Collaborating with the Long-Tail: Tackling the Sparsity and Skew of Behavioral Data in Recommendation and User Modeling

Adit Krishnan

University of Illinois at Urbana-Champaign, USA
aditk2@illinois.edu

1 An Overview of the Proposed Thesis

The proposed thesis is centered on personalized recommendation and user profiling on a wide-range of dynamic online platforms where behavioral observations for users may be limited in volume and quality. In recent times, a wide-range of media platforms, on-demand services, e-commerce and other consumer facing platforms have incorporated social elements and content-creation, soliciting user participation in dynamic interactive settings. For instance, users on the Yelp platform participate in a follower-followee network [1] where users may create and interact with review content. Similarly, community question-answer platforms (CQA) incoporate interactions between users and collaboratively authored content [2], often over diverse domains and discussion threads.

This setting poses new and exciting challenges in dealing with the scale and multi-modality of behavioral telemetry. Further, the economic aspects and potential of these platforms are changing and small transactions are increasingly profitable at scale. Thus, consumer models prioritize serving a diverse large pool of users, most of who produce limited interaction data for inferencing, with an ever-expanding inventory of content and products in a context-driven personalized environment. Dealing with skew in the user population also has ties to the emerging domain of fairness in AI. We identify two critical avenues for progress.

First, we must generate personalized inference with limited user-level interaction data, although aggregate volumes of data are often very extensive. Second, our frameworks and models must be malleable and adaptive to keep pace with the rapid influx of users, new and varied content, and the addition of new services and applications on these platforms. These two challenges form the basis of this dissertation. The unifying theme is to *do more with less* in the context of user profiling with the confluence of new and emerging themes in Machine Learning, Information Retrieval and AI.

1.1 Tackling the Sparsity and Skew of Behavioral Data

In the presence of dynamic, multi-faceted observations of users (where facets are data modalities such as clicks, image views, video playbacks etc.), the sparsity

[1] https://neo4j.com/docs/graph-algorithms/current/yelp-example/
[2] https://stackoverflow.com/

problem is exacerbated by the large resultant space of user activity [21]. Further, no one view is sufficient, we might need to consider different subsets of the available behavioral data and the associated data facets for different applications. While a pre-determined set of data facets could help us leverage domain knowledge to address sparsity (e.g., how are clicked images linked to videos the user watches), it does not generalize to other facets or a different prediction objective.

Further, long tailed distributions are a fundamental characteristic of human activity, owing to the bursty nature of human attention [1]. As a result, skew is often observed in data facets that involve human interaction. For instance, specialized topics have a smaller number of followers and even fewer active authors in Community Q&A forums [28]. This has a strong impact on profiling models, they are effective on the active subset of users who display more common behavioral traits while proving ineffective on long-tail users.

The third and fourth chapters are dedicated primarily to data-driven modeling solutions to these challenges. In Section 3, we identify the key connection between sparsity and skew. Identifying more informative groups in the presence of skew helps us bridge the lack of data for individual users, while the converse is also true, better inference for sparse users would help us create more coherent groups to begin with. In Section 4, we zoom in on the skew challenge, focused on the inventory side of recommendation. We propose an architecture agnostic adversarial framework to guide neural models with time-evolving penalties when the recommender fails to identify personalized niche (or long-tail) items for users, given their purchase histories and global item co-occurrences. In effect, we learn the hardest aspects of the entity association structure as the model is trained, and then apply this learned knowledge towards bridging these gaps in the training process. This strategy learns-to-learn, generalizes classical neighbor models [38], i.e., it adaptively identifies and focusses on the hardest samples in the item association structure.

1.2 Malleable and Adaptive Recommendation Frameworks

Malleable frameworks for model development are easier to adapt to new application scenarios and recommendation objectives. Under model malleability and adaptability, we focus on two key angles in Section 5 and Section 6. The first is the explicit *multi-modal* data setting where users interact and generact multiple discrete modes of data [85]. On most platforms, there is a central or primary mode such as item purchases on e-commerce platforms, however social links and reviews could be secondary or auxiliary modes of purchase data. The second angle is that of generalizing to more than one platform or dataset, i.e., *platform agnostic modeling*, where there is a shared mode of data. For instance, two very different e-commerce platforms could still share contextual purchase data as a common data modality even if the user and item sets do not overlap.

Simulataneous progress in both directions, multi-modal integration as well as platform agnostic modeling, will result in the most malleable or flexibile frameworks to build and train user-profiling models. Not only in terms of the types of user-generated data, but also across datasets, such as sparse and dense datasets

or platforms. Finally, for future work we aim to explore the gradient feedback obtained by training recommender models as a transferrable latent factor in the absence of explicitly shared data modalities. In effect, our work aims to overcome the weaknesses of brittle one-time models. As recommendation morphs into contextualization, personalized search and behavior modeling at scale, we expect the central themes of this thesis to be increasingly relevant in such a setting.

2 Related Work

While collaborative recommendation itself has attracted massive volumes of work, mostly directed towards neural recommendation models in recent times [16], [34], addressing data sparsity has proceeded along a few traditional routes which we discuss below. On the other hand, the modeling implications of the pareto nature of behavioral data, especially when segmented by user preferences are relatively unexplored. Malleability and adaptation of the trained models is a central theme in gradient-based meta-learning [10], but we will discuss in greater detail the challenges associated with gradient-based meta-learning in recommendation.

Clustering is one common way to address activity sparsity by modeling behaviors at the level of entity groups; representative methods include cluster-based smoothing [78], user-item co-clustering [76], and joint clustering and collaborative filtering [43]. However, clustering in the presence of behavior skew can lead to uninformative results, *e.g.*, Sato et al. [58] show that when topic models do not account for activity skew, the resulting topics are less descriptive. In contrast to explicit clustering, we explore implicit data-driven entity groups to regularize representations learnt by base neural recommenders. Beutel et al. [3] propose a bayesian approach with Pitman-Yor priors to group users with limited history and capture skewed product ratings; while this approach can capture aggregate skew in cluster size, it does not alleviate interaction sparsity.

Cross-domain recommendations via shared entities is a popular route to alleviate sparsity, where transfer-learning methods have found partial success in mitigating interaction sparsity. In the pairwise user-shared (or item-shared) cross-domain setting, the interaction structure in the dense domain is leveraged to improve recommendations in the sparse domains. State-of-the-art techniques include co-clustering via shared entities [45], [69], structure transfer to align the principal components of the user and item subspaces [49], [11], [51], or a hybrid approach involving both [18]. However, existing methods are limited to pairs of domains with shared entities, and do not scale to the many-sparse-target setting. In this proposal, we move beyond shared entities, to investigate the more ambitious non-overlapping scenario, *e.g.,* meta-transfer grounded on interaction context, and moment consistencies to facilitate nothing-shared model transfer.

Recent work to address activity skew with external data include social [40], group-based [6], knowledge-aware recommendations [67]. Jiang et al. [22] propose sparsity-aware tensor factorization for user behavior analysis, to regularize user representations with auxiliary data source(s) (*e.g.*, author-author citations in academic networks); however quadratic scaling (in the number of

entities) imposes severe computational limits on such methods. Prior efforts to integrate social structure in the latent interest space primarily employed static hypotheses [38], [20], while being unable to explicitly prioritize specific preferences originating from different contacts based on the context. [6], [82] address group recommendation through multi-task learning over individual and group interactions; these methods do not account for skew in group (and user) interactions, which results in models that over-fit to either data source. Our approach is fundamentally different: we propose regularization strategies that are not only agnostic to the models used in each source, but also enhance expressivity to contextually utilize relevant information from each data source.

There is limited work that addresses recommendation systems using n-ary ($n \geq 3$) information sources (i.e., **multi-modal data**). Zhang et al. [85], [35] introduce multi-task learning frameworks that integrate source-specific neural representation models though static regularizers. Instead, we adopt contextually weighted regularizers that either align or disentangle the central information source with the auxiliary sources.

An alternative view of developing models for information sources, is to define a **Heterogeneous Information Network** (HIN) [61], that includes the interactions among the entities of different types, across multiple sources. A few recent efforts by Wang et al. [68], [67] utilize Graph Neural Networks (GNNs) [77] to synthesize information from the connectivity in HINs, thus enriching entity representations. However, existing GNN implementations cannot scale to large-scale recommendation settings with multiple sources, since they either store the entire graph in GPU memory [24], [65] (infeasible for real-world applications) or incur expensive neighbor sampling costs at each layer [14].

Building malleable representation models is also a central theme in gradient-based meta-learning. Recent work [10], [33] employs the gradient magnitude of a base-learner model as a measure of its plasticity for few-shot adaptation (i.e., with a small number of samples) to multiple semantically similar tasks. However, the base-learner is often constrained to simpler architectures (such as shallow neural networks) to prevent overfitting [62] and furthermore, requires multi-task gradient feedback at training time [10]. This strategy does not scale to the embedding learning problem in Collaborative Filtering. This strategy does not scale to the embedding learning problem in latent-factor collaborative recommendation, especially in the *many-sparse-target* setting. Instead, in Section 6, we incorporate the core strengths of meta-learning and transfer learning grounded on contextual predicates in recommendation.

3 Unified Mitigation of Behavioral Skew and Sparsity

This chapter addresses the challenge of learning robust statistical representations of participant behavior on online social networks. Graphical behavior models have found success in several social media applications: content recommendation [54], [80], behavior prediction [55], [84], user characterization [44] and community profiling [5]. Despite the large sizes of these social networks (e.g. several

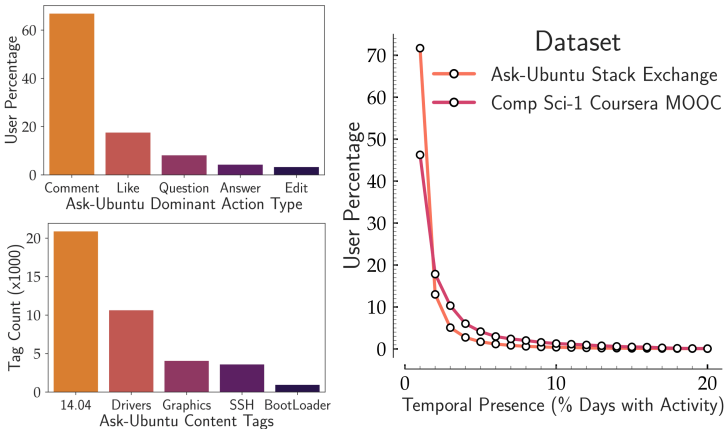
million users), developing robust behavior profiles is challenging due to the heavy tailed nature (a small set of users account for most interactions) with temporally sparse users. Furthermore, user activity styles and topical interests are highly skewed within the population, complicating the inference of prototypical behavior types. Figure 1 shows a typical example of behavior skew and temporal sparsity in `AskUbuntu`[3], a popular online Q&A forum.

Past work addresses one of the challenges (either sparsity or skew) separately in graphical behavior models, but do not adopt a unified approach to learn representations. Clustering is one common way to address sparsity [79], [56]. However, using clustering techniques in the presence of behavior skew can lead to uninformative results. For example, when topic models do not account for skew (e.g. Zipf's law), the resulting topics are less descriptive [59]. The use of suitable priors such as the Pitman-Yor process [52] (visualized via Chinese Restaurant Process; CRP) over the cluster sizes is a way to deal with skew [4]. However, a direct application of the CRP prior to behavior models cannot address sparsity since behavior profiles are still learnt at the user-level and inactive users degrade the ability to learn robust latent representations.

Our main technical insight is to simultaneously address behavior skew and temporal sparsity of inactive users. Our key innovation in addressing sparsity and behavior skew lies in how we "seat" users onto tables. In effect, we adopt three concrete lines of attack. Profiles should be learned from data at the granularity of a table (or equivalently, a group of users), *not* at the user-level, behavioral similarity should guide user seating on these tables and we discount common behavioral profiles to identify niche behaviors in the presence of skew. We refer to our model as CMAP (CRP-based Multi-facet Activity Profiling) in the rest of this paper.

[3] `https://askubuntu.com/`

Fig. 1: Dominant Action Types and Content are highly skewed in Ask-Ubuntu. User presence also exhibits steep power-law ($\eta \approx 3$) indicating several inactive users. Behavioral skew and data sparsity are both prominent challenges.



To summarize, we propose a partitioning scheme that adapts to varying levels of behavior skew to uncover niche behavior profiles and simultaneously addresses user-level sparsity. Our framework can be adapted to a large class of graphical behavior models that incorporate different facets of data. It is hard to account for the distributional properties of different combinations of facets that varied applications require us to model. We thus employ a non parametric approach, while traditional LDA based models [54] (a popular thread in text mining and behavior modeling) are imherently unsuited to skewed data facets.

3.1 Problem Definition

Let $\mathcal{U}$ denote the aggregate user set. Users employ a set of discrete actions $\mathcal{A}$ to interact with content generated from vocabulary $\mathcal{V}$. A user interaction d (atomic unit of participant activity) is a tuple $d = (a, W, t)$, where the user performs action $a \in \mathcal{A}$ on content $W = \{w_1, w_2 \ldots \mid w_i \in \mathcal{V}\}$ at time-stamp $t \in [0, 1]$ (normalized appropriately). We denote the set of all interactions of $u \in \mathcal{U}$ as $\mathcal{D}_u$. Thus the collection of interactions in the dataset is $\mathcal{D} = \bigcup_{u \in \mathcal{U}} \mathcal{D}_u$.

Inter-participant social links are represented by a directed multigraph $G = (\mathcal{U}, E)$. A directed labeled edge $(u, v, \ell) \in E$ represents an interaction of user u, $d_u \in \mathcal{D}_u$(e.g. "answer") in response to an interaction of user v, $d_v \in \mathcal{D}_v$ (e.g. "ask question") with label $\ell \in \mathcal{L}$ indicating the nature of the exchange (e.g. "answer" $\rightarrow$ "question"). We denote the set of all social interactions in which user u is involved by L_u, so that $E = \bigcup_{u \in \mathcal{U}} L_u$. Our goal is to obtain a set of activity profiles R describing discrete observed behavior types, and infer user representations $\mathcal{P}_u, u \in \mathcal{U}$ as mixtures over the inferred profiles $r \in R$.

