

Latent Unexpected Recommendations

PAN LI and ALEXANDER TUZHILIN, New York University, Stern School of Business

Unexpected recommender system constitutes an important tool to tackle the problem of filter bubbles and user boredom, which aims at providing unexpected and satisfying recommendations to target users at the same time. Previous unexpected recommendation methods only focus on the straightforward relations between current recommendations and user expectations by modeling unexpectedness in the feature space, thus resulting in the loss of accuracy measures to improve unexpectedness performance. In contrast to these prior models, we propose to model unexpectedness in the latent space of user and item embeddings, which allows us to capture hidden and complex relations between new recommendations and historic purchases. In addition, we develop a novel Latent Closure (LC) method to construct a hybrid utility function and provide unexpected recommendations based on the proposed model. Extensive experiments on three real-world datasets illustrate superiority of our proposed approach over the state-of-the-art unexpected recommendation models, which leads to significant increase in unexpectedness measure without sacrificing any accuracy metric under all experimental settings in this article.

CCS Concepts: • **Information systems → Recommender systems;**

Additional Key Words and Phrases: Unexpected recommendation, beyond-accuracy objectives, latent closure, latent embeddings, latent space

ACM Reference format:

Pan Li and Alexander Tuzhilin. 2020. Latent Unexpected Recommendations. *ACM Trans. Intell. Syst. Technol.* 11, 6, Article 70 (September 2020), 25 pages.

<https://doi.org/10.1145/3404855>

1 INTRODUCTION

Recommender systems have been playing an important role in the process of information dissemination and online commerce, which assist users in filtering the best content while shaping their consumption behavior patterns at the same time. However, classical recommender systems are facing the problem of filter bubbles [43, 46], which means that target users only get recommendations of their most familiar items, while losing reach to many other available items. They also lead to the problem of user boredom [28, 29], which significantly deteriorates user satisfaction with recommender systems. For example, even a Harry Potter fan may feel unsatisfied if the system keeps recommending the Harry Potter series all the time.

Authors' addresses: P. Li, New York University, Stern School of Business, Henry Kaufman Management Center, Room 8-185B, 44 West 4th Street, New York, NY, 10012; email: pli2@stern.nyu.edu; A. Tuzhilin, New York University, Stern School of Business, Henry Kaufman Management Center, Room 8-85, 44 West 4th Street, New York, NY, 10012; email: atuzhili@stern.nyu.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2020 Association for Computing Machinery.

2157-6904/2020/09-ART70 \$15.00

<https://doi.org/10.1145/3404855>

To address these two problems, researchers have introduced recommendation objectives beyond accuracy, including unexpectedness, serendipity, novelty, and diversity [53], the goal of which is to provide novel, surprising, and satisfying recommendations. Among them, unexpectedness is of particular interest for its close relation with user satisfaction and ability to improve recommendation performance [1, 2]. Therefore, we focus on modeling unexpectedness and providing unexpected recommendations in this article.

In prior literature, researchers have proposed to define unexpectedness in multiple ways, including deviations from primitive prediction results [14, 42], unexpected combination of feature patterns [5], and feature distance from previous consumptions [2]. They subsequently provide unexpected recommendations based on these definitions and achieve significant performance improvements in terms of certain unexpectedness measures.

However, as shown in the prior literature [76, 78], improvements in unexpectedness come at the cost of sacrificing accuracy measures, which severely limits practical use of unexpected recommendations, since the major goal of recommender system is to enhance overall user satisfaction. This is the case for the following reasons. First, previous models only focus on the straightforward relations between current recommendation and user expectations by modeling unexpectedness in the *feature space*, while not taking into account deep, complex, and heterogeneous relations between users and items. Second, prior modeling of unexpectedness relies completely on the explicit user and item information and may not work well in the case when the consumption records are sparse, noisy, or even missing. And, finally, the distance metric between discrete items, which is crucial for defining unexpectedness, is hard to formulate in the discrete feature space, and this may lead to unintentional biases in the estimation of user preferences. Therefore, prior unexpected recommendation models can be further improved, and this constitutes the main topic of this article.

To address the aforementioned concerns, in this article we propose to define unexpectedness in the *latent space* containing latent embeddings of users and items, as opposed to the *feature space* that only has the explicit information about them. Specifically, we propose a novel *Latent Closure* (*LC*) method to model unexpectedness that:

- captures latent, complex, and heterogeneous relations between users and items to effectively model the concept of unexpectedness.
- provides unexpected recommendations without sacrificing any performance accuracy.
- efficiently computes unexpectedness for large-scale recommendation services.

The proposed unexpected recommendation model follows the following three-stage procedure. First, we map the features of users and items into the latent space and represent users and items as latent embeddings there. These embeddings are obtained using several state-of-the-art mapping approaches, including *Heterogeneous Information Network Embeddings* (HINE) [12, 54, 62], *AutoEncoder* (AE) [21, 52], and *MultiModal Embeddings* (ME) [45] methods. We subsequently utilize the concept of “closure” from differential geometry and formulate the definition of unexpectedness of a new item as the distance between the embedding of that item and the closure of all the previously consumed item embeddings. And, finally, we combine this unexpectedness measure with the estimated rating of the item to construct the hybrid utility function for providing unexpected recommendations.

In this article, we make the following contributions:

- (1) We propose *latent* modeling of unexpectedness. Although many papers have recently explored latent spaces for recommendation purposes, it is not clear how to do it for *unexpected* recommendations, which constitutes the topic of this work.

- (2) We construct hybrid utility function based on the proposed unexpectedness measure and provide unexpected recommendations accordingly. We also demonstrate that this approach would significantly outperform all other unexpected recommendation baselines considered in this article.
- (3) We conduct extensive experiments in multiple settings and show that it is indeed the latent modeling of unexpectedness that leads to significant increase in unexpectedness measures without sacrificing any accuracy performance. Thus, the proposed method helps users to break out of their filter bubbles without sacrificing recommendation performance.

