A Modular Adversarial Approach to Social Recommendation

Adit Krishnan, Hari Cheruvu, Cheng Tao, Hari Sundaram
University of Illinois at Urbana-Champaign, IL-61801, USA
{aditk2,hcheruv2,ctao6,hs1}@illinois.edu

ABSTRACT

This paper proposes a novel framework to incorporate social regularization for item recommendation. Social regularization grounded in ideas of homophily and influence appears to capture latent user preferences. However, there are two key challenges: first, the importance of a specific social link depends on the context and second, a fundamental result states that we cannot disentangle homophily and influence from observational data to determine the effect of social inference. Thus we view the attribution problem as inherently adversarial where we examine two competing hypothesis—social influence and latent interests—to explain each purchase decision.

We make two contributions. First, we propose a modular, adversarial framework that decouples the architectural choices for the recommender and social representation models, for social regularization. Second, we overcome degenerate solutions through an intuitive contextual weighting strategy, that supports an expressive attribution, to ensure informative social associations play a larger role in regularizing the learned user interest space. Our results indicate significant gains (5-10% relative Recall@K) over state-of-the-art baselines across multiple publicly available datasets.

KEYWORDS

Social Recommendation; Neural Collaborative Filtering; Adversarial Machine Learning; Generative Adversarial Networks

ACM Reference Format:
Adit Krishnan, Hari Cheruvu, Cheng Tao, Hari Sundaram. 2019. A Modular Adversarial Approach to Social Recommendation. In *The 28th ACM International Conference on Information and Knowledge Management (CIKM '19), November 3–7, 2019, Beijing, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3357384.3357898

1 INTRODUCTION

This paper proposes a novel framework to incorporate social regularization for item recommendation. The motivating idea is to appropriately leverage social relation structure to capture unseen user preferences. Social correlation theories such as homophily [24] and notions of influence or conversely, susceptibility [3, 23] lend support to the idea of social regularization.

The social recommendation problem has received significant attention in the research community. The social connections among users (in the form of explicit social networks) and among items (such as induced co-occurrence graphs [36]) can play a critical role in improving recommendation quality in the presence of data sparsity and in addressing long-tail concerns [4, 14, 15]. The use of homophily encodes the assumption that social connections share similar preferences [10, 21]. This assumption constrains our ability to combine user interests and social factors effectively [33]. Exposure models [18, 33] are more nuanced and adopt an *exposure precedes action* lens. Each user's exposure to his contacts' preferences limits her potential actions. A weakness of the exposure approach is that it cannot explicitly prioritize specific preferences originating from different contacts based on the available context. For instance, Alice may prefer her connection Bob's suggestions on books, but follow Mary (another connection) for music. Thus social contacts can vary in the extent of influence they assert. Their relative importance depends on a contextual mixture of factors that we can infer from their interest representations and social structure.

Shalizi and Thomas [29] 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.

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. Our contributions are as follows:

Modular Adversarial Formulation: To the best of our knowledge, ours is the first work to address the social recommendation problem with an architecture-agnostic formulation. In contast to prior work, we integrate *state-of-the-art* recommender architectures and social representations models.

Expressive Attribution Strategy: We unify the interest and social distributions of users by contextually attributing their purchase decisions across these two representations. Thus, we incorporate diversity across users' social links and the varied impact of each link on their purchase decisions, enabling a more expressive interest space. Our qualitative analysis in *Section* 5 indicates we can preferentially select important social relations to improve recommendations.

CIKM '19, November 3–7, 2019, Beijing, China

ACM ISBN 978-1-4503-6976-3/19/11…$15.00
https://doi.org/10.1145/3357384.3357898

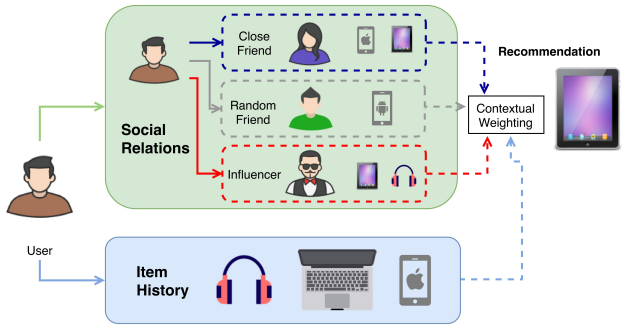


Figure 1: Social contacts and item histories of users must be contextually weighted to evaluate their potential impact on future purchases

Robust Experimental Results: We integrate three state-of-the-art social-agnostic recommender models in our adversarial framework and observe significant gains with adversarial training across multiple public datasets (4-10% relative Recall@K). Further, we categorize and study the extent of regularization imposed by social samples. We find that relations between influential users tend to play an important role in regularizing interests. Further, links across peers (similar levels of activity) are better regularizers than those with highly active users. Finally, our stochastic optimization approach is resilient to lossy social data.

We organize the rest of the paper as follows. In Section 2 we discuss related work. We formally define the problem and propose our approach in Section 3 and Section 4. We then present our experimental results in Sections 5, perform qualitative analysis of our model in Section 5.4, Section 5.5, Section 5.6 and discuss it's limitations in Section 5.7, finally concluding in Section 6.

2 RELATED WORK

Historically, matrix factorization (MF) has been the most popular collaborative filtering approach [22, 25] and forms the basis for efficient modern recommenders [7] and effective deep-learning strategies [6, 20, 38]. Prior efforts to integrate social structure in the latent interest space employed static hypotheses [10, 21] that do not incorporate additional context. Incorrect prioritization of social links could hurt recommendation quality. A second line of work has looked at transfer learning [26], auxiliary facet integration in MF [17] and trust propagation [9]. While these approaches augment [22], they are expensive and incompatible with neural methods [6].

More recently, *exposure* models [18, 33] view user actions as subsets of their social exposure. However, they do not separate sources of exposure; an item exposed by a subject expert is likely to have a greater impact, for instance. Wu et al. [37] propose a multi-armed bandit (MAB) solution to contextually pick *one-of-many* factors to explain purchases. Although it incorporates context, it is intuitive to explore a continuous version of [37] that differentially combines factors rather than pick just one.

In recent times, neural social-agnostic recommenders obtained *state-of-the-art* results with user-item rating information [20, 30, 38]. Further, a wide range of formulations and convolutional models have been proposed to effectively embed social networks [2, 12, 28, 31] with diverse link semantics. Our work unifies these two lines of work. While we address the weaknesses of static social integration models with a dynamic contextual regularization approach, our primary focus is to enable diverse recommenders to effortlessly integrate with the most suitable social models, enabling more interesting and relevant recommendations.

3 OUR APPROACH

In this section, we describe relevant preliminaries and formalize our problem definition. We discuss the implications of structurally regularizing user representations and provide an intuitive solution to avoid converging to degenerate solutions. Finally in Section 3.2, we describe our approach with a modular adversarial framework for social recommendation.