3.2 Model Description

Attacking the Skew-Sparsity Challenge: We begin by formally discussing the Pitman-Yor process [52] and then highlight challenges in the presence of sparsity. The conventional Chinese Restaurant arrangement induces a non-parametric prior over integer partitions (or indistinguishable entities), with concentration γ, discount δ, and base distribution G_0, to seat users across tables (partitions). Each user is either seated on an existing table $x \in \{1, \ldots, \chi\}$, or assigned a new table $\chi + 1$ as follows:

$$p(x \mid u) \propto \begin{cases} \frac{n_x - \delta}{N + \gamma}, & x \in \{1, \ldots, \chi\}, \text{ existing table,} \\ \frac{\gamma + \chi\delta}{N + \gamma}, & x = \chi + 1, \text{new table,} \end{cases} \tag{1}$$

where n_x is the user-count on existing tables $x \in \{1, \ldots, \chi\}$, $\chi + 1$ denotes a new table and $N = \sum_{x \in \{1, \ldots, \chi\}} n_x$ is the total user-count. A direct application of Equation (1) as a simple prior can address skew in profile proportions, but not sparsity. To address sparsity we identify three concrete lines of attack: Profiles need to be learned from data at the granularity of a table (or equivalently, a group of users), *not* at the level of an individual; Behavioral similarity should guide seating on these tables; We should discount common behavioral profiles to encourage identification of niche behaviors and improve profile resolution.

Symbol	Description
N, R	Number of seated users, Set of profiles
$\{1,\ldots,\chi\}$, $\chi+1$	Set of existing tables, New table
n_x, r_x	User count on table x, profile served on x
χ_r, N_r	Number of tables serving profile r, Total users seated on tables serving profile r

Our Profile-Driven Seating approach builds upon CRP to simultaneously generate partitions of similar users and learn behavior profiles for these partitions. Consider profiles $r \in R$ describing observed facets of user data with conditional likelihood $p(u \mid r)$ for $u \in \mathcal{U}$. We "serve" a profile $r_x \in R$ to users seated on each table $x \in \{1,\ldots,\chi\}$. A user u is seated on an existing table $x \in \{1,\ldots,\chi\}$ serving profile r_x or a new table $\chi+1$ as follows,

$$p(x \mid u) \propto \begin{cases} \frac{n_x-\delta}{N+\gamma} \times p(u \mid r_x), & x \in \{1,\ldots,\chi\}, \\ \frac{\gamma+\chi\delta}{N+\gamma} \times \frac{1}{|R|}\sum_{r\in R} p(u \mid r), & x = \chi+1. \end{cases} \tag{2}$$

The likelihood $p(x \mid u)$ of choosing an *existing* table $x \in \{1,\ldots,\chi\}$ for user u depends on the conditional $p(u \mid r_x)$ of the profile r_x served on the table and the number of users seated on table x. Further, the seating likelihoods for existing tables depend on the latent profiles served, while the latent profiles r_x are learned from the table x they are served on. This process introduces a mutual coupling between seating and profile learning.

The likelihood of assigning the user to a new table $x = \chi+1$ depends on the sum of conditionals $p(u \mid r)$ with a uniform prior $\frac{1}{|R|}$, and the number of existing tables χ. Notice the effect of the discount factor δ: increasing δ favors exploration by forming new tables. Long-tail users are more likely to be seated separately with a different profile served to them.

Unlike CRP Equation (1), we seat users based on the table size distribution, the profiles served on those tables, and the conditional probability of the user for the behavioral profile. Equation (2) reduces to Equation (1) when all profiles $r \in R$ are equally likely for every user. We can show that our seating process is exchangeable i.e., seating likelihoods are stochastically agnostic to the order of users. When user u is seated on a new table $\chi+1$, we draw profile variable $r_{\chi+1} \in R$ on the new table as follows:

$$p(r_{\chi+1} \mid u) \sim p(u \mid r)p(r),$$

where $p(r)$ is the Pitman-Yor base distribution G_0, or prior over the set of profiles. We set G_0 to be uniform to avoid bias.

The likelihood $p(r \mid u)$ of assigning profile r when seating user u, is proportional to the sum of likelihoods of seating the user on an existing table $x \in \{1,\ldots,\chi\}$ serving profile r (i.e. $r_x = r$), or seating on a new table $\chi+1$

with the profile $r_{\chi+1} = r$. That is:

$$p(r \mid u) \propto \left(\sum_{\substack{x \in \{1,\ldots,\chi\}, \\ r_x = r}} \frac{n_x - \delta}{N + \gamma} p(u \mid r) \right) + \frac{1}{|R|} \cdot \frac{\gamma + \chi\delta}{N + \gamma} p(u \mid r), \tag{3}$$

$$\propto \left(\frac{N_r - \chi_r \delta}{N + \gamma} + \frac{\gamma + \chi\delta}{|R|(N + \gamma)} \right) p(u \mid r), \tag{4}$$

where χ_r is the number of existing partitions serving profile r and N_r is the total number of users seated on tables serving profile r.

Three insights stem from Equation (4). First, the skew in profile sizes depends on the counts of users exhibiting similar behavior patterns ($\propto p(u \mid r)$) enabling adaptive fits unlike [4]. Second, we discount common profiles served on multiple tables by the product $\chi_r\delta$. Since χ_r is larger for common profiles drawn on many tables, we discount common profiles more than niche profiles. This "common profile discounting" enables us to learn behavioral profile variations. Finally, not constraining the number of tables introduces stochasticity in profile learning and encourages exploration. In the next subsection, we introduce our temporal activity profiles $r \in R$ for representing user activity in our datasets.

3.3 Latent Profile Description

Our profiles have two constituents: Actions-word associations ("action-topics"), and temporal distributions over action topics. Each action-topic $k \in K$ models user actions and the associated words, with $\phi_k^{\mathcal{V}}$ (multinomial over vocabulary $\mathcal{V}$) and $\phi_k^{\mathcal{A}}$ (multinomial over actions $\mathcal{A}$). We employ a continuous time model, Beta$(\alpha_{r,k}, \beta_{r,k})$ distributions, over a normalized time span to capture the temporal trend of each action-topic k within *each* profile r. Thus, for any interaction $d = (a, W, t)$, the probability $p(d \mid r, k)$ of a user interaction d given a profile r and topic k is:

$$p(d \mid r, k) \propto \underbrace{\phi_k^{\mathcal{A}}(a) \prod_{w \in W} \phi_k^{\mathcal{V}}(w)}_{\text{'what': profile independent}} \times \underbrace{\frac{t^{\alpha_{r,k}-1}(1-t)^{\beta_{r,k}-1}}{\mathrm{B}(\alpha_{r,k}, \beta_{r,k})}}_{\text{'when': profile dependent}}, \tag{5}$$

where B refers to the beta function. There are K action topics, but $R \times K$ temporal distributions to allow users with different overall behavior to employ the same action-topic. The likelihood $p(d \mid r)$ of user interaction d (as defined in section 3.1) for profile r is:

$$p(d \mid r) \propto \sum_k p(d \mid r, k) \times \phi_r^K(k), \tag{6}$$

where $\phi_r^K(k)$ is a K dimensional multinomial mixture over action-topics for each profile.

Table 1: Reputed User Prediction ($\mu \pm \sigma$ across Stack-Exchanges). We obtain improvements of 6.65-21.43% AUC.

Method	Precision	Recall	F1-score	AUC
LRC [30]	0.73 ± 0.04	0.69 ± 0.04	0.72 ± 0.03	0.73 ± 0.03
DMM [83]	0.69 ± 0.05	0.65 ± 0.04	0.66 ± 0.04	0.70 ± 0.04
LadFG [53]	0.86 ± 0.03	0.75 ± 0.03	0.79 ± 0.02	0.80 ± 0.03
FEMA [21]	0.79 ± 0.04	0.73 ± 0.03	0.77± 0.03	0.79 ± 0.04
BLDA [54]	0.75 ± 0.04	0.71 ± 0.04	0.74 ± 0.03	0.74 ± 0.04
CMAP (Ours)	0.85 ± 0.02	**0.83 ± 0.03**	**0.84 ± 0.02**	**0.86 ± 0.02**

We model social linkages between pairs of behavioral profiles (r, r') (rather than users) motivated by sparsity. Label $\ell \in \mathcal{L}$ describes link type (e.g. Question $\rightarrow$ Answer, Comment $\rightarrow$ Answer etc.) between users (u, v). We set-up $|R|^2$ multinomial distributions over link types $\phi^{\mathcal{L}}_{r,r'}$ between ordered profile pairs (r, r').

Let L_u denote all links from and to user u.

$$p(L_u \mid r) \propto \underbrace{\prod_{(s,u,\ell)\in L_u} \phi^{\mathcal{L}}_{r_s,r}(\ell)}_{\text{inbound exchange}} \times \underbrace{\prod_{(u,y,\ell)\in L_u} \phi^{\mathcal{L}}_{r,r_y}(\ell)}_{\text{outbound exchange}}, \tag{7}$$

where $\phi^{\mathcal{L}}_{r_s,r}(\ell)$ is for an in-link from source user s (profile r_s) to u, and $\phi^{\mathcal{L}}_{r,r_y}(\ell)$ for an out-link from u to target user y (profile r_y).

The overall conditional $p(u \mid r)$ is the product of links $p(L_u \mid r)$ and content interactions $p(D_u \mid r)$:

$$P(u \mid r) \propto p(L_u \mid r) \times \prod_{d\in D_u} p(d \mid r). \tag{8}$$

We combine $p(u \mid r)$ from Equation (8) with $p(x \mid u)$ (Equation (2)) to seat users u on tables x, serving profile r_x.

3.4 Qualitative Evaluation and Analysis

We show strong quantitative and qualitative results on diverse datasets (public Stack-Exchange datasets and Coursera MOOCs[4]). We chose our datasets across technical/non-technical subject domains and varying population sizes, with all datasets seen to exhibit significant behavioral skew and sparsity. We evaluate our model (CMAP) against state-of-the-art baselines and observe that our ability to discover more distinct and descriptive user clusters even with the same latent dimensions as baselines is the primary reason for our performance gains. Our method improves on the baselines in the *reputation prediction task* by 6.26-15.97% AUC averaged across the Stack-Exchanges; Table 1 shows the results with statistically significant improvements in bold. Similarly, we improve on *certification prediction* (see Table 2) by 6.65-21.43% AUC averaged over MOOCs.

[4] `https://stackexchange.com`, `https://coursera.org`

Method	Precision	Recall	F1-score	AUC
LRC [30]	0.76 ± 0.04	0.71 ± 0.05	0.74 ± 0.04	0.72 ± 0.03
DMM [83]	0.77 ± 0.03	0.74 ± 0.04	0.75 ± 0.03	0.74 ± 0.03
LadFG [53]	0.81 ± 0.02	0.78 ± 0.02	0.79 ± 0.02	0.79 ± 0.02
FEMA [21]	0.78 ± 0.03	0.75 ± 0.04	0.76 ± 0.03	0.78 ± 0.03
BLDA [54]	0.80 ± 0.04	0.75 ± 0.03	0.77 ± 0.03	0.77 ± 0.04
CMAP (Ours)	**0.86 ± 0.02**	**0.81 ± 0.03**	**0.83 ± 0.02**	**0.84 ± 0.02**

Table 2: Certificate Earner Prediction ($\mu \pm \sigma$ across MOOCs); CMAP improves upon baselines by 6.65-21.43% AUC

Cluster	Cluster Action Style	Cluster Topics
1	+31% Answer, +24% Edit, −09% Question	Drivers, Boot, Disk Partition
2	+67% Answer, −03% Edit, −21% Question	Gnome, Desktop, Package Install
3	+11% Answer, −04% Edit, +47% Question	Script, Application, Sudo Access

Table 3: Actions and Content in the most reputed user clusters discovered by CMAP on Ask-Ubuntu, +/-% against the average Ask-Ubuntu user.

The Impact of Profile Driven Seating We now compare clusters obtained through CMAP seating against generative assignments in BLDA [54] on StackExchanges. Both models group users best described by the same profile to form clusters. We use average user reputations of the clusters (appropriately normalized) as an external validation metric for cluster quality.

The Dirichlet-Multinomial setting in BLDA tends to merge profiles and hence shift cluster sizes and average participant reputation closer to the mean. Our cluster assignments appear to mirror the behavior skew for `Ask-Ubuntu` in Figure 1. Our approach (CMAP) learns finer variations in the topic affinities and actions of expert users. We can observe these variations in Figure 2 and Table 3. The top three profiles are more reputed, smaller in sizes and each cluster shows distinct user activity (Table 3).

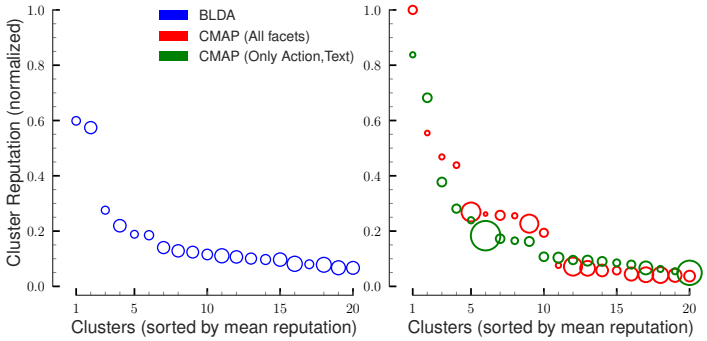


Fig. 2: Bubbles denote user clusters discovered by each model in the Ask-Ubuntu dataset (Bubble size $\propto$ Users). CMAP discovers fine distinctions of reputed users (Table 3) while BLDA clusters show a mean-shift in both size and reputation. Our assignments are reflective of the behavioral skew in the dataset.