The rest of the article is organized as follows. We discuss the related work in Section 2 and present our proposed latent modeling of unexpectedness in Section 3. The unexpected recommendation model is introduced in Section 4. Experimental design on three real-world datasets are described in Section 5, and the results as well as discussions are presented in Section 6. Finally, Section 7 summarizes our contributions and concludes the article.

2 RELATED WORK

In this section, we provide an overview on the related work covering three fields: beyond-accuracy metrics, unexpected recommendations, and latent embeddings for recommendations. We highlight the importance of combining unexpected recommendations with latent modeling approaches to achieve superb recommendation performance.

2.1 Beyond-Accuracy Metrics

As researchers have pointed out, accuracy is not the only important objective of recommendations [41], while other beyond-accuracy metrics should also be taken into account, including unexpectedness, serendipity, novelty, diversity, coverage, and so on [14, 27]. Note that these metrics are closely related to each other but still different in terms of definition and formulation. Therefore, prior literature have proposed multiple recommendation models to optimize each of these metrics separately.

Serendipity measures the positive emotional response of the user about a previously unknown item and indicates how surprising these recommendations are to the target users [9, 53]. Representative methods to improve serendipity performance include Serendipitous Personalized Ranking (SPR) [39], which extends traditional personalized ranking methods by considering serendipity information in the AUC optimization process, and Auralist [74], which utilizes the topic modeling approach to capture serendipity information and provide serendipitous recommendations accordingly.

Novelty measures the percentage of new recommendations that the users have not seen before or known about [41]. It is computed as the percentage of unknown items in the recommendations. Researchers have proposed multiple methods to improve novelty measure in recommendations, including clustering of long-tail items [47], innovation diffusion [25], graph-based algorithms [56], and ranking models [44, 67].

Diversity measures the variety of items in a recommendation list, which is commonly modeled as the aggregate pairwise similarity of recommended items [77]. Typically, models to improve diversity of recommendations include Determinantal Point Process (DPP) [10, 13], which proposes a novel algorithm to greatly accelerate the greedy MAP inference and provides diversified recommendations accordingly; Greedy Re-ranking methods [8, 32, 59, 64, 77], which provide diversified recommendations based on the combination of the item's relevance and its average distance to items already in the recommended list; and Latent Factor models to optimize diversity measures [22, 57, 61].

Table 1. Classification of Unexpected Recommendation Research

Model	Literature	Strength	Weakness
Rule-Based Approaches • K-Furthest Neighbor • Frequency Discount • Taxonomy-Based Similarity • Partial Similarity • Social Network • Graph Theory	[50], [11], [26], [34], [63]	• Easy to implement • Allow for model control • Improves interpretability	• Require pre-defined rules • Lack of scalability • Lack of generalizability
Model-Based Approaches • Matrix Factorization • Learning to Rank • Re-Ranking • Clustering • Graph Theory	[31], [30], [39], [56], [67]	• Robust and generalizable • Mathematical foundation • Efficient optimization	• Lack of interpretability • Restricted model control • Limited model input
Utility-Based Approaches • Weighted Sum Model • Weighted Product Model • Probabilistic Model • Neural Network Model	[69], [23], [20], [74], [2], [35]	• Balance between objectives • Allow for model control • Model agnostic	• Require hyperparameter optimization • Explicit information only

Coverage measures the degree to which recommendations cover the set of available items [3, 14, 19]. To improve coverage measure, researchers propose to use coverage optimization [3, 4] and popularity reduction methods [65] to balance between relevance and coverage objectives [70].

Over all beyond-accuracy metrics, in this article we only focus on the unexpectedness measure and aim at providing unexpected recommendations for its close relation with user satisfaction and ability to improve recommendation performance [1, 2]. Moreover, the proposed unexpected recommendation algorithm is capable of improving serendipity and diversity measures as well, as shown in our experiment results.

2.2 Unexpectedness in Recommendations

Different from other beyond-accuracy metrics, unexpectedness measures those recommendations that are not included in user expectations and depart from what they would expect from the recommender system. Researchers have shown the importance of incorporating unexpectedness in recommendations, which could overcome the overspecialization problem [2, 24], broaden user preferences [19, 74, 75], and increase user satisfaction [2, 39, 74]. Unexpectedness captures the deviation of a particular recommender system from the results obtained from other primitive prediction models [5, 14, 42] and the deviation from user expectations [2, 35]. To improve unexpectedness measure in the final recommendations, existing models can be classified into three categories: rule-based approaches, model-based approaches, and utility-based approaches, as we show in Table 1.

Rule-based approaches typically involve pre-definition of a set of rules or recommendation strategies for unexpected recommendations, including partial similarity [26], k-furthest-neighbor [50], and graph-based approaches [34, 63]. Rule-based approaches are generally simple to implement and easy to put into actual practice, as most of the approaches incorporate unexpectedness into the classical models instead of starting from scratch. Besides, rule-based approaches allow for more control in the model, as the rules and recommendation strategies are often explicitly specified by the designers. It also improves the explainability and interpretability of the proposed unexpected recommendation model. However, they require pre-defined strategies to be set prior to recommendations. Also, scalability is a big concern for the usage of rule-based methods. In

addition, these models typically lack of generalizability, for they focus only on specific domains and specific applications.

Model-based approaches aim to improve novelty and unexpectedness of the recommended items by proposing new models and data structures that go beyond the traditional collaborative filtering paradigm. Representative models that optimize the unexpectedness objective include personalized ranking [67], innovator identification [30, 31], and transition cost graph [56]. Model-based approaches are backed with mathematical foundations that guarantee either convergence or stability of the learning process, thus making them robust to different settings and with greater potential of generalizability. However, they are often hard to interpret, for there is no natural way to transfer mathematical formulations into explicit rules or recommendation strategies. Therefore, it is relatively hard to control the degree of unexpectedness that we aim to incorporate into the recommendation model. And, finally, model-based approaches might not take full advantages of all available information due to the restrictions of specific model form.

