'haves' at a fixed point in time.[IX]

One may note in Table 5.1 that the shortage of eligible swapping partners proves to be a problem even for large user bases, like that of **Bookmooch**. An exception to the rule is seen for **Ratebeer**, which may be explained by the fact that the platform is several years older than the others, and has a global community of users.

An implication of the aforementioned scarcity is that approaches that match users exclusively according to wish list and give-away list content, such as **CSEM** [2] and **BVEM** [119], do not perform well on this data, yielding too few (or zero) recommendations per user. In practice, as we show next, many trades take place between users who are not strictly 'eligible.'

[VIII] All the datasets are available at *http://swapit.github.io/*.

[IX] For example, if u_j posts their item lists on the thread at time t, and u_k does the same d days later, one may be wrong to assume that u_j's preferences have not changed in the meantime (they may have exchanged items, rendering the lists stale, because the system has no way of updating them). Therefore, a snapshot taken at time $t + d$ can only include the subset of users who posted on the thread at that time.

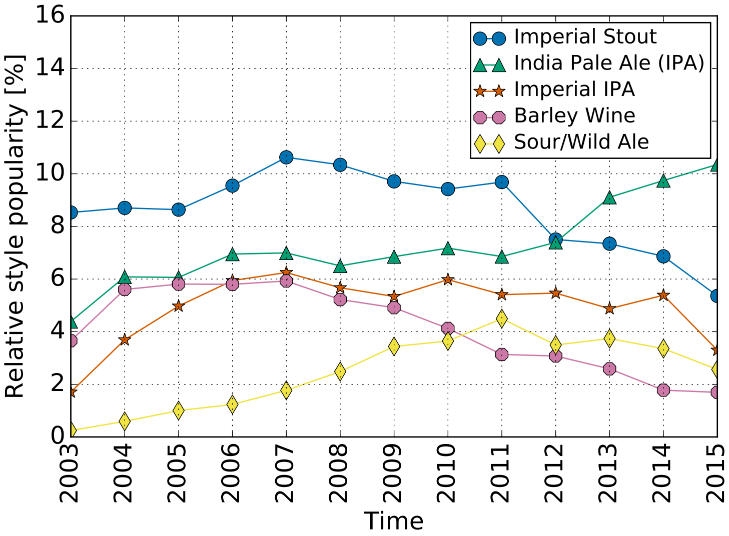


Figure 5.3 – Evolution in popularity of the top-5 beer styles (using the **Ratebeer** dataset).

Preferences are not exhaustively listed in wish lists. Since **Bookmooch** surfaces weekly database snapshots, it is possible to evaluate the extent to which books that a user receives while trading on the platform were present in their wish lists *before* the transaction took place. Using these snapshots from **Bookmooch**, we computed this percentage to be, on average, 33.2% per user. This directly implies the need for a recommender which can infer a user's preferences toward items which they may not be aware of (or did not explicitly declare in their wish list), and may instead discover serendipitously. Critically, this issue is one not addressed in previous work.

Users trade multiple times with the same peer. The intuition that pairs of users who successfully traded in the past are likely to trade again is supported by an observation we make about transaction events. On average, a pair of users trades 1.35 times on **Bookmooch**, 3.56 times on **Ratebeer** and 1.19 times on **/r/gameswap**. This suggests that social ties may play an important role in determining the trading partner of a user, and that pairs of users who successfully function as trading partners are likely to trade again in the future.

Item popularity and trade frequency are time dependent. A highly dynamic environment such as that of a bartering platform is subject to time-dependent trends. In Fig. 5.3 we observe how beer styles evolve in popularity, measured as their trade frequency over time. For example, we see how IPAs steadily gain popularity, surpassing all the other styles by the year 2013, whereas before that Imperial Stouts were the most popular among the beer styles being traded.

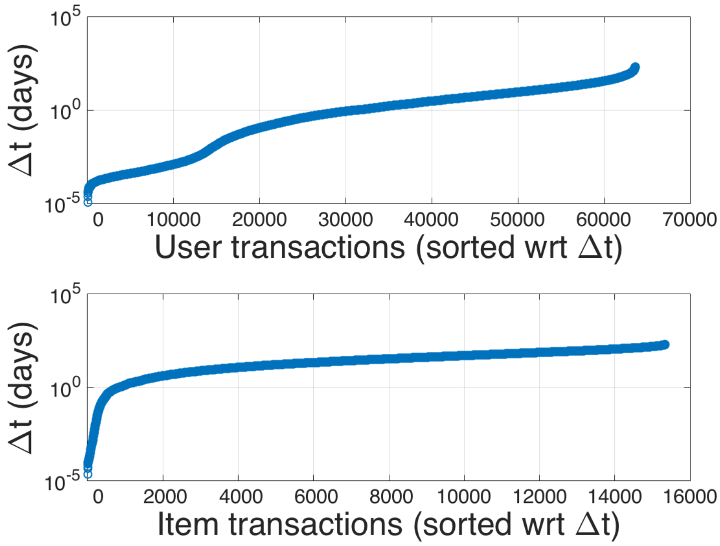


Figure 5.4 – Cumulative frequency plot of the transactions performed on Bookmooch. Note that while there is a core of power users who perform multiple transactions per day (i.e., $\Delta t < 10^0$), most of the items are swapped infrequently (i.e., a few times per year).

Fig. 5.4, on the other hand, suggests a different type of time-dependent behavior. Every point corresponds to a transaction, focused on either the users transacting the item (top plot), or the item being transacted (bottom plot). The Y-value is given by the number of days passed since the particular user previously transacted (respectively the number of days passed since the last time the same item was transacted on the platform). Fig. 5.4 shows the existence of items and users more actively involved in trading, as opposed to others with less frequent interaction.

5.3.1 Limitations of Previous Work

The main disadvantages of the previously mentioned approaches come from the restrictions on which they rely. For example the Circular Single-item Exchange Model (**CSEM**) [2] requires that a user and their item be recommended to only one user at a time; this is disadvantageous as it reduces the probability of an item being traded. Such a restriction would accentuate the scarcity of eligible swapping partners, as a recommendation of an item to a user would be further conditioned on whether the same item has already been recommended to somebody else. Ideally, an item should be recommended to as many users as are potentially interested.

While the Binary Value Exchange Model (**BVEM**) [119] more realistically models the trade recommendation problem, in order to be tractable it requires an assumption that the length of the item lists be bounded to some small number (say, less than 50). This is

	$\beta = 0.6$	$\beta = 0.7$	$\beta = 0.8$	$\beta = 0.9$
Total recommendations	113	111	110	110
Distinct users	155	152	150	150

Table 5.2 – Number of **BVEM** recommendations for various values of the price matching parameter β. Note that each recommendation is made to two distinct users.

contradicted by our findings across all collected datasets, showing that the size of the item lists varies between one and several thousand items, approximately following a power-law. This assumption, however, only affects time performance rather than the quality of the recommendations.

The most important drawback—common to both previous approaches—is that they only consider explicit user preferences, which are shown to be far from complete. Neither **BVEM** [119], nor **CSEM** [2] make use of the implicit preference information encoded in users' transaction histories, but base their recommendations solely on the items which a user explicitly lists in their wish list. Not only does this prevent serendipitous item discovery (which makes up for a majority of trades in real transaction histories), but it also implies a much too rigid definition of 'eligible swapping partners,' yielding very few trade opportunities (as seen in Table 5.1). A system aiming to pair users based on matching wish lists and give-away lists can only make recommendations to an extremely limited number of users in these datasets.

To support the latter statement, we tested the performance of **BVEM** [119] on the **Bookmooch** dataset, as it is the only one providing item pricing information as required by the model. Table 5.2 summarizes the number of recommendations produced using this approach for a dataset of 84,989 users, based on a snapshot from September 2015. Extremely few users (a maximum of 155) receive recommendations under **BVEM**, due to the scarcity in the coincidence of 'wants' (see Table 5.1). Having recorded the trade history for the four months following the September snapshot, we observed that 3,864 distinct users received books via trades, a much higher number than that of users being made recommendations. Also, the total number of recommendations made system-wide is very low (with a maximum of 113), compared to the size of the user base; this effect would be even more drastic on **Swapadvd**, where **BVEM** would fail to make any recommendations, as there are no pairs of eligible swapping partners.

Moreover, we assessed the predictive power of the recommendations produced by **BVEM**, with respect to the transactions recorded during the four months following the database snapshot. None of the users who were predicted to interact based on **BVEM**'s matching have actually exchanged any item in the concerned time frame, therefore yielding zero recall.

CSEM with cycles of length two [2] and **BVEM** [119] are particular instances of a more general approach, where matching users depends on wish list and give-away list intersections, and the problem boils down to finding the maximum matching on a bipartite graph in which nodes represent users, and edges exist between eligible swapping partners. Edges can further be weighed according to various quantities to be optimized platform-wide (e.g. an aggregate reciprocal preference score for the involved users). Computing a maximum weight matching on this graph retrieves an optimal set of user pairs with respect to the previously established criterion. Such an approach can at best produce a number of recommendations equal to the number of users having at least one eligible swapping partner (see Table 5.1). Also, a user may receive at most as many recommendations as they have eligible swapping partners. Applying *any such technique* to the aforementioned datasets would yield few recommendations, and generally few options to any user receiving a recommendation.

Our observations point to the need for a more flexible approach, which better models user preferences, and is capable of surfacing recommendations to a larger fraction of users.

5.4 Model

5.4.1 Problem Definition and Notation

The setting of the bartering platforms presently considered is described by a set of users $U = \{u_1, u_2, \dots u_m\}$, and a set of items $I = \{i_1, i_2, \dots i_n\}$ known at any time t. Each user u_j has a *wish list* W_j and a *give-away list* G_j, both of which are available for all members to see. W_j is a subset of I containing items which u_j wishes to obtain, while G_j is a subset of I with items to be given away by u_j.

The key difference between bartering compared to traditional recommendation is that users are both suppliers and consumers. Thus, every item i_k has an associated owner u_j, in whose give-away list G_j it appears, and a set of users who desire it. Note that there might be items that are listed as give-aways but are not wished for by any user, just as there may be items that are wished for but not available. Also, each user u_j has an associated history of transactions, which we will denote by H_j. Since transactions are bidirectional, we define H_j^g to contain all the items that u_j gives away in transactions, and similarly, H_j^r to be all those which u_j receives via transactions:

$$H_j^g = \{(u_l,\, i_{j\to l}) \,|\, u_j \text{ gives item } i_{j\to l} \text{ to } u_l\}$$
$$H_j^r = \{(u_l,\, i_{j\leftarrow l}) \,|\, u_j \text{ receives item } i_{j\leftarrow l} \text{ from } u_l\}$$

In the following sections, we use the notation $\hat{y}_{u_j i_k}$ to denote the estimate of user u_j's preference for item i_k.

5.4.2 Modeling Basic User Preferences

The first goal of our model is to estimate a user's preference for an individual item. As our data contains wish lists and past transactions, we use them as implicit feedback signals [84] when building the preference model.

Following the approach proposed by Hu *et al.* [52], the user-item interaction matrix R is built based on implicit feedback signals as follows:

$$r_{u_j i_k} = \begin{cases} 1, \text{ if } i_k \in W_j, \text{ or } (*, i_k) \in H_j^r \\ 0, \text{ otherwise} \end{cases}$$

(i.e., 1 iff the item belongs to u_j's wish list, or there exists a past transaction in which u_j receives i_k).

We want to model the preference $\hat{y}_{u_j,i_k}$ that a user u_j exhibits toward item i_k. We start with a low-rank model of users and items:

$$\hat{y}_{u_j,i_k} = p_{u_j}^T q_{i_k} \tag{5.1}$$

where p_{u_j} and q_{i_k} are vectors describing the 'preferences' of the user u_j and the 'properties' of the item i_k. Although we defer details of our optimization procedure until Section 5.5.1, our goal is that $\hat{y}_{u_j,i_k}$ should be large if and only if the user is 'compatible' with the item.

As this formulation of the optimization problem is the simplest one, we will reference it as a baseline during our experiments.

5.4.3 Incorporating Social Bias

The effect of incorporating social information into collaborative filtering models has been shown to improve prediction accuracy and alleviate data sparsity (e.g., Ma et al. [71], [29]). As discussed in Section 6.3, users tend to repeatedly trade with a selected subgroup of peers on the observed bartering platforms, suggesting that their choices have a strong social (or simply trust) component. This further points to the fact that a plain low-rank decomposition of the interaction matrix R as in Section 5.4.2 cannot fully capture the dynamics of users' behavior. Thus, we incorporate a *directed social bias* $S \in \mathbb{R}^{|U| \times |U|}$ as part of the predictor where $s_{u_j u_l}$ models the bias for user u_j toward user u_l. The extended model to be optimized is described as follows:

$$\hat{y}_{u_j,u_l,i_k} = p_{u_j}^T q_{i_k} + s_{u_j u_l} \tag{5.2}$$

where the preference score $\hat{y}_{u_j,u_l,i_k}$ is now in terms of an item i_k *and* a user u_l with whom the item is being traded. Note that this relation is *asymmetric*, as the bias from u_j toward

u_l may differ from the bias from u_l towards u_j.