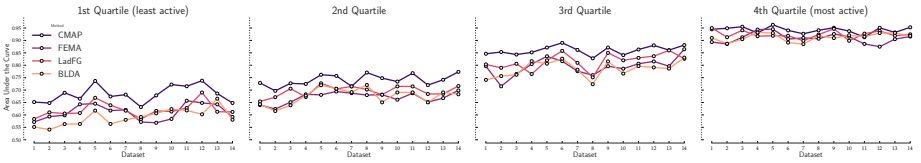


Fig. 3: Effects of activity sparsity on prediction tasks (AUC) for Stack Exchanges (datasets 1-10) and MOOCs (datasets 11-14). CMAP has greatest performance gains in Quartile-1 (sparse users) and minor gains for active users (Quartile-4).

We observe a similar trend in the aggregate clusters obtained on other Stack-Exchange datasets as well. Our performance gains in the prediction and recommendation tasks reflect these underlying improvements in profile quality (Table 1, Table 2).

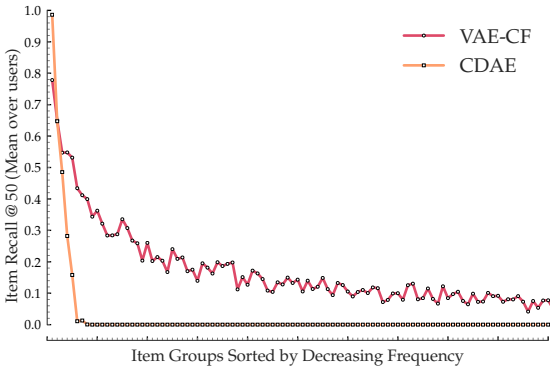
Making gains on inactive users We split users in each Stack-Exchange and MOOC into four quartiles based on interaction count (Quartile 1 is least active, 4 most). Then, we evaluate each method on Reputation and Certificate Prediction AUC in each quartile separately. Our model shows significant performance gains (Figure 3) in Quartiles 1,2 that contain sparse users. We attribute these gains to joint profile learning to describe similar users seated on tables. The decision to address skew and sparsity jointly has two advantages: better profile fits for sparse users; more distinct and informative profiles in skewed scenarios. In contrast, models building representations at the user level perform weakly in Quartiles-1,2 since these methods rely on interaction volume. As expected, performance differences between all models are smaller in the data-rich quartiles 3,4.

4 Tackling Skew via Adversarial Association Learning

Collaborative filtering (CF) methods personalize item recommendations based on historic interaction data (implicit feedback setting), with matrix-factorization being the most popular approach [27]. In recent times, Neural CF methods have transformed simplistic inner-product representations with non-linear interactions, parametrized by deep neural networks [16]. Although performance gains over conventional approaches are significant, a closer analysis indicates skew towards popular items that frequently appear in the feedback, resulting in poor niche (long-tail) item recommendations to users (see fig. 4). This stifles user experience, recommendation diversity and could hurt platform revenue and online market fairness.

Given the diversity of NCF architectures and applications [34],[16],[32], architectural solutions are hard to generalize. Instead, we propose to augment NCF training to levy penalties when the recommender fails to identify suitable niche items for users, given their history and global item co-occurrence. Conventional neighbor models do this via static pre-computed links between entities

Fig. 4: CDAE[74] and VAE-CF[34] recall for item-groups (decreasing frequency) in MovieLens (*ml-20m*). CDAE overfits to popular item-groups, falls very rapidly. VAE-CF has better long-tail recall due to representational stochasticity.



[41] to regularize their representations. While we can add a similar term to the NCF objective, we aim to learn the association structure rather than imposing it on the model. Towards this goal, we introduce an adversary network, trained in tandem with the recommender, to infer the inter-item association structures unlike link-based models, guided by item co-occurrences in the feedback data. It can condition the feedback on auxiliary data if required or be extended to incorporate other associations.

For each user, a penalty is imposed on the recommender if the suggested niche items do not correlate with the user's history. The adversary is trained to distinguish the recommender's niche item suggestions against actual item pairings sampled from the data. The more confident this distinction, the higher the penalty imposed. As training proceeds, the adversary learns the inter-item association structure guided by the item pairs sampled from user records while the recommender incorporates these associations, until mutual convergence. Further, our approach is completely architecture and application agnostic, thus satistfying our broad malleable framework objective.

4.1 Problem Definition

We consider the implicit feedback setting with interaction matrix $\mathcal{X} \in Z_2^{M_\mathcal{U} \times M_\mathcal{I}}$,

$$Z_2 = \{0, 1\}$$

given users $\mathcal{U} = \{u_1, \ldots, u_{M_\mathcal{U}}\}$, items $\mathcal{I} = \{i_1, \ldots, i_{M_\mathcal{I}}\}$. Items $\mathcal{I}$ are partitioned apriori into two disjoint sets, $\mathcal{I} = \mathcal{I}^\mathcal{P}$ (*popular items*) $\cup\, \mathcal{I}^\mathcal{N}$ (*niche items*) based on their frequency in $\mathcal{X}$. We use $\mathcal{X}_u$ to denote the set of items for $u \in \mathcal{U}$, split into popular and niche subsets $\mathcal{X}_u^\mathcal{P}$, $\mathcal{X}_u^\mathcal{N}$. The base neural recommender $\mathbf{G}$ learns a scoring function $f_\mathbf{G}(i \mid u, \mathcal{X}), i \in \mathcal{I}, u \in \mathcal{U}$ to rank items given u's history $\mathcal{X}_u$ and global feedback $\mathcal{X}$, by minimizing CF objective $\mathcal{O}_\mathbf{G}$ over recommender $\mathbf{G}$'s parameters θ via stochastic gradient methods. Typically, $\mathcal{O}_\mathbf{G}$ is composed of a reconstruction loss (like the conventional inner product loss [27]) and a regularizer depending on the architecture. We adopt $\mathbf{O_G}$ as the starting point

in our training process. Our goal is to enhance the long-tail performance of recommender $\mathbf{G}$ on the niche items $\mathcal{I}^{\mathcal{N}}$.

4.2 Adversarial Formulation

Our key insight is that generating item recommendations for user u, and modeling the associations of recommended niche items to his history $\mathcal{X}_u$ are mutually linked tasks. The adversarial paradigm [13] fits our application well, we seek to balance the tradeoff between the biased reconstruction objective against the recall and accuracy of long-tail recommendations.

Towards the above objective, we introduce an adversary model $\mathbf{D}$ to learn the inter-item association structure in the feedback data and correlate $\mathbf{G}$'s niche item recommendations with popular items in the user's history, $\mathcal{X}_u^{\mathcal{P}}$. The adversary $\mathbf{D}$ is trained to distinguish "fake" or synthetic popular-niche item pairs sampled from $X_u^{\mathcal{P}}$ and $f_{\mathbf{G}}(i \mid u, \mathcal{X})$ against "real" popular-niche pairs sampled from global co-occurrences in $\mathcal{X}$. The more confident this distinction by $\mathbf{D}$, the stronger the penalty on $\mathbf{G}$. To overcome the applied penalty, $\mathbf{G}$ must produce niche item recommendations that are correlated with the user's history. The model converges as the synthetic and true niche-popular pairs align.

True & Synthetic Pair Sampling "True" popular-niche item pairs $(i^p, i^n) \in \mathcal{I}^{\mathcal{P}} \times \mathcal{I}^{\mathcal{N}}$ are sampled from their global co-occurrence counts in $\mathcal{X}$. To achieve efficiency, we use the alias table method [31] ($O(1)$ amortized cost) compared to $O(\mathcal{I}^{\mathcal{P}} \times \mathcal{I}^{\mathcal{N}})$ for standard sampling. We will denote the true distribution of pairs from $\mathcal{X}$ as $p_{true}(i^p, i^n)$.

Synthetic pairs $(\tilde{i^p}, \tilde{i^n}) \in \mathcal{I}^{\mathcal{P}} \times \mathcal{I}^{\mathcal{N}}$ are drawn per user with $\tilde{i^n} \propto f_{\mathbf{G}}(\tilde{i^n} \mid u, \mathcal{X})$, and $\tilde{i^p}$ randomly drawn from $\mathcal{X}_u^{\mathcal{P}}$. The number of synthetic pairs drawn for user u is in proportion to $|\mathcal{X}_u^{\mathcal{P}}|$. We denote the resulting synthetic pair distribution $p_\theta(\tilde{i^p}, \tilde{i^n} \mid u)$ since it depends on u and parameters θ of the recommender $\mathbf{G}$.

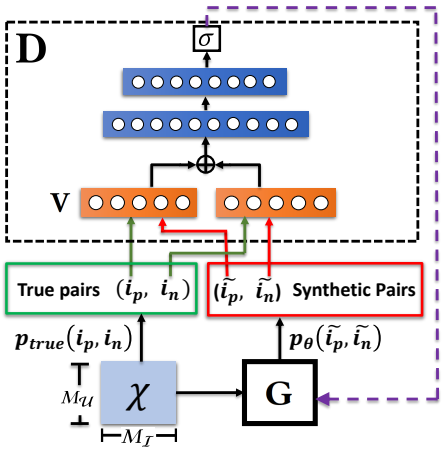
Discriminative Adversary Training The adversary $\mathbf{D}$ compares synthetically generated item pairs $(\tilde{i^p}, \tilde{i^n})$ across all users with an equal number of true pairs (i^p, i^n) sampled as above. It learns latent representations $\mathbf{V} = [\mathbf{v}_i, i \in \mathcal{I}]$ for all items with dimensionality d and a discriminator function $f_\phi(i^p, i^n)$, simultaneously with $\mathbf{V}$, to estimate the probability of a pair (i^p, i^n) being drawn from $p_{true}(i^p, i^n)$.

$$\mathbf{D}_\phi(i^p, i^n) = \sigma(f_\phi(i^p, i^n)) = \frac{1}{1 + \exp(-f_\phi(\mathbf{v}_{i^p}, \mathbf{v}_{i^n}))}$$

We implement $\mathbf{D}_\phi$ via two simple symmetric feedforward ladders followed by fully connected layers (Figure 5). With the parameters of $\mathbf{G}$ (*i.e.,* θ) fixed, ϕ and $\mathbf{V}$ are optimized by stochastic gradient methods to maximize the log-likelihood of true pairs, while minimizing that of synthetic pairs with balance parameter μ,

$$\phi^*, \mathbf{V}^* = \arg\max_{\phi} \sum_{u \in \mathcal{U}} E_{(i^n, i^p) \sim p_{true}(i^p, i^n)} \left[\sigma(f_\phi(i^p, i^n))\right] + \mu . E_{(\tilde{i^p}, \tilde{i^n}) \sim p_\theta(\tilde{i^p}, \tilde{i^n} | u)} \left[\log(1 - \sigma(f_\phi(\tilde{i^p}, \tilde{i^n})))\right] \quad (9)$$

Fig. 5: Architecture details for the discriminative adversary D trained in tandem with base recommender G



Recommender Model Training The more confident the distinction of the fake pairs generated as $(\tilde{i^p}, \tilde{i^n}) \sim p_\theta(\tilde{i^p}, \tilde{i^n} \mid u)$ by adversary $\mathbf{D}$, the stronger the penalty applied to $\mathbf{G}$. As previously described, synthetic pairs $(\tilde{i^p}, \tilde{i^n})$ are drawn as $\tilde{i^n} \propto f_\mathbf{G}(\tilde{i^n} \mid u, \mathcal{X})$, and $\tilde{i^p}$ randomly drawn from $\mathcal{X}_u^\mathcal{P}$. Thus,

$$p_\theta(\tilde{i^p}, \tilde{i^n} \mid u) \propto \frac{1}{|\mathcal{X}_u^\mathcal{P}|} f_\mathbf{G}(\tilde{i^n} \mid u, \mathcal{X}) \quad (10)$$

For sanity, we shrink $p_\theta(\tilde{i^p}, \tilde{i^n} \mid u)$ as $p_\theta(u)$ in the following equations. We reinforce the associations of niche items recommended by $\mathbf{G}$ to the popular items in user history. This is achieved by maximizing $\mathbf{D}_\phi(\tilde{i^p}, \tilde{i^n})$, i.e., the synthetic pairs cannot be distinguished from the true ones. Thus, there are two terms in the recommender's loss, the base objective $\mathcal{O}_\mathbf{G}$ and the weighted adversary term. $\mathbf{D}$'s parameters $\phi, \mathbf{V}$ are held constant as $\mathbf{G}$ is optimized (alternating optimization schedule).

$$\begin{aligned} \theta^* &= \arg\max_\theta -\mathcal{O}_\mathbf{G} + \lambda \sum_{u \in \mathcal{U}} E_{(\tilde{i^p}, \tilde{i^n}) \sim p_\theta(u)} \left[\log D(\tilde{i^p}, \tilde{i^n})\right] \quad (11) \\ &= \arg\min_\theta \mathcal{O}_\mathbf{G} + \lambda \sum_{u \in \mathcal{U}} E_{(\tilde{i^p}, \tilde{i^n}) \sim p_\theta(u)} \left[\log(1 - D(\tilde{i^p}, \tilde{i^n}))\right] \end{aligned}$$

Table 4: Composition of top-100 item recommendations to users in item popularity quartiles (Q1-Most Popular Items)

Method	ml-20m				Ask-Ubuntu			
	Q-1	Q-2	Q-3	Q-4	Q-1	Q-2	Q-3	Q-4
CDAE ($\mathbf{G}_1$)	74%	26%	0%	0%	97%	3%	0%	0%
$\mathbf{D}$+$\mathbf{G}_1(\lambda = 0.1)$	61%	23%	10%	6%	76%	14%	7%	3%
$\mathbf{D}$+$\mathbf{G}_1(\lambda = 1)$	62%	21%	11%	6%	73%	16%	6%	5%
$\mathbf{D}$+$\mathbf{G}_1(\lambda = 10)$	61%	19%	12%	8%	65%	19%	11%	5%
VAE-CF ($\mathbf{G}_2$)	64%	24%	8%	4%	60%	25%	9%	6%
$\mathbf{D}$+$\mathbf{G}_2(\lambda = 0.1)$	58%	23%	12%	7%	53%	25%	12%	10%
$\mathbf{D}$+$\mathbf{G}_2(\lambda = 1)$	59%	21%	13%	7%	55%	21%	13%	11%
$\mathbf{D}$+$\mathbf{G}_2(\lambda = 10)$	59%	20%	13%	8%	54%	22%	14%	10%

Before adversarial training, $\mathbf{G}$ can be pre-trained with loss $\mathcal{O}_{\mathbf{G}}$, while $\mathbf{D}$ can be pre-trained with just the maximization term for true pairs. Our overall objective can be given by combining eq. (9), eq. (11),

$$\mathcal{O} = \min_{\theta} \max_{\phi} \mathcal{O}_{\mathbf{G}} + \lambda \sum_{u \in \mathcal{U}} E_{(i^p, i^n) \sim p_{true}(i^p, i^n)} \left[\log D_{\phi}(i^p, i^n) \right] + \mu . E_{(\tilde{i^p}, \tilde{i^n}) \sim p_{\theta}(\tilde{i^p}, \tilde{i^n} | u)} \left[\log(1 - D_{\phi}(\tilde{i^p}, \tilde{i^n})) \right]$$

On the whole, our framework is a minimax strategy for iterative refinement: As the adversary identifies finer distinctions between true and synthetic pairs and refines its inter-item association structure, the recommender incorporates it in the user recommendations.