Utility-based approaches involve the construction of a hybrid utility function as the combination of estimated relevance and degree of unexpectedness. References [20, 23, 69] have followed this direction of research. Specifically, Reference [2] proposed to include user expectation into the hybrid utility function and achieves state-of-the-art unexpected recommendation performance. Utility-based methods allow for more control of the recommendation strategy, and it is easier to implement and put into practice as well. Especially, the construction of unexpectedness does not depend on the estimation of user preferences toward the candidate item, thus making it model agnostic. However, the unexpected hyperparameter plays an important role in determining the recommendation performance of the hybrid-based model, thus requiring proper hyperparameter optimization.

One important limitation of all prior unexpected recommendation models lies in that they only focus on the straightforward relations between users and items and define unexpectedness in the feature space, without taking into account the deep, complex interactions underlying their feature information. Therefore, previous unexpected recommendations might not reach the optimal recommendation performance, as discussed in References [54, 71, 72]. In addition, they are facing the tradeoff dilemma between optimizing the accuracy and unexpectedness objectives. To address these limitations, in this article we propose to define unexpectedness instead in the latent space, thus obtaining significant improvements over previous models.

2.3 Latent Embeddings for Recommendation

Another body of related work is around embedding approaches that effectively map users and items into the latent space and extract their deep, complex, and heterogeneous relations among each other. Specifically, different embedding methods fit for different recommendation applications. In the case where heterogeneous feature data are available, HINE [12, 54, 55] utilizes the data structure of a heterogeneous information network (HIN) to extract complex heterogeneous relations between user and item features and thus provide better recommendations to the target users. In the case where rich interactions between users and items are available, the AE approach [17, 21, 36, 37, 49, 52] utilizes deep neural network (DNN) techniques and obtains the semantic-aware representations of users and items as embeddings in the latent space to model their relationship and provide recommendations accordingly. Finally, in the case where a multimodal dataset is available, researchers propose to use an ME approach [45] to combine information from different sources and obtain superb recommendation performance.

Compared with classical approaches, latent embedding methods have several important advantages that enable recommender systems to provide more satisfying recommendations [38, 73], as discussed in Section 1. Therefore, in this article we provide the definition of unexpectedness

utilizing these latent embedding methods, which contributes to the strong recommendation performance.

3 LATENT MODELING OF UNEXPECTEDNESS

In this section, we introduce the proposed latent modeling of unexpectedness. We compare the new definition with feature-based definitions and illustrate superiority and benefits of the proposed approach.

3.1 Latent Space

As introduced in prior literature [2, 14, 42], an important component for modeling unexpectedness is the *expected set*, which contains previous consumptions of the user. The idea is that users should have no unexpectedness toward those recommended items that they have purchased before or very similar to their purchases, for they understand that typical recommender systems collect their historic behaviors and thus provide similar recommendations based on these records.

To construct the user expectations, Reference [2] proposes to form the expected set by taking into account explicit feature information of users and items. For example, in the book recommendation, the expected set is constructed based on the features of alternative editions, in the same series, with same subjects and classifications, with the same tags, and so on. Unexpectedness is subsequently defined by a positive, unbounded function of the distance of the recommended item from the set of expected items.

However, this definition only focuses on the straightforward relations between users and items, but falls short of addressing deeper correlations beyond the explicit feature information. For example, if a certain user has been a frequent consumer of McDonald and Carl's Jr, then the recommendation of Burger King might not be unexpected to that user, although these restaurants belong to different franchise and offer different menus, as shown in their feature information.

Besides, feature-based modeling of unexpectedness typically assumes the same importance for each feature during the calculation of unexpectedness, while in reality it is not necessarily the case. A natural example is that for music recommendations, genre information plays a more important role in determining the degree of unexpectedness than profile information, such as time of release. In addition, the distance function is also hard to define in the discrete feature space.

Therefore, in this article, we propose to construct the expected set in the latent space by taking the closure of item embeddings. Unexpectedness is subsequently defined as the distance between the new item embedding and the closure of the expected set in the latent space. Comparing with feature-based definitions, latent modeling of unexpectedness obtains several important advantages, as discussed in Section 1. Especially, we point out that the proposed LC model is capable of utilizing richer information of user reviews and multi-modal data to determine the degree of unexpectedness, as previous models typically do not take this information into account, as shown in Table 2. These benefits are also supported by strong experiment results.

In the next section, we will introduce the idea of latent closure and how to construct user expectations based on the proposed latent closure method.

3.2 Latent Closure

As discussed in the previous section, we propose to compute user expectations in the latent space rather than in the original feature space. In addition, we point out that the modeling of user expectations should go beyond the direct aggregation of previous consumptions and should also take into account those items that are similar to the consumed items, while similarities between items are captured by the Euclidean distance in the latent space. Therefore, it is natural to take the

Table 2. Comparison of Unexpected Recommendation Methods

Algorithms	LC	Feature Modeling			
		SPR	Auralist	HOM-LIN	DPP
Latent Embeddings	✓	✗	✗	✗	✗
Explicit Features	✓	✗	✗	✓	✗
User Reviews	✓	✗	✓	✗	✗
Pre-Defined Rules	✓	✗	✓	✓	✗
Past Transactions	✓	✓	✓	✓	✓
User Ratings	✓	✓	✓	✓	✓

“closure” of all consumed item embeddings to model the expected set, as opposed to using individual item embedding in the latent space.

According to mathematical theories in differential geometry [18], there are three common geometric structures in high-dimensional latent spaces that can be naturally extended to modeling the closure of latent embeddings, namely Hypersphere, Hypercube, and Convex Hull. The particular choice of latent closure depends on the assumption we make toward the relations between users and items in the latent space.