Also note that the optimization remains tractable after adding the social bias term, as in practice users trade with a limited number of peers, rendering the matrix S sparse.

5.4.4 Adding Temporal Dynamics

User tastes may shift over time, or vary periodically (for example, following certain holidays). Temporal dynamics have previously been exploited in collaborative filtering settings, e.g. to build temporally-aware models of preferences on Netflix [60]. Motivated by the observations made in Section 6.3, we extend our model from Eqn. 5.2 to capture the temporal dynamics of bartering platforms.

There are two key aspects we wish to consider. Firstly, the model should capture the activity 'density' of users. For example, a user with high activity level at time t is probably more likely to trade at time $t + \epsilon$ than a user who hasn't traded during the same period. Secondly, the model should capture seasonality, i.e., that certain items tend to be more frequently requested during specific periods of the year (e.g. Christmas beers at Christmas). The activity level of users and the frequency with which items are traded can be observed by analyzing the timestamps available in the transaction histories of all three datasets.

In order to work with a smooth function, we approximate the trade time density using a *Kernel Density Estimator* [110]. **KDE** is a non-parametric method for estimating the Probability Density Function from a set of i.i.d. samples, under weak smoothness assumptions. Eqn. 5.3 represents such a density estimator for the sample set $\{x_1, x_2, \dots x_n\}$:

$$\delta(x; \bar{x}) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{5.3}$$

For our purposes, we set $K(x) = \frac{1}{\sqrt{2\pi}} exp(-\frac{1}{2}x^2)$, i.e., the Gaussian kernel [110, 74]. Parameter h represents the bandwidth, and can be set according to Silverman's rule of thumb [116], i.e., $h \approx 1.06\hat{\sigma}n^{-1/5}$, where $\hat{\sigma}$ is the standard deviation of the samples. This quantity is then incorporated into our predictor by modulating it with a parameter per item τ_{i_k} for item i_k, and a parameter τ_{u_j} for user u_j, which are to be learned. Eqn. 5.4 represents our final model, which includes the social bias and the temporal terms:

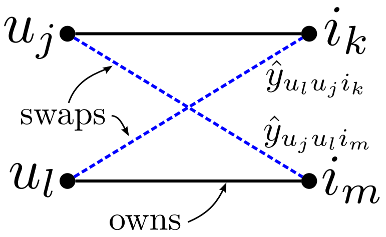
$$\hat{y}_{u_j,u_l,i_k} = p_{u_j}^T q_{i_k} + \overbrace{s_{u_j u_l}}^{\text{social bias}} + \underbrace{\tau_{u_j}\delta(t; \bar{t}_{u_j}) + \tau_{i_k}\delta(t; \bar{t}_{i_k})}_{\text{temporal dynamics}} \tag{5.4}$$

where t is the timestamp of the transaction sample, and $\bar{t}_{u_j}$ and $\bar{t}_{i_k}$ are time points of activities for trades involving user u_j, and item i_k (respectively).

5.4.5 Modeling Reciprocal Interest

In a bartering context, recommendations need to be addressed not just to a user, but to a user *and* each of the items that they own. This reflects the idea that for each owned item, a user may have different swapping opportunities. It follows that for each $(u_j,\, i_k)$, where i_k is owned by u_j, we will generate a ranking of all possible pairs (u_l, i_m), where u_l owns i_m, according to some preference score. It is worth stressing that in this context, u_j will have a different preference score for item i_m owned by u_l, than for the same item owned by $u_n \neq u_l$. In order to model the bidirectionality of user preferences within each pair of potential swapping partners, we will aggregate the preference of u_j for i_m and the preference of u_l for i_k into one meaningful score. Note that we do not impose the constraint that i_m should be in W_j, nor that i_k belongs to W_l, so as to allow serendipity.

We evaluate potential transactions by defining an aggregate score given by a function $f : \mathbb{R}^2 \to \mathbb{R}$. In the following, we consider u_j to be the owner of i_k, and u_l to be the owner of i_m. We want to evaluate the strength of the cross preference within the (u_j, u_l) pair, with respect to items i_m and i_k (respectively). We therefore aggregate interest scores into a single value, which quantifies the 'strength' of the pair's potential interaction. For this purpose we use the arithmetic mean:



$$\begin{aligned} \hat{y}_{u_j, i_m, u_l, i_k} &= f(\hat{y}_{u_j u_l i_m}, \hat{y}_{u_l u_j i_k}) \\ &= \frac{1}{2}(\hat{y}_{u_j u_l i_m} + \hat{y}_{u_l u_j i_k}) \end{aligned} \tag{5.5}$$

(the basic idea of the reciprocal interest model is depicted on the left). In this case, a strong preference from one user compensates for a potentially weaker one coming from the other. We also considered other aggregating functions, such as the harmonic mean [91], but found that the arithmetic mean was consistently the best performing one.

5.5 Experiments and Discussion

5.5.1 Parameter Learning

Since our input data consists of implicit preference signals, our methods' performance should be oriented towards correctly ranking items relative to each other, rather than accurately predicting missing values from the interaction matrix R. The BPR optimization

	Bookmooch	**/r/gameswap**	**Ratebeer**
(1) MF	0.758	0.790	0.824
(2) MF+B	0.798 (+2.0%)	0.842 (+5.19%)	0.892 (+6.79%)
(3) MF+B+S	0.849 (+9.15%)	0.863 (+7.31%)	0.962 (+13.84%)
(4) MF+B+T	0.938 (+18.06%)	0.890 (+9.99%)	0.969 (+14.55%)
(5) MF+B+S+T	**0.958 (+19.98%)**	**0.903 (+11.29%)**	**0.983 (+15.87%)**

Table 5.3 – Results of our approach in terms of the AUC (higher is better): The best performing method on each dataset is boldfaced. MF (1) stands for plain Matrix Factorization used as a baseline, B (2) stands for the bidirectional model, S (3) stands for the social bias term and T (4) stands for the temporal dynamics term.

technique introduced by Rendle *et al.* [97] is designed for this type of optimization problem. Following the notation of Rendle *et al.* [97] the update rules for this setting are defined as

$$\theta \leftarrow \theta + \alpha \cdot (\sigma(-\hat{x}_{u_j i_k i_m}) \frac{\partial \hat{x}_{u_j i_k i_m}}{\partial \theta} + \lambda_\theta \Omega'(\theta)), \tag{5.6}$$

where $\hat{x}_{u_j i_k i_m} = \hat{y}_{u_j i_k} - \hat{y}_{u_j i_m}$, and θ represents the set of parameters to be learned. $\Omega(\theta)$ denotes a regularizer, and in our case we opted for ℓ_2 regularization, i.e., $\Omega(\theta) = \|\Theta\|_2^2$.

The term $\hat{x}_{u_j i_k i_m}$ denotes the *difference* in preference score of user u_j for two items i_k and i_m. Should the difference be negative, the user is assumed to prefer i_m over i_k, and should it be positive the user is assumed to prefer i_k over i_m. In other words, the framework optimizes the fraction of times that the model ranks a traded item higher than a (randomly sampled) non-traded item, which approximates the AUC [97]. The update from Eqn. 6.5 is repeated with a large number of random samples until convergence.

The item preference terms $\hat{y}_{u_j i_k}$ and $\hat{y}_{u_j i_m}$ in the above can be adapted with any of the previously described models (Sections 5.4.2—5.4.5).

5.5.2 Evaluation Methodology

Our datasets contain *one-for-one* item exchanges between user pairs, directly from the transaction history. We express such transactions as quadruplets (u_j, i_k, u_l, i_m), where u_j owns i_k and u_l owns i_m, and define I^+ to be the set of all positive interactions extracted from the considered transactions:

$$\begin{aligned} I^+ = \{(u_j, i_k, u_l, i_m) | (u_l, i_m) \in H_j^r \wedge (u_j, i_m) \in H_l^g \wedge \\ (u_j, i_k) \in H_l^r \wedge (u_l, i_k) \in H_j^g \ \} \end{aligned}$$

Our evaluation set E consists of triplets (u_j, i_m, i_n), where $(u_j, *, *, i_m) \in I^+$ means that we have observed a positive signal from user u_j towards item i_m, while i_n is randomly chosen from the set of items which do not belong to either W_j or H_j^r, meaning that u_j did not express a preference towards it. A model that performs well should rank unseen items which received positive feedback from u_j (like i_m) higher than items with no observed interaction (like i_n). We formally define E below:

$$E = \left\{ (u_j, i_m, i_n) | (*, i_m) \in H_j^r \wedge i_n \notin W_j \wedge (*, i_n) \notin H_j^r \right\} .$$

To assess the effectiveness of our approach, we select the widely used metric *Area Under the Curve* (AUC) [113] as our measure of performance:

$$\text{AUC} = \frac{1}{|E|} \sum_{(u_j, i_m, i_n) \in E} \nVdash(\hat{y}_{u_j i_m} - \hat{y}_{u_j i_n}) = \frac{1}{|E|} \sum_{(u_j, i_m, i_n) \in E} \nVdash(2\hat{x}_{u_j i_m i_n}), \tag{5.7}$$

where $\nVdash$ is the Heaviside function (the latter formula uses the notation introduced in Section 5.5.1). Negative user-item pairs (u_j, i_n) are randomly sampled from a set of unobserved interactions for user u_j. This metric shows how well the model ranks items that the user has actually received from transactions that are withheld during training, versus items that the user has not interacted with, or does not explicitly desire.

Note that above we have expressed the AUC in terms of our simplest preference model ($\hat{y}_{u_j i_m}$), in order to avoid excessive notation. However, the above expression can be adapted to include any of the previously described models.

5.5.3 Experiments

Experimental Setup. Experiments were performed on a single machine running Matlab R2015b. Following the methodology proposed by Rendle *et. al* [97], the hyperparameters of all the described methods have been tuned based on the expected error estimated on a randomly drawn initial train/test split. To create the split, positive samples were randomly selected from I^+ for each user, and set aside for testing. The negative samples from the triplets of E were randomly sampled on the fly. Afterwards, the hyperparameters are kept constant during the testing phase, where a new train/test split is drawn at every round in the same fashion. The results in Table 5.3 are averaged over 5 rounds. We found that the optimal models have 40-dimensional latent factors for both **/r/gameswap** and **Ratebeer**, and 100-dimensional factors for **Bookmooch**.

All code is available at `http://swapit.github.io`

Results. Table 5.3 summarizes the performance of the various instances of our approach. On average, our method outperforms 'vanilla' matrix factorization by 15.71% across the

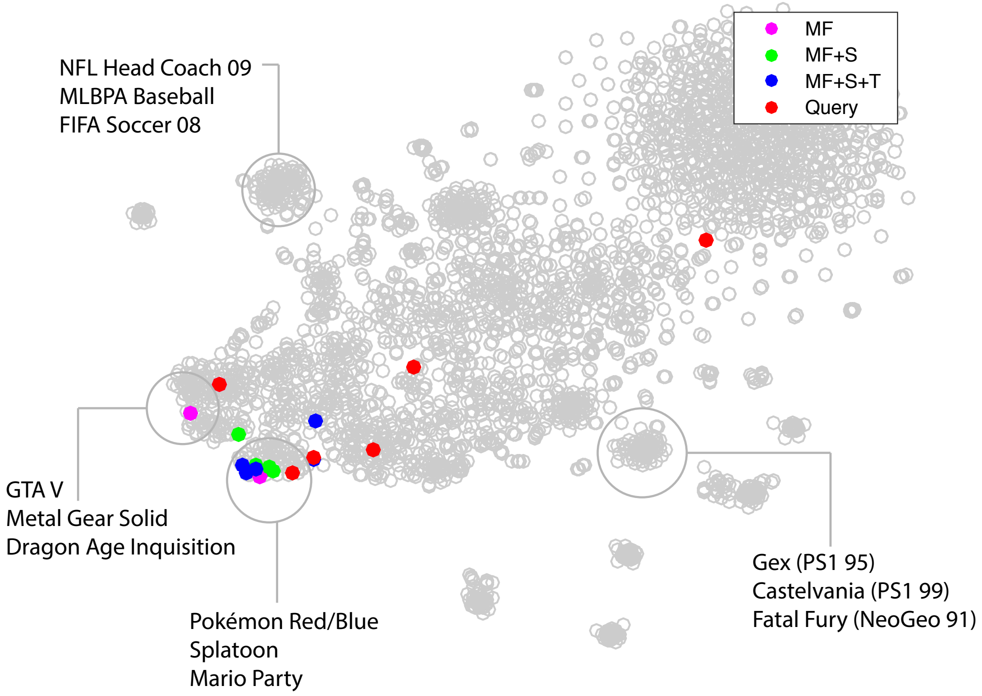


Figure 5.5 – t-SNE [126] embedding of items' latent factors from the */r/gameswap* dataset. Colored dots show the projection of recommendations in Table 5.4.

three datasets we consider. Each of the model extensions (bidirectionality, social bias, and temporal dynamics) makes a substantial contribution to the performance of our method, yielding cumulative performance gains of 4.66%, 5.44%, and 5.61% (respectively). AUC scores of the final model (**MF+B+S+T**) are above 0.9 on all three datasets.