4.3 Experimental Validation

Variational Auto-Encoders [34] and Denoising Auto-Encoder (CDAE) [74] as the base recommender $\mathbf{G}$. Results on the *ml-20m* dataset already indicate strong long-tail performance of stochastic VAE-CF (fig. 6) in comparison to deterministic CDAE [74]. Thus, performance gains in niche-item recall for VAE-CF with our adversarial training are particularly significant. Models are trained with training user interactions, while the interactions in the validation and test sets are split in two. One subset is fed as input to the trained model, while the other is used to evaluate the system output (ranked list) on $NDCG@100, Recall@K$, $K = 20, 50$. The architecture and training procedure is adopted from [34] for comparison. We set tradeoff parameter λ to multiple values and explore it's effect on recommendation over different sets of items, grouped by popularity. The balance parameter μ was set to 1 and $\mathbf{D}$ used a feed-forward network with 2 hidden layers (300, 100) as in fig. 5 (*tanh* activations and sigmoid output layer) and 300-dimensional embedding layers.

We first analyze the composition of the `top` 100 recommendations of $\mathbf{D} + \mathbf{G}$, against $\mathbf{G}$ trained in isolation. All items are split into four quartiles based on

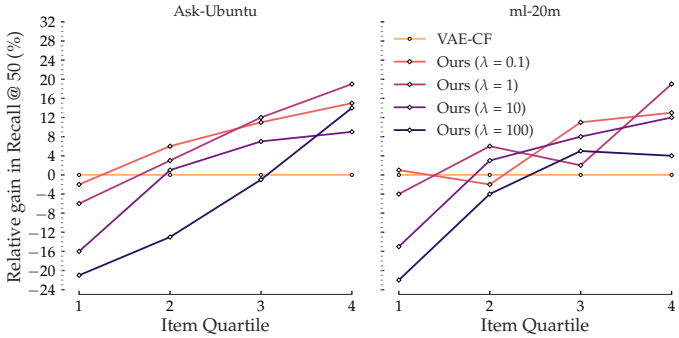
their popularity. We demonstrate the effect of the tradeoff λ on the `top` 100 items for validation set users, by analyzing the quartiles they appear from (Table 4). Clearly, the recommendations from our model with higher values of λ improve the niche-tag coverage and diversity. The overall recommendation performance against VAE-CF and CDAE in Table 5 show that diversity is not hurting our performance.

Table 5: Overall recommender performance on ml-20m and Ask-Ubuntu datasets

Method	**ml-20m**			**Ask-Ubuntu**		
	N@100	**R@20**	**R@50**	**N@100**	**R@20**	**R@50**
CDAE ($\mathbf{G}_1$)	0.34	0.27	0.37	0.29	0.30	0.46
VAE-CF ($\mathbf{G}_2$)	0.51	0.44	0.57	0.42	0.45	0.59
$\mathbf{D}$+$\mathbf{G}_2(\lambda = 0.1)$	0.53	0.45	0.59	0.43	0.46	0.61
$\mathbf{D}$+$\mathbf{G}_2(\lambda = 1)$	0.52	0.44	0.58	0.42	0.46	0.59
$\mathbf{D}$+$\mathbf{G}_2(\lambda = 10)$	0.48	0.41	0.55	0.40	0.43	0.56
$\mathbf{D}$+$\mathbf{G}_2(\lambda$=100)	0.42	0.37	0.51	0.38	0.41	0.53

Note that CDAE does not make *any* niche item recommendations (Q3 and Q4). Integrating our adversary to train CDAE results in a significant jump in long-tail coverage. To further dissect the above results, we will now observe our relative gains in *Recall*@50 compared to VAE-CF for each item quartile (Figure 6). We compare with VAE-CF due to it's stronger long-tail performance.

Fig. 6: Relative improvement over VAE-CF with adversary training, measured for each item popularity quartile (R@50)



As expected, our strongest gains are observed in Quartiles-3 and 4, which constitute long-tail items. Although there is a slight loss in popular item performance for $\lambda = 1$, this loss is not significant owing to the ease of recommending popular items with auxiliary models if required. We observe the values of tradeoff λ between 0.1 and 1 to generate balanced results.

We now analyze overall recommendation performance against VAE-CF and CDAE in Table 5 ($\mathbf{N}$ = NDCG, $\mathbf{R}$ = Recall). Even though our models recommend very different compositions of items (table 4), the results exhibit modest overall improvements for $\lambda = 0.1$ and $\lambda = 1$ over both the base recommenders. Clearly, the additional niche recommendations are coherent since there is no

performance drop. However, larger λ values hurt the recommender performance. It is thus essential to balance the adversary objective and base recommender to obtain strong overall results.

5 Integrating Primary and Auxiliary Behavioral Facets for Sparsity-Regularized Recommendation

One of the two ways that we build malleable frameworks is to enable flexibility in the types/modes of behavioral data and how we could apply them towards a profiling or recommendation objective. In this chapter, we demonstrate the application scenario of social recommendation, where the central mode of data is the purchase history of each user which can be applied to collaborative inference. However, we have a secondary mode of data in the form of social links between users, which can be viewed as an auxiliary mode or regularizer data. The key question we answer is, how do we incorporate diverse auxiliary data modalities to reinforce the central modality and the overall objective. We build an adaptive adversarial framework to balance the contribution of the modalities of behavioral data towards recommendation.

Social regularization is grounded in correlation theories such as homophily [48] and notions of influence or conversely, susceptibility [47]. The social connections among users (in the form of explicit social networks) and among items (such as induced co-occurrence graphs [72]) can play a critical role in improving recommendation quality in the presence of data sparsity and in addressing long-tail concerns [81]. However, a direct application of homophily [39], [42] wihtout contextualization constrains effective combination of user interests and social factors. Exposure models [66] adopt a *exposure precedes action* perspective to improve homophily. However, they do not contextually prioritize specific different social contacts. For instance, Alice may prefer her connection Bob's suggestions on books, but follow Mary (another connection) for music. Their relative importance depends on a contextual mixture of factors that we can infer from their interest representations and social structure.

Shalizi et al. [60] proved a key negative result—homophily and influence are fundamentally confounded in observational studies. In other words, we cannot disentangle peer influence from latent interests using observational data. Thus the attribution problem is inherently adversarial where we examine two competing hypothesis—social influence and latent interests—to explain each purchase decision. In a broader sense, this is true of any multi-modal observational setting. For instance, users could be influenced by viewing review content or prefer a certain cuisine on the Yelp platform. Our framework attempts to address the contextual attribution question in such settings to obtain regularized cumulative predictions, while also overcoming sparsity that a single data modality might exhibit.

The social regularization problem is readily amenable to a Generative Adversarial Network (GAN) formulation, whereby the social and interest factors of each user complete to explain each user's observed actions. As a result of such

a training process, the most contextually relevant social information regularizes the interest space of each user. Furthermore, an adversarial formulation provides a modular framework to decouple the architectural choices for the recommender and social representation models, enabling a wide range of recommender applications. Degenerate solutions are a significant challenge in vanilla GAN implementations that lack a sufficiently expressive attribution strategy. We overcome this challenge through an intuitive contextual weighting strategy to ensure informative social associations play a larger role in regularizing the learned user interest space.

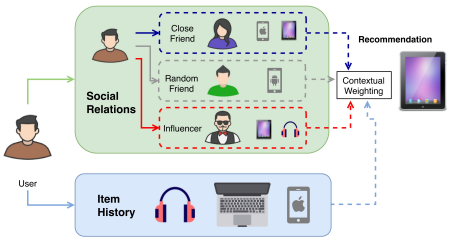


Fig. 7: Social contacts and item histories of users must be contextually weighted to evaluate their potential impact on future purchases

5.1 Problem Formulation

We consider the implicit feedback setting with users $\mathcal{U}$, items $\mathcal{I}$ and interaction matrix $\mathcal{Z} \in B^{|\mathcal{U}|\times|\mathcal{I}|}$ $(B = \{0, 1\})$. $\mathcal{N} \in B^{|\mathcal{U}|\times|\mathcal{U}|}$ denotes the social links between users, we abuse $\mathcal{N}$ to denote both, the social network and its user adjacency matrix. The total number of user-item interactions and social links are denoted $|\mathcal{Z}|$, $|\mathcal{N}|$ respectively.

Latent-factor social recommenders learn the latent social and interest representations for each user. Without loss of generality, let us denote social embeddings $\mathbf{S} \in R^{|\mathcal{U}|*d_S}$ and interest embeddings $\mathbf{X} \in R^{|\mathcal{U}|*d_X}$. Note that $\mathbf{X}_u, \mathbf{S}_u$ denote rows for user u. Further, we denote item embeddings $\mathbf{I} \in R^{|\mathcal{I}|*d_I}$. Given any user embedding matrix $\mathbf{E}$, we can compute user-user similairities in $\mathbf{E}$'s latent space as,

$$p_{\mathbf{E}}(u, v) \propto \sigma(\mathbf{E}_u \cdot \mathbf{E}_v) \tag{12}$$

where $u, v \in \mathcal{U}$ and $\sigma(x) = 1/(1 + e^{-x})$. The social and interest embedding spaces $\mathbf{S}$, $\mathbf{X}$ induce different user-user proximities p_S, p_X in Equation (12). Social regularization of $\mathbf{X}$ involves sharing the coordinate structure across $\mathbf{S}$ and $\mathbf{X}$. At the heart of this problem is the choice of a suitable distance metric in the embedding space. Historically metric learning approaches have learned effective distance functions in similarity, distance-based tasks [29], and recently in Collaborative Filtering [17]. Thus, the question follows,

Can we learn a distance metric to regularize interest embeddings X with social structure S? Let us consider a metric embedding space M with

metric distance measure $\mathbf{D}_M$ without any form assumptions. To transfer structure under $\mathbf{D}_M$, for each user-item interaction $(u, i) \in \mathcal{Z}$ we obtain pairwise loss $\|\mathbf{X}_u - \mathbf{I}_i\|_{\mathbf{D}_M} \rightarrow 0$ (with user interest embeddings $\mathbf{X}$ and item embeddings $\mathbf{I}$). Similarly, for social links $(u, v) \in \mathcal{N}$, we obtain $\|\mathbf{S}_u - \mathbf{S}_v\|_{\mathbf{D}_M} \rightarrow 0$ (with social embeddings $\mathbf{S}$). When we convert the above pairwise losses to equalities, it is easy to show that we obtain an over-specified system with only degenerate solutions (i.e., assigning the same interest embedding $\mathbf{X}_u$ to all $u \in \mathcal{U}$) due to the identity property of any $\mathbf{D}_M$. No solution can perfectly satisfy the above system if any pair of connected users have different item ratings. The continuous loss version of this system (optimized via gradient methods) moves towards some degenerate solution collapsing the user embeddings $\mathbf{X}_u$ (*interest space collapse*).

Can we transfer the structure of S to X without affecting interest space expressivity? The user-user similarities (or pairwise proximities) $p_{\mathbf{S}}(u, v)$ and $p_{\mathbf{X}}(u, v)$ from Equation (12) represent the structures of the embedding spaces $\mathbf{S}$ and $\mathbf{X}$. Ideally, we must converge $p_{\mathbf{S}}$ and $p_{\mathbf{X}}$ to a *meaningful*, i.e. *non-degenerate* equilibrium to avoid interest space collapse.

We avoid the over-specification problem in section 5.1 by introducing pair-specific translations for each pairwise constraint, i.e, the system is now of the form $\|\mathbf{S}_u - \mathbf{S}_v\|_{\mathbf{D}_M} \rightarrow w(u, v)$ where w is a learned function of the user context. This added expressivity enables a non-degenerate encoding in interest space $\mathbf{X}$, while retaining a contextually transformed version of the social structure via $w(u, v)$. We formalize the continuous version of the above regularization in a GAN framework [13] to regularize any gradient optimizable recommender, agnostic to its specific architecture.

5.2 Adversarial Social Regularization

The Generator ($\mathbf{G}$) in the GAN framework synthesizes data samples $\mathbf{y}_G \in R^d$ from a source distribution $P_G(\mathbf{Y})$ over R^d induced by $\mathbf{G}$. Conversely, discriminator ($\mathbf{D}$) attempts to construct a decision boundary to distinguish synthetic samples $\mathbf{y}_G$ drawn from $P_G(\mathbf{Y})$ against positive labeled samples drawn from an unknown target distribution that we wish to mimic. In our formulation, the social-agnostic base recommender learns the scoring function $f_{\mathbf{G}}(i \mid u, \mathcal{Z}), i \in \mathcal{I}, u \in \mathcal{U}$ to rank items given u's history $\mathcal{Z}_u$ by minimizing continuous, differentiable objective $\mathcal{O}_{\mathbf{G}}$ over parameters θ_G. It learns the interest embeddings $\mathbf{X}$, and the source user-user similarity $p_{\mathbf{X}}(u, v)$ in the interest space $\mathbf{X}$ (Equation (12)). We will refer to the base recommender as the generator $\mathbf{G}$ in our formulation.