- **Latent HyperSphere (LHS).** The hypersphere in the R^n space is defined as the set of n -tuples points (x_1, x_2, \dots, x_n) such that $x_1^2 + x_2^2 + \dots + x_n^2 = r^2$, where r is the radius of the hypersphere. Under this definition, we assume that the expected set of items for each user grows homogeneously in all directions in the latent space.
- **Latent HyperCube (LHC).** The hypercube is a closed, compact, convex figure, whose 1-skeleton consists of groups of opposite parallel line segments aligned in each of the space’s dimensions, perpendicular to each other and of the same length. Under this definition, we assume that the expected set of each user grows homogeneously in the n perpendicular directions.
- **Latent Convex Hull (LCH).** The convex hull of a set of points X in the Euclidean space is the smallest convex set that contains all points in X . Under this definition, we assume that the expected set maintains its convexity in the growing process. In addition, if we construct the expected set as the convex hull of consumed item embeddings, then the convexity property will guarantee the feasibility of the recommendation as an optimization problem given by the Slater’s Condition [58].

We visualize the definition of unexpectedness based on these geometric structures in Figure 1 (a), (b), and (c). These latent closure approaches capture latent semantic interactions between users and items and construct the expected set for each user accordingly. Compared to feature-based definitions [2], latent closures utilize richer information including user and item features to model user expectations more precisely. The process for finding closures in high-dimensional latent spaces is not significantly different from the process in the two-dimensional space. For LHS and LHC, we only need to find the furthest two points in the latent space to identify the centroid of the latent closure. For LCH, we follow the QuickHull algorithm [7] to identify the latent structure. Experiment results show that all three geometric structures consistently obtain significant improvements over baseline models, while no structure dominates the other two.

To summarize, in this article we utilize the latent closure method to model unexpectedness in the latent space. We hereby propose the following definition of unexpectedness:

Definition 3.1. Unexpectedness of a new item as the distance between the embedding of that item and the closure of all previously consumed item embeddings.

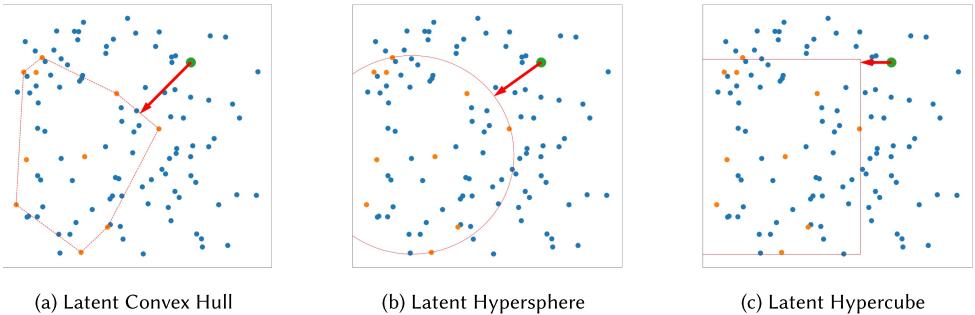


Fig. 1. Visualization of Latent Closure and the Unexpectedness. Blue points denote all the available items; orange points represent the consumed items; green points refer to the newly recommended item. We define unexpectedness as the distance between the new item and the latent closure generated by all consumed items.

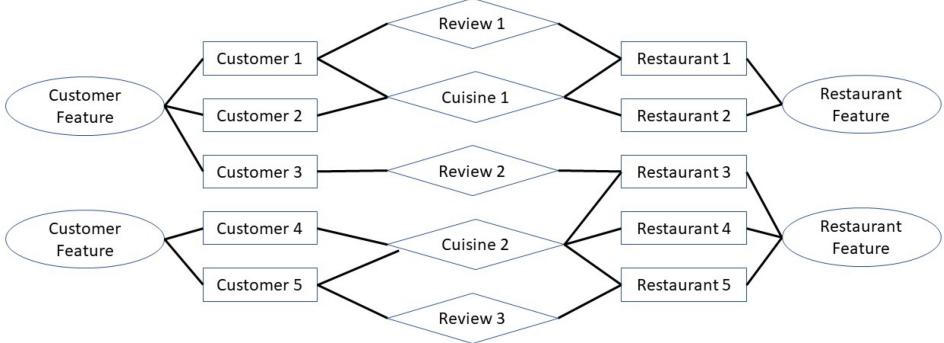


Fig. 2. Heterogeneous Information Network Embedding Method.

In the next section, we will discuss the specific techniques for obtaining latent embeddings and methods to provide unexpected recommendations accordingly.

4 UNEXPECTED RECOMMENDATION MODEL

4.1 Latent Embeddings

To effectively model unexpectedness in the latent space and demonstrate the robustness of the proposed model, we utilize three state-of-the-art latent embedding approaches, namely HINE, AE, and ME, to map users and items into the latent space and calculate the unexpectedness subsequently. The mapping procedures for these three approaches have been visualized in Figure 2, Figure 3 and Figure 4 respectively.

4.1.1 Heterogeneous Information Network Embeddings (HINE). To capture the complex and multi-dimensional relations in the data record, Heterogeneous Information Network (HIN) [62] has become an effective data structure for recommendations, which models multiple types of objects and multiple types of links in one single network. It includes users, items, transactions, ratings, entities extracted from reviews, and the feature information. We link the associated entities with corresponding users and items in the network and utilize meta-path embedding approach [12] to obtain node embeddings.