3.1 Preliminaries

We consider the implicit feedback setting with users $\mathcal{U}$, items $\mathcal{I}$ and binary user-item interaction matrix $\mathcal{Z} \in \mathbb{B}^{|\mathcal{U}|\times|\mathcal{I}|}$ ($\mathbb{B} = \{0, 1\}$). Further, $\mathcal{N} \in \mathbb{B}^{|\mathcal{U}|\times|\mathcal{U}|}$ denotes the explicit social link matrix between the users, we abuse $\mathcal{N}$ to denote both, the social network and its user adjacency matrix. Although we assume undirected social links, the extension to the directed case is straightforward. 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 item interest representations for each user. Without loss of generality, let us denote the social embedding matrix $\mathbf{S} \in \mathbb{R}^{|\mathcal{U}|*d_S}$ and the interest embeddings $\mathbf{X} \in \mathbb{R}^{|\mathcal{U}|*d_X}$. Note that $\mathbf{X}_u, \mathbf{S}_u$ denote the rows for user u. Further, we denote item embeddings $\mathbf{I} \in \mathbb{R}^{|\mathcal{I}|*d_I}$. Given any user embedding matrix $\mathbf{E}$, we can compute user-user similairities in $\mathbf{E}$'s latent space as,

$$p_{\mathbf{E}}(u, v) \propto \sigma(\mathbf{E}_u \cdot \mathbf{E}_v) \tag{1}$$

where $u, v \in \mathcal{U}$ and $\sigma(x) = 1/(1 + e^{-x})$. The social and interest embedding spaces $\mathbf{S}$, $\mathbf{X}$ model the social neighborhoods and item interactions of users, and thus induce different user-user proximities p_S, p_X when placed in Equation (1). Social regularization of interest space $\mathbf{X}$ is achieved by introducing a shared coordinate structure between $\mathbf{S}$ 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 [16], and recently in Collaborative Filtering [8]. Thus, the question follows,

3.1.1 *Can we learn a distance metric to regularize interest embeddings* $\mathbf{X}$ *with social structure* $\mathbf{S}$*?*

Let us consider the embeddings to lie in metric space $\mathbb{M}$ with any metric distance measure $\mathbf{D}_M$. This is the most general form with no constraint on the form of $\mathbf{D}_M$. To transfer structure under metric $\mathbf{D}_M$, for each user-item interaction $(u, i) \in \mathcal{Z}$ we obtain pairwise loss $\|\mathbf{X}_u - \mathbf{I}_i\|_{\mathbf{D}_M} \to 0$ (with user interest embeddings $\mathbf{X}$ and item embeddings $\mathbf{I}$). Similarly, for social links $(u, v) \in \mathcal{N}$, we obtain $\|\mathbf{S}_u - \mathbf{S}_v\|_{\mathbf{D}_M} \to 0$ (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 $\mathbf{D}_M$.

Note the fundamental adversarial nature of the regularization problem in any metric embedding space. 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 with user embeddings $\mathbf{X}_u$ collapsing inwards. The resulting loss in expressivity of interest space $\mathbf{X}$ causes reduced diversity in recommendations (especially for users sharing first, second-order connections in $\mathcal{N}$). We refer to this as *interest space collapse*.

3.1.2 ***Can we transfer the structure of*** $\mathbf{S}$ ***to*** $\mathbf{X}$ ***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 (1) 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 3.1.1 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 now describe and motivate our modular stochastic approach to solve the continuous version of the above regularization problem in an adversarial framework similar to GANs [5]. Note that social regularization is naturally amenable to such an approach due to the competing interest and social spaces. Further, we can socially regularize any gradient optimizable recommender model with our approach, agnostic to its architecture.

3.2 Adversarial Social Regularization

The Generator (**G**) in the GAN framework is a neural model that synthesizes data samples, $\mathbf{y}_G \in \mathbb{R}^d$, drawn from the source distribution $P_G(\mathbf{Y})$ over $\mathbb{R}^d$ induced by **G**. The Discriminator (**D**) on the other hand, attempts to construct a decision boundary to distinguish synthetic samples $\mathbf{y}_G$ drawn from the source distribution against true (positive labeled) samples drawn from an unknown target distribution. The generator is trained to synthesize data points that mimic target samples, hence encoding the target distribution.

In our formulation, the social-agnostic base recommender model learns a 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 $O_{\mathbf{G}}$ over its parameters θ_G. As a result, 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 (1)). We will refer to the base recommender as the generator **G** in our formulation.

On the other hand, social network $\mathcal{N}$ induces a target user-user similarity that the generator must learn to imitate to regularize its interest space $\mathbf{X}$. To compute the target user-user similarity, we apply a Graph Auto-Encoder [13] on network $\mathcal{N}$ and place the learned embeddings in Equation (1). We will denote this as $p_{\mathcal{N}}(u, v)$, the target or *true* user-user similarity from $\mathcal{N}$.

Finally, discriminator **D** learns an independent social embedding space $\mathbf{S}$ for users separate from social network $\mathcal{N}$. The discriminator induces social proximity, $p_{\mathbf{S}}(u, v)$ of users in its latent social space, forming the link between the target $p_{\mathcal{N}}(u, v)$ and source $p_{\mathbf{X}}(u, v)$, and attempts to move them closer. We highlight two key advantages of the adversarial regularization strategy —

1) It enables our modular optimization strategy (Section 3.2.1), providing flexibility in the recommender **G** and discriminator **D**'s architectures. In our experiments, we substitute and show gains for multiple strong neural recommenders as **G** with a convolutional discriminator [12] to capture social representations.

2) We enable pair-specific expressivity in Section 3.2.2 as motivated in section 3.1.2 to provide a wider choice of target $p_{\mathbf{X}}$ given source $p_{\mathcal{N}}$, hence reducing the likelihood of interest-space collapse and providing contextual social structure integration in $\mathbf{X}$.

3.2.1 ***Structure Regularization:*** We propose a robust stochastic approach to represent source $p_{\mathcal{X}}$ and target $p_{\mathcal{N}}$ with a finite number 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, the discriminator must assign higher likelihoods to the *true-pairs* sampled from target $p_{\mathcal{N}}$ (denoted (u_+, v_+)), and lower likelihoods to *fake-pairs* sampled from the source $p_{\mathbf{X}}$ (denoted (u_-, v_-)), while the generator's goal is to confuse the discriminator, i.e., maximize expected fake-pair likelihood $\mathbb{E}(p_{\mathbf{S}}(u_-, v_-))$. Thus, we obtain overall objective O,

$$O = \min_{\mathbf{X}} \max_{\mathbf{S}} \Big(\mathbb{E}_{(u_+, v_+) \sim p_{\mathcal{N}}} \log p_{\mathbf{S}}(u_+, v_+) + \mu . \mathbb{E}_{(u_-, v_-) \sim p_{\mathbf{X}}} \log \big(1 - p_{\mathbf{S}}(u_-, v_-)\big) \Big) \tag{2}$$

where μ is the balance parameter. When we optimize O, **G** learns X so that *fake-pairs* $(u_-, v_-) \sim p_{\mathbf{X}}$ confuse the discriminator i.e., maximize $\log p_{\mathbf{S}}(u_-, v_-)$. Conversely, the discriminator attempts to maximize expected *true-pair* likelihood $\log p_{\mathbf{S}}(u_+, v_+)$ and minimize *fake-pair* likelihood $\log p_{\mathbf{S}}(u_-, v_-)$. The expectations $E_{(u,v)}$ are averaged over ϵ *fake* and *true-pair* samples each (policy-gradient approximation) [32, 34].

We find in Section 5.5 that the number of *fake* and *true* user pair samples ϵ required for robust convergence is $\leq 2\%$ of the distinct user pair count ($|\mathcal{U}|^2$), enabling much faster training than Coordinate Transfer Learning [26]. Further, our approach is observed to be robust to lossy social data (Figure 9). We perform stratified sampling to equally represent all users in the *fake* and *true-pair* sample sets, denoted ϵ_-, ϵ_+ respectively ($|\epsilon_-| = |\epsilon_+| = \epsilon$).