In summary, a successful model for generating trade recommendations for online bartering requires flexibility in modeling users' preferences; approaches that are too 'rigid' (e.g., which use strict matching criteria) are unsuccessful, as discussed in Section 5.3.1. Beyond predicting users' preferences for items, a strong approach should also model the social and temporal dynamics at play. We further analyze our results and give examples below.

5.5.4 Discussion

The astute reader will notice that **BVEM** is missing from Table 5.3. Due to the requirements of this method, its application is only possible on the **Bookmooch** dataset, which is the only one of the three containing pricing information (see Section 5.3.1). More importantly, the **BVEM** approach does not produce a preference estimate of a user for a given item, which makes it impossible to evaluate its performance under the same metric as that used for our approach.

User's wish list
Super Mario World
Sonic Generation
Kirby's Dream Land
Metroid: Zero Mission
Super Mario 64
Mario Kart: Super Circuit
Sly 3: Honor Among Thieves

MF			MF + S			MF + S + T		
Recommendations (ranked)	#own	most recent activity	Recommendations (ranked)	owner activity	past trans.	Recommendations (ranked)	owner activity	past trans.
Sonic Generations	19	56 wks.	Kid Icarus	24 wks.	2	Fire Emblem	<1 wk.	0
Earthbound	14	22 wks.	Final Fantasy	24 wks.	2	Contra	<1 wk.	0
Super Mario Sunshine	26	22 wks.	Beyond: Two Souls	24 wks.	2	Monster Hunter	<1 wk.	1
Grand Theft Auto V	253	<1 wk.	Fire Emblem	92 wks.	1	Bayonetta 2	<1 wk.	1
Fire Emblem	28	<1 wk.	Paper Mario	92 wks.	1	Mario Kart 7	<1 wk.	1

Table 5.4 – An example of recommendations produced by the models from Table 5.3. **Left:** The set of items in a user's wish list; most are *Nintendo* console games. **Right:** Recommendations. All methods correctly identify related games, as depicted in Fig. 6.6. However, matrix factorization (**MF**) alone suggests a heterogeneous set of games belonging to multiple users; once social terms are added (**MF+S**), the system suggests trades with prior trading partners, but many of them have been inactive for some time; once the temporal term is added (**MF+S+T**), the system finally identifies relevant games, amongst active users, several of whom were prior trading partners.

Users' decision processes can be influenced by external factors, such as social ties and item availability. In such a scenario, the success of a trade cannot be fully explained by a low-rank decomposition (i.e., **MF**) that captures unilateral preferences of users toward items. Bidirectionality (**MF+B**) substantially improves the score over **MF**, and leads to similar improvements in combination with all other models. This suggests that a strong signal coming from one of the traders can compensate for a weaker signal coming from the other party. Note that the aggregation does not depend on the predictor and, thus, can be applied to various recommender systems techniques. Using the socially-aware model **MF+S** described in Section 5.4.3, our predictor is able to partially explain the observed variance by a social bias that users exhibit toward their prior trading partners. Considering the same item with different owners, the model will favor exchanges with users that already traded in the past. Proposing viable exchanges with this additional constraint consistently improved the score on our three datasets. The biggest improvement can be seen for **Ratebeer**, the platform with the highest percentage of recurrent trades between pairs of users. Beyond social biases, temporal bias (**MF+T**) acts as a gating function as it decreases the score of users/items that exhibit a long period of inactivity.

As with social bias, temporal bias can partially explain the score of a trade by the recent burst of activity of the considered user/item. Again, this addition in our model provides an improvement on all our datasets.

A recommendation example (for **/r/gameswap**) is shown in Table 5.4, illustrating the performance of our method with temporal and social constraints. These recommendations are also visualized in terms of a t-SNE embedding [126] in Fig. 6.6. Note that while even the simplest method (**MF**) already generates semantically meaningful trades, they are with unlikely trading partners due to lack of recent activity and social ties. These latter two features are important to ensure that plausible trades are recommended.

Advantages over previous approaches. Compared to existing methods, there are several factors that make our approach better suited to real-world bartering platforms.

First, our model outputs recommendations chosen from a ranked list of all swap opportunities available on the platform. This list contains not only items that a user mentioned explicitly in their wish list, but also items that are likely to be preferred by the user, based on preference modeling. **CSEM** [2] does not produce such a ranking, and while **BVEM** [119] aims to output a ranked list of recommendations, it fails due to the strictness of its assumptions, as previously described.

Second, our approach works even when the user base contains few compatible swap candidates, i.e., when few (or zero) users exist such that their wish lists and give-away lists are cross-compatible. Such a case occurs on the **Swapadvd** platform, as noted in Table 5.1. Applying **BVEM** on this dataset would yield no recommendations at all, as there are no two users with a bidirectional coincidence of 'wants'. This is also a plausible scenario for newly emerged platforms with a small user base, where eligible swapping partners are likely to be rare. We show that through preference modeling via **MF** techniques (applied to implicit user feedback), our method can recommend meaningful swap transactions from early on, in order to support platform activity and growth. This is also the reason why we can output long recommendation lists, as opposed to very few recommendations per user obtained under **BVEM**.

Finally, our model does not restrict the recommended trades to contain only items from the users' wish lists, consistent with our observation that only 33.2% of the items that users receive are explicitly listed. By modeling user preferences with the help of Matrix Factorization, our method allows us to estimate the users' preferences for items they have not explicitly desired, therefore allowing for potentially serendipitous recommendations.

5.6 Summary and Future Work

We introduced a new approach to recommending trades in the context of online bartering platforms. We presented several bartering datasets with transaction histories, covering books (**Bookmooch**), video games (**/r/gameswap**), and beers (**Ratebeer**). We analyzed their properties, revealing important reference points to be considered in the design of a bartering recommender. By considering real-world datasets, we found that previous approaches based on matching algorithms face severe performance limitations, due to the shortage of eligible swapping partners.

The approach we introduce builds upon well established recommender systems techniques, including matrix factorization, social regularization and temporally-aware models. Our design is data-driven, following observations that are consistent across the three aforementioned datasets, namely that i) successful trades require reciprocal user interest, ii) users develop 'trust' and trade according to their social ties, and iii) activity density varies over time, for both items and users. Our method is more flexible than existing approaches, due to the use of preference modeling, allowing coherent recommendations to be computed even in cases where few swapping partners are strictly 'eligible.' This allows users to receive potentially serendipitous item recommendations, due to the fact that we do not impose them to exclusively contain items from their wish lists.

As future work, we are interested in evaluating the performance of our approach on different scenarios in which reciprocal interest plays a considerable role, e.g. e-dating platforms, partner match-up in online video games, etc. Additionally, we hope to study the problem of bartering with heterogeneous item types (i.e. items with large price differences) and to explore more complex preference aggregation schemes for modeling the bidirectionality of interest between potential trade partners.

6 Personalized Collaboration

Many Web platforms rely on user collaboration to generate high-quality content: Wiki, Q&A communities, etc. Understanding and modeling the different collaborative behaviors is therefore critical. However, collaboration patterns are difficult to capture when the relationships between users are not directly observable, since they need to be inferred from user actions. In this chapter, we propose a solution to this problem by adopting a systemic view of collaboration. Rather than modeling users as independent actors in the system, we capture their coordinated actions with personalization methods which can, in turn, identify shared objectives and predict future user actions.

To validate our approach, we perform a study on a dataset comprising more than 16M user actions, recorded on the online collaborative sandbox Reddit *r/place*. Participants had access to a drawing canvas where they could change the color of one pixel at every fixed time interval. Users were not grouped in teams nor were given any specific goals, yet they organized themselves into a cohesive social fabric and collaborated to the creation of a multitude of artworks. Our contribution in this chapter is multi-fold: i) we perform an in-depth analysis of the Reddit *r/place* collaborative sandbox, extracting insights about its evolution over time; ii) we propose a predictive method that captures the latent structure of the emergent collaborative efforts; and iii) we show that our method provides an interpretable representation of the emergent social structure.

6.1 Introduction

Human beings left free to act according to their own will seem to often produce spontaneous order. This phenomenon has been observed in various contexts, ranging from city traffic flow to self-organizing economy. Those examples suggest that, even with different goals, a multitude of individuals interacting with one another often tend to avoid disorder, letting some underlying structure emerge. The Web is no exception: many Web initiatives rely on this principle, such as Question Answering platforms, collaborative code platforms, or even larger platforms such as Wikipedia. Characterizing the way people interact and organize themselves is a necessary step to understand users' behavior when little or no rules are enforced. Broadening our understanding of collaboration and self-organization

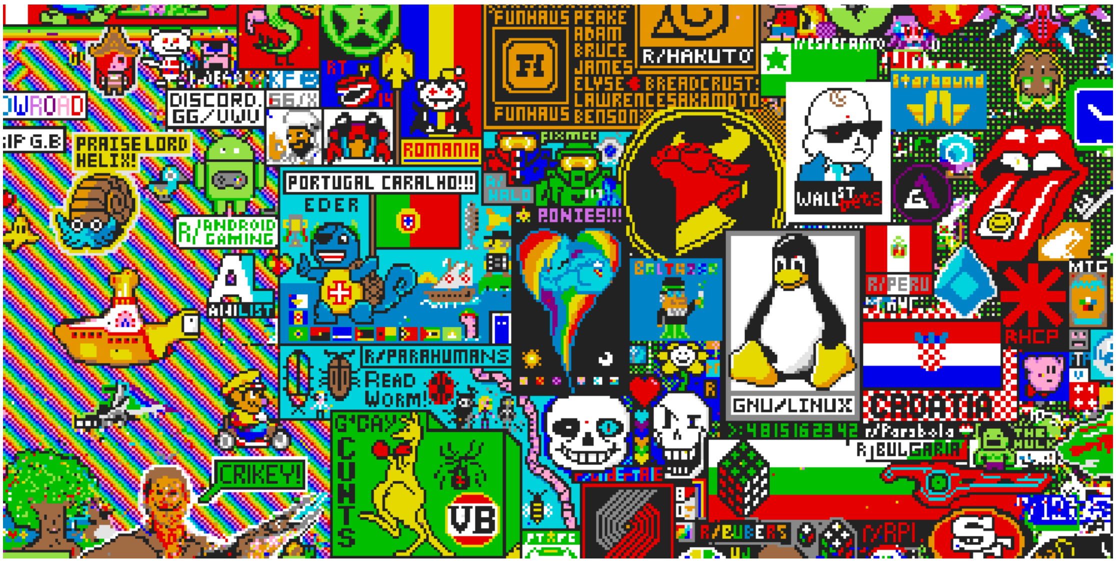


Figure 6.1 – A portion of the final canvas produced by the contributors of Reddit r/place

can, therefore, have a practical impact on the design of applications that are built for large populations of users, which underlines the importance of expanding our ensemble of methods to study such phenomena.

The study of collaboration is, however, being hampered by the inherent complexity of the task: the phenomenon is hard to be analyzed in realistic scenarios as it requires to capture complex interactions between actors. Therefore, the dynamics of such efforts have been mostly studied in scenarios where the relationships among users are directly observable. The problem of community detection in collaboration networks, for example, relies on an existing graph to segment it into subgroups of users. However, in practical scenarios, the relationships between users are often not explicit. In such cases, the analysis can rely on human judgment [72], however with severe limitations in terms of size of the environment that can be analyzed, both in terms of number of users and interactions. Therefore, we propose in this work a data-driven method to infer the user relationships from their interactions in the environment.

