Bartering Books to Beers: a Recommender System for Exchange Platforms

Jérémie Rappaz*
LSIR, EPFL
jeremie.rappaz@epfl.ch

Maria-Luiza Vladarean*
LSIR, EPFL
ml.vladarean@gmail.com

Julian McAuley
UC San Diego
jmcauley@eng.ucsd.edu

Michele Catasta
LSIR, EPFL
michele.catasta@epfl.ch

ABSTRACT

Bartering is a timeless practice that is becoming increasingly popular on the Web. Recommending trades for an online bartering platform shares many similarities with traditional approaches to recommendation, in particular the need to model the preferences of users and the properties of the items they consume. However, there are several aspects that make bartering problems interesting and challenging, specifically the fact that users are both suppliers and consumers, and that the trading environment is highly dynamic. Thus, a successful model of 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.

We propose new models for bartering-based recommendation, for which we introduce three novel datasets from online bartering platforms. Surprisingly, we find that existing methods (based on matching algorithms) perform poorly on real-world platforms, as they rely on idealized assumptions that are not supported by real bartering data. We develop approaches based on Matrix Factorization in order to model the reciprocal interest between users and each other's items. We also 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. We evaluate our approach on trades covering books, video games, and beers, where we obtain promising empirical performance compared to existing techniques.

Keywords

barter; swap; exchange; reciprocity; collaborative filtering; social dynamics; temporal dynamics; matrix factorization

*These two authors contributed equally

WSDM 2017, February 06-10, 2017, Cambridge, United Kingdom

© 2017 ACM. ISBN 978-1-4503-4675-7/17/02...$15.00

DOI: http://dx.doi.org/10.1145/3018661.3018696

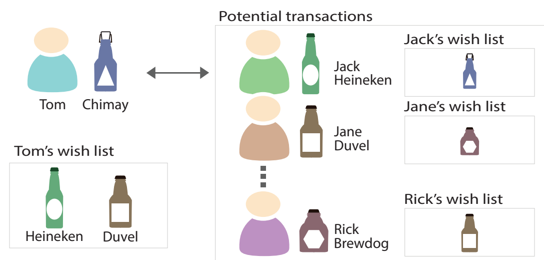


Figure 1: Illustration of the problem setting in which a user (Tom) can exchange an item with owners of other items available on the platform (assuming the recipient has a reciprocal interest in the item being given away).

1. INTRODUCTION

At its inception, the economy is assumed to have been barter-based [6]. 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 21^{st} century consumers [9]. 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,[1] 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.

[1]There exist some other platforms that employ a trade matching method, like **barterquest.com** [5], but their data was inaccessible. Their matching method, however, does not involve user preference modeling.

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 [1, 2, 27] 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 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 paper we propose a model based on Matrix Factorization [12] 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 [1, 2, 13, 19, 27]. We compare against a state-of-the-art item exchange method [27], 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.

Notation	Description
R	Interaction matrix $\in \mathbb{R}^{\lvert U\rvert \times \lvert I\rvert}$
I	Item set
U	User set
u_j	User $u_j \in U$
i_k	Item $i_k \in I$
$r_{u_j i_k}$	Entry in R (for user u_j and item i_k)
W_j	Wish list of user u_j
G_j	Give-away list of user u_j
H_j^g	History of given item for user u_j
H_j^r	History of received item for user u_j
$\hat{y}_{u_j i_k}$	Predicted preference of user u_j for item i_k
$\mathbb{1}$	Heaviside step function

Table 1: Notation

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 [22, 23]. 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.* [22], using the The Top Trading Cycles and Chains mechanism. Another relevant work is that of Haddawy *et al.* [8], 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 [16] 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.* [27], 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 [10]. **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 [14], and temporal dynamics in recommender systems [11].

Bayesian Personalized Ranking (BPR). Bayesian Personalized Ranking is a *pairwise* optimization procedure proposed by Rendle *et al.* [21], 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 [21].

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.

3. DATA ANALYSIS

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

1. **Swapacd**[2] is a CD exchange platform.
2. **Swapadvd**[3] is a DVD exchange platform.
3. **ReaditSwapit**[4] is a book exchange platform.