Equation (2) 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 3.1.2. We now describe an intuitive pair weighting strategy to enable a wider choice of the target $p_{\mathbf{X}}$ by learning to prioritize the most important parts of $p_{\mathcal{N}}$ (contextual social regularization).

3.2.2 ***User Pair Weighting to Avoid Interest Space Collapse.*** In our formulation, interest space collapse can cause **G** to learn interest space $\mathbf{X}$ with shallow variety, moving towards degenerate solutions to the Min-Max game in Equation (2). 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, such as close friend links vs. celebrity-follower links, correlation of the interersts of each social contact to a user, expertise etc. The augmented Min-Max objective is as follows —

$$O = \min_{\mathbf{X}} \max_{\mathbf{S}} \Big(\mathbb{E}_{(u_+, v_+) \sim p_N} \ \log p_{\mathbf{S}}(u_+, v_+) + \mu . \mathbb{E}_{(u_-, v_-) \sim p_{\mathbf{X}}} \ w(u_-, v_-) \log \big(1 - p_{\mathbf{S}}(u_-, v_-)\big) \Big) \tag{3}$$

Note that the above transformation regularizes the product $w(u, v) \times p_{\mathbf{X}}(u, v)$ against $p_{\mathbf{S}}$ (instead of just $p_{\mathbf{X}}$ against $p_{\mathbf{S}}$), enabling a much wider choice of $\mathbf{X}$. The contextual weighting function $w(u, v)$ accounts for diverse social relations with varying levels of interest sharing. 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 only for the ϵ *fake-pairs* in sample set ϵ_- and adds limited overhead ($\epsilon \ll |\mathcal{U}|^2$).

4 MODEL DETAILS

We now describe the architectural details of $\mathbf{G}$, contextual pair weighting function $w(u, v)$, discriminator $\mathbf{D}$ and an alternating optimization approach to train these modules.

4.1 Generator Architecture

We limit our assumptions on the generator to the most general hypothesis, namely $\mathbf{G}$ learns the user interest embeddings $\mathbf{X}$ (and any other parameters θ_G) by optimizing differentiable continuous objective $O_{\mathbf{G}}$. In our experiments, we show gains by socially regularizing the three best performing neural baselines with our framework.

Fake-pair Sampling: Fake-pairs (u_-, v_-) are sampled by first choosing u_-, and then sampling $v_- \propto p_{\mathbf{X}}(u_-, v_-)$. We stratify the samples per user, so that each user appears in at least $\epsilon/|\mathcal{U}|$ pairs.

True pair Sampling: True pairs are representative of the underlying social network structure. They are sampled similar to the fake pairs above by replacing the generator embeddings with Graph Auto-Encoder [13] embeddings from social network $\mathcal{N}$.

We now describe the parametrization of the contextual weighting function $w(u_-, v_-)$.

4.2 Attentive Hadamard Pair Weighting

Multiplicative cross-factors between the context features of a pair of users are natural indicators of homogeneity and heterogeneity. For instance, the multiplicative cross-factors across appropriate dimensions of interest embeddings $\mathbf{X}_u$ and $\mathbf{X}_v$ can help us infer shared interests and differences between pair (u, v). A similar intuition generalizes across other user-features.

Towards this transformation, we propose a simple Hadamard projection approach to achieve low-rank bilinear pooling of user features in the contextual weight function $w(u, v)$. We learn a projector matrix $\mathbf{P} \in \mathbb{R}^{N * d_w}$, where d_w is the dimensionality of contextual user features. Each row of the projector matrix, $\mathbf{P}_i$, $i \in [1, ..., N]$, represents a unique transformation on the user context. For each user pair sample (u, v), the input representations are projected as (using interest embeddings $\mathbf{X}_u$ as the contextual features) —

$$\mathbf{X}_u^i = \mathbf{X}_u \odot \mathbf{P}_i, \ \mathbf{X}_v^i = \mathbf{X}_v \odot \mathbf{P}_i$$

where $\odot$ denotes the Hadamard product operation. We then compute attention weights for each projector to represent the alignment of the users under it's projected dimensions, i.e.,

$$a_n(u, v) = \frac{\exp(\mathbf{X}_u^n . \mathbf{X}_v^n)}{\sum_{i=1}^{N} \exp(\mathbf{X}_u^i . \mathbf{X}_v^i)}$$

The higher weight $a_n(u, v)$, stronger the multiplicative cross-factors for pair (u, v) across dimensions projected by $\mathbf{P}_n$. We then compute pair alignment vector $\mathbf{A}(u, v)$ as a weighted projector sum,

$$\mathbf{A}(u, v) = \sum_{n=1}^{N} a_n(u, v) \mathbf{P}_n$$

Alignment vector $\mathbf{A}(u, v)$ denotes the nature of the relation between users (u, v). It is then transformed to the pair weight value $w(u, v)$ through a single feed-forward layer. Additionally, we introduce a batch sparsity regularizer across the N projectors to incentivize sparsity and hence diversity in their projected dimensions.

There is a loss in expressivity moving from $\mathbf{A}(u, v)$ to weight $w(u, v)$ for a user pair. We can address this by transforming each projection and their interactions separately to obtain a fine-grained joint expression. We leave this investigation to future work.

4.3 Discriminator Architecture

The discriminator architecture $\mathbf{D}$ learns social representations $\mathbf{S}$ by optimizing the Min-Max objective in eq. (3). It hence parametrizes the proximity $p_{\mathbf{S}}(u, v)$. We explore a few simple architectural choices to keep the computational overhead to a minimum—

- **Inner Product Discriminator:** The inner product discriminator parametrizes the likelihood $p_{\mathbf{S}}(u, v)$ as $1/(1 + e^{-\mathbf{S}_u . \mathbf{S}_v})$. We also expore a bilinear form $p_{\mathbf{S}}(u, v) = 1/(1 + e^{-(\mathbf{S}_u)^{\mathrm{T}} \mathbf{W_B} \mathbf{S}_v})$. Thus the embeddings $\mathbf{S}$ (and bilinear weight parameter $\mathbf{W_B}$) are learned directly by optimizing eq. (3) with these functional forms of $p_{\mathbf{S}}$.
- **MLP:** We apply a RelU bi-layer perceptron to encode the normalized Laplacian matrix $\mathbf{L}$ of the social network $\mathcal{N}$ to the latent social embeddings $\mathbf{S}$. Note that $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{1/2}$ where $\mathbf{A}$ and $\mathbf{D}$ denote the adjacency and degree matrices of $\mathcal{N}$. Once again, $p_{\mathbf{S}}(u, v) = 1/(1 + e^{-\mathbf{S}_u . \mathbf{S}_v})$ where $\mathbf{S}_u = \mathbf{MLP}(\mathbf{L}_u)$.
- **Graph Convolutional Network:** The convolution operations on the social network $\mathcal{S}$ is given by the product of input user features $\mathbf{F}_u \in \mathbb{R}^n$ with learned filter g_θ in the fourier domain,

  $$g_\theta * \mathbf{F}_u = g_\theta(\mathbf{Q} \Lambda \mathbf{Q}^T) \mathbf{F}_u = \mathbf{Q} g_\theta(\Lambda) \mathbf{Q}^T \mathbf{F}_u$$

  where rows of $\mathbf{Q}$ are the eigenvectors of Laplacian $\mathbf{L}$.

To circumvent the expensive eigen-decomposition of the Laplacian, Defferrard et al. [1] proposed to approximate filter $g_\theta(\Lambda)$ with truncated Chebyshev polynomials $T_k(x)$ to the kth order. This approximation results in k-localization, i.e. node representations incorporate k-hop neighborhoods. Kipf and Welling [12] further simplified this to a first-order linear form (GCN). We stack k GCN layers to condition $\mathbf{S}$ on the k-hop social neighborhoods of users.