In this chapter, we propose a methodology to analyze user activities in a simple virtual environment, with the goal of characterizing collaboration patterns that emerged from it. In a second step, we propose a predictive method that captures the latent structure of user cooperation, and evaluate our approach on its capacity to predict future user actions. Our study focuses on a virtual sandbox in which users receive no particular directives, and are given the freedom to act as they want. In particular, we propose to investigate the behavioral patterns observed in Reddit[I] *r/place*[II], an online canvas where users were allowed to change the color of only one pixel at every fixed time interval (during a total of

[I] https://www.reddit.com

[II] https://www.reddit.com/r/place/

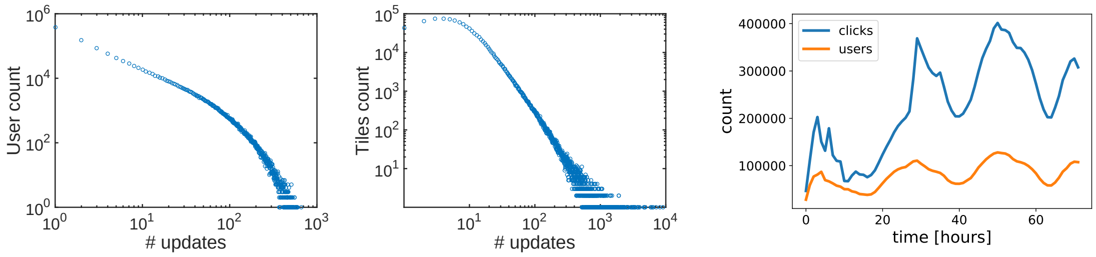


Figure 6.2 – Distribution of the number of updates for each user (left). Distribution of the number of updates per tile (middle). Evolution of the number of clicks and the number of unique users, computed per hour (right)

72 hours). Being part of Reddit, a discussion platform with more than 230M unique users per month, this collaborative project received a massive engagement of over a million users (see Fig. 6.1 for a glimpse of the final canvas).

In order to establish a predictive model of user behavior, we consider the sandbox as a complex social system, i.e., a system inherently difficult to model due to the large amount of interdependencies between its parts. Previous research in the field of Complexity Science [12] hypothesized that the nature of such systems is favorable to the emergence of global behaviors, arising from the local interactions of the actors. Following this evidence, we propose a model that assesses the likelihood of a user interaction by observing its social context. In other terms, we propose a predictive model that captures inter-user relationships instead of modeling independent behaviors.

6.2 Related Work

Networks of collaboration between scientists have been studied by Newman [83]. The authors argue that simple unweighted networks are unable to capture the strength of collaboration ties and propose a method to model the strength of collaboration by relying on the number of co-authored papers. The same author has later studied various properties of such networks [83]. Ramasco et al. [94] have studied collaboration networks from an evolving and self-organizing perspective. Behavioral experiments on the ability to solve problems collaboratively have been conducted by Kearns [57]. Online collaboration with different network topologies has been studied by Suri and Watts [121]. The exploration-exploitation trade-off in a collaborative problem solving task has been discussed by Mason and Watts [75]. Kittur and Kraut [58] studied various types of collaboration taking place between Wikipedia editors and measured the impact on quality of the resulting articles.

The study, as well as the interpretation of proximity data from a social perspective has been a prolific research area. Recent studies [111], have extracted social network properties from proximity sensor data. In particular, the authors propose a method to distinguish between strong and weak social ties, using the Bluetooth signal strength of users' cellphones. The authors observe that weak links, i.e. the interactions that have been observed less than once per day, have a lower probability of being observed at later times. Collaboration patterns between university students, collaborating in teams for their course assignments, have also been studied [25]. The authors consider the time spent in physical proximity, using university wifi logs, as a proxy to measure ties between students. Their analysis suggests that only strong ties matter in order to predict team performances. We also notice that the study of social properties from positional tracking is not limited to the human species, as a colony of ants as been recently tracked, at individual level, revealing complex hierarchical social structures [78].

The phenomenon of emergence has been studied in different domains, notably in the field of Complexity Science and in the context of agent based modeling. The term emergence has various definition across fields [64], alike complexity [36] from which emergence has been suggested to arise from. Emergence generally refers to system-wide behaviors that cannot be explained by the sum of individual behaviors. Moreover, means of modeling emergence are still subject to debate. Counting interaction between agents is, however, a widely used method to infer complex behaviors in a system [76].

The analysis of virtual behavior has been suggested as being a valid proxy for the study of real-life behaviors. High-level social behaviors, such as the bystander effect, have been observed inside a simple video-game based virtual environment [63]. Social interactions in Massively Multilayer Online games have been studied by Cole et al. [19]. They suggest such games to be favorable for teamwork.

The task of community detection has been a well-studied problem, whose goal is to assign users to communities [103]. The most relevant line of research is probably the task of detecting *overlapping* communities, whose members can be part of multiple groups. Those lines of research have made use of Matrix Factorization methods in order to relax the assumption of communities being disjoint [145] [142]. Methods providing a direct way to embed the nodes of a network, thus generalizing the notion of network neighborhood, have recently been proposed [40].

The modeling of personalized decision processes has many application, notably in Collaborative Filtering, which generally assumes that future user actions can be predicted by collecting historical data from many users. Such applications have made extensive use of Matrix Factorization methods that have been popularized during the Netflix Prize [13]. Traditional techniques were mostly relying on ratings data but recent advances of the field have focused their attention on the problem of One-Class Collaborative Filtering [87], that is the task of learning from positive interactions only. More recently, Rendle et al. [96] proposed a variation of the method that is particularly suited for the modeling of one-class datasets. A link between recommendation and collaboration has been explored by Tang et al. [123]. In particular, the authors tackled the problem of scientific collaboration recommendation.

Research Questions: Given the work above, several research questions have remained unanswered:

RQ1: Are local coordinated user activities predictive of future user actions?

RQ2: Is the latent representation of our model interpretable?

RQ3: Is the product of users' collaboration segmentable relying on implicit social signal exclusively?

6.3 Data

In April 2017, the discussion platform *Reddit* launched *Place*, an online canvas of 1000-by-1000 pixels, designed as a social experiment. Reddit users were allowed to change the color of one pixel at every fixed time interval (the interval varied from 5 to 20 minutes during the events). The event lasted 72 hours and received a massive engagement from more than 1.2M unique users. Users collaborated to create various artworks by either directly interacting with the canvas or by coordinating their actions from the discussion platform.

The full dataset of events has been made publicly available. It contains more than 16M events and includes, for each event, the position of the click in the canvas, the chosen color and a unique user identifier. Note that usernames have been anonymized using a hash function, making impossible to link users to their respective *Reddit* profile.

After the end of the event, Reddit users launched an initiative to segment and detail the various artworks of the final canvas. Their crowd-sourced effort resulted in an atlas publicly available online [III]. In total, 1493 artworks have been identified by the community. In their original format, those annotations contained overlapping regions, coming from noisy segmentations or from users annotating subparts of the artworks. In this work, we do not consider the hierarchy of the artworks and, therefore, we have to further process the set of artworks through manual annotation with the following strategy: if two artworks have a non-zero intersection, we give priority to the smallest one and remove the overlapping region from the largest one. Then, we manually and iteratively select coherent artworks made of a single region. The result is a set of 830 non-overlapping artworks covering 84.2% of the canvas.

6.4 Analysis

In this section, we propose a general analysis of the event by shining a light on user behaviors within the online canvas.

[III] https://draemm.li/various/place-atlas/

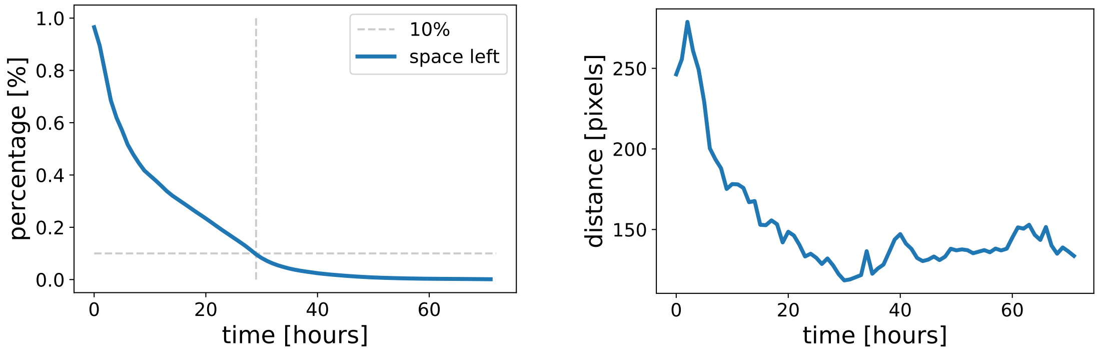


Figure 6.3 – Percentage of unused space in the canvas, over time (left). Average distance between two subsequent clicks from the same user, over time (right).

We first observe, in Fig. 6.2 (left), the activity distribution of the users. This distribution highlight the presence of few power-users and a vast majority of users performing a moderate number of clicks. In Fig. 6.2 (middle), we observe the same type of distribution for the number of updates performed on every pixel. As few pixels have been highly disputed, the large majority of them have only been updated a few times. For example, 4.35% of the pixels have been updated only once during the entire event. The fact that the number of updates is not uniform over the set of pixels suggests that there exists a latent structure in the users' decision process.

We pay special attention to the system activity level over time, from its initial chaotic state to the first signs of its convergence. The evolution of the activity level of the entire system is observable through the variations of the click rate. We first notice the level of user activity being clearly influenced by the circadian cycle of American users. Note that, according to Alexa [IV], United States alone represent 57% of the traffic on Reddit. Beyond the temporal variations of the overall activity within the canvas, the relationship between the number of active users and the number of clicks is of our interest, as it is subject to significant variations that characterize the average activity of single users (see Fig. 6.2 right). We observe a rapidly increasing number of click-per-user after the 24 hour. A first potential cause is the coverage of the event (both in social and mainstream media). Moreover, we observe the term “Reddit place” having growth by a factor of 1.8 from the first to the second day, according to Google Trends. Second, we compare this sudden raise of the activity level with variations of the userbase. At peak value, the number of concurrent unique users per hour was subject to an increase of 27% from the first to the second day.

Starting from an empty canvas, users have been continuously filling the space. After the first 24 hours, 90% of the pixels have already been clicked on at least once (see Fig. 6.3 left). With less blank space at their disposal, users were forced to overwrite existing

[IV] https://www.alexa.com/siteinfo/reddit.com

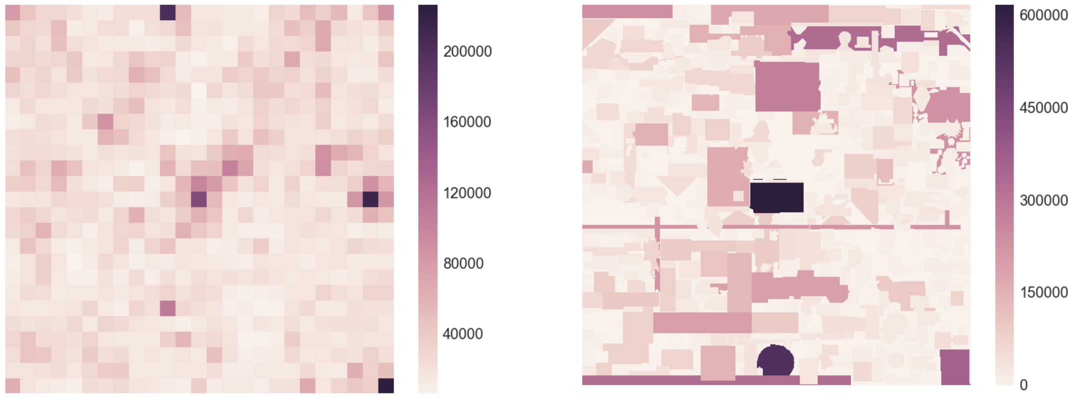


Figure 6.4 – Activity heatmap on the canvas. On the **left**, the canvas is partitioned in uniformly-sized tiles. On the **right**, the canvas regions reflect the artwork shapes, as described in Section 6.3. Both heatmaps use a comparable number of regions.

structure. We observe people having focusing their actions on a similar distribution: the distance between two subsequent clicks performed by the same user has, on average, largely decreased during the first 24 hours (see Fig. 6.3 right).

We observe the activity level with different partitioning schemes of the canvas. Without a clear segmentation following the shape of the artworks, though, only limited observation can be made on the activity taking place on the canvas, i.e., we mainly observe spikes of localized activity (see Fig. 6.4 left). However, when considering the artwork shapes, the activity from a group dynamics perspective is revealed (see Fig. 6.4 right), therefore motivating the development of an effective segmentation strategy.

6.5 Model

In this section, we introduce a predictive method that models collaboration between users in order to predict future user actions. In this regard, we train a model to evaluate the likelihood of a user u_i to perform a particular action at a given moment in time. We use the term action as a shorthand, with a slight abuse of notation, to denote the click of a user u_i on a pixel at coordinates (x, y) at time t.