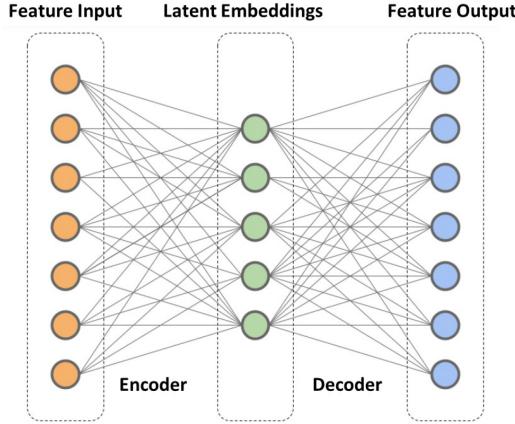


Fig. 3. AutoEncoder Embedding Method.

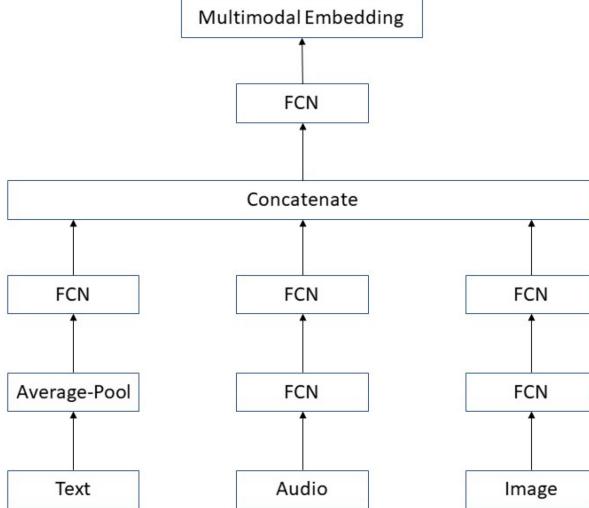


Fig. 4. Multimodal Embedding Method.

We denote the heterogeneous network as $G = (V, E, T)$, in which each node v and each link e are assigned with specific type T_v and T_e . To effectively learn node representations, we enable the skip-gram mechanism to maximize the probability of each context node c_t within the neighbors of v , denoted as $N_t(v)$, where we add the subscript t ($t \in T_v$) to limit the node to a specific type:

$$\arg \max_{\theta} \sum_{v \in V} \sum_{t \in T_v} \sum_{c_t \in N_t(v)} \log P(c_t | v; \theta). \quad (1)$$

Thus, it is important to calculate $P(c_t | v; \theta)$, which represents the conditional probability of context node c_t given node v . Therefore, we follow Reference [16] and revise the network embedding model accordingly for dealing with heterogeneous information network. Specifically, we propose to use heterogeneous random walk to generate paths of multiple types of nodes in the network. Given a heterogeneous information network $G = (V, E, T)$, the metapath of the network is generated in the form of $V_1 \xrightarrow{R_1} V_2 \xrightarrow{R_2} V_3 \dots V_n$, wherein $R = R_1 \circ R_2 \circ \dots \circ R_n$ defines

the composite relations between the start and the end of the heterogeneous random walk. The transition probability within each random walk between two nodes is defined as follows:

$$p(V_{t+1}|V_t) = \begin{cases} \frac{C(T_{V_t}, T_{V_{t+1}})}{|N_{t+1}(V_t)|}, & (V_t, V_{t+1}) \in E \\ 0, & (V_t, V_{t+1}) \notin E \end{cases}, \quad (2)$$

where $C(T_{V_t}, T_{V_{t+1}})$ stands for the transition coefficient between the type of node V_t and the type of node V_{t+1} . We have six different transition coefficients that correspond to 6 different relations in the network $C_{UU}, C_{UE}, C_{UI}, C_{EI}, C_{EE}$, and C_{II} (U: User, I: Item, E: Entity/Feature). $|N_{t+1}(V_t)|$ stands for the number of nodes of type V_{t+1} in the neighborhood of V_t . We apply heterogeneous random walk iteratively to each node and generate the collection of meta-path sequences. The user and item embeddings are therefore obtained through the aforementioned skip-gram mechanism.

4.1.2 AutoEncoder (AE). Apart from modeling interactions between users and items through HIN, the AE approach also constitutes an important tool to learn the latent representations of user and item features and transform discrete feature vectors into continuous feature embeddings.

We denote the feature information for user a as $u_a = \{u_{a_1}, u_{a_2}, \dots, u_{a_m}\}$ and the feature information for item b as $i_b = \{i_{b_1}, i_{b_2}, \dots, i_{b_n}\}$, where m and n stand for the dimensionality of user and item feature vectors, respectively. The goal is to train two separate neural networks: encoder, which maps feature vectors into latent embeddings, and decoder, which reconstructs feature vectors from latent embeddings. Due to effectiveness and efficiency of the training process, we formulate both the encoder and the decoder as multi-layer perceptron (MLP). MLP learns the hidden representations using the following equations:

$$y_a = \Phi(u_a), y_b = \Phi(i_b), \quad (3)$$

where y_a, y_b represents the latent embeddings and Φ denotes the fully connected layer with activation functions. We apply another layer of fully connected network for reconstruction and optimization. Note that in this step we train the global autoencoder for users and items in the entire dataset simultaneously to obtain the hidden representation.

4.1.3 Multimodal Embeddings (ME). In addition to the aforementioned approaches, when dealing with datasets that include multiple modalities, such as movie and video data (which are usually associated with images and subtitles), multimodal embeddings [45, 68] constitute an efficient tool to combine the information from different sources.

Specifically, in the video recommendation task, we illustrate the model for obtaining video embeddings in Figure 4. First, we initialize the embeddings for text, audio, and image data through Fully Convolutional Network (FCN) with L2-Norm as a regularization term. For the text data, we use the average pooling technique as a special treatment to obtain the semantic information as the average of word embeddings. Then we concatenate these embeddings and apply another layer of Fully Convolutional Network to obtain multimodal embeddings for the input video that captures joint information of subtitles, sound, and graphics.