The feature inputs to the kth GCN layer are the user representations from the previous layer, $\mathbf{F}^{k-1} \in \mathbb{R}^{|\mathcal{U}| \times d_{k-1}}$, where d_{k-1} is the dimensionality of the $(k\text{-}1)^{th}$ GCN layer. Thus,

$$\mathbf{F}^k = \sigma(\hat{\mathbf{A}} \mathbf{F}^{k-1} \mathbf{W}), \ \ \hat{\mathbf{A}} = \mathbf{D}^{-1/2} \hat{\mathbf{A}} \mathbf{D}^{-1/2} + \mathbf{I}$$

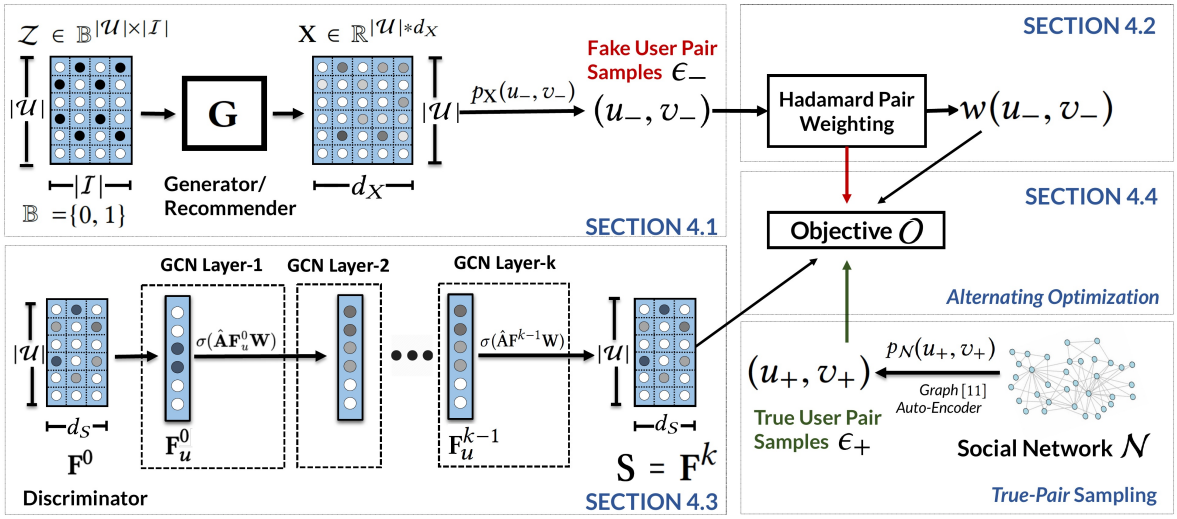


Figure 2: Architecture diagram illustrating the model components and computation of the loss terms that appear in the adversarial objective in Equation (3). We do not place any restrictions on the architecture of recommender G.

Note that inputs $\mathbf{F}^0$ are the node features of the users in $\mathcal{N}$. We used one-hot feature inputs in our experiments. The social embedding matrix $\mathbf{S}$ is k^{th} layer ouput i.e., $\mathbf{S} = \mathbf{F}^k$. Thus,

$$p_{\mathbf{S}}(u, v) = 1/(1 + e^{-\mathbf{F}_u^k \cdot \mathbf{F}_v^k})$$

We set the dimensions of all GCN layers F^k and social embeddings S to the same value d_S. In our experiments we find two and three layer GCNs (k=2,3) to significantly outperform the Inner-product and MLP variants. Although we expect further improvements with architectures such as Graph Attention [31], we leave this investigation to future work.

4.4 Model Optimization

We now describe our alternation optimization approach and the specific objective functions for each of the previous three modules. The optimization objective for each module is obtained by separating out the relevant terms from Equation (3).

4.4.1 Generator Objective: In the absence of our adversarial framework, recommender (generator) $\mathbf{G}$ optimizes $O_{\mathbf{G}}$ to learn $\mathbf{X}$ and associated parameters θ_D. The adversarial term optimizes the discriminator likelihood of G's *fake-pair* samples,

$$\mathbf{X}, \theta_D = \arg\min_{\mathbf{X}, \phi} \left(O_{\mathbf{G}} + \frac{\lambda}{\epsilon} \sum_{\epsilon_-} w(u_-, v_-) \log \left(1 - p_{\mathbf{S}}(u_-, v_-)\right) \right) \tag{4}$$

Note that constant λ controls the adversary weight (i.e., overall regularization strength). Also note that minimizing the second term is equivalent to maximizing $w_(u_-, v_-) \log \left(p_{\mathbf{S}}(u_-, v_-)\right)$. The generator updates $\mathbf{X}$ to increase the likelihood of generating *fake-pairs* with higher contextual weights and discriminator likelihoods.

4.4.2 Discriminator Objective: The discriminator learns social space $\mathbf{S}$ and associated parameters θ_D, to maximize the similarity or likelihood $p_{\mathbf{S}}$ of the *true-pairs* and minimize that of the *fake-pairs* sampled from the generator's interest space $\mathbf{X}$,

$$\mathbf{S}, \phi = \arg\max_{\mathbf{S}, \phi'} \left(\frac{1}{\epsilon} \sum_{\epsilon_+} \log p_{\mathbf{S}}(u_+, v_+) + \frac{\mu}{\epsilon} \sum_{\epsilon_-} w_{(u_-, v_-)} \log \left(1 - p_{\mathbf{S}}(u_-, v_-)\right) \right) \tag{5}$$

As a result, the discriminator progressively learns finer distinctions between samples from $p_{\mathbf{X}}$ and $p_{\mathcal{N}}$. In response, $\mathbf{G}$ selectively embeds the social structure to generate harder *fake-pair* samples. Note that pair weighting enables, in theory, an infinitely wide choice for $p_{\mathbf{X}}$ to differ from $p_{\mathcal{N}}$. In practice however, model expressivity depends on the context features provided to the weighting module.

4.4.3 Pair Weighting Objective: The Hadamard network learns to prioritize pairs that result in minimizing $\mathbf{G}$'s loss while keeping $\mathbf{X}, \mathbf{S}$ fixed. This translates to the following objective.

$$\mathbf{P}, \theta_w = \arg\min_{\mathbf{P}, \theta_w} \left(\frac{\lambda}{\epsilon} \sum_{\epsilon} w(u_-, v_-) \log \left(1 - p_{\mathbf{S}}(u_-, v_-)\right) \right) + \sum_{n=1}^{N} \|\mathbf{P}_n\|_2 \tag{6}$$

We impose group Lasso (each $\mathbf{P}_n$ is a group) regularization to avoid over-fitting and incentivize sparse projectors. By combining objectives section 4.4.1, eq. (5), eq. (6) we can re-obtain eq. (3) with minor modifications. Each module is trained alternately holding the other two constant via *ADAM* gradient updates [11].

Computational Complexity: On the whole, our model complexity is $O(\mathbf{G}) + O(|\mathcal{N}| \times d_{\mathbf{S}}) + O(\epsilon \times N \times d_w)$, where $O(\mathbf{G})$ is the recommender complexity, $|\mathcal{N}|$ is the social link count, N is the number of Hadamard projectors, $d_{\mathbf{S}}$ the social space dimensionality, d_w the user context feature dimensionality and ϵ is the *fake/true-pair* sample count. In practice, our modules are highly parellel and discriminator $\mathbf{D}$ is implemented with sparse optimizations [12].