[2] http://www.swapacd.com
[3] http://www.swapadvd.com
[4] http://www.readitswapit.co.uk

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 2: 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.

4. **Bookmooch**[5] is a book exchange platform.
5. **Ratebeer**[6] is a beer exchange (and rating) platform.
6. **/r/gameswap**[7] is a self-organized subreddit made for users to exchange video games.

Basic statistics of the datasets are shown in Table 2. 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 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).[8]

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 Figure 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' [17] on the platforms. **Swapacd**, **Swapadvd** and **Read-**

[5] http://www.bookmooch.com
[6] http://www.ratebeer.com
[7] http://www.reddit.com/r/gameswap
[8] All the datasets are available at *http://swapit.github.io/*.

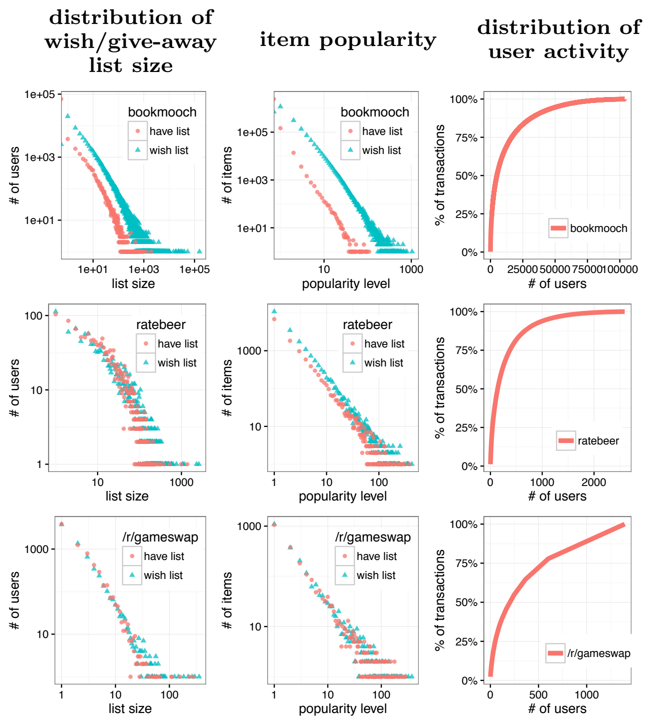


Figure 2: The distribution of item list sizes and item popularity approximately follows a power-law for the three considered platforms (top: **Bookmooch**; middle: **Ratebeer**; bottom: **/r/gameswap**). The CDF plots on the right show user activity in terms of the number of swap transactions each user participates in. The presence of power users, who account for a majority of the transactions, is apparent on all three platforms.

itSwapit, 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 2, 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.[9]

[9]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.

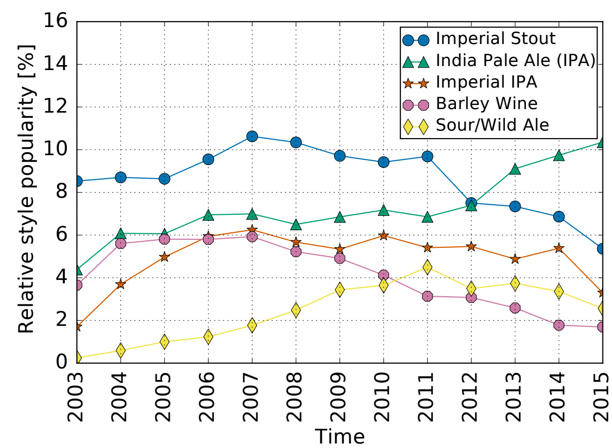


Figure 3: Evolution in popularity of the top-5 beer styles (using the **Ratebeer** dataset).

One may note in Table 2 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** [27], 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.'