On the other hand, social network $\mathcal{N}$ induces a target user-user similarity that $\mathbf{G}$ must imitate to regularize interest space $\mathbf{X}$. To compute the target user-user similarity, we apply a Graph Auto-Encoder [25] on $\mathcal{N}$ in Equation (12) and denote this as $p_{\mathcal{N}}(u, v)$, the target user-user similarity for $\mathbf{G}$. Finally, discriminator $\mathbf{D}$ learns an independent social space $\mathbf{S}$ for users separate from network $\mathcal{N}$. The discriminator induces social proximity, $p_{\mathbf{S}}(u, v)$ of users via $\mathbf{S}$ to link the target $p_{\mathcal{N}}(u, v)$ and source $p_{\mathbf{X}}(u, v)$ and *contextually* move them closer.

Structure Regularization: We develop a robust stochastic approach to represent $p_{\mathcal{X}}$ and $p_{\mathcal{N}}$ with a finite set of user-user pair samples drawn from each space. We evaluate the likelihood of each sampled user pair (u,v) with the discriminator embeddings $\mathbf{S}$, i.e., $p_{\mathbf{S}}(u, v)$. Ideally, $\mathbf{D}$ should consider *true-pairs* $(u_+, v_+) \sim p_{\mathcal{N}}$ more likely than *fake-pairs* $(u_-, v_-) \sim p_{\mathbf{X}}$. Conversely, $\mathbf{G}$ acts adversarial to $\mathbf{D}$ by maximizing expected fake-pair likelihood $E(p_{\mathbf{S}}(u_-, v_-))$. Thus, we obtain the overall objective $\mathcal{O}$,

$$\mathcal{O} = \min_{\mathbf{X}} \max_{\mathbf{S}} \Big(E_{(u_+,v_+)\sim p_{\mathcal{N}}} \log p_{\mathbf{S}}(u_+, v_+) + \mu . E_{(u_-,v_-)\sim p_{\mathbf{X}}} \log \big(1 - p_{\mathbf{S}}(u_-, v_-)\big) \Big) \tag{13}$$

where μ is a balance parameter. When we optimize $\mathcal{O}$, $\mathbf{G}$ learns X to maximize $\log p_{\mathbf{S}}(u_-, v_-)$. Conversely, the $\mathbf{D}$ maximizes $\log p_{\mathbf{S}}(u_+, v_+)$ and minimizes $\log p_{\mathbf{S}}(u_-, v_-)$. The expectations $E_{(u,v)}$ are averaged over ϵ *fake* and *true-pair* samples each (policy-gradient approximation) [70]. Empirically, we need $\leq 2\%$ of the distinct user pair count ($|\mathcal{U}|^2$), enabling much faster training than Coordinate Transfer Learning [50]. Equation (13) stochastically moves the user interest structure in $p_{\mathbf{X}}$ closer to $p_{\mathcal{N}}$. However, it may still lead to partial collapse of the interest space $\mathbf{X}$ since it lacks the pairwise expressivity defined in Section 5.1.

We can prevent interest space collapse by varying the regularization induced by each user pair sample, thus increasing model expressivity. This effectively differentiates social and interest context at the pair sample level in the objective,

$$\mathcal{O} = \min_{\mathbf{X}} \max_{\mathbf{S}} \Big(E_{(u_+,v_+)\sim p_{\mathcal{N}}} \log p_{\mathbf{S}}(u_+, v_+) + \mu . E_{(u_-,v_-)\sim p_{\mathbf{X}}} w(u_-, v_-) \log \big(1 - p_{\mathbf{S}}(u_-, v_-)\big) \Big) \tag{14}$$

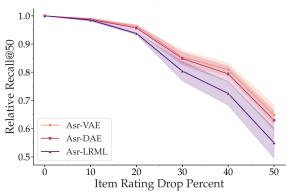
In this equation, we regularize $w(u, v) \times p_{\mathbf{X}}(u, v)$ against $p_{\mathbf{S}}$ (instead of just $p_{\mathbf{X}}$), enabling a much wider choice for $\mathbf{X}$. The contextual weighting function $w(u, v)$ accounts for diversity in the social links. Also note that contextually weighting *fake-pairs* is sufficient to expand the expressivity of $\mathbf{X}$, we do not need to weight the *true-pairs*. Thus, $w(u, v)$ needs to be computed on ϵ *fake-pairs* and adds a small overhead ($\epsilon \ll |\mathcal{U}|^2$).

5.3 Empirical Analysis

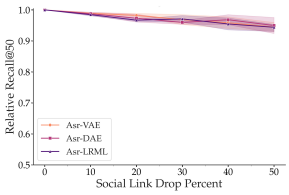
We evaluate and analyse our framework by regularizaing three diverse neural recommenders (**DAE** [75], **VAE-CF** [34] and **LRML** [63]) in our framework (Table 6) on multiple platforms *Ciao, Epinions, Delicious*, *Ask-Ubuntu* and *Yelp*. We refer to these variants as *Asr-DAE* etc.

We analyze the effect of adversary weight λ on the diversity of items recommended to users (Figure 9) and find that more regularization causes interest space collapse while too little results in overfitting to the training data (both

cases lack diversity in recommendations). We also examine the robustness of each adversarial model by separately sub-sampling the social links and item ratings of each user in the respective training sets (Figure 8) and find our stochastic user-pair sampling to be robust. Performance drop is measured vs. the best performance (e.g., 0.98 $\sim$ 2% loss).



(a) Performance with Item Drop



(b) And Social Link Drop

Fig. 8: We observe $\leq$ 6% *R@50* degradation at 20% item drop indicating our models are fairly robust in practice.

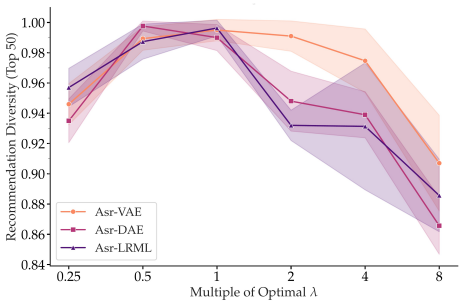
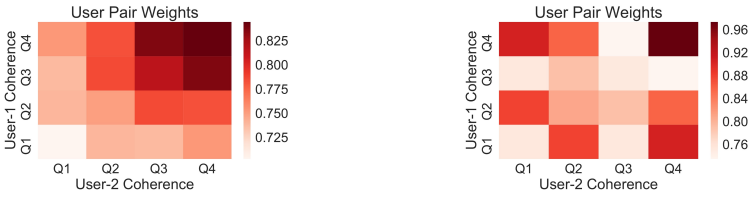


Fig. 9: Smaller λs overfit to the supervised term $\mathcal{O}_{\mathbf{G}}$, while larger multiples collapse the interest space i.e., less diverse aggregate recommendations. Units relative to highest diversity achieved.

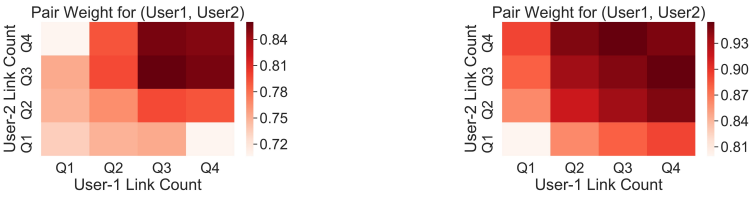
We observe from Figure 11, Figure 12 that our model prioritizes pairs of users where both users have numerous social connections or longer item histories to regularize their neighborhoods. Intuitively, pair samples where both users are influencers or prolific consumers, are likely to be regularize their social and interest neighborhoods respectively (they may act as cluster centers). We observe a similar trend against user coherence (coherence is a measure of how specialized or generic their item lists are) in the *Ciao* dataset (Figure 10). More coherent users act as better regularizers.

In the overall recommendation task (Table 6), conventional social recommenders are outperformed by social-agnostic neural methods. Uncontextualized regularization is detrimental to aggregate quality. Expressive representations (like in **DAE** [75]) gain more from regularization than conventional representations (e.g., mean *R@50* gains of **SBPR** vs. **BPR** are smaller than those of **Asr-VAE** vs. **VAE**). **VAE** representations are inherently stochastic unlike **DAE** and **LRML** resulting in greater recommendation diversity (less interest space collapse) with **Asr-VAE**. While **SEREC** permits for the exposed item



(a) *Ciao* dataset (b) *Epinions* dataset

Fig. 10: Pair weights against user coherence for pair samples in the *Ciao* and *Epinions* datasets

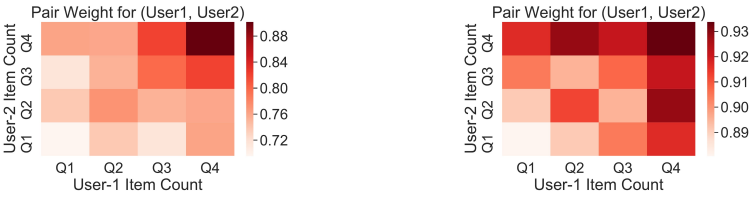


(a) Smaller Datasets (*Ciao, Epinions, Del*) (b) Large Datasets (*Ask-Ubuntu, Yelp*)

Fig. 11: We measure the Pair-Weight allocations to sampled pairs of users by our weight module. The x and y-axis denote the social link count quartiles of each user in pair (User-1, User-2), Q1 contains the lower values. E.g., The top-right box of the heatmap is the average weight alloted to samples where both users have many social links (Q4, Q4)

set to be prioritized differently, **CB** [73] flexibly attributes puchases, however picking a single factor (interest vs social) instead of a contextual combination.

While our model is well-suited to bi-modal observations with two data facets, extension to more than n-modes of data (n $\geq$ 3) is quadratic since there are $\binom{n}{2}$ ways to combine pairs of facets. We aim to build a linear solution to this scenario as future work.



(a) Small Datasets (*Ciao, Epinions, Del*) (b) Large Datasets (*Ask-Ubuntu, Yelp*)

Fig. 12: We create these heatmaps similar to Figure 11 with user item count quartiles, i.e., Q4 denotes long item histories

Table 6: *R@K* and *N@K* denote the Recall and NDCG ranked-list metrics

	Epinions				**Ask-Ubuntu**				**Yelp**			
Rec Model	*R@20*	*N@20*	*R@50*	*N@50*	*R@20*	*N@20*	*R@50*	*N@50*	*R@20*	*N@20*	*R@50*	*N@50*
						Social-Agnostic Recommenders						
BPR [57]	0.264	0.141	0.440	0.176	0.377	0.199	0.514	0.264	0.228	0.125	0.431	0.170
NCF [16]	0.310	0.138	0.462	0.181	0.420	0.215	0.538	0.281	0.196	0.118	0.488	0.209
DAE [75]	0.324	0.164	0.498	0.198	0.416	0.301	0.569	0.392	0.270	0.158	0.473	0.213
VAE [34]	0.336	0.161	0.510	0.204	0.408	0.317	0.576	0.383	0.281	0.164	0.479	0.208
LRML [63]	0.329	0.173	0.509	0.219	0.405	0.366	0.564	0.405	0.272	0.160	0.483	0.196
						Social Rec						
SBPR [86]	0.271	0.138	0.446	0.185	0.368	0.206	0.528	0.287	0.230	0.143	0.449	0.196
SNCF	0.306	0.189	0.468	0.202	0.414	0.371	0.541	0.403	0.198	0.103	0.493	0.202
SGCN [71]	0.318	0.153	0.481	0.198	0.397	0.343	0.526	0.395	0.288	0.160	0.492	0.176
CB [73]	0.337	0.171	0.436	0.202	0.399	0.365	0.559	0.382	0.282	0.154	0.471	0.196
SEREC [66]	0.348	0.167	0.496	0.213	0.415	0.362	0.584	0.414	0.306	0.173	**0.508**	0.211
						Adversarial						
Asr-DAE	0.339	0.168	0.513	0.207	0.434	0.347	0.585	0.412	0.272	0.158	0.489	0.201
Asr-VAE	**0.358**	0.173	0.532	0.216	0.431	0.350	0.592	0.401	0.298	0.161	0.496	0.218
Asr-LRML	0.340	0.166	0.527	0.220	0.411	**0.375**	0.578	**0.419**	0.287	0.172	0.481	**0.233**

* The Asr variants denote the DAE, VAE-CF and LRML base models integrated in our framework. Bold numerals indicate statistically significant gains over the next best model at $p = 0.05$. When there are two or more strong performers under, we underline them.

6 Context Invariants for Cross-Domain Recommendation

While the previous chapters primarily focused on addressing skew and sparsity within a single recommendation domain such as an online platform, there are scenarios where profiling models could benefit from those that are already learned on a different platform. While the ideal scenario is direct reuse, in most practical situations, both domain-invariant and domain-specific components are necessary for holistic recommendation. In this chapter, we introduce a highly scalable neural transfer approach to extract and reuse multi-linear contextual invariants that describe user behavior across domains that do not share users or items.

Cross-domain transfer learning is a well-studied paradigm to address sparsity in recommendation. In the most common pairwise cross-domain setting, we can employ co-clustering via shared users or items [46], latent structure alignment [12], or hybrid approaches using both [19]. However, cases with limited or no user-item overlap are pervasive in real-world applications, such as geographic region based domains (e.g., cities or states), where we face disparities in data quality and volume. Our work focuses on the *few-dense-source*, *multiple-sparse-target* setting, where prior approaches are mostly inapplicable.

Combinations of contextual predicates prove critical in *learning-to-organize* the user and item latent factors. For instance, an *Italian wine restaurant* is a good recommendation for a *high spending* user on a *weekend evening* unlike a *Monday*

afternoon. The intersection of restaurant type (an attribute), historical patterns (historical context), and interaction time (interaction context) jointly describe the likelihood of this interaction. Our key intuition is to infer such *behavioral invariants* from a *dense-source* domain where we have ample interaction histories of users with wine restaurants, and apply (or adapt) these learned invariants to improve inference in *sparse-target* domains.