In practice, the discriminator $\mathbf{D}$ and pair weight module $w(u, v)$ add 50% overhead to train Auto-Encoder based recommenders [20, 38] (less percentage overhead for more complex recommender

architectures), if the dimensions of $\mathbf{S}$, $\mathbf{X}$ are equal, i.e., $d_{\mathbf{S}} = d_{\mathbf{X}}$. The overheads are reduced further if $d_{\mathbf{S}} < d_{\mathbf{X}}$.

5 EXPERIMENTAL RESULTS

In this section, we present extensive quantitative and qualitative analysis of our model. We begin by introducing datasets and baseline methods in Section 5.1, followed by the primary recommendation task in Section 5.2, and quantitative results by integrating three diverse neural recommenders in our framework (Table 1). Then in Section 5.3, we analyze the user segments where our model exhibits gains, and study the pair samples that were important in the model's learning process in Section 5.4. In Section 5.5 we examine two important questions: Q1—*What is the effect of adversary weight λ on interest space collapse and does this depend on the generator architecture?* and Q2—*Is adversarial training robust to missing social or item history user data?* Finally, we analyze parameter sensitivity in Section 5.6 and discuss limitations in Section 5.7.

5.1 Datasets and Baselines

We evaluated all models over five publicly available datasets, *Delicious*, *Ciao*, *Epinions*, *Ask-Ubuntu* and *Yelp*.

Ciao[1]: The Ciao dataset contains user's ratings on DVDs, the user social network and DVD category data.

Epinions[1]: The Epinions dataset provides user ratings to purchased items, the user social network, and item categories.

Ask-Ubuntu[2]: Ask-Ubuntu is a popular online Q&A forum. We predict tags for users' posts. Social links are interactions between users via comments, answers or edits.

Delicious[3]: The Delicious dataset contains user bookmarks, social links, and tags. We predict bookmarks in our experiments.

Yelp[4]: The Yelp dataset contains user ratings to restaurants and their social network.

We pre-process smaller datasets (*Ciao, Epinions, Delicious*) to retain users and items with 10 or move reviews. For *Ask-Ubuntu* and *Yelp*, we set the threshold to 30. We compare our framework against recent state-of-the-art baselines. We present gains by integrating the three strongest social-agnostic recommender baselines as the generators in our framework.

BPR [27]: BPR is a first-cut baseline for all implicit feedback recommendation methods.

SBPR [39]: SBPR aguments personalized ranking by assuming users assign higher ranks to their friends' preferences.

NCF [6]: NCF is a state-of-the-art neural ranking model combining matrix factorization and neural representation learning. NCF outperforms most conventional baselines.

SNCF: We modify NCF by concatenating social network embedding representations (as in [30]) in the neural inputs. We refer to this variant as Social NCF (SNCF).

Social-GCN [35]: Social-GCN convolves user neighbor features and optimizes a personalized ranking objective function.

SEREC [33]: SEREC assumes users are exposed to items reviewed by their contacts, some leading to purchases. SEREC is competitive on most datasets due to its flexible item choices.

CB [37]: Contextual-Bandit (CB) uses dual graph-attention networks to compute user interest and social embeddings, and selects one of the factors to explain each purchase.

DAE [38]: Denoising Auto-Encoders learn a low-dimensional user interest representation by decoding a noised version of his item history. We incorporate DAE as an adversarial variant.

VAE-CF [20]: Variational Auto-Encoders eliminate noisy inputs by introducing stochasticity in the user interest space. We incorporate VAE and evaluate its gains in our framework.

LRML [30]: LRML is a memory network architecture to learn relation vectors between user-item pairs. We incorporate LRML in our adversarial framework.

We tested our framework by incorporating **DAE**, **VAE-CF** and **LRML** as generators $\mathbf{G}$ in our framework. We refer to these variants as **Asr-DAE**, **Asr-VAE** and **Asr-LRML** (**Asr** denotes adversarial social regularization). Experiments were performed on a Nvidia Tesla V100 GPU with *TensorFlow* implementations on the Linux platform. Our implementations will be made publicly available[5].

5.2 Recommendation

To evaluate the performance of the recommender models listed above, we compute the *NDCG@K* (*N@K*) and *Recall@K* (*R@K*) metrics [19]. *Recall@K* is a measure of the percentage of relevant items in the top-K recommendations to each user; it considers true and false positives in the list and is thus more descriptive than Precision. The *NDCG@K* metric is position sensitive and considers the order of the ranked list against the ideal case (only relevant items placed at the top). We evaluate each ranked list at $K = 20, 50$ (Table 1).

We randomly split each dataset into Training (80%), Validation (10%) and Test (10%). We tune the baselines with parameter ranges centered at the author provided values, to obtain the best performance on our datasets. For fair comparison, we set the representation dimensions to 128 for all models. For our model, adversary weight λ, balance μ were both tuned in the range $(0, 10]$ and we set Hadamard projectors $N = 10$ across all experiments.

Comparative Analysis : We make several observations from the experimental results obtained with the baseline recommenders and our adversarial variants (Table 1). First, conventional social recommenders are outperformed by social-agnostic neural methods that efficiently leverage the rating information. Non-linear transformations of ineterest representations are more expressive than linear or bi-linear operations [6].

Second, expressive interest spaces (like in **DAE** [38]) benefit more from social regularization than conventional interest representations. The gains achieved by integrating neural models in our framework are stronger than those adding social information to older methods (e.g., *R@50* gains of **SBPR** vs **BPR** are smaller on average than those of **Asr-VAE** vs **VAE**). Also, note that a direct integration of pre-trained embeddings (as in **SNCF**) does not produce a noticeable gain in performance. Pre-trained graph embeddings

[1]https://www.cse.msu.edu/ tangjili/datasetcode/truststudy.htm
[2]https://archive.org/details/stackexchange
[3]https://grouplens.org/datasets/hetrec-2011/
[4]https://www.yelp.com/dataset/challenge