To train the model according to the decision structure of user u_i, we train the model to discriminate an action a_i performed by user u_i from a randomly sampled action a_k performed by another user $u_k \neq u_i$. We define it as a probability

$$\Pr(a_i >_{u_i} a_j | \Theta), \tag{6.1}$$

where Θ represents the set of parameters of an arbitrary predictor and $>_{u_i}$ represents the preference scheme of user u_i. Specifically, we use the notation $a_i >_{u_i} a_j$ to indicate user u_i preferring action a_i over action a_j. A predictor that would perfectly model the latent preference structure $>_{u_i}$ of user u_i would thus predict a probability of 1 for $\Pr(a_i >_{u_i} a_j|\Theta)$ and a probability of 0 for $\Pr(a_i <_{u_i} a_j|\Theta)$. Defining $\hat{x}_{u_i,a_i}$ as the predicted score for user u_i and action a_i, the same ideal predictor would systematically predict a higher score for $\hat{x}_{u_i,a_i}$ than for $\hat{x}_{u_i,a_j}$, thus making the quantity $\hat{x}_{u_i,a_i,a_j} := \hat{x}_{u_i,a_i} - \hat{x}_{u_i,a_j}$ consistently and strictly positive. Therefore, training the model to discriminate between the two actions can be achieved by maximizing the difference between their predicted scores. In order to make the operation differentiable, we follow the procedure introduced by Rendle et al. [96] by maximizing the quantity $\ln \sigma(\hat{x}_{u_i,a_i,a_j})$. In particular, we maximize the following criterion

$$\text{BPR-OPT} := \sum_{(u_i,a_i,a_j)\in D} \ln \sigma(\hat{x}_{u_i,a_i,a_j}) - \lambda_\Theta ||\Theta||^2 \tag{6.2}$$

where λ_Θ is a parameter controlling the strength of the regularization. In our case, we opted for ℓ_2-regularization.

So far, we described our optimization criterion without specifying the underlying predictive model. Our choice of predictor is driven by its capability to model users-users relationships. We opt for an embedding method, since we hypothesize less independent behaviors than individuals in the system. Embedding methods are especially adapted to produce *personalized* predictions (e.g. collaborative filtering applications), by making the assumption that the behavior from an individual can be predicted by collecting data from many users and by projecting each of them in a common latent space. We therefore represent every user in the system by a latent representation: a real-valued vector p_{u_i} of size K where K is the chosen dimensionality of the latent space. We define a notion of distance between any pair of users in the considered population, where the distance metric represents the strength of collaboration between users. If two users are actively collaborating, the response produced by the combination of their respective vectors (typically by using dot product) should be high.

As suggested by previous studies (see Section 6.2), the choice of input features is determinant to capture complex inter-user dependencies. Assuming users performing actions in a fully-observable environment, we construct our input signal by observing the proximity of their actions. In our scenario, the notion of proximity could be constructed by looking at the tiles surrounding the one on which the considered user u_i clicked (see Fig. 6.5). Assuming every user being represented by a latent representation vector p_{u_i}, we define q_{a_i} to be a combination of the users' latent features vectors that updated the eight adjacent cells at last, before action a_i. We then specifically train our model to produce a high predicted score $\hat{x}_{u_i,a_i}$ of user u_i performing action a_i. The score $\hat{x}_{u_i,a_i}$ is computed as

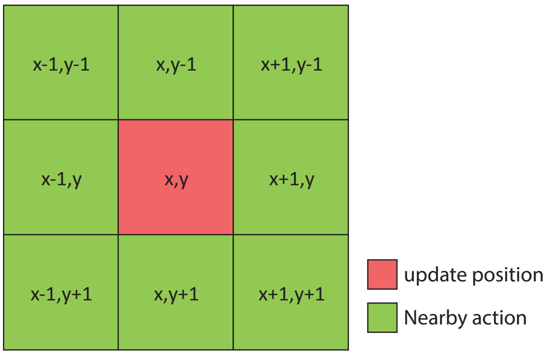


Figure 6.5 – Illustration of the input features. For an action performed on pixel (x,y), we consider the users having updated the 8 adjacent pixels at last.

follows

$$\hat{x}_{u_i,a_i} = p_{u_i}^T \cdot q_{a_i} \tag{6.3}$$

where q_{a_i} is a combination of users' embeddings. This combination could be computed in many ways, using a differentiable operation. We combined the embedding vectors using a simple sum over users' latent representations, defined as follows

$$q_{a_i} = \sum_{k \in A_{a_i}} p_{u_k} \tag{6.4}$$

where A_{a_i} is the list of users having updated the adjacent tiles at last, before action a_i. During our experiments, we observed the normalization of each vector p_{u_k} to benefit from normalization (in the sense of ℓ_2 normalization).

6.5.1 Optimization

As we aim to discriminate positive from negative examples, we devise our optimization procedure as a ranking problem. In particular, we train the predictor to discriminate an observed interaction from a randomly sampled negative (an action performed by another user). The BPR optimization scheme can be optimized using the following update step

$$\theta \leftarrow \theta + \alpha \cdot (\sigma(-\hat{x}_{u_i,a_i,a_j})\,\frac{\partial \hat{x}_{u_i,a_i,a_j}}{\partial \theta} + \lambda_\theta \Omega'(\theta)), \tag{6.5}$$

where $\hat{x}_{u_i,a_i,a_j} = \hat{x}_{u_i a_i} - \hat{x}_{u_i a_j}$, and θ represents the set of parameters to be learned. $\Omega(\theta)$ denotes a regularizer. We opted for a ℓ_2 regularization $\Omega(\theta) = \|\Theta\|_2^2$.

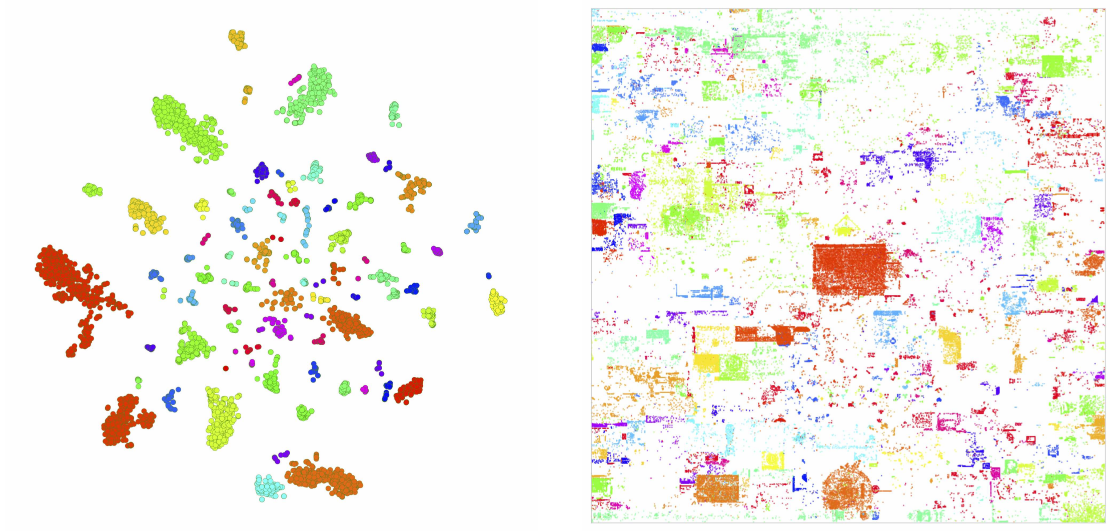


Figure 6.6 – Collaboration patterns on Reddit r/place. Groups of users have been identified by exploiting the computed latent representation (**left**), and their activities have been localized over the canvas (**right**). Best seen in colors. *Left:* t-SNE [127] projection of user embeddings. For visualization purposes, user groups have been colored using the resulting clustering from DBSCAN [30]. *Right:* Traces of activity performed by selected users. Colors correspond to the left figure. Results are computed for the last 1M interactions (around 3 hours of activity).

6.5.2 Experimental Setting

We adopt a *leave-one-out* methodology to assess the accuracy of the model, thus making every user having the same weight in the evaluation. Specifically, we constitute our evaluation set by sampling one action for each user. As an experimental setting, we discard users having performed less than 10 interactions from our dataset and filter the first quarter of interactions. We apply this filtering to avoid a cold-start data regime, a scenario that is outside of the scope of this work.

Reproducibility: We ran our experiment on a single computer, running a 3.2 GHz Intel Core i7 CPU, using PyTorch version 0.2.0.4 [V]. We run the optimization on GPU NVIDIA GTX 670. We trained our model with the following parameters: $\alpha = 0.04$, $\lambda_\theta = 0.01$, $K = 120$. All code will be made available at publication time [VI].

6.5.3 Baselines:

We describe the baselines to which our approach is compared to. Those methods are split in two different categories: methods that model the interaction of users with their

[V] http://pytorch.org/

[VI] https://github.com/JRappaz/placemf

environment and methods that model users interrelationships.

Median: As discussed in Section 6.4, users have rapidly focused their actions on specific areas of the canvas. We therefore compare our results to a simple baseline that compute the likelihood of interaction as a linear function of the euclidean distance between two points p_1 and p_2, where p_1 is the position of the considered click and p_2 is the median position of the user activity in the canvas (computed from the training set). We also tried with the average position but exclude the results as they were consistently lower.

MF: Matrix Factorization (MF) methods typically model the preferences from users interacting with a large number of items. We trained the MF baseline to model the interactions between users and pixels, in a setting similar to a preference problem. We used the Spark version 2.2 ALS implementation, a scalable model of Collaborative Filtering method. [VII].

Count: We compare our method to a simple count of user interactions. We count the number of adjacent clicks between users and rank the available actions based on the location having the largest sum of users' interactions.

Community Detection: We use Infomap[VIII], a scalable, state-of-the-art community detection algorithm optimizing an information-theoretic criterion. We model collaboration between user as a weighted undirected graph. Edges weights are computed from the number of past adjacent clicks between two users. The algorithm provides a single assignment for each user to a cluster. The predicted score for an action and a user u_i is then computed by counting the number of users having updated the adjacent tiles at last and being part of the same community than u_i.

6.5.4 Evaluation

To evaluate our approach, we select the widely used metric *Area Under the Curve* (AUC) [113] as our measure of performance:

$$\text{AUC} = \frac{1}{|D|} \sum_{(u_i,a_i,a_j)\in D} H(\hat{x}_{u_i a_i} - \hat{x}_{u_i a_j}) = \frac{1}{|D|} \sum_{(u_i,a_i,a_j)\in D} H(\hat{x}_{u_i,a_i,a_j}), \tag{6.6}$$

where $H(\cdot)$ is the Heaviside step function (equal to 1 for a positive input, zero otherwise) and D is our evaluation set composed of one triplet (u_i, a_i, a_j) per source where u_i is a user, a_i is a randomly sampled action that has been performed by user u_i and a_j is a randomly sampled action from another user $u_j \neq u_i$. This metric assesses the ability of

[VII] https://spark.apache.org/

[VIII] http://www.mapequation.org/code.html

Method		AUC
Environment	Median	0.8413 ± 0.0006
	MF	0.7921 ± 0.0006
Social	Count	0.8383 ± 0.0003
	CD	0.8382 ± 0.0006
	Ours	**0.8792** ± 0.0019

Table 6.1 – Results: scores are reported with the Area Under the Curve (AUC) metric (CI=0.95).

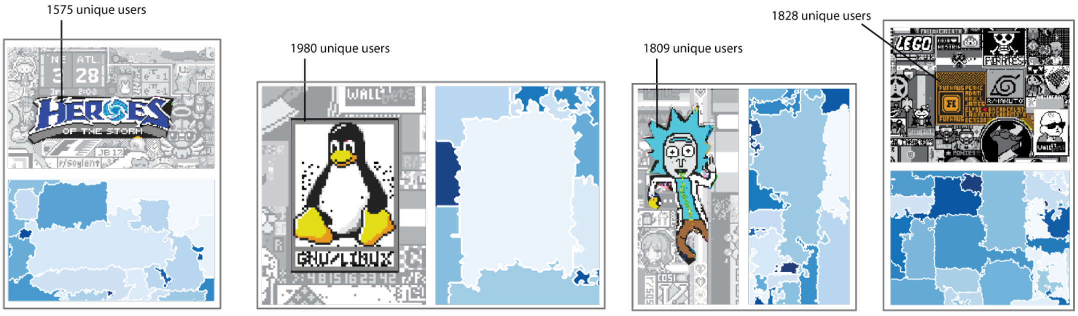


Figure 6.7 – Segmentation of the artworks using the proposed method which leverages exclusively social signals. We report the number of users that contributed to the final version of the highlighted artworks. Even with artworks produced by a large user base, our segmentation method can correctly identify the boundaries, as shown in the above examples.

the predictor to correctly rank a positive interaction withheld during training against a random negative example. Negative examples are sampled from the training set during training and from the testing set during testing. An ideal predictor would obtain a score of $AUC = 1$, while a random selection would output a score around $AUC = 0.5$.