4.2 Hybrid Utility Function

Based on the latent embedding approaches introduced in the previous section, we map the users and items into the continuous latent space and model the expected set for each user as the latent closure of item embeddings. Specifically, we feed the user and item features as input into the latent embedding models and obtain their latent representations. We subsequently formulate the unexpectedness as the distance between the embedding of new item and the latent expected set as

$$U_{u,i} = d(i; LC(N_i)), \quad (4)$$

where $N_i = (i_1, i_2, \dots, i_n)$ contains the embeddings of all consumed items. This unexpectedness metric is well defined as the minimal distance from the new item to the boundaries of the closure in the latent space. We then perform the unexpected recommendation based on the hybrid utility function:

$$Utility_{u,i} = EstRating_{u,i} + \alpha * Unexp_{u,i}, \quad (5)$$

which incorporates the linear combination of estimated ratings and unexpectedness. The key idea lies in that, instead of recommending the similar items that the users are very familiar with as the classical recommenders do, we recommend unexpected and useful items to the users that they might have not thought about but indeed fit well to their satisfactions. The two adversarial forces of accuracy and unexpectedness work together to get the optimal recommendation and thus obtain the best recommendation performance and user satisfaction. We present the entire framework in Algorithm 1.

ALGORITHM 1: Latent Unexpected Recommendation

Data: Users; Items; Historic Actions; Other feature information

Result: List of Recommended Items

```

1 Map users and items into the latent space;
2 for each user  $u$  in Users do
3   for each item  $i$  in Items do
4      $Unexp(u, i) = d(i; LC(N_i));$ 
5      $Utility_{u,i} = EstRating_{u,i} + \alpha * Unexp_{u,i}$ 
6   end
7   Recommend Top-N(Utility);
8 end
```

5 EXPERIMENTS

To validate the performance of our approach, we conduct extensive experiments on three large-scale real-world applications and compare the results of our model with the state-of-the-art baselines. The experimental setup is introduced in this section. Specifically, we design the experiments to address the following research questions:

- RQ1:** *How does the proposed model perform compared to baseline unexpected recommendation models?*
- RQ2:** *Can we achieve significant improvements in unexpectedness measure while keeping the same level of accuracy performance?*
- RQ3:** *Are the improvements robust to different experimental settings?*

5.1 Datasets

We implement our model on three real-world datasets: the Yelp Challenge Dataset Round 12,¹ which contains ratings and reviews of users and restaurants; the TripAdvisor Dataset,² which contains check-in information of users and hotels; and the Video Dataset, which includes the traffic logs we collected from a large-scale industrial video platform. Specifically, we use four days of traffic logs for the training process and the following day for the evaluation process. We list the descriptive statistics of these datasets in Table 3. To avoid the cold-start and sparsity issues, we filter out users and items that appear less than 5 times in all three datasets.

¹<https://www.yelp.com/dataset/challenge>.

²<http://www.cs.cmu.edu/~jiweil/html/hotel-review.html>.

Table 3. Descriptive Statistics of Three Datasets

Dataset	Yelp	TripAdvisor	Video
# of Records	5,996,996	878,561	1,155,987
# of Items	188,593	576,689	287,607
# of Users	1,518,169	3,945	5,241
Sparsity	0.002%	0.039%	0.077%

5.2 Parameter Settings

We perform Bayesian optimization [60] to select optimal hyperparameters for the proposed method as well as baseline models. The α is selected as 0.03, where we achieve the optimal balance between the accuracy and unexpectedness measures. In addition, the dimension of the latent embeddings is 128, which is efficient to capture the relations between users and items, as shown in Reference [73]. Detailed parameter settings are further introduced in the next section.

As discussed in Section 4, for three different datasets we select three state-of-the-art embedding approaches accordingly to model the unexpectedness in the latent space. Specifically, the Yelp dataset contains information about explicit users, items, and ratings, as well as substantial amounts of meta-information, including text reviews, friendship network, user demographics, and geolocation. Thus, it is suitable to be analyzed using a HINE approach to address the heterogeneous relationships within the Yelp dataset. Meanwhile, due to the multimodality of video data structure, we utilize the ME approach to calculate the unexpectedness between users and videos in the Video dataset. Meanwhile, the TripAdvisor dataset only includes users, items, and their associated feature information, which makes the AE approach a reasonable choice for obtaining latent embeddings.

We point out that, although it could further increase the validity of our approach if we test the same embedding approach on the three datasets, it is not practical to do so. By implementing our model through three different embedding approaches, we illustrate the strength of modeling unexpectedness in the latent space. Note that illustration of this point does not rely on the specific design of embedding approaches.

5.3 Training Procedure

Our proposed latent unexpected recommendation model follows a three-step training procedure: First, we utilize the latent embedding approaches to map users and items into the latent space; then we subsequently calculate the unexpectedness and construct the hybrid utility function for each user; finally, we provide unexpected recommendations based on the hybrid utility function and update our model accordingly.

To obtain the heterogeneous information network embeddings from the Yelp dataset, we extract the users, restaurants, and feature labels from the dataset to construct the nodes in the heterogeneous information network. We link the user nodes and items nodes with their associate feature nodes, and we also link the user node with the item node if the user has visited that restaurant before. We conduct heterogeneous random walk [54] with length 100 starting from each node to generate the sequences of nodes. We repeat this process 10 times. Then we enable skip-gram mechanism following the procedures in Reference [16] with window size 2, minimal term count 1, and iterations 100 to map the nodes into the latent space and obtain the corresponding latent embeddings.

To obtain the autoencoder embeddings from the TripAdvisor dataset, we utilize one layer of MLP as the encoder to generate latent representations for each user and item, and then use one layer of MLP as decoder to reconstruct the original information. We jointly optimize encoder and decoder to generate the latent embeddings.

To obtain the multimodal embeddings from the Video dataset, we decompose the input videos into texts, audios, and images, where we subsequently apply FCN with L2-Norm as regularization term to obtain the latent embeddings separately. Then we concatenate text embeddings, audio embeddings, and image embeddings to go through another layer of FCN to generate the final multimodal embeddings.