[5]https://github.com/CrowdDynamicsLab/Adversarial-Social-Recommendation

Method	Smaller Datasets												Larger Datasets							
	Epinions				Ciao				Delicious				Ask-Ubuntu				Yelp			
	R@20	*N@20*	*R@50*	*N@50*	*R@20*	*N@20*	*R@50*	*N@50*	*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 [27]	0.264	0.141	0.440	0.176	0.232	0.128	0.428	0.162	0.363	0.271	0.590	0.328	0.377	0.199	0.514	0.264	0.228	0.125	0.431	0.170
NCF [6]	0.310	0.138	0.462	0.181	0.282	0.147	0.471	0.193	0.498	0.283	0.753	0.401	0.420	0.215	0.538	0.281	0.196	0.118	0.488	0.209
DAE [38]	0.324	0.164	0.498	0.198	0.290	0.143	0.493	0.191	0.572	0.340	0.762	0.393	0.416	0.301	0.569	0.392	0.270	0.158	0.473	0.213
VAE-CF [20]	0.336	0.161	0.510	0.204	0.299	0.152	0.496	0.197	0.585	0.327	0.771	0.416	0.408	0.317	0.576	0.383	0.281	0.164	0.479	0.208
LRML [30]	0.329	0.173	0.509	0.219	0.317	**0.165**	0.526	0.206	0.482	0.310	0.698	0.384	0.405	0.366	0.564	0.405	0.272	0.160	0.483	0.196
Social Recommenders																				
SBPR [39]	0.271	0.138	0.446	0.185	0.217	0.140	0.439	0.174	0.381	0.292	0.625	0.362	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	0.284	0.151	0.478	0.196	0.520	0.296	0.747	0.380	0.414	0.371	0.541	0.403	0.198	0.103	0.493	0.202
SGCN [35]	0.318	0.153	0.481	0.198	0.275	0.142	0.470	0.179	0.546	0.295	0.718	0.401	0.397	0.343	0.526	0.395	0.288	0.160	0.492	0.176
CB [37]	0.337	0.171	0.436	0.202	0.288	0.153	0.491	0.180	0.572	0.287	0.753	0.384	0.399	0.365	0.559	0.382	0.282	0.154	0.471	0.196
SEREC [33]	0.348	0.167	0.496	0.213	0.303	0.158	0.513	0.202	0.589	0.314	0.762	0.398	0.415	0.362	0.584	0.414	0.306	0.173	**0.508**	0.211
Adversarial Social Recommenders (Ours)																				
Asr-DAE	0.339	0.168	0.513	0.207	0.301	0.144	0.519	0.189	0.603	0.322	0.784	0.418	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	0.312	0.138	0.528	0.196	**0.617**	**0.379**	**0.797**	**0.442**	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	**0.328**	0.160	**0.544**	**0.214**	0.495	0.357	0.743	0.426	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 as the generator in our adversarial framework. Our model can substitute recommender (generator) and discriminator architectures owing to the modular formulation. The performance numbers in bold numerals indicate statistically significant gains over the second best model at $p = 0.05$. When there are two or more strong performers under a specific metric, we underline them. Our adversarial variants exhibit strong gains over competing social recommenders as well as their respective base models.

Table 1: $R@K$ and $N@K$ denote the Recall and NDCG metrics for all models. Our models outperform competing baselines by upto 35% Recall@50 and 25% NDCG@50. Asr-VAE was found to be the best overall model.

cannot contextually distinguish the influence of a user's neighbors by their interests.

We find our adversarial variants and **SEREC** to outperform older social recommender baselines by significant margins. While **SEREC** permits for the exposed item set to be prioritized differently, **CB** [37] flexibly attributes puchases, however picking a single factor (interest vs social) instead of a contextual combination. **Asr-VAE** was found to achieve the best overall performance. The **VAE** user representations are inherently stochastic unlike **DAE** and **LRML**, we also observed greater recommendation diversity (less interest space collapse) with **Asr-VAE** (Section 5.5).

Neighbor Diversity: Unlike exposure models, we condition each item on the specific social context i.e., a phone exposed by an android expert has a greater effect than from other social contacts. To verify this, we measure the diversity of each user's friends. As an example, if a user's four friends have 5, 10, 15, and 20 items, their item distribution is $(5/50, 10/50, 15/50, 20/50)$. We estimate the KL-divergence of this distribution against the uniform case to measure diversity. We then split all users into four quartiles based on their neighbor diversity (Q4 has users with high neighbor diversity), and compare *R@50* relative gains of **Asr-VAE** over **SEREC** on samples from each quartile. Ideally, we expect our model to make gains on later quartiles since context is more important to distinguish diverse social contacts.

Neighbor Diversity Quartile	**Q1**	**Q2**	**Q3**	**Q4**
% Gain R@50 (Asr-VAE vs SEREC)	3.82%	3.16%	3.45%	4.23%

Table 2: Performace gains of Asr-VAE against SEREC on user neighbor diversity. We see stronger gains for Q4 (high neighbor item-count diversity)

5.3 Interpreting our Results

We now study our results more closely to understand the source of **Asr-VAE**'s gains over base recommender **VAE**. We observe the *R@50* performance values of **Asr-VAE** against the base recommender **VAE** to observe the source of our gains. We analyze users along three axes -

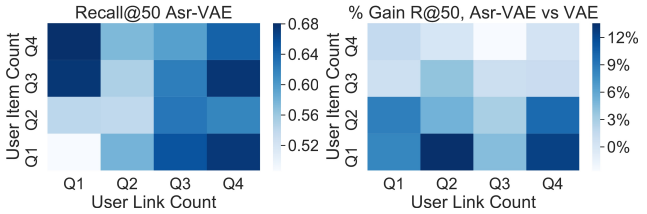
Item Count Quartile: We separate the test users into 4 quartiles based on the number of items in their histories.

Social Links Quartile: We again separate test users into 4 quartiles depending on their social link counts.

User Coherence Quartile: We define user coherence as the mean pair-wise correlation of item categories purchased by the user. Thus if a user were to purchase items that are often bought together, he receives greater coherence. We partition test users in 4 quartiles by coherence scores. We can compute coherence only for the *Epinions* and *Ciao* datasets.

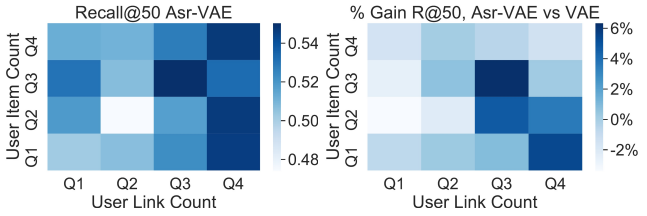
We first study the overall performance variations and performance gains for users in different social and item count quartiles.

Overall Results: The heatmaps on the left in Figure 3, Figure 4 indicate the performance (*R@50*) achieved by Asr-VAE for users grouped under each quartile (Q1 - Lower values), averaged over the smaller and larger datasets respectively. We observe weaker performance for users at the bottom-left of the plot (i.e., users with sparse links and items). For the small datasets, stronger results appear at the other three corners (i.e., users who have either have a long item history, or several social connections). On the large datasets, results are concentrated towards users with greater link counts (social link quartiles Q3 and Q4). These gains are consistent with our intuitions, users with a large item history obtain accurate



(a) Asr-VAE $Recall@50$ (b) % Gain Asr-VAE vs VAE $R@50$

Figure 3: Overall Performance and Percentage Gains of Asr-VAE (by R@50), measured across social link count and item count user quartiles (Q1 = lowest values, Q4 = highest values). Heatmap values are averaged over the smaller datasets (*Ciao, Epinions, Delicious*)



(a) Asr-VAE $Recall@50$ (b) % Gain Asr-VAE vs VAE $R@50$

Figure 4: Overall Performance and Percentage Gains of Asr-VAE (R@50), mean over larger datasets (*Ask-Ubuntu, Yelp*)

interest representations while those with more social links can socially regularize their interest embeddings.

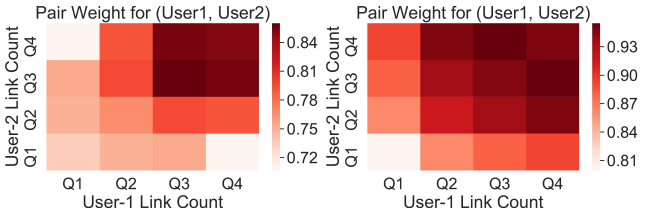
Difference between Asr-VAE and VAE-CF: The heatmaps on the right of each figure (Figure 3, Figure 4) indicate the relative performance gains of **Asr-VAE** against its base recommender **VAE** for users in the respective quartiles. We observe stronger performance gains in the bottom half of each heatmap, indicating improvements for users in the 25% and 50% item quartiles (i.e., users with sparse interest representations). As expected, social regularization benefits users with limited purchase histories. Surprisingly, we also see gains in the bottom left corner for the smaller datasets. It is feasible that users in these quartiles have fewer informative social links (since the datasets are smaller), making gains in **Asr-VAE** vs **VAE**.