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 Figure 3 we observe how beer styles evolve in popularity, measured as their trade frequency over time. For example,

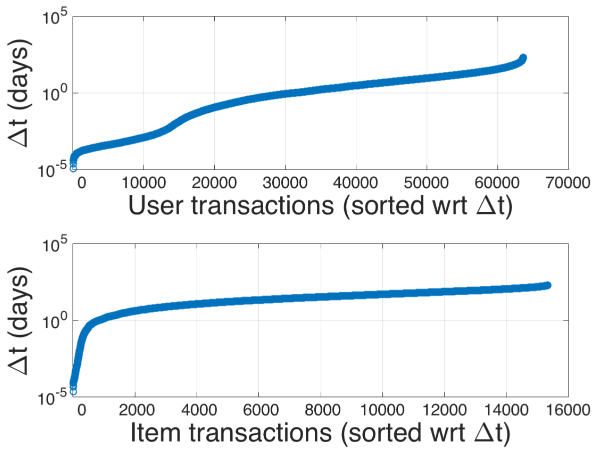


Figure 4: Cumulative frequency plot of the transactions performed on Bookmooch. Note that while there is a core of power users who perform multiple transactions per day (i.e., $\Delta t < 10^0$), most of the items are swapped infrequently (i.e., a few times per year).

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 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). Figure 4 shows the existence of items and users more actively involved in trading, as opposed to others with less frequent interaction.

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**) [27] 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 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.

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

Table 3: Number of **BVEM** recommendations for various values of the price matching parameter β. Note that each recommendation is made to two distinct users.

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** [27], 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 2). 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** [27] on the **Bookmooch** dataset, as it is the only one providing item pricing information as required by the model. Table 3 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 2). 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** [27] 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 2). 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.

4. MODEL

4.1 Problem Definition and Notation

Our notation is defined in Table 1.

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.

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 [18] when building the preference model.

Following the approach proposed by Hu *et al.* [10], 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:

$$\hat{y}_{u_j,i_k} = p_{u_j}^T q_{i_k} \tag{1}$$

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 5.1, 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.

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. [14], [7]). As discussed in Section 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 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:

$$\hat{y}_{u_j,u_l,i_k} = p_{u_j}^T q_{i_k} + s_{u_j u_l} \tag{2}$$

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.

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 [11]. Motivated by the observations made in Section 3, we extend our model from Equation 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* [24]. **KDE** is a non-parametric method for estimating the Probability Density Function from a set of i.i.d. samples, under weak smoothness assumptions. Equation 3 represents such a density estimator for the sample set $\{x_1, x_2, \dots x_n\}$:

$$\delta(x; \bar{x}) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i) = \frac{1}{nh}\sum_{i=1}^{n} K\Big(\frac{x - x_i}{h}\Big) \tag{3}$$

For our purposes, we set $K(x) = \frac{1}{\sqrt{2\pi}} exp(-\frac{1}{2}x^2)$, i.e., the Gaussian kernel [15, 24]. Parameter h represents the bandwidth, and can be set according to Silverman's rule of thumb [26], 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. Equation 4 represents our final model, which includes the social bias and the temporal terms:

$$\hat{y}_{u_j,u_l,i_k} \;=\; p_{u_j}^T q_{i_k} + \overbrace{s_{u_j u_l}}^{\text{social bias}} + \underbrace{\tau_{u_j}\delta(t;\bar{t}_{u_j}) + \tau_{i_k}\delta(t;\bar{t}_{i_k})}_{\text{temporal dynamics}} \quad (4)$$

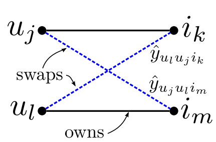
where t is the timestamp of the transaction sample, and $\bar{t}_{u_j}$ and $\bar{t}_{i_k}$ are time points of activities for trades involving user u_j, and item i_k (respectively).

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:

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



(the basic idea of the reciprocal interest model is depicted on the right). 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 [20], but found that the arithmetic mean was consistently the best performing one.

5. EXPERIMENTS AND DISCUSSION

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 technique introduced by Rendle *et al.* [21] is designed for this type of optimization problem. Following the notation of Rendle *et al.* [21] the update rules for this setting are defined as

$$\theta \leftarrow \theta + \alpha \cdot (\sigma(-\hat{x}_{u_j i_k i_m}) \frac{\partial \hat{x}_{u_j i_k i_m}}{\partial \theta} + \lambda_\theta \Omega'(\theta)), \quad (5)$$

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 [21]. The update from Equation 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 4.2—4.5).