6.6 Results

In this section, we discuss the results summarized in Table 6.1. We divide the aforementioned methods in two categories: methods that capture the relation between users and their environment, and methods that capture users' inter-relationships.

User-environment relationship: The virtual environment is constituted by a set of clickable tiles. Moreover, this environment is structured, since pixels have clearly defined positions. MF-based methods are unaware of the structure of the environment and did not model any social aspect of the event. We therefore observe them to exhibit a lower score than the other methods. The score obtained from a median-based method is, however, performing relatively well, despite its relative simplicity. This suggest the locality of user actions being a critical aspect in the design of a method to predict collaborations.

User-user relationship: We observe the Community Detection method being on par with a method based on raw interactions count. We therefore suggest that a strict segmentation of the users does not result on a gain in performance. Since those methods were able to capture relatively good social proximity between users, they were not optimized (and not parametrized) to directly model user interactions. From Table 6.1, we can observe the performance gain obtained from our model, being optimized to model user interactions data. Specifically, our methods outperformed the best performing baseline with a margin of 4.5% of relative improvement. We therefore conclude that, from the considered models, the parametrization of user interrelationships is the most predictive method of user actions in a sandbox environment.

Systems are generally considered as complex if the sum of the individual behaviors of its subparts cannot explain the overall behavior of the ensemble. This consideration encourages the modeling of the inter-relationships between subparts, instead of modeling them as independent behaviors. Our results reinforce this perspective, as we report this modeling approach to be more predictive of users' decisions. Moreover, our approach has the advantage to be interpretable, as described in the following section.

6.7 Segmentation

In this section, we exploit the learned representation of our model to identify user groups and segment their respective artworks.

First we show that groups of users can be identified from the latent representation obtained through the proposed method. As described in Section 6.5, each user is represented by a single low-dimensional embedding vector p_{u_i}. The total set of vectors represents a distributed representation of the collaboration strength between any two considered users (all the vectors share the same latent space but the triangle inequality is not necessarily respected). This representation can be further reduced in dimensionality in order to be visualized. We observe the resulting representation to exhibit sufficient cohesiveness to be easily labeled by an unsupervised clustering approach. The traces left in the canvas by the different groups of users is shown in Fig. 6.6.

In a second step, we propose a method to segment the product of user collaboration, i.e. to attribute every pixel to a single partition of the final image. To this end, we propose a simple method that detects abrupt variations of the social activity on the canvas. We first assume that users leave a social *signature* in the canvas, and suggest that abrupt variations of this signature could reveal the artworks edges. To model it, we attach to each pixel a fingerprint of the social activity that took place on it. We

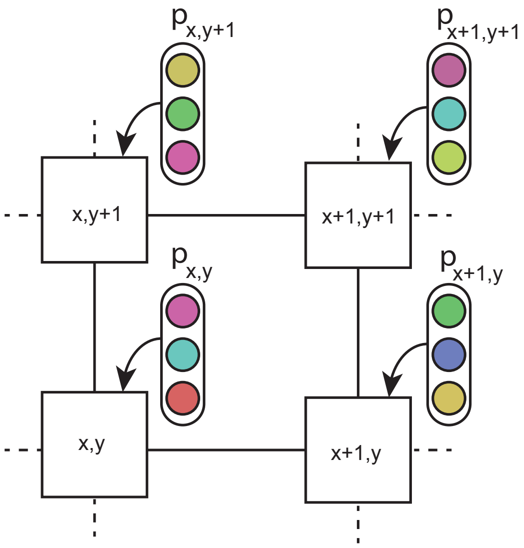


Figure 6.8 – Illustration of the agglomerative clustering procedure setting. Each pixel has a latent features vector attached to it. Black lines represent the connectivity constraint applied to the procedure: only adjacent pixels can be merged.

opt for the latent representation of the user that updated the pixel at last, i.e., the user whose action colored a pixel of the final canvas. We then segment the canvas by using an agglomerative clustering procedure. At start, the algorithm attributes each pixel to its own cluster. Then, the procedure recursively merges clusters by minimizing the sum of squared differences within all clusters. Note that the algorithm is constrained by the connectivity of the grid as only adjacent pixels could be merged (see Fig. 6.8). The procedure terminates when the target number of clusters C is obtained. To find the optimal value of the C parameter, we rely on the crowd-sourced segmentation of user artwork that we treat as ground truth. We search for the optimal number of clusters, restricting our evaluation to the portions of the space being annotated. We use Adjusted Rand Index (ARI), as a metric comparing the results of the clustering procedure with the annotations. We found the best value to be $C = \mathbf{840}$, a value close to the 830 artworks identified in human-curated atlas. Examples of the resulting clustering are shown in Fig. 6.7.

6.8 Discussion

The method introduced in Section 6.5 exhibits a significant improvement over the baselines in a purely predictive task. We suggest that this result is due to its capability of modeling user-user relationships, while still capturing local consistency of the environment. Indeed, our approach directly models a 8-to-1 relationship between the considered user and the users having updated the adjacent tiles. Moreover, thanks to the common latent space in which all the users are projected, our method models transitive relations between participants. As an example, if user A collaborated with user B and user B with user C, a potential collaboration between A and C should still exhibit a high predicted score since they would be contained in the same local manifold of the embedding space. This is especially important in the case of large artworks, to which many users contributed.

We propose a qualitative interpretation of the learned parameters in Fig. 6.7, where we show the traces left by participants on the canvas. We observe the method to capture a relative proximity of actions performed by users belonging to the same group. This result is coherent with the observation of user actions being relatively localized (see Section 6.4).

The proposed method captures the proximity of users' actions on the canvas. One might question the ability of such approach to establish a clear segmentation of the artworks from proximity signals. For example, two different user populations, working on two different adjacent artworks, are difficult to be distinguished, as the actions of the two groups would appear in close vicinity. However, we give evidences that a latent representation, computed from a large set of actions, is sufficiently expressive to recover the artworks boundaries.

6.9 Summary and Future Work

We introduce a generic method to infer collaboration patterns in environments where only user interactions are observable. We show, through experiments, that the local proximity of users' actions represents a sufficiently expressive signal for the study of collaboration. Indeed, we report it to be more predictive than the modeling of the interactions between users and their environment. This finding reinforces previous results in the domain, that suggest the study of emergent phenomenons requiring the modeling of interrelationships between the parts of a system, rather than modeling their individual behaviors.

Our method finds immediate applications in the analysis of large-scale collaborative efforts, such as Wikipedia or Github, in which users don't always have explicit or observable links between them. In such cases, the proper segmentation of user contributions could lead to a more fine-grained quantitative analysis of the various portions of the contribution. In concrete terms, the method would allow assessing the quality of the contribution produced by a subgroup of the population instead of measuring the quality of the article/repository as a whole. Beyond the analysis of collaboration, such methods could also be used to produce recommendations of partnership as a way to engage participants through direct collaboration recommendation. Our method can finally find applications in malicious collaboration. In such context, the result of a collaboration could be isolated using similar segmentation techniques to limit the impact of adversarial activities on large projects.

We foresee several directions as future work.

- Given the generic nature of our method, we want to test it on other collaborative platforms (e.g., Wikipedia, Github, etc.) as they represent a natural extension of our current effort on Reddit r/place. Moreover, we believe that other types of virtual environments, such as multi-player video-games, would represent an interesting testbed for the proposed method.
- We want to tackle the problem of learning in a low data regime (i.e., cold-start scenario). We plan to leverage side-information, inherent to both users and their environment, as it represents a promising resource to generate prediction for newly introduced users.
- We want to make our model temporally-aware, as further insights can be gathered by analyzing the temporal dynamics of the user interactions.
- We plan to study the dual problem, that is to infer antisocial behaviors from localized actions, and to propose a method to distinguish them from collaboration patterns.

7 Conclusion

The range of activities that can be performed online continues to expand. People increasingly use Web platforms to discuss with friends, consume movies and songs, look for employment, housing, or even romantic partners. The large volume of information and options available on these platforms can become overwhelming and make navigation difficult. To tackle this problem, personalized predictive systems are often used to guide users through these services, by learning from their preferences and helping them access the right information at the right time. However, predicting user actions is a difficult task, because of the high interactivity of the modern Web, the spread of user-generated content, and the many social and temporal factors that affect user decisions.

In this thesis, we proposed novel personalized ranking methods suited to the interactive and social Web, that incorporate contextual signals into their predictions and that maintain a consistent embedding space over multiple time epochs. Our work situates itself in recent lines of research that, instead of predicting user interests on static databases of items, predict user interactions with pieces of content or with other users. Our model designs are motivated by the study of large datasets involving millions of users and items. From the careful study of user dynamics, we identified temporal and social patterns that we incorporated into our models for improving predictions. To encourage future research in this domain, we shared multiple datasets containing rich interaction signals with the research community.

In Chapter 3, we studied recommendation on a live-streaming platform, *Twitch*, for which we designed a method to rank available channels. In this setting, the meaning of positive interactions differs from its traditional definition, since users could interact multiple times with the same item; we observed that repeated interactions account for more than 50% of our dataset. To discriminate among positive instances, we added a temporal embedding layer, to learn from the timing of these repeated interactions. Negative examples also have a different meaning, since "non-interaction" could reflect implicit user preferences, as much as the unavailability of an item. Therefore, we proposed a novel model architecture that only samples and processes items from a pool of available options. We observed that the modeling of these two aspects led to an increase in performance, which paves the way for future research in real-time media.

In Chapter 4, we proposed a methodology, based on collaborative filtering, to model the selection of news stories performed by digital media and predict their coverage of world events. First, we showed that the knowledge captured by this model could be exploited to *mitigate* popularity biases in an aggregation of news articles, by reducing the weight of articles published by similar media. Then, we proposed a mechanism to maintain this model temporally consistent over successive time epochs, which enabled us to *monitor* the evolution of news coverage and to detect strong variations at channel-level. We observed those variations to be frequently associated with the acquisition of media channels by broadcast groups. We believe that our methods represent a first step towards large-scale monitoring and mitigation of event misrepresentation in the news, and could be applied to other aggregated information streams coming from diverse sources.

In Chapter 5, we proposed a recommender system for bartering platforms that models reciprocal interest: each of two users needs to have what the other wants in order for a trade to occur. Compared to traditional e-commerce platforms, trade opportunities are extremely scarce on exchange platforms, which encourages the use of a bidirectional recommendation method. We showed that pairs of users who successfully traded in the past are likely to trade again as they build trust (for example, a pair of users trade 3.56 times on average on *Ratebeer*, a beer exchange community). Therefore, we proposed a socially-aware model that favors exchanges with users that already traded together. With the rise of two-sided online platforms, we believe that matching recommendation systems could be applied to new scenarios, both for the symmetric case (e.g. dating platforms) and asymmetric case (e.g. matching workers to jobs).

In Chapter 6, we studied a collaborative creation of artworks on a virtual canvas hosted on Reddit. We showed that a personalized ranking approach can predict user clicks by modeling the proximity of their actions on the canvas. We believe that this method could be applied to infer coordinated activities in other virtual environments such as video games. More generally, capturing rich social signals, such as collaboration patterns, represents a unique opportunity to study complex social phenomenons. With virtual environments being increasingly used as a proxy to study specific social-psychology aspects, our method paves the way for the analysis of user behavior in contexts in which only user actions are observable, and the collaboration patterns are emergent rather than predefined.

In this thesis, we leveraged collaborative filtering methods to predict and characterize user decisions. We explored various scenarios on digital and social media, analyzed how people produce and consume online content, and how they interact together on social platforms. We believe that this research line could be pursued, by applying personalized predictive methods to new online scenarios. We give a few promising research directions below.

Live settings: the recent rise of streaming media platforms encourages the development of novel recommendation techniques. Many known dynamics remain to be adapted to the real-time setting. For example, the popularity of an item is generally parametrized with a single weight, but modeling the evolution of item popularity in real-time would require a more sophisticated approach. Further research is also needed to include new types of signals from a live audience. For example, a model could account for the navigation path from an item to another (e.g. browsing video channels), and for the influence (or competition) of concurrently available items. In general, making recommendations of streaming content requires combining historical data with real-time signals, which is a challenging problem that remains unaddressed.