5.4 Pair-Weight Allocations

The Hadamard projection vectors in our weighting function $w(u, v)$ are hard to interpret, since we do not know what each latent dimension is, however pair weights assigned to pair samples can be aggregated to analyze the training process.

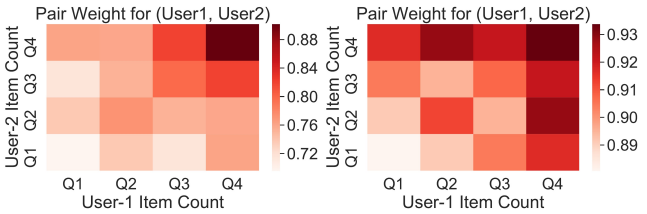
We observe from Figure 5, Figure 6 that our model prioritizes pairs of users where both users have numerous social connections or longer item histories to regularize their neighborhoods. Intuitively, pair samples where both users are influencers or prolific consumers, are likely to be regularize their social and interest neighborhoods respectively (they may act as cluster centers). We observe a similar trend against user coherences in the *Ciao* dataset (Figure 7). In *epinions*, the model also prioritizes quartiles where one user in a pair has more coherent purchases than the other (note that we can only compute coherence for the *Ciao, Epinions* datasets using their item category labels).

Finally, we also analyze pair weights by considering differences within user pairs. We look at the difference in the number of social



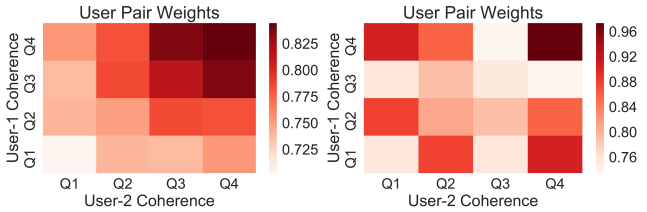
(a) Smaller Datasets (*Ciao, Epinions, Del*) (b) Large Datasets (*Ask-Ubuntu, Yelp*)

Figure 5: 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)



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

Figure 6: We create these heatmaps similar to Figure 5 with user item count quartiles, i.e., Q4 denotes long item histories



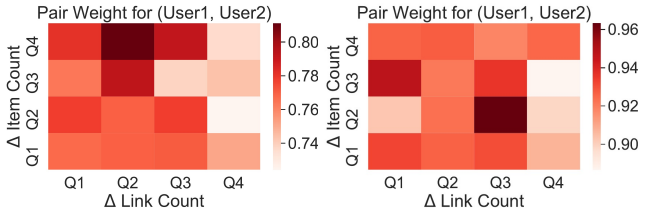
(a) *Ciao* dataset (b) *Epinions* dataset

Figure 7: Pair weights against user coherence for pair samples in the *Ciao* and *Epinions* datasets

counts and length of item histories of the two users. Figure 8 indicates a slight drop in pair weights at the extreme right of each plot (large difference in social link count). Such connections are unlikely to represent friend-friend links and hence may not effectively regularize preferences. However, in *Yelp*, *Ask-Ubuntu* we observe a more uniform distribution of pair weights, potentially due to the information-seeking requirements of users on these websites.

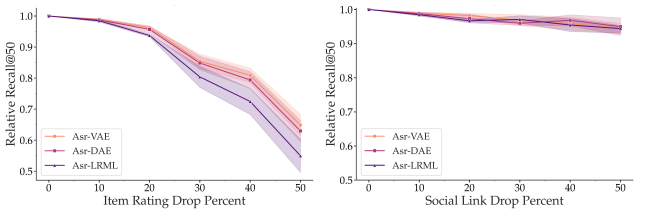
5.5 Robustness and Interest Space Collapse

We study the robustness of each adversarial model to lossy data by separately sub-sampling the social links and item ratings of each user in the respective training sets (Figure 9). Performance is measured as a fraction of the peak performance (e.g., 0.98 indicates the model degraded by 2%). We observe an average performance degradation $\leq 3\%$ by *R@50* with 10% item ratings dropped and $\leq 6\%$ at 20% drop, indicating our models are fairly robust to lossy item ratings. Asr-LRML shows a slightly steeper drop compared to the



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

Figure 8: Each pair sample (User 1, User 2) is binned in quartiles by item and social link count differences between the two users. We then plot the average pair weights assigned to the pair samples within the respective quantiles.



(a) Performance with Item Drop (b) And Social Link Drop

Figure 9: We observe $\leq$ 6% *R@50* degradation at 20% item drop indicating our models are fairly robust in practice. Dropping social links results in much smaller performance drops, indicating the effectiveness of stochastic user pair sampling. Performance values are averaged across datasets.

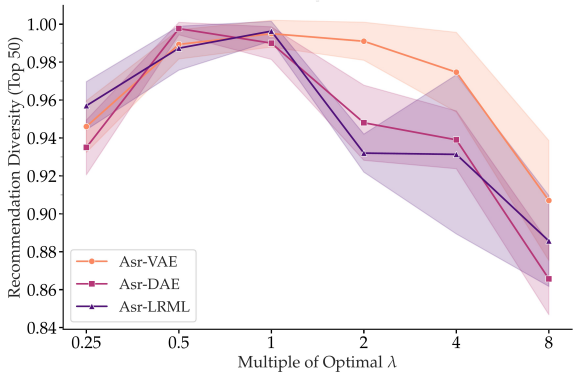


Figure 10: Recommendation diversity is observed to drop on either side of λ_{opt}, but due to different causes. The smaller λ multiples result in overfitting to the supervised term O_G, while larger multiple result in interest space collapse, i.e., less diverse recommendations to socially clustered users. Diversity values are averaged across datasets.

auto-encoder variants. Further, we observe our models are highly robust to social link drop, degrading by 5% *R@50* even with 50% social links dropped, owing to their stochastic pair sample based gradient updates.

We also analyze the effect of adversary weight λ on the diversity of items recommended to users (Figure 10). Specifically, we apply k-means clustering to the GAE [13] embeddings for each social network, pick the median user cluster by average degree and

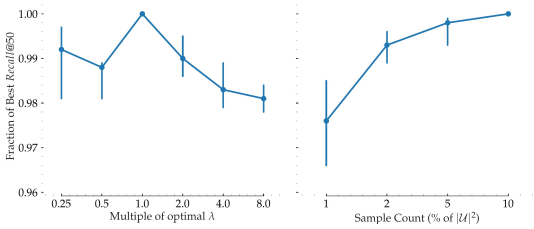


Figure 11: Asr-VAE is fairly robust in a wide range of values ($\leq$ 2.5% *R@50* variation). λ is varied as multiples of the best performing value λ_{opt}, larger multiples result in a performance drop. Robust performance is obtained for user pair sample count $\epsilon = 0.02|\mathcal{U}|^2$, further samples provide small gains ($\leq$ 1%). *R@50* values are averaged across datasets.

measure recommendation diversity as the union of their top-50 recommendations. λ_{opt} indicates the optimal λ setting by *R@50* for each dataset. As λ is varied, the variation in diversity is measured as a percentage of the largest union set obtained (i.e., less diversity implies a smaller union set and hence lower percentage).

In general, larger values of λ result in less diverse recommendations. **Asr-VAE**'s recommendations are slightly more diverse at greater values of λ owing to the stochasticity of the user representations in the **VAE** generator. On the opposite end, smaller multiples of λ also produce lower recommendation diversity by over-fitting to the supervised loss term O_G in the generator objective in Equation (4). Values close to λ_{opt} produce the most diverse set of top-50 recommendations.