For performance comparison, we select the deep-learning based Neural Collaborative Filtering (NCF) model [17] as well as five popular collaborative filtering algorithms, including the k-Nearest Neighborhood approach (KNN) [6], Singular Value Decomposition approach (SVD) [51], Co-Clustering approach [15], Non-Negative Matrix Factorization approach (NMF) [33], and Factorization Machine approach (FM) [48] to verify robustness of the proposed model. We implement the model in the Python environment using the “Surprise,” “SciPy,” and “Gensim” packages. All experiments are performed on a laptop with 2.50-GHz Intel Core i7 and 8-GB RAM. We show that the training procedure is time efficient: It takes 3 hours, 0.5 hours, and 1 hours, respectively, for our proposed model to obtain latent embeddings in the Yelp dataset, the TripAdvisor dataset, and the Video dataset. The subsequent unexpected recommendation process takes less than 1 hour to complete.

5.4 Evaluation Metrics: Accuracy and Unexpectedness

To compare the performance of the proposed LC method and baseline models, we measure the recommendation results along two dimensions: *accuracy*, in terms of RMSE, MAE, Precision@N, and Recall@N metrics [19], and *unexpectedness*, in terms of Unexpectedness, Serendipity, and Diversity metrics [14]. Specifically, we calculate unexpectedness through Equation (5) following our proposed definition, while serendipity and diversity are computed following the standard measures in the literature [14, 77],

$$\text{Serendipity} = \frac{\text{RS\&PM\&USEFUL}}{\text{RS}}. \quad (6)$$

Serendipity is computed as the percentage of serendipitous recommendations, where *RS* stands for the recommended items using the target model, *PM* stands for the recommendation items using a primitive prediction algorithm (usually selected as the linear regression), and *USEFUL* stands for the items whose utility is above the average level. Diversity is computed as the average intra-list distance,

$$\text{Diversity} = \sum_{i \in \text{RS}} \sum_{j \neq i \in \text{RS}} \text{sim}(i, j). \quad (7)$$

5.5 Baseline Models

We implement several state-of-the-art unexpected recommendation models as baselines and report their performance in terms of aforementioned metrics. The baseline models include SPR, Auralist, DPP, and HOM-LIN. Note that we do not include the neural network approach, because there is no deep-learning based model for unexpected recommendations in the literature.

- **SPR** [39]. Serendipitous Personalized Ranking is a simple and effective method for serendipitous item recommendation that extends traditional personalized ranking methods by considering item popularity in AUC optimization, which makes the ranking sensitive to the popularity of negative examples.
- **Auralist** [74]. Auralist is a personalized recommendation system that balances among the desired goals of accuracy, diversity, novelty, and serendipity simultaneously. Specifically, in the music recommendation, the authors combine the Artist-based LDA recommendation

with two novel components: Listener Diversity and Musical Bubbles. We adjust the algorithm accordingly to fit in our restaurant and hotel recommendation scenario.

- **DPP [10].** The determinantal point process (DPP) is an elegant probabilistic model of repulsion with applications in various machine learning tasks. The authors propose a fast greedy MAP inference approach for DPP to generate relevant and diverse recommendations.
- **HOM-LIN [2].** HOM-LIN is the state-of-the-art unexpected recommendation algorithm, where the author propose to define unexpectedness as the distance between items and the expected set of users in the feature space and linearly combine unexpectedness with estimated ratings to provide recommendations.

5.6 Significant Testing

To illustrate the differences of recommendation performance between our proposed model and the baseline methods, we conduct significant testing over the experiment results. Specifically, the significance level is determined through rerunning the unexpected recommendation models with random initialization multiple times and conduct Student's *t*-test to compute the p-value. We report the significance level together with our results in the next section.

5.7 Cold-Start Problem

Note that the cold start problem is very important in recommender systems. We point out that our proposed unexpected recommender system does not encounter this problem due to the following reasons: First, for the user-side cold start problem, we do not provide unexpected recommendations, as the new users have very few interactions and normally do not face the problem of boredom. Instead, we suggest to provide classical recommendations, which aim to produce similar recommendations to help the users identify and reinforce their interested contents. Second, for the item-side cold start problem, the new item embeddings could be obtained through classical cold start embedding methods [66], and then we could subsequently calculate the unexpectedness and provide unexpected recommendations accordingly.

6 RESULTS

In this section, we report the experimental results on three real-world datasets to answer the research questions in Section 5.

6.1 Unexpected Recommendation Performance

To start with, we compare the recommendation performance of proposed latent unexpectedness with baseline unexpected recommendation models. Specifically, the proposed LC method provides unexpected recommendations through $Utility_{u,i} = EstRating_{u,i} + \alpha * Unexp_{u,i}$, where unexpectedness is calculate using Latent HyperSphere introduced in Section 3.2 and estimated ratings are computed through deep-learning based method NCF and five other popular collaborative filtering algorithms: FM, CoClustering (CC), Singular Value Decomposition (SVD), NMF, and KNN. We denote the corresponding unexpected recommendations provided through hybrid utility functions as NCF+LC, FM+LC, CC+LC, SVD+LC, NMF+LC, and KNN+LC accordingly.