6.1 Problem Definition

Consider a set of recommendation domains $D = \{\mathbf{D}_i\}$ where each domain is a tuple $\{\mathcal{U}_{\mathbf{D}_i}, \mathcal{V}_{\mathbf{D}_i}, \mathcal{T}_{\mathbf{D}_i}\}$, with $\mathcal{U}_{\mathbf{D}_i}$, $\mathcal{V}_{\mathbf{D}_i}$ denoting the user and item sets of $\mathbf{D}_i$, and $\mathcal{T}_{\mathbf{D}_i}$, the set of contextual interactions between them. There is no overlap between the user and item sets of any two recommendation domains. $|\mathcal{U}|, |\mathcal{V}|$. In the implicit feedback setting, each interaction $t \in \mathcal{T}_{\mathbf{D}_i}$ is a tuple $t = (u, \mathbf{c}, v)$ where $u \in \mathcal{U}_{\mathbf{D}_i}, v \in \mathcal{V}_{\mathbf{D}_i}$ and context vector $\mathbf{c} \in R^{|\mathbf{C}|}$. For the explicit feedback setting, $\mathcal{T}_{\mathbf{D}_i}$ is replaced by ratings $\mathcal{R}_{\mathbf{D}_i}$, where each rating is a tuple $r = (u, \mathbf{c}, v, r_{uv})$, with the rating value r_{uv} (other notations are the same).

For simplicity, all interactions in all domains have the same set of context features. In our datasets, the context feature set $\mathbf{C}$ contains three different types of context features, interactional features $\mathbf{C_I}$ (such as time of interaction), historical features $\mathbf{C_H}$ (such as a user's average spend), and attributional features $\mathbf{C_A}$ (such as restaurant cuisine or user age). Thus each context vector $\mathbf{c}$ contains these three types of features for that interaction, i.e., $\mathbf{c} = [\mathbf{c_I}, \mathbf{c_H}, \mathbf{c_A}]$.

Under implicit feedback, we rank items $v \in \mathcal{V}_{\mathbf{D}}$ given user $u \in \mathcal{U}_{\mathbf{D}}$ and context $\mathbf{c}$. In the explicit feedback scenario, we predict rating r_{uv} for $v \in \mathcal{V}_{\mathbf{D}}$ given $u \in \mathcal{U}_{\mathbf{D}}$ and $\mathbf{c}$. Our transfer objective is to reduce the rating or ranking error in a set of disjoint sparse target domains $\{\mathbf{D}_t\} \subset D$ given the dense source domain $\mathbf{D}_s \in D$.

6.2 Model Architecture

In this section, we develop a scalable, modular architecture to extract pooled contextual invariants and guide the learned latent factor representations. We achieve this via four synchronized neural modules with complementary semantic objectives. We define and construct these modules to maintain a clear demarcation between the context-driven transferrable modules and the domain-specific non transferrable recommendation modules. This separation is critical to model scalability.

Context Module $\mathcal{M}^1$: User-item interactions are driven by context feature intersections that are inherently *multiplicative*, missed in the implicit Naive-Bayes assumption of additive models such as feature attention [15], [2]. The first layer in $\mathcal{M}^1$ transforms context $\mathbf{c}$ of an interaction $(u, \mathbf{c}, v)$ as follows:

$$\mathbf{c}^2 = \sigma(\underbrace{\mathbf{W}^2\mathbf{c} \oplus (\mathbf{b}^2 \otimes \mathbf{c})}_{\text{Weighted linear transform}}) \otimes \underbrace{\mathbf{c}}_{\text{Element-wise interaction}} \tag{15}$$

Fig. 13: Our overall recommender architecture highlighting $\mathcal{M}^1$ to $\mathcal{M}^4$

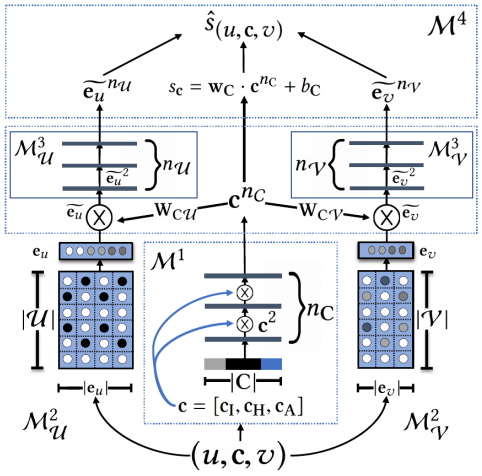


Table 7: Modules and Learned Parameter Notations

Modules	**Learned Parameters**
Domain-Specific $(\mathcal{M}^2_\mathcal{U}, \mathcal{M}^2_\mathcal{V})$	**Embeddings** $\mathbf{e}_u \forall u \in \mathcal{U}_\mathbf{D}$, $\mathbf{e}_v \forall v \in \mathcal{V}_\mathbf{D}$ **Biases** (only under explicit feedback) $s, s_u \forall u \in \mathcal{U}_\mathbf{D}, s_v \forall v \in \mathcal{V}_\mathbf{D}$
Shared Modules $(\mathcal{M}^1, \mathcal{M}^3, \mathcal{M}^4)$	$\mathcal{M}^1$ eq. (17) $(\mathbf{W}^i, \mathbf{b}^i) \forall i = [1, \cdots, n_\mathbf{C}]$ $\mathcal{M}^3$ eq. (19) $\mathbf{W}_{\mathbf{C}\mathcal{U}}, \mathbf{W}_{\mathbf{C}\mathcal{V}}$ $\mathcal{M}^3_\mathcal{U}$ eq. (21) $(\mathbf{W}^i_\mathcal{U}, \mathbf{b}^i_\mathcal{U}) \forall i = [1, \cdots, n_\mathcal{U}]$ $\mathcal{M}^3_\mathcal{V}$ eq. (21) $(\mathbf{W}^i_\mathcal{V}, \mathbf{b}^i_\mathcal{V}) \forall i = [1, \cdots, n_\mathcal{V}]$ $\mathcal{M}^4$ eq. (22) $\mathbf{W}_\mathbf{C}, \mathbf{b}_\mathbf{C}$

where $\oplus, \otimes$ denote element-wise product and sum, i.e.,

$$\mathbf{c}^2_i \propto \mathbf{c}_i \times \sigma(\mathbf{b}^2_i \mathbf{c}_i + \sum_j \mathbf{W}^2_{ij} \mathbf{c}_j) \tag{16}$$

Thus, $\mathbf{c}^2_i$ (i^{th}-component of $\mathbf{c}^2$) incorporates a weighted bivariate interaction between $\mathbf{c}_i$ and other context factors $\mathbf{c}_j$, including itself. We then repeat this transformation over multiple stacked layers with each layer using the previous output:

$$\mathbf{c}^n = \sigma(\mathbf{W}^n \mathbf{c}^{n-1} \oplus (\mathbf{b}^n \otimes \mathbf{c}^{n-1})) \otimes \mathbf{c} \tag{17}$$

Each layer interacts *n-variate* terms from the previous layer with $\mathbf{c}$ to form *n+1-variate* terms. However, since each layer has only $|\mathbf{C}|$ outputs (i.e., low-rank), $\mathbf{W}^n$ prioritizes the most effective *n-variate* combinations of $\mathbf{c}$

Context Conditioned Clustering $\mathcal{M}^3$: We combine domain-specific embeddings $\mathcal{M}^2$ with the context combinations extracted by $\mathcal{M}^1$ to generate context-conditioned representations,

$$\widetilde{\mathbf{e}_u} = \mathbf{e}_u \otimes \sigma(\mathbf{W}_{\mathbf{C}\mathcal{U}} \times \mathbf{c}^{n_\mathbf{C}}) \tag{18}$$

$$\widetilde{\mathbf{e}_v} = \mathbf{e}_v \otimes \sigma(\mathbf{W}_{\mathbf{C}\mathcal{V}} \times \mathbf{c}^{n_\mathbf{C}}) \tag{19}$$

where, $\mathbf{W}_{\mathbf{C}\mathcal{U}} \in R^{|\mathbf{e}_u| \times |\mathbf{C}|}$, $\mathbf{W}_{\mathbf{C}\mathcal{V}} \in R^{|\mathbf{e}_v| \times |\mathbf{C}|}$ are learned parameters that map the most relevant context combinations to the user and item embeddings. We further introduce $n_{\mathcal{U}}$ feedforward *RelU* layers to cluster the representations,

$$\widetilde{\mathbf{e}_u}^2 = \sigma(\mathbf{W}_{\mathcal{U}}^2 \widetilde{\mathbf{e}_u} + \mathbf{b}_{\mathcal{U}}^2) \tag{20}$$

$$\widetilde{\mathbf{e}_u}^n = \sigma(\mathbf{W}_{\mathcal{U}}^n \widetilde{\mathbf{e}_u}^{n-1} + \mathbf{b}_{\mathcal{U}}^n) \tag{21}$$

Analogously, we obtain context-conditioned item representations $\widetilde{\mathbf{e}_v}^2, \cdots, \widetilde{\mathbf{e}_v}^{n_{\mathcal{V}}}$ with $n_{\mathcal{V}}$ feedforward *RelU* layers. The bilinear transforms in eq. (19) introduce *dimension alignment* for both $\widetilde{\mathbf{e}_u}^{n_{\mathcal{U}}}$ and $\widetilde{\mathbf{e}_v}^{n_{\mathcal{V}}}$ with the context otuput $\mathbf{c}^{n_{\mathbf{C}}}$. Thus, when $\mathcal{M}^3$ and $\mathcal{M}^1$ layers are transferred to a sparse target domain, we can backpropagate to guide the target domain user and item embeddings if we have the same set of context features.

6.3 Source Domain Training Algorithm

Focusing on harder data samples accelerates and stabilizes stochastic gradients [37], [7]. Since our learning process is grounded on context, novel interactions display *interesting* context combinations. Let $\mathcal{L}_{(u,\mathbf{c},v)}$ denote the loss function for an interaction $(u, \mathbf{c}, v)$. We propose an inverse novelty measure referred as the context-bias, $s_{\mathbf{c}}$, which is self-paced by the context combinations of $\mathcal{M}^1$ in Equation (17),

$$s_{\mathbf{c}} = \mathbf{w}_{\mathbf{C}} \cdot \mathbf{c}^{n_{\mathbf{C}}} + b_{\mathbf{C}} \tag{22}$$

We then attenuate the loss $\mathcal{L}_{(u,\mathbf{c},v)}$ for this interaction as,

$$\mathcal{L}'_{(u,\mathbf{c},v)} = \mathcal{L}_{(u,\mathbf{c},v)} - s_{\mathbf{c}} \tag{23}$$

The resulting novelty loss $\mathcal{L}'_{(u,\mathbf{c},v)}$ decorrelates interactions [8], [23] by emulating variance-reduction in the *n-variate* pooled space of $\mathbf{c}^{n_{\mathbf{C}}}$. $\mathcal{L}'_{(u,\mathbf{c},v)}$ determines the user and item embedding spaces, inducing a novelty-weighted training curriculum focused on harder samples as training proceeds. We now describe loss $\mathcal{L}_{(u,\mathbf{c},v)}$ for the explicit and implicit feedback scenarios.

In the *implicit feedback setting*, predicted likelihood $\hat{s}_{(u,\mathbf{c},v)}$ is computed with the context-conditioned embeddings (Equation (21)) and context-bias (Equation (23)) as,

$$\hat{s}_{(u,\mathbf{c},v)} = \widetilde{\mathbf{e}_u}^{n_{\mathcal{U}}} \cdot \widetilde{\mathbf{e}_v}^{n_{\mathcal{V}}} + s_{\mathbf{c}} \tag{24}$$

The loss for all the possible user-item-context combinations in domain $\mathbf{D}$ is,

$$\mathcal{L}_{\mathbf{D}} = \sum_{u \in \mathcal{U}_{\mathbf{D}}} \sum_{v \in \mathcal{V}_{\mathbf{D}}} \sum_{\mathbf{c} \in R^{|\mathbf{c}|}} ||I_{(u,\mathbf{c},v)} - \hat{s}_{(u,\mathbf{c},v)}||^2 \tag{25}$$

where I is the binary indicator $(u, \mathbf{c}, v) \in \mathcal{T}_{\mathbf{D}}$. $\mathcal{L}_{\mathbf{D}}$ is intractable due to the large number of contexts $\mathbf{c} \in R^{|\mathbf{c}|}$. We develop a negative sampling approximation for implicit feedback with two learning objectives - identify the likely item given

the user and interaction context, and identify the likely context given the user and the item. We thus construct two negative samples for each $(u, \mathbf{c}, v) \in \mathcal{T}_\mathbf{D}$ at random: Item negative with the true context, $(u, \mathbf{c}, v^-)$ and context negative with the true item, $(u, \mathbf{c}^-, v)$. $\mathcal{L}_\mathbf{D}$ then simplifies to,

$$\mathcal{L}_\mathbf{D} = \sum_{\mathcal{T}_\mathbf{D}} ||1 - \hat{s}_{(u,\mathbf{c},v)}||^2 + \sum_{(u,\mathbf{c},v^-)} ||\hat{s}_{(u,\mathbf{c},v^-)}|| + \sum_{(u,\mathbf{c}^-,v)} ||\hat{s}_{(u,\mathbf{c}^-,v)}|| \tag{26}$$

In the explcit feedback setting, we introduce two additional bias terms, one for each user, s_u and one for each item, s_v. These terms account for user and item rating eccentricities (e.g., users who always rate well), so that the embeddings are updated with the relative rating differences. Finally, global bias s accounts for the rating scale, e.g., 0-5 vs. 0-10. Thus the predicted rating is given as,

$$\hat{r}_{(u,\mathbf{c},v)} = \widetilde{\mathbf{e}_v}^{n\mathcal{V}} \cdot \widetilde{\mathbf{e}_u}^{n\mathcal{U}} + s_\mathbf{c} + s_u + s_v + s \tag{27}$$

Negative samples are not required in the explicit feedback setting,

$$\mathcal{L}_\mathbf{D}^{explicit} = \sum_{(u,\mathbf{c},v,r_{uv}) \in \mathcal{R}_\mathbf{D}} ||r_{uv} - \hat{r}_{(u,\mathbf{c},v)}||^2 \tag{28}$$

Our formulation enables training the shared modules $\mathcal{M}^1, \mathcal{M}^3$ and $\mathcal{M}^4$ on a dense source domain, and transferring them to sparse target domains to guide their embedding module $\mathcal{M}^2$. We can view each shared module $\mathcal{M}$ as an encoder receiving inputs $\mathbf{x}_\mathcal{M}$ and generating output representations $\mathbf{y}_\mathcal{M}$. In each recommendation domain, module $\mathcal{M}$ determines the joint input-output distribution,

$$p(\mathbf{y}_\mathcal{M}, \mathbf{x}_\mathcal{M}) = p(\mathbf{y}_\mathcal{M} | \mathbf{x}_\mathcal{M}) \times p(\mathbf{x}_\mathcal{M}) \tag{29}$$

where the parameters of $\mathcal{M}$ determine the conditional $p(\mathbf{y}_\mathcal{M}|\mathbf{x}_\mathcal{M})$, while marginal $p(\mathbf{x}_\mathcal{M})$ describes the nature of inputs $\mathbf{x}_\mathcal{M}$ in that domain. We could modify inputs $\mathbf{x}_\mathcal{M}$ in that domain without changing $\mathcal{M}$ i.e., alter $p(\mathbf{x}_\mathcal{M})$, or adapt the parameters of $\mathcal{M}$, i.e., alter $p(\mathbf{y}_\mathcal{M}|\mathbf{x}_\mathcal{M})$ without changing the inputs.