Incorporating rich signals: Web platforms typically generate a large volume of rich signals that represent a trove of knowledge to learn from. Recommender systems can enrich interaction data with various content, social and temporal features to make more accurate recommendations. Moreover, modeling user and item attributes generally helps make more interpretable recommendations, for example by providing the specific factors that led to a prediction. In particular, with the recent advance of vision and language models, recommendations can be enhanced with textual and visual item data. The wide availability of pre-trained models creates new opportunities to transfer knowledge acquired by these models to recommendation tasks. This approach is particularly appealing for scenarios in which user historical data are not accessible (e.g. for privacy reasons) since predictions could be made from item features exclusively.

Users in control: there exist risks in blindly optimizing personalization models from preference signals. Recommender systems have been associated with various detrimental effects such as filter bubbles, popularity bias, lack of diversity in content exposure or loss of trust from users. More worryingly, recommending content of political nature has been suspected to be a source of polarization and to create "pathways" from moderate to extreme (e.g. alt-right) communities. Those phenomena are often discussed anecdotally since they are difficult to measure online (they have been mostly investigated through simulations) and further research is needed to characterize these dynamics empirically. We believe that future research should create more transparent, explainable, and interactive systems to put users in control of the content they are exposed to. Personalization methods could then be seen as a discovery mechanism or a navigation paradigm rather than a black-box predictive system. The structure and interface of such systems constitute important research challenges.

Bibliography

[1] Z. Abbassi, N. Haghpanah, and V. Mirrokni. Exchange market mechanisms without money. In *WINE*, 2015.

[2] Z. Abbassi and L. V. Lakshmanan. On efficient recommendations for online exchange markets. In *ICDE*, pages 712–723, 2009.

[3] Z. Abbassi, L. V. Lakshmanan, and M. Xie. Fair recommendations for online barter exchange networks. In *WebDB*, pages 43–48, 2013.

[4] D. J. Abraham, A. Blum, and T. Sandholm. Clearing algorithms for barter exchange markets: enabling nationwide kidney exchanges. In *EC*, pages 295–304, 2007.

[5] E. Acar, D. M. Dunlavy, and T. G. Kolda. Link prediction on evolving data using matrix and tensor factorizations. In *Data Mining Workshops, 2009. ICDMW'09. IEEE International Conference on*, pages 262–269. IEEE, 2009.

[6] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6):734–749, 2005.

[7] J. An and H. Kwak. Two tales of the world: comparison of widely used world news datasets gdelt and eventregistry. In *Proceedings of the 10th International AAAI Conference on Web and Social Media (ICWSM16)*, pages 619–622, 2016.

[8] A. Anderson, R. Kumar, A. Tomkins, and S. Vassilvitskii. The dynamics of repeat consumption. In *Proceedings of the 23rd international conference on World wide web*, pages 419–430, 2014.

[9] S. P. Anderson and S. Coate. Market provision of broadcasting: a welfare analysis. *The Review of Economic Studies*, 72(4):947–972, 2005. ISSN: 00346527, 1467937X. URL: http://www.jstor.org/stable/3700696.

[10] E. Bakshy, S. Messing, and L. A. Adamic. Political science. exposure to ideologically diverse news and opinion on facebook. *Science*, 348 6239:1130–2, 2015.

[11] T. Baldwin, D. Bergan, F. Fico, S. Lacy, and S. S. Wildman. News media coverage of city governments in 2009. *Retrieved June*, 11:2012, 2010.

[12] Y. Bar-Yam. General features of complex systems. *Encyclopedia of Life Support Systems (EOLSS), UNESCO, EOLSS Publishers, Oxford, UK*, 2002.

[13] R. M. Bell and Y. Koren. The bellkor solution to the netflix prize. In 2007.

[14] J. Bennett, S. Lanning, et al. The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35. New York, NY, USA., 2007.

[15] P. Bocheck. System and method for finding potential trading partners in both two-party and multi-party scenarios, Feb. 2014. URL: https://www.google.com/patents/US8645203. US Patent 8,645,203.

[16] S. Chang, Y. Zhang, J. Tang, D. Yin, Y. Chang, M. A. Hasegawa-Johnson, and T. S. Huang. Streaming recommender systems. In *Proceedings of the 26th international conference on world wide web*, pages 381–389, 2017.

[17] J. Chen, H. Zhang, X. He, L. Nie, W. Liu, and T.-S. Chua. Attentive collaborative filtering: multimedia recommendation with item-and component-level attention. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 335–344, 2017.

[18] I. A. Christensen and S. Schiaffino. Social influence in group recommender systems. *Online Information Review*, 2014.

[19] H. Cole and M. D. Griffiths. Social interactions in massively multiplayer online role-playing gamers. *CyberPsychology & Behavior*, 10(4):575–583, 2007.

[20] R. Crane and D. Sornette. Robust dynamic classes revealed by measuring the response function of a social system. *Proceedings of the National Academy of Sciences*, 105(41):15649–15653, 2008.

[21] G. Davies. *History of Money.* University of Wales Press, 2010.

[22] N. De Cao, L. Wu, K. Popat, M. Artetxe, N. Goyal, M. Plekhanov, L. Zettlemoyer, N. Cancedda, S. Riedel, and F. Petroni. Multilingual autoregressive entity linking. *arXiv preprint arXiv:2103.12528*, 2021.

[23] C. de Juan Carbonell and J. Goldstein-Stewart. The use of mmr, diversity-based reranking for reordering documents and producing summaries. *SIGIR Forum*, 51:209–210, 1998.

[24] S. DellaVigna, E. Kaplan, A. B. . Krueger, M. Manacorda, E. Moretti, T. Persson, S. Popkin, R. Puglisi, M. Rabin, J. M. Shapiro, U. Simonsohn, L. Stoker, D. Stromberg, T. Deryugina, M. Deza, D. Fox, M. Galicia, C. W.-L. Ho, S. Khanchanawong, R. M. Kim, M. Kohan, V. S. Kumar, J. J. Leung, C. Li, T. Y. Lim, M. Mai, S. Parekh, S. Radakrishnan, R. Relan, D. Acland, S. Bhargava, A. Ebenstein, and D. G. Pope. The fox news effect: media bias and voting. In 2005.

[25] Y.-A. de Montjoye, A. Stopczynski, E. Shmueli, A. Pentland, and S. Lehmann. The strength of the strongest ties in collaborative problem solving. In *Scientific reports*, 2014.

[26] S. Djankov, C. McLiesh, T. Nenova, and A. Shleifer. Who owns the media? *Journal of Law and Economics*, 46(2):341–381, 2003.

[27] G. Doyle. *Media Ownership: The Economics and Politics of Convergence and Concentration in the UK and European Media.* 2002. DOI: http://dx.doi.org/10.4135/9781446219942.

[28] D. M. Dunlavy, T. G. Kolda, and E. Acar. Temporal link prediction using matrix and tensor factorizations. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(2):10, 2011.

[29] L. Eberhard and C. Trattner. Recommending sellers to buyers in virtual marketplaces leveraging social information. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pages 559–564. International World Wide Web Conferences Steering Committee, 2016.

[30] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In 1996.

[31] S. Flaxman, S. Goel, and J. M. Rao. Filter bubbles, echo chambers, and online news consumption. In 2015.

[32] M. Gentzkow and J. Shapiro. What drives media slant? evidence from u.s. daily newspapers. *Econometrica*, 78(1):35–71, 2010. URL: https://EconPapers.repec.org/RePEc:ecm:emetrp:v:78:y:2010:i:1:p:35-71.

[33] M. Gentzkow, J. M. Shapiro, and M. Sinkinson. The Effect of Newspaper Entry and Exit on Electoral Politics. Working Paper 15544, National Bureau of Economic Research, Nov. 2009. DOI: 10.3386/w15544. URL: http://www.nber.org/papers/w15544.

[34] L. George and J. Waldfogel. National media and local political participation: the case of the new york times:33–48, Jan. 2008.

[35] D. J. Gerner, R. Abu-Jabr, P. A. Schrodt, and Ö. Yilmaz. Conflict and mediation event observations (cameo): a new event data framework for the analysis of foreign policy interactions. In *of Foreign Policy Interactions. Paper presented at the International Studies Association*, 2002.

[36] C. Gershenson and N. Fernández. Complexity and information: measuring emergence, self-organization, and homeostasis at multiple scales. *Complexity*, 18:29–44, 2012.

[37] J. S. Goldstein. A conflict-cooperation scale for weis events data. *The Journal of Conflict Resolution*, 36(2):369–385, 1992. ISSN: 00220027, 15528766. URL: http://www.jstor.org/stable/174480.

[38] T. Groseclose and J. Milyo. A measure of media bias. *The Quarterly Journal of Economics*, 120(4):1191–1237, 2005. ISSN: 00335533, 15314650. URL: http://www.jstor.org/stable/25098770.

[39] T. Groseclose and J. Milyo. A measure of media bias. *The Quarterly Journal of Economics*, 120(4):1191–1237, 2005. URL: https://EconPapers.repec.org/RePEc:oup:qjecon:v:120:y:2005:i:4:p:1191-1237..

[40] A. Grover and J. Leskovec. Node2vec: scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM, 2016.

[41] P. Gupta, D. Garg, P. Malhotra, L. Vig, and G. Shroff. Niser: normalized item and session representations to handle popularity bias. *arXiv preprint arXiv:1909.04276*, 2019.

[42] P. Haddawy, N. Rujikeadkumjorn, K. Dhananaiyapergse, and C. Cheng. Balanced matching of buyers and sellers in e-marketplaces: the barter trade exchange model.

[43] J. Hamari, M. Sjöklint, and A. Ukkonen. The sharing economy: why people participate in collaborative consumption. *Journal of the Association for Information Science and Technology*, 2015.

[44] W. A. Hamilton, O. Garretson, and A. Kerne. Streaming on twitch: fostering participatory communities of play within live mixed media. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pages 1315–1324. ACM, 2014.

[45] R. He, C. Fang, Z. Wang, and J. McAuley. Vista: a visually, socially, and temporally-aware model for artistic recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 309–316. ACM, 2016.

[46] R. He and J. McAuley. Ups and downs: modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, pages 507–517, 2016.

[47] R. He and J. McAuley. Vbpr: visual bayesian personalized ranking from implicit feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30 of number 1, 2016.

[48] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182, 2017.

[49] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, 2015.

[50] Z. Hilvert-Bruce, J. T. Neill, M. Sjöblom, and J. Hamari. Social motivations of live-streaming viewer engagement on twitch. *Computers in Human Behavior*, 84:58–67, 2018.

[51] G. Hu, Y. Zhang, and Q. Yang. Mtnet: a neural approach for cross-domain recommendation with unstructured text. *KDD Deep Learning Day*:1–10, 2018.

[52] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*, pages 263–272. Ieee, 2008.

[53] H. Jenkins. The cultural logic of media convergence. *International journal of cultural studies*, 7(1):33–43, 2004.

[54] W.-C. Kang and J. McAuley. Learning consumer and producer embeddings for user-generated content recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pages 407–411, 2018.

[55] W.-C. Kang and J. McAuley. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 197–206. IEEE, 2018.

[56] M. Kaytoue, A. Silva, L. Cerf, W. Meira Jr, and C. Raıssi. Watch me playing, i am a professional: a first study on video game live streaming. In *Proceedings of the 21st International Conference on World Wide Web*, pages 1181–1188. ACM, 2012.

[57] M. Kearns. Experiments in social computation. *Communications of the ACM*, 55(10):56–67, 2012.

[58] A. Kittur and R. E. Kraut. Harnessing the wisdom of crowds in wikipedia: quality through coordination. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, pages 37–46. ACM, 2008.

[59] Y. Koren. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 447–456, 2009.

[60] Y. Koren. Collaborative filtering with temporal dynamics. *Communications of the ACM*, 2010.

[61] Y. Koren and R. Bell. Advances in collaborative filtering. *Recommender systems handbook*:77–118, 2015.

[62] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[63] M. D. Kozlov and M. K. Johansen. Real behavior in virtual environments: psychology experiments in a simple virtual-reality paradigm using video games. *Cyberpsychology, behavior and social networking*, 13 6:711–4, 2010.

[64] A. KubÃ. Toward a formalization of emergence. *Artificial Life*, 9(1):41–65, 2003. DOI: 10.1162/106454603321489518. eprint: https://doi.org/10.1162/106454603321489518. URL: https://doi.org/10.1162/106454603321489518.

[65] E. J. Laurance. Events data and policy analysis: *Policy Sciences*, 23(2):111–132, May 1990. ISSN: 1573-0891. DOI: 10.1007/BF00175597. URL: https://doi.org/10.1007/BF00175597.