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) [25] as our measure of performance:

$$\text{AUC} = \frac{1}{|E|} \sum_{(u_j,i_m,i_n)\in E} \mathbb{1}(\hat{y}_{u_j i_m} - \hat{y}_{u_j i_n}) = \frac{1}{|E|} \sum_{(u_j,i_m,i_n)\in E} \mathbb{1}(\hat{x}_{u_j i_m i_n}), \quad (6)$$

where $\mathbb{1}$ is the Heaviside function (the latter formula uses the notation introduced in Section 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.

5.3 Experiments

Experimental Setup. Experiments were performed on a single machine running Matlab R2015b. Following the methodology proposed by Rendle *et. al* [21], 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

Dataset	(1) MF	(2) MF+B	(3) MF+B+S	(4) MF+B+T	(5) MF+B+S+T	(6) B impr.	(7) B+S impr.	(8) B+T impr.	(9) Total impr.
Bookmooch	0.758	0.798	0.849	0.938	**0.958**	+2.0%	+9.15%	+18.06%	+19.98%
/r/gameswap	0.790	0.842	0.863	0.890	**0.903**	+5.19%	+7.31%	+9.99%	+11.29%
Ratebeer	0.824	0.892	0.962	0.969	**0.983**	+6.79%	+13.84%	+14.55%	+15.87%

Table 4: 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.

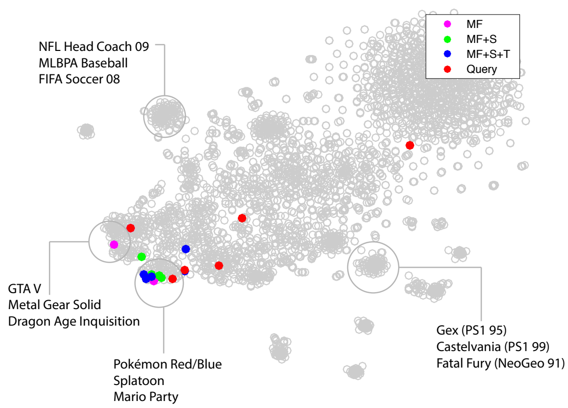


Figure 5: t-SNE [28] embedding of items' latent factors from the */r/gameswap* dataset. Colored dots show the projection of recommendations in Table 5.

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 4 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 4 summarizes the performance of the various instances of our approach. On average, our method outperforms 'vanilla' matrix factorization by 15.71% across the 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 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.

5.4 Discussion

The astute reader will notice that **BVEM** is missing from Table 4. 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 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.

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 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, illustrating the performance of our method with temporal and social constraints. These recommendations are also visualized in terms of a t-SNE embedding [28] in Figure 5. 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

User's wish list	MF			MF + S			MF + S + T		
	Recommendations (ranked)	#own	most recent activity	Recommendations (ranked)	owner activity	past trans.	Recommendations (ranked)	owner activity	past trans.
Super Mario World									
Sonic Generation									
Kirby's Dream Land	Sonic Generations	19	56 wks.	Kid Icarus	24 wks.	2	Fire Emblem	<1 wk.	0
Metroid: Zero Mission	Earthbound	14	22 wks.	Final Fantasy	24 wks.	2	Contra	<1 wk.	0
Super Mario 64	Super Mario Sunshine	26	22 wks.	Beyond: Two Souls	24 wks.	2	Monster Hunter	<1 wk.	1
Mario Kart: Super Circuit	Grand Theft Auto V	253	<1 wk.	Fire Emblem	92 wks.	1	Bayonetta 2	<1 wk.	1
Sly 3: Honor Among Thieves	Fire Emblem	28	<1 wk.	Paper Mario	92 wks.	1	Mario Kart 7	<1 wk.	1

Table 5: An example of recommendations produced by the models from Table 4. **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 Figure 5. 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.

likely to be preferred by the user, based on preference modeling. **CSEM** [2] does not produce such a ranking, and while **BVEM** [27] 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 2. 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.

6. CONCLUSIONS 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.

References

[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] P. Bocheck. System and method for finding potential trading partners in both two-party and multi-party scenarios, Feb. 4 2014. US Patent 8,645,203.

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

[7] 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.

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

[9] 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.

[10] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM*, pages 263–272, 2008.

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

[12] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, pages 30–37, 2009.

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

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

[15] 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.

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

[17] 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.

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

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

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

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

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

[23] 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.

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

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

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

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

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