5.6 Parameter Sensitivity

In this section, we study the sensitivity of our model to two key parameter values, first the adversary weight λ, and second the user pair sample count ϵ (Section 5.6) measured as a fraction of the total number of unique user pairs (e.g., 5% denotes $0.05 \times \mathcal{U}^2$).

Varying the adversary weight λ results in a performance drop on either side of the optimal value. We find $\epsilon = 0.02 \times \mathcal{U}^2$ to provide an efficient tradeoff between compute-cost and performance. In practice ϵ does not significantly change the overall compute time, since the pair weight module is inexpensive. Also note that λ_{opt} varies across datasets, with values on either side of λ_{opt} resulting in weaker and less diverse recommendations (Figure 10).

5.7 Limitations

We identify a few key limitations of our model. First, although the model performance is stable around multiples of the optimal values of λ_{opt}, the optimal weight varies across datasets and applications. The optimal strength of social regularization depends on the data semantics as well the generator and discriminator architectures. Second, the pair-weighting strategy performs best when the provided context features are meaningfully correlated to the interests and social indicators of users. Thus, depending on the application, context features should be picked to enhance social inference and prevent loss of diversity in the generated recommendations.

6 CONCLUSION

This paper proposes an approach to integrate the social network between users to improve the quality of their recommendations. Unlike prior work, we adopt a modular architecture-agnostic framework enabling a broad range of recommender applications. Further, we show that a direct application of metric learning approaches or equivalent formulations could result in interest space collapse. Our pair weighting approach enhances the expressivity of the user interest space and permits for contextual integration of their social structure. Extensive experimental results over five real-world datasets reveal the strengths of our approach.

We identify three rewarding future directions - developing smart samplers to produce informative *fake-pairs* to regularize the interest space, enhancing contextual weighting with a fine-grained combination of the context projections, and finally, developing efficient and expressive discriminator architectures.

7 ACKNOWLEDGMENTS

Our research was funded by an Amazon Research Awards grant[6].

REFERENCES

[1] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*. 3844–3852.

[2] Tyler Derr, Yao Ma, and Jiliang Tang. 2018. Signed graph convolutional networks. In *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 929–934.

[3] Shanshan Feng, Gao Cong, Arijit Khan, Xiucheng Li, Yong Liu, and Yeow Meng Chee. 2018. Inf2vec: Latent Representation Model for Social Influence Embedding. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*. IEEE, 941–952.

[4] David Fullagar, Christopher Newton, and Laurence R Lipstone. 2015. Handling long-tail content in a content delivery network (CDN). US Patent 8,930,538.

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.

[6] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 173–182.

[7] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 549–558.

[8] Cheng-Kang Hsieh, Longqi Yang, Yin Cui, Tsung-Yi Lin, Serge Belongie, and Deborah Estrin. 2017. Collaborative metric learning. In *Proceedings of the 26th international conference on world wide web*. International World Wide Web Conferences Steering Committee, 193–201.

[9] Mohsen Jamali and Martin Ester. 2010. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 135–142.

[10] Meng Jiang, Peng Cui, Rui Liu, Qiang Yang, Fei Wang, Wenwu Zhu, and Shiqiang Yang. 2012. Social contextual recommendation. In *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM, 45–54.

[11] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[12] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[13] Thomas N Kipf and Max Welling. 2016. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308* (2016).

[14] Adit Krishnan, Ashish Sharma, Aravind Sankar, and Hari Sundaram. 2018. An Adversarial Approach to Improve Long-Tail Performance in Neural Collaborative Filtering. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 1491–1494.

[15] Adit Krishnan, Ashish Sharma, and Hari Sundaram. 2018. Insights from the Long-Tail: Learning Latent Representations of Online User Behavior in the Presence of Skew and Sparsity. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 297–306.

[6]https://ara.amazon-ml.com/

[16] Brian Kulis et al. 2013. Metric learning: A survey. *Foundations and Trends® in Machine Learning* 5, 4 (2013), 287–364.

[17] Defu Lian, Cong Zhao, Xing Xie, Guangzhong Sun, Enhong Chen, and Yong Rui. 2014. GeoMF: joint geographical modeling and matrix factorization for point-of-interest recommendation. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 831–840.

[18] Dawen Liang, Laurent Charlin, James McInerney, and David M Blei. 2016. Modeling user exposure in recommendation. In *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 951–961.

[19] Dawen Liang, Laurent Charlin, James McInerney, and David M Blei. 2016. Modeling user exposure in recommendation. In *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 951–961.

[20] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 689–698.

[21] Hao Ma, Haixuan Yang, Michael R Lyu, and Irwin King. 2008. Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management*. ACM, 931–940.

[22] Benjamin M Marlin. 2004. Modeling user rating profiles for collaborative filtering. In *Advances in neural information processing systems*. 627–634.

[23] Peter V Marsden and Noah E Friedkin. 1993. Network studies of social influence. *Sociological Methods & Research* 22, 1 (1993), 127–151.

[24] Miller McPherson, Lynn Smith-Lovin, and James M Cook. 2001. Birds of a feather: Homophily in social networks. *Annual review of sociology* 27, 1 (2001), 415–444.

[25] Andriy Mnih and Ruslan R Salakhutdinov. 2008. Probabilistic matrix factorization. In *Advances in neural information processing systems*. 1257–1264.

[26] Weike Pan, Evan Wei Xiang, Nathan Nan Liu, and Qiang Yang. 2010. Transfer learning in collaborative filtering for sparsity reduction. In *Twenty-fourth AAAI conference on artificial intelligence*.

[27] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. AUAI Press, 452–461.

[28] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*. Springer, 593–607.

[29] Cosma Rohilla Shalizi and Andrew C Thomas. 2011. Homophily and contagion are generically confounded in observational social network studies. *Sociological methods & research* 40, 2 (2011), 211–239.

[30] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Latent relational metric learning via memory-based attention for collaborative ranking. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 729–739.

[31] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[32] Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. 2017. Irgan: A minimax game for unifying generative and discriminative information retrieval models. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 515–524.

[33] Menghan Wang, Xiaolin Zheng, Yang Yang, and Kun Zhang. 2018. Collaborative filtering with social exposure: A modular approach to social recommendation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[34] Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8, 3-4 (1992), 229–256.

[35] Le Wu, Peijie Sun, Richang Hong, Yanjie Fu, Xiting Wang, and Meng Wang. 2018. SocialGCN: An Efficient Graph Convolutional Network based Model for Social Recommendation. *arXiv preprint arXiv:1811.02815* (2018).

[36] Qitian Wu, Hengrui Zhang, Xiaofeng Gao, Peng He, Paul Weng, Han Gao, and Guihai Chen. 2019. Dual Graph Attention Networks for Deep Latent Representation of Multifaceted Social Effects in Recommender Systems. *arXiv preprint arXiv:1903.10433* (2019).

[37] Qitian Wu, Hengrui Zhang, Xiaofeng Gao, Peng He, Paul Weng, Han Gao, and Guihai Chen. 2019. Dual Graph Attention Networks for Deep Latent Representation of Multifaceted Social Effects in Recommender Systems. *arXiv preprint arXiv:1903.10433* (2019).

[38] Yao Wu, Christopher DuBois, Alice X Zheng, and Martin Ester. 2016. Collaborative denoising auto-encoders for top-n recommender systems. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. ACM, 153–162.

[39] Tong Zhao, Julian McAuley, and Irwin King. 2014. 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*. ACM, 261–270.