6.4 Module Transfer to Sparse Domains

Under Direct Layer-Transfer, we train all four modules on the source and each target domain in isolation. Let us denote these pretrained modules as $(\mathcal{M}^i)_\mathbf{S}$ and $(\mathcal{M}^i)_\mathbf{T}$ for source domain $\mathbf{S}$ and a target domain $\mathbf{T}$ respectively. We then replace the shared modules in all the target state models with their source-trained version, i.e., $(\mathcal{M}^1)_\mathbf{T} = (\mathcal{M}^1)_\mathbf{S}$, $(\mathcal{M}^3)_\mathbf{T} = (\mathcal{M}^3)_\mathbf{S}$, $(\mathcal{M}^4)_\mathbf{T} = (\mathcal{M}^4)_\mathbf{S}$, while the domain-specific embeddings/embedding-layers in $(\mathcal{M}^2)_\mathbf{T}$ are not changed.

Simulated Annealing is a stochastic local-search algorithm, that implicitly thresholds parameter variations in the gradient space by decaying the gradient learning rates [26]. As a simple and effective adaptation strategy, we anneal each

transferred module $\mathcal{M}$ in the target domain with exponentially decaying learning rates to stochastically prevent overfitting. While annealing the transferred modules, domain-specific module $\mathcal{M}^2$ is updated with the full learning rate η_0. Clearly, annealing modifies the conditional in Equation (29), i.e., it changes $p(\mathbf{y}_\mathcal{M}|\mathbf{x}_\mathcal{M})$ without changing the inputs (conditional adaptation). However, annealing transferred modules to each target domain is somewhat expensive, and the annealed parameters are not shareable. Target domains effectively retrain separate models causing scalability limitations in the one-to-many transfer scenario. We now describe a lightweight residual adaptation strategy to overcome these scalability challenges.

Distributionally Regularized Residuals (DRR) reuses source modules with target-specific input modifications (i.e., input adaptation), thus addressing the scalability concerns of parameter modification methods. Each module $\mathcal{M}$ implements the conditional $p(\mathbf{y}_\mathcal{M}|\mathbf{x}_\mathcal{M})$. To share the conditionals across our recommendation domains, we introduce target-specific residual perturbations to account for its eccentricities [36] and smooth the input distribution $p(\mathbf{x}_\mathcal{M})$. Target-specific feature adaptation overcomes the need for an expensive end-to-end parameter search. Our adaptation problem thus reduces to learning an input modifier,

$$x^{\mathbf{T}}_\mathcal{M} = f^{\mathbf{T}}_\mathcal{M}(x_\mathcal{M}) \tag{30}$$

for each target domain $\mathbf{T}$ and shared module $\mathcal{M} \in [\mathcal{M}^1, \mathcal{M}^3, \mathcal{M}^4]$.

Residual transformations enable the flow of information between layers without the distortion or gradient attenuation of inserting new non-linear layers, resulting in numerous optimization advantages. Given the module-input $\mathbf{x}_\mathcal{M}$ to the shared module $\mathcal{M}$, we introduce the following target-specific residual transform:

$$x^{\mathbf{T}}_\mathcal{M} = x_\mathcal{M} + \delta^{\mathbf{T}}_\mathcal{M}(x_\mathcal{M}) \tag{31}$$

The form of the residual function δ is flexible. We choose a non-linear feed-forward transformation, $\delta(\mathbf{x}_\mathcal{M}) = tanh(\mathbf{W}\mathbf{x}_\mathcal{M} + \mathbf{b})$. We also experimented with a more expressive bilinear form, $\delta(\mathbf{x}_\mathcal{M}) = \mathbf{h} \otimes tanh(\mathbf{W}\mathbf{x}_\mathcal{M} + \mathbf{b})$. An intuitive trade-off can be made to balance the complexity and number of residual layers.

6.5 Empirical Analysis

When we adapt modules trained on a rich source domain to the sparse target domains, we signficantly reduce the computational costs and improve performance in comparison to learning directly on the sparse domains. We chose two review datasets that were split over states in the United States with no shared users or businesses across any two states. Our source model was trained on states with ample volumes and density of data, while the targets were sparse regions with limited review data (Section 6.5).

We evaluate module transfer methods by the drop in **RMSE** (Table 9) for the sparse target states in each dataset when we transfer the $\mathcal{M}^1, \mathcal{M}^3$ and $\mathcal{M}^4$

Dataset		State	Users	Items	Interactions
	S	**Pennsylvania**	**10.3 k**	**5.5 k**	**170 k**
Yelp[5]	$\mathbf{T}_1$	Alberta, Canada	5.10 k	3.5 k	55.0 k
$\lvert\mathbf{C}\rvert = 120$	$\mathbf{T}_2$	Illinois	1.80 k	1.05 k	23.0 k
	$\mathbf{T}_3$	S.Carolina	0.60 k	0.40 k	6.20 k
Google	**S**	**California**	**46 k**	**28 k**	**320 k**
Local[6]	$\mathbf{T}_1$	Colorado	10 k	5.7 k	51.0 k
$\lvert\mathbf{C}\rvert = 90$	$\mathbf{T}_2$	Michigan	7.0 k	4.0 k	29.0 k
	$\mathbf{T}_3$	Ohio	5.4 k	3.2 k	23.0 k

Table 8: *Source* and *Target* statistics for our datasets

modules from the source state rather than training all four modules from scratch on that target domain. Similarly, the meta-learning baselines were evaluated by comparing their joint meta-model performance on the target state against our model trained only on that state. We start with an analysis of the training process for module transfer with simulated annealing and DRR adaptation.

Transfer Details: On each target state in each dataset, all four modules of our MMT-Net model are pretrained over two gradient epochs on the target samples. The layers in modules $\mathcal{M}^1, \mathcal{M}^3$ and $\mathcal{M}^4$ are then replaced with those trained on the source state, while retaining module $\mathcal{M}^2$ without any changes (in our experiments $\mathcal{M}^2$ just contains user and item embeddings, but could also include neural layers if required). This is then followed by either simulated annealing or DRR adaptation of the transferred modules.

On the target states, module transfer is 3x faster then training a new model from scratch Figure 15. We analyze the training loss curves in Section 6.5 to better understand the fast adaptation of the transferred modules. Further, the sizes and densities of the target states were not always correlated to the gains we achieved. Skew (e.g., few towns vs. one big city) and other data factors played a significant role. For simplicity, we aggregated our target domains by state, although we expect a finer resolution (such as town) to yield better transfer performance.

Invariant Quality: A surprising result was the similar performance of *direct layer-transfer* with no adaptation to training all modules on the target state from scratch (Table 9). The transferred source state modules were directly applicable to the target state embeddings. This helps us validate the generalizability of context-based modules across independently trained state models even with no user or item overlap.

Computational Gains: We also plot the total training times including pre-training for DRR and annealing against the total number of target state interactions in Figure 15. There is a significant reduction in the overall training time and computational effort in the *one-to-many* setting. Simulated annealing and DRR adaptation converge in fewer epochs when applied to the pre-trained target model, and outperform the target-trained model by significant margins (Table 9). Further, these computational gains facilitate moving towards a finer

Table 9: Percentage RMSE improvements on the Yelp and Google Local target states with module transfer approaches and meta-learning baselines against training all modules on the target state directly.

Dataset		Direct %RMSE	Anneal %RMSE	DRR %RMSE	LWA [64] %RMSE	NLBA [64] %RMSE	s^2-Meta [9] %RMSE
Yelp	T_1	-2.2%	**7.7%**	7.2%	2.6%	4.1%	3.7%
	T_2	-2.6%	**9.0%**	7.9%	1.8%	3.6%	3.1%
	T_3	0.8%	**8.5%**	8.1%	0.3%	5.3%	1.8%
Google Local	T_1	-1.2%	**11.2%**	11.0%	3.3%	4.3%	3.1%
	T_2	-1.7%	**12.1%**	10.9%	4.6%	4.9%	2.8%
	T_3	-2.0%	**9.6%**	8.8%	2.4%	6.3%	3.9%

target domain granularity for effective module adaptation (e.g., adapt to towns or counties rather than states).

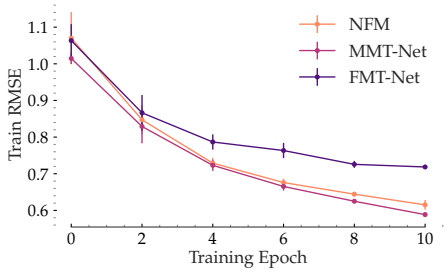


Fig. 14: MMT-Net (ours) convergence compared to NFM [15] and FMT-Net (our variant with additive context transform) on the Google Local Colorado target

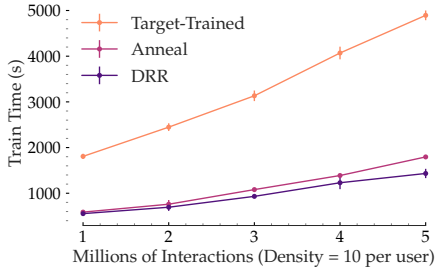


Fig. 15: MMT-Net(ours) training duration with and without module transfer vs. target domain interaction volume

Training without Context-Bias To understand the importance of decorrelating training samples in the training process, we repeat the performance analysis on our MMT-Net model with and without the adaptive context-bias term in the training objective in Section 6.3. We observe a 15% performance drop across the Yelp and Google Local datasets, although this does not reflect in the Train-RMSE convergence (Figure 16) of the two variations. In the absence of context-bias, the model overfits uninformative transactions to the user and item bias terms (s_u, s_v) in Equation (27), Equation (28) and thus achieves

comparable Train-RMSE values. However, the overfit user and item terms are not generalizable, resulting in the observed drop in test performance.

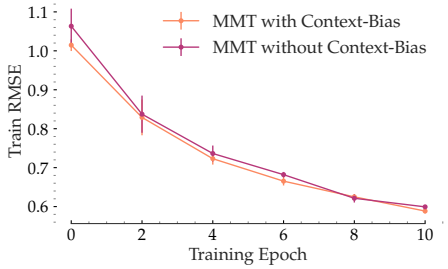


Fig. 16: MMT-Net (ours) trained with & without context-bias (Equation (23)) on the Google Local source exhibits similar Train-RMSE, but registers $> 10\%$ drop in test performance.

Model Training and Convergence Analysis We compare the Train-RMSE convergence for the MMT-Net model fitted from scratch to the Google Local target state, Colorado ($\mathbf{T}_1$) vs. the training curve under DRR and annealing adaptation with two pretraining epochs on the target state in Figure 17. Clearly, the target-trained model takes significantly longer to converge to a stable Train-RMSE in comparison to the Anneal and DRR adaptation. Although the final Train-RMSE is comparable (Figure 15), there is a significant performance difference between the two approaches on the test dataset, as observed in Table 9. Training loss convergence alone is not indicative of the final model performance; the target-only training method observes lower Train-RMSE by overfitting to the sparse data. We also compare the Train-RMSE convergence for target-trained models with and without pooled context factors (MMT-Net, NFM [15] vs. FMT-Net) in Figure 14. We observe the NFM [15], MMT-Net models to converge faster to a better optimization minima than FMT-net.

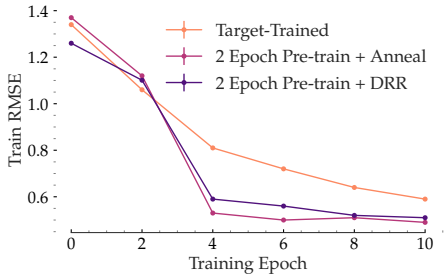


Fig. 17: MMT-Net (our model) convergence under target-training vs. Annealing/DRR adaptation after 2 epochs of pretraining on the Google Local Colorado target

While contextual invariants are effective when the compared domains have the same or similar context features, in future work, we aim to adapt model architectures with gradient feedback alone, i.e., treat the gradient feedback tensors as the key contextual factors of user interactions across recommendation

domains. We expect these advances to significantly broaden the application of the neural layer transfer approaches proposed in this chapter.