[66] G. Lederrey and R. West. When sheep shop: measuring herding effects in product ratings with natural experiments. In *Proceedings of the 2018 World Wide Web Conference*, pages 793–802, 2018.

[67] J. Li, Y. Wang, and J. McAuley. Time interval aware self-attention for sequential recommendation. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 322–330, 2020.

[68] Y.-R. Lin, J. P. Bagrow, and D. Lazer. More voices than ever? quantifying media bias in networks. *CoRR*, abs/1111.1227, 2011.

[69] Y. Liu, L. Zhou, and J. Huang. A research to improve collaborative filtering recommendation algorithm oriented barter electronic trading platform. In *ICCNCE*, 2013.

[70] R. Lorimer and Scannell. *Mass communications: a comparative introduction.* M. U. Press, editor. 1994, pp. 86–87. ISBN: 978-0-7190-3946-1.

[71] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King. Recommender systems with social regularization. In *WSDM*, 2011.

[72] A. Majchrzak, A. Malhotra, and R. John. Perceived individual collaboration know-how development through information technology - enabled contextualization: evidence from distributed teams. *Information Systems Research*, 16:9–27, 2005.

[73] M. Mansoury, H. Abdollahpouri, M. Pechenizkiy, B. Mobasher, and R. Burke. Feedback loop and bias amplification in recommender systems. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2145–2148, 2020.

[74] F. Martins, J. Magalhães, and J. Callan. Barbara made the news: mining the behavior of crowds for time-aware learning to rank. *CoRR*, abs/1602.03101, 2016. URL: http://arxiv.org/abs/1602.03101.

[75] W. Mason and D. J. Watts. Collaborative learning in networks. *Proceedings of the National Academy of Sciences*, 109(3):764–769, 2012.

[76] M. J. Mataric. Designing emergent behaviors: from local interactions to collective intelligence. In *Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, pages 432–441, 1993.

[77] S. Mathieu. *Match-making in bartering scenarios.* PhD thesis, Citeseer, 2006.

[78] D. P. Mersch, A. Crespi, and L. Keller. Tracking individuals shows spatial fidelity is a key regulator of ant social organization. *Science*, 340(6136):1090–1093, 2013. ISSN: 0036-8075. DOI: 10.1126/science.1234316. eprint: http://science.sciencemag.org/content/340/6136/1090.full.pdf. URL: http://science.sciencemag.org/content/340/6136/1090.

[79] J. Mondak. Media exposure and political discussion in u.s. elections. English (US). *Journal of Politics*, 57(1):62–85, May 1995. ISSN: 0022-3816. DOI: 10.2307/2960271.

[80] L. Muchnik, S. Pei, L. C. Parra, S. D. Reis, J. S. Andrade Jr, S. Havlin, and H. A. Makse. Origins of power-law degree distribution in the heterogeneity of human activity in social networks. *Scientific reports*, 2013.

[81] G. Nascimento, M. Ribeiro, L. Cerf, N. Cesário, M. Kaytoue, C. Raıssi, T. Vasconcelos, and W. Meira. Modeling and analyzing the video game live-streaming community. In *2014 9th Latin American Web Congress*, pages 1–9. IEEE, 2014.

[82] T. Neumayr and M. Augstein. A systematic review of personalized collaborative systems. *Frontiers in Computer Science*:43, 2020.

[83] M. E. Newman. Scientific collaboration networks. ii. shortest paths, weighted networks, and centrality. In volume 64 of number 1, page 016132. APS, 2001.

[84] D. W. Oard, J. Kim, et al. Implicit feedback for recommender systems. In *AAAI workshop on recommender systems*, pages 81–83, 1998.

[85] F. Oberholzer-Gee and J. Waldfogel. Media Markets and Localism: Does Local News en Español Boost Hispanic Voter Turnout? Working Paper 12317, National Bureau of Economic Research, June 2006. DOI: 10.3386/w12317. URL: http://www.nber.org/papers/w12317.

[86] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang. One-class collaborative filtering. In *2008 Eighth IEEE International Conference on Data Mining*, pages 502–511. IEEE, 2008.

[87] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. M. Lukose, M. Scholz, and Q. Yang. One-class collaborative filtering. *2008 Eighth IEEE International Conference on Data Mining*:502–511, 2008.

[88] M. S. Pera and Y.-K. Ng. A recommendation-based book-exchange system without using wish lists, 2015.

[89] P. Pipergias Analytis, D. Barkoczi, P. Lorenz-Spreen, and S. Herzog. The structure of social influence in recommender networks. In *Proceedings of The Web Conference 2020*, pages 2655–2661, 2020.

[90] K. Pires and G. Simon. Youtube live and twitch: a tour of user-generated live streaming systems. In *Proceedings of the 6th ACM multimedia systems conference*, pages 225–230. ACM, 2015.

[91] L. Pizzato, T. Rej, T. Chung, I. Koprinska, and J. Kay. Recon: a reciprocal recommender for online dating. In *RecSys*, pages 207–214, 2010.

[92] D. Potter and K. E. Matsa. State of the News Media 2014: A Boom in Acquisitions and Content Sharing Shapes Local TV News in 2013. Technical report, Pew Research Center, Mar. 2014.

[93] D. Pritchard. Viewpoint diversity in cross-owned newspapers and television stations: a study of news coverage of the 2000 presidential campaign, Sept. 2002. URL: https://docs.fcc.gov/public/attachments/DOC-226838A7.pdf.

[94] J. J. Ramasco, S. N. Dorogovtsev, and R. Pastor-Satorras. Self-organization of collaboration networks. *Physical review E*, 70(3):036106, 2004.

[95] S. Rendle and C. Freudenthaler. Improving pairwise learning for item recommendation from implicit feedback. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 273–282. ACM, 2014.

[96] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: bayesian personalized ranking from implicit feedback. In *UAI*, 2009.

[97] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. BPR: bayesian personalized ranking from implicit feedback. In *UAI*, pages 452–461, 2009.

[98] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*, 2012.

[99] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 811–820, 2010.

[100] F. Ricci, L. Rokach, and B. Shapira. Introduction to recommender systems handbook. In *Recommender systems handbook*, pages 1–35. Springer, 2011.

[101] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor. *Recommender Systems Handbook*. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition, 2010, pages 148–149, 161–168. ISBN: 0387858199, 9780387858197.

[102] E. Rich. User modeling via stereotypes. *Cognitive science*, 3(4):329–354, 1979.

[103] M. Rosvall and C. T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences of the United States of America*, 105 4:1118–23, 2008.

[104] A. E. Roth, T. Sonmez, and M. U. Unver. Efficient kidney exchange: Coincidence of wants in a structured market. Technical report, National Bureau of Economic Research, 2005.

[105] A. E. Roth, T. Sonmez, and M. U. Unver. Kidney exchange. Technical report, National Bureau of Economic Research, 2003.

[106] M. Rudolph and D. Blei. Dynamic bernoulli embeddings for language evolution. *arXiv preprint arXiv:1703.08052*, 2017.

[107] M. Rudolph, F. Ruiz, S. Mandt, and D. Blei. Exponential family embeddings. In *Advances in Neural Information Processing Systems*, pages 478–486, 2016.

[108] D. Sáez-Trumper, C. Castillo, and M. Lalmas. Social media news communities: gatekeeping, coverage, and statement bias. In *CIKM*, 2013.

[109] P. A. Schrodt, S. G. Davis, and J. L. Weddle. Political Science: KEDS–A Program for the Machine Coding of Event Data. *Social Science Computer Review*, 12(4):561, 1994.

[110] D. W. Scott. Multivariate density estimation: theory, practice, and visualization, 2015.

[111] V. Sekara and S. Lehmann. The strength of friendship ties in proximity sensor data. In *PloS one*, 2014.

[112] H. Semetko and P. Valkenburg. Framing european politics: a content analysis of press and television news. 50:93–109, June 2000.

[113] G. Shani and A. Gunawardana. Evaluating recommendation systems. In *Recommender systems handbook*, pages 257–297. Springer, 2011.

[114] J. Shlens. A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100*, 2014.

[115] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, London, 1986.

[116] B. W. Silverman. Density estimation for statistics and data analysis, 1986.

[117] P. O. Steiner. Program patterns and preferences, and the workability of competition in radio broadcasting. *The Quarterly Journal of Economics*, 66(2):194–223, 1952. URL: https://EconPapers.repec.org/RePEc:oup:qjecon:v:66:y:1952:i:2:p:194-223..

[118] A.-A. Stoica, C. Riederer, and A. Chaintreau. Algorithmic glass ceiling in social networks: the effects of social recommendations on network diversity. In *Proceedings of the 2018 World Wide Web Conference*, pages 923–932, 2018.

[119] Z. Su, A. K. Tung, and Z. Zhang. Supporting top-K item exchange recommendations in large online communities. In *EDBT*, pages 97–108, 2012.

[120] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang. Bert4rec: sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1441–1450, 2019.

[121] S. Suri and D. J. Watts. Cooperation and contagion in web-based, networked public goods experiments. *PloS one*, 6(3):e16836, 2011.

[122] J. Tan, X. Wan, and J. Xiao. A neural network approach to quote recommendation in writings. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 65–74, 2016.

[123] J. Tang, S. Wu, J. Sun, and H. Su. Cross-domain collaboration recommendation. In *KDD*, 2012.

[124] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[125] L. van der Maaten and G. Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9: 2579–2605, Nov 2008.

[126] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[127] L. van der Maaten, G. Hinton, and Y. Bengio. Visualizing data using t-sne. In 2008.

[128] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[129] T. Vizcarrondo. Measuring concentration of media ownership: 1976–2009. *International Journal on Media Management*, 15(3):177–195, 2013.

[130] K. Wallsten. Political blogs and the bloggers who blog them: is the political blogosphere and echo chamber? In 2005.

[131] M. Wan, D. Wang, M. Goldman, M. Taddy, J. Rao, J. Liu, D. Lymberopoulos, and J. McAuley. Modeling consumer preferences and price sensitivities from large-scale grocery shopping transaction logs. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1103–1112, 2017.

[132] M. Wan, D. Wang, J. Liu, P. Bennett, and J. McAuley. Representing and recommending shopping baskets with complementarity, compatibility and loyalty. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1133–1142, 2018.

[133] C. Wang, M. Zhang, W. Ma, Y. Liu, and S. Ma. Modeling item-specific temporal dynamics of repeat consumption for recommender systems. In *The World Wide Web Conference*, pages 1977–1987, 2019.

[134] S. Wang, L. Hu, L. Cao, X. Huang, D. Lian, and W. Liu. Attention-based transactional context embedding for next-item recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32 of number 1, 2018.

[135] W. Wang, W. Zhang, S. Liu, Q. Liu, B. Zhang, L. Lin, and H. Zha. Beyond clicks: modeling multi-relational item graph for session-based target behavior prediction. In *Proceedings of The Web Conference 2020*, pages 3056–3062, 2020.

[136] J. Wu, R. Cai, and H. Wang. Déjà vu: a contextualized temporal attention mechanism for sequential recommendation. In *Proceedings of The Web Conference 2020*, pages 2199–2209, 2020.

[137] L. Wu, F. Petroni, M. Josifoski, S. Riedel, and L. Zettlemoyer. Zero-shot entity linking with dense entity retrieval. In *EMNLP*, 2020.

[138] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, and T. Tan. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33 of number 01, pages 346–353, 2019.

[139] J. Xiao, H. Ye, X. He, H. Zhang, F. Wu, and T.-S. Chua. Attentional factorization machines: learning the weight of feature interactions via attention networks. *arXiv preprint arXiv:1708.04617*, 2017.

[140] L. Xiong, X. Chen, T.-K. Huang, J. Schneider, and J. G. Carbonell. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *Proceedings of the 2010 SIAM international conference on data mining*, pages 211–222. SIAM, 2010.

[141] C. Xu, P. Zhao, Y. Liu, V. S. Sheng, J. Xu, F. Zhuang, J. Fang, and X. Zhou. Graph contextualized self-attention network for session-based recommendation. In *IJCAI*, volume 19, pages 3940–3946, 2019.

[142] J. Yang and J. Leskovec. Overlapping community detection at scale: a nonnegative matrix factorization approach. In *WSDM*, 2013.

[143] W. Ye, S. Wang, X. Chen, X. Wang, Z. Qin, and D. Yin. Time matters: sequential recommendation with complex temporal information. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1459–1468, 2020.

[144] W. Yu, C. C. Aggarwal, and W. Wang. Temporally factorized network modeling for evolutionary network analysis. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 455–464. ACM, 2017.

[145] Y. Zhang and D.-Y. Yeung. Overlapping community detection via bounded nonnegative matrix tri-factorization. In *KDD*, 2012.