References

1. Barabasi, A.-L. The origin of bursts and heavy tails in human dynamics. *Nature 435*, 7039 (2005), 207–211.
2. Beutel, A., Covington, P., Jain, S., Xu, C., Li, J., Gatto, V., and Chi, E. H. Latent cross: Making use of context in recurrent recommender systems. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining* (2018), ACM, pp. 46–54.
3. Beutel, A., Murray, K., Faloutsos, C., and Smola, A. J. Cobafi: collaborative bayesian filtering. In *Proceedings of the 23rd international conference on World wide web* (2014), ACM, pp. 97–108.
4. Beutel, A., Murray, K., Faloutsos, C., and Smola, A. J. Cobafi: collaborative bayesian filtering. In *Proceedings of the 23rd international conference on World wide web* (2014), ACM, pp. 97–108.
5. Cai, H., Zheng, V. W., Zhu, F., Chang, K. C.-C., and Huang, Z. From community detection to community profiling. *Proceedings of the VLDB Endowment 10*, 7 (2017), 817–828.
6. Cao, D., He, X., Miao, L., An, Y., Yang, C., and Hong, R. Attentive group recommendation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval* (2018), ACM, pp. 645–654.
7. Chang, H.-S., Learned-Miller, E., and McCallum, A. Active bias: Training more accurate neural networks by emphasizing high variance samples. In *Advances in Neural Information Processing Systems* (2017), pp. 1002–1012.
8. Cogswell, M., Ahmed, F., Girshick, R., Zitnick, L., and Batra, D. Reducing overfitting in deep networks by decorrelating representations. *arXiv preprint arXiv:1511.06068* (2015).
9. Du, Z., Wang, X., Yang, H., Zhou, J., and Tang, J. Sequential scenario-specific meta learner for online recommendation. *arXiv preprint arXiv:1906.00391* (2019).
10. Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70* (2017), JMLR. org, pp. 1126–1135.
11. Gao, S., Luo, H., Chen, D., Li, S., Gallinari, P., and Guo, J. Cross-domain recommendation via cluster-level latent factor model. In *Joint European conference on machine learning and knowledge discovery in databases* (2013), Springer, pp. 161–176.
12. Gao, S., Luo, H., Chen, D., Li, S., Gallinari, P., and Guo, J. Cross-domain recommendation via cluster-level latent factor model. In *Joint European conference on machine learning and knowledge discovery in databases* (2013), Springer, pp. 161–176.
13. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems* (2014), pp. 2672–2680.
14. Hamilton, W., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems* (2017), pp. 1024–1034.

15. He, X., and Chua, T.-S. Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval* (2017), ACM, pp. 355–364.
16. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., and Chua, T.-S. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web* (2017), International World Wide Web Conferences Steering Committee, pp. 173–182.
17. Hsieh, C.-K., Yang, L., Cui, Y., Lin, T.-Y., Belongie, S., and Estrin, D. Collaborative metric learning. In *Proceedings of the 26th international conference on world wide web* (2017), International World Wide Web Conferences Steering Committee, pp. 193–201.
18. Hu, G., Zhang, Y., and Yang, Q. Transfer meets hybrid: A synthetic approach for cross-domain collaborative filtering with text. In *The World Wide Web Conference* (2019), ACM, pp. 2822–2829.
19. Hu, G., Zhang, Y., and Yang, Q. Transfer meets hybrid: A synthetic approach for cross-domain collaborative filtering with text. In *The World Wide Web Conference* (2019), ACM, pp. 2822–2829.
20. Jiang, M., Cui, P., Liu, R., Yang, Q., Wang, F., Zhu, W., and Yang, S. Social contextual recommendation. In *Proceedings of the 21st ACM international conference on Information and knowledge management* (2012), ACM, pp. 45–54.
21. Jiang, M., Cui, P., Wang, F., Xu, X., Zhu, W., and Yang, S. Fema: flexible evolutionary multi-faceted analysis for dynamic behavioral pattern discovery. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining* (2014), ACM, pp. 1186–1195.
22. Jiang, M., Cui, P., Wang, F., Xu, X., Zhu, W., and Yang, S. Fema: flexible evolutionary multi-faceted analysis for dynamic behavioral pattern discovery. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining* (2014), ACM, pp. 1186–1195.
23. Johnson, R., and Zhang, T. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems* (2013), pp. 315–323.
24. Kipf, T. N., and Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).
25. Kipf, T. N., and Welling, M. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308* (2016).
26. Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. Optimization by simulated annealing. *science 220*, 4598 (1983), 671–680.
27. Koren, Y., Bell, R., and Volinsky, C. Matrix factorization techniques for recommender systems. *Computer 42*, 8 (2009).
28. Krishnan, A., Sharma, A., and Sundaram, H. Insights from the long-tail: Learning latent representations of online user behavior in the presence of skew and sparsity. In *To appear in Proceedings of the 2018 ACM on Conference on Information and Knowledge Management* (2018), ACM.
29. Kulis, B., et al. Metric learning: A survey. *Foundations and Trends® in Machine Learning 5*, 4 (2013), 287–364.
30. Leskovec, J., Huttenlocher, D., and Kleinberg, J. Predicting positive and negative links in online social networks. In *Proceedings of the 19th international conference on World wide web* (2010), ACM, pp. 641–650.
31. Li, A. Q., Ahmed, A., Ravi, S., and Smola, A. J. Reducing the sampling complexity of topic models. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining* (2014), ACM, pp. 891–900.

32. Li, X., and She, J. Collaborative variational autoencoder for recommender systems. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2017), ACM, pp. 305–314.
33. Li, Z., Zhou, F., Chen, F., and Li, H. Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835* (2017).
34. Liang, D., Krishnan, R. G., Hoffman, M. D., and Jebara, T. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 World Wide Web Conference* (2018), International World Wide Web Conferences Steering Committee, pp. 689–698.
35. Liu, F., Cheng, Z., Sun, C., Wang, Y., Nie, L., and Kankanhalli, M. User diverse preference modeling by multimodal attentive metric learning. In *Proceedings of the 27th ACM International Conference on Multimedia* (2019), ACM, pp. 1526–1534.
36. Long, M., Zhu, H., Wang, J., and Jordan, M. I. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems* (2016), pp. 136–144.
37. Loshchilov, I., and Hutter, F. Online batch selection for faster training of neural networks. *arXiv preprint arXiv:1511.06343* (2015).
38. Ma, H., Yang, H., Lyu, M. R., and King, I. Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management* (2008), ACM, pp. 931–940.
39. Ma, H., Yang, H., Lyu, M. R., and King, I. Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management* (2008), ACM, pp. 931–940.
40. Ma, H., Zhou, D., Liu, C., Lyu, M. R., and King, I. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining* (2011), ACM, pp. 287–296.
41. Ma, H., Zhou, D., Liu, C., Lyu, M. R., and King, I. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining* (2011), ACM, pp. 287–296.
42. Ma, H., Zhou, D., Liu, C., Lyu, M. R., and King, I. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining* (2011), ACM, pp. 287–296.
43. Ma, J., Wen, J., Zhong, M., Liu, L., Li, C., Chen, W., Yang, Y., Tu, H., and Li, X. Dbrec: Dual-bridging recommendation via discovering latent groups. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (2019), ACM, pp. 1513–1522.
44. Ma, Z., Sun, A., Yuan, Q., and Cong, G. A tri-role topic model for domain-specific question answering. In *AAAI* (2015), pp. 224–230.
45. Man, T., Shen, H., Jin, X., and Cheng, X. Cross-domain recommendation: An embedding and mapping approach. In *IJCAI* (2017), pp. 2464–2470.
46. Man, T., Shen, H., Jin, X., and Cheng, X. Cross-domain recommendation: An embedding and mapping approach. In *IJCAI* (2017), pp. 2464–2470.
47. Marsden, P. V., and Friedkin, N. E. Network studies of social influence. *Sociological Methods & Research 22*, 1 (1993), 127–151.
48. McPherson, M., Smith-Lovin, L., and Cook, J. M. Birds of a feather: Homophily in social networks. *Annual review of sociology 27*, 1 (2001), 415–444.
49. Pan, W., Xiang, E. W., Liu, N. N., and Yang, Q. Transfer learning in collaborative filtering for sparsity reduction. In *Twenty-fourth AAAI conference on artificial intelligence* (2010).

50. Pan, W., Xiang, E. W., Liu, N. N., and Yang, Q. Transfer learning in collaborative filtering for sparsity reduction. In *Twenty-fourth AAAI conference on artificial intelligence* (2010).
51. Perera, D., and Zimmermann, R. Cngan: Generative adversarial networks for cross-network user preference generation for non-overlapped users. In *The World Wide Web Conference* (2019), ACM, pp. 3144–3150.
52. Pitman, J., and Yor, M. The two-parameter poisson-dirichlet distribution derived from a stable subordinator. *The Annals of Probability* (1997), 855–900.
53. Qiu, J., Tang, J., Liu, T. X., Gong, J., Zhang, C., Zhang, Q., and Xue, Y. Modeling and predicting learning behavior in moocs. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining* (2016), ACM, pp. 93–102.
54. Qiu, M., Zhu, F., and Jiang, J. It is not just what we say, but how we say them: Lda-based behavior-topic model. In *Proceedings of the 2013 SIAM International Conference on Data Mining* (2013), SIAM, pp. 794–802.
55. Qu, Q., Chen, C., Jensen, C. S., and Skovsgaard, A. Space-time aware behavioral topic modeling for microblog posts. *IEEE Data Eng. Bull. 38*, 2 (2015), 58–67.
56. Quan, X., Kit, C., Ge, Y., and Pan, S. J. Short and sparse text topic modeling via self-aggregation. In *IJCAI* (2015), pp. 2270–2276.
57. Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence* (2009), AUAI Press, pp. 452–461.
58. Sato, I., and Nakagawa, H. Topic models with power-law using pitman-yor process. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining* (2010), ACM, pp. 673–682.
59. Sato, I., and Nakagawa, H. Topic models with power-law using pitman-yor process. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining* (2010), ACM, pp. 673–682.
60. Shalizi, C. R., and Thomas, A. C. Homophily and contagion are generically confounded in observational social network studies. *Sociological Methods & Research 40*, 2 (2011), 211–239.
61. Shi, C., Li, Y., Zhang, J., Sun, Y., and Philip, S. Y. A survey of heterogeneous information network analysis. *IEEE Transactions on Knowledge and Data Engineering 29*, 1 (2016), 17–37.
62. Sun, Q., Liu, Y., Chua, T., and Schiele, B. Meta-transfer learning for few-shot learning. *CoRR abs/1812.02391* (2018).
63. Tay, Y., Anh Tuan, L., and Hui, S. C. Latent relational metric learning via memory-based attention for collaborative ranking. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web* (2018), International World Wide Web Conferences Steering Committee, pp. 729–739.
64. Vartak, M., Thiagarajan, A., Miranda, C., Bratman, J., and Larochelle, H. A meta-learning perspective on cold-start recommendations for items. In *Advances in neural information processing systems* (2017), pp. 6904–6914.
65. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).
66. Wang, M., Zheng, X., Yang, Y., and Zhang, K. Collaborative filtering with social exposure: A modular approach to social recommendation. In *Thirty-Second AAAI Conference on Artificial Intelligence* (2018).

67. Wang, X., He, X., Cao, Y., Liu, M., and Chua, T.-S. Kgat: Knowledge graph attention network for recommendation. *arXiv preprint arXiv:1905.07854* (2019).
68. Wang, X., He, X., Wang, M., Feng, F., and Chua, T.-S. Neural graph collaborative filtering. In *Proceedings of the 42Nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (New York, NY, USA, 2019), SIGIR'19, ACM, pp. 165–174.
69. Wang, Y., Feng, C., Guo, C., Chu, Y., and Hwang, J.-N. Solving the sparsity problem in recommendations via cross-domain item embedding based on co-clustering. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining* (2019), ACM, pp. 717–725.
70. Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning 8*, 3-4 (1992), 229–256.
71. Wu, L., Sun, P., Hong, R., Fu, Y., Wang, X., and Wang, M. Socialgcn: An efficient graph convolutional network based model for social recommendation. *arXiv preprint arXiv:1811.02815* (2018).
72. Wu, Q., Zhang, H., Gao, X., He, P., Weng, P., Gao, H., and Chen, G. Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems. *arXiv preprint arXiv:1903.10433* (2019).
73. Wu, Q., Zhang, H., Gao, X., He, P., Weng, P., Gao, H., and Chen, G. Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems. *arXiv preprint arXiv:1903.10433* (2019).
74. Wu, Y., DuBois, C., Zheng, A. X., and Ester, M. Collaborative denoising auto-encoders for top-n recommender systems. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining* (2016), ACM, pp. 153–162.
75. Wu, Y., DuBois, C., Zheng, A. X., and Ester, M. Collaborative denoising auto-encoders for top-n recommender systems. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining* (2016), ACM, pp. 153–162.
76. Wu, Y., Liu, X., Xie, M., Ester, M., and Yang, Q. Cccf: Improving collaborative filtering via scalable user-item co-clustering. In *Proceedings of the ninth ACM international conference on web search and data mining* (2016), ACM, pp. 73–82.
77. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Yu, P. S. A comprehensive survey on graph neural networks. *arXiv preprint arXiv:1901.00596* (2019).
78. Xue, G.-R., Lin, C., Yang, Q., Xi, W., Zeng, H.-J., Yu, Y., and Chen, Z. Scalable collaborative filtering using cluster-based smoothing. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval* (2005), ACM, pp. 114–121.
79. Xue, G.-R., Lin, C., Yang, Q., Xi, W., Zeng, H.-J., Yu, Y., and Chen, Z. Scalable collaborative filtering using cluster-based smoothing. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval* (2005), ACM, pp. 114–121.
80. Yang, C., Yan, H., Yu, D., Li, Y., and Chiu, D. M. Multi-site user behavior modeling and its application in video recommendation. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval* (2017), ACM, pp. 175–184.
81. Yin, H., Cui, B., Li, J., Yao, J., and Chen, C. Challenging the long tail recommendation. *Proceedings of the VLDB Endowment 5*, 9 (2012), 896–907.
82. Yin, H., Wang, Q., Zheng, K., Li, Z., Yang, J., and Zhou, X. Social influence-based group representation learning for group recommendation. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)* (2019), IEEE, pp. 566–577.

83. Yin, J., and Wang, J. A dirichlet multinomial mixture model-based approach for short text clustering. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining* (2014), ACM, pp. 233–242.
84. Yin, Z., Cao, L., Han, J., Zhai, C., and Huang, T. Geographical topic discovery and comparison. In *Proceedings of the 20th international conference on World wide web* (2011), ACM, pp. 247–256.
85. Zhang, Y., Ai, Q., Chen, X., and Croft, W. B. Joint representation learning for top-n recommendation with heterogeneous information sources. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management* (2017), ACM, pp. 1449–1458.
86. Zhao, T., McAuley, J., and King, I. Leveraging social connections to improve personalized ranking for collaborative filtering. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management* (2014), ACM, pp. 261–270.