As shown in Tables 4, 5, and 6, by utilizing the proposed latent modeling of unexpectedness, all six unexpected recommendation models consistently and significantly outperform the baseline methods in both accuracy and unexpectedness measures. Specifically, we observe an average increase of 5.21% in RMSE, 8.11% in MAE, 1.14% in Precision, 1.57% in Recall, 48.77% in Unexpectedness, 8.30% in Serendipity, and 27.69% in Diversity compared to the second best baseline model in the Yelp dataset. That is, the proposed latent modeling of unexpectedness enables us to provide more unexpected and more useful recommendations at the same time. Also, we show that

Table 4. Comparison of Unexpected Recommendation Performance in the Yelp Dataset

Yelp Dataset							
Model	RMSE	MAE	Pre@5	Rec@5	Unexp	Ser	Div
NCF+LC	0.9169	0.7078	0.7783*	0.6291*	0.1450	0.4905*	0.4178*
FM+LC	0.9180	0.6888*	0.7704	0.6278	0.1378	0.4603	0.4164
CC+LC	0.9514	0.7007	0.7626	0.5926	0.1355	0.4793	0.3961
SVD+LC	0.9136	0.7039	0.7722	0.6212	0.1214	0.4630	0.3511
NMF+LC	0.9522	0.7026	0.7781	0.6238	0.1466*	0.4894	0.4045
KNN+LC	0.9133*	0.7715	0.7674	0.6287	0.1288	0.4380	0.3388
SPR	1.0351	0.7729	0.7692	0.6188	0.0668	0.3720	0.2532
Auralist	1.0377	0.7799	0.7678	0.6000	0.0663	0.3637	0.2047
HOM-LIN	0.9609	0.7447	0.7621	0.6150	0.0751	0.4329	0.3011
DPP	1.0288	0.7702	0.7598	0.6012	0.0670	0.4488	0.2488

The asterisk denotes 95% statistical significance.

Table 5. Comparison of Unexpected Recommendation Performance in the TripAdvisor Dataset

TripAdvisor Dataset							
Model	RMSE	MAE	Pre@5	Rec@5	Unexp	Ser	Div
NCF+LC	0.9624*	0.7310*	0.7201*	0.9810*	0.0586	0.4635	0.0472
FM+LC	1.0230	0.7450	0.7031	0.9638	0.0581	0.4637*	0.0388
CC+LC	1.0230	0.7539	0.6887	0.9754	0.0587	0.4629	0.0491*
SVD+LC	0.9908	0.7519	0.7093	0.9569	0.0585	0.4614	0.0477
NMF+LC	1.0280	0.7594	0.6864	0.9735	0.0584	0.4629	0.0488
KNN+LC	0.9981	0.7493	0.6909	0.9743	0.0588*	0.4625	0.0488
SPR	1.0328	0.8008	0.6395	0.9325	0.0474	0.3593	0.0375
Auralist	1.0318	0.7997	0.6460	0.9390	0.0473	0.3462	0.0355
HOM-LIN	1.0298	0.7902	0.6420	0.9418	0.0572	0.3729	0.0411
DPP	1.0304	0.8158	0.6264	0.9303	0.0464	0.3245	0.0311

The Asterisk Denotes 95% Statistical Significance.

Table 6. Comparison of Unexpected Recommendation Performance in the Video Dataset

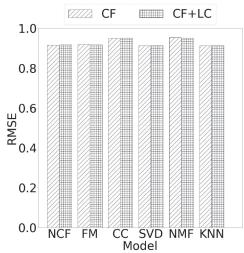
Video Dataset							
Model	RMSE	MAE	Pre@5	Rec@5	Unexp	Ser	Div
NCF+LC	0.3810*	0.2854*	0.2560	0.3615	0.7070	0.9830	0.2538
FM+LC	0.3924	0.3044	0.2498	0.3265	0.7096*	0.9833*	0.2510
CC+LC	0.4167	0.3296	0.2569	0.3676*	0.7053	0.9815	0.2519
SVD+LC	0.3888	0.2862	0.2455	0.3253	0.7018	0.9810	0.2412
NMF+LC	0.4405	0.3330	0.2494	0.3439	0.6999	0.9792	0.2450
KNN+LC	0.4088	0.3091	0.2608*	0.3212	0.9814	0.9801	0.2558*
SPR	0.4610	0.3638	0.2298	0.2870	0.6300	0.9593	0.2137
Auralist	0.4515	0.3610	0.2304	0.2890	0.6462	0.9462	0.1980
HOM-LIN	0.4498	0.3608	0.2310	0.2912	0.6732	0.9473	0.2154
DPP	0.4770	0.3670	0.2271	0.2870	0.6593	0.9328	0.2154

The Asterisk Denotes 95% Statistical Significance.

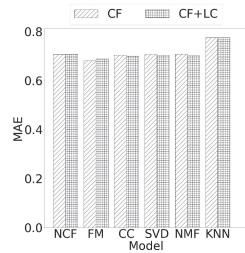
Table 7. Comparison of Recommendation Performance with and without Unexpectedness in the Yelp Dataset

Model	RMSE	MAE	Pre@5	Rec@5	Unexp	Ser	Div
NCF	0.9154	0.7070	0.7761	0.6318	0.0492	0.1666	0.3492
NCF+LC	0.9169	0.7078	0.7783	0.6291	0.1450*	0.4905*	0.4178*
FM	0.9197	0.6815	0.7699	0.6223	0.0326	0.0978	0.0135
FM+LC	0.9180	0.6888	0.7704	0.6278	0.1378*	0.4603*	0.4164*
CC	0.9499	0.7040	0.7655	0.5913	0.0338	0.1595	0.3106
CC+LC	0.9514	0.7007	0.7626	0.5926	0.1355*	0.4793*	0.3961*
SVD	0.9132	0.7071	0.7792	0.6244	0.0457	0.1352	0.0479
SVD+LC	0.9136	0.7039	0.7722	0.6212	0.1214*	0.4630*	0.3511*
NMF	0.9533	0.7081	0.7797	0.6318	0.0333	0.1954	0.3268
NMF+LC	0.9522	0.7026	0.7781	0.6238	0.1466*	0.4894*	0.4045*
KNN	0.9123	0.7748	0.7687	0.6285	0.0448	0.0977	0.0129
KNN+LC	0.9133	0.7715	0.7674	0.6287	0.1288*	0.4380*	0.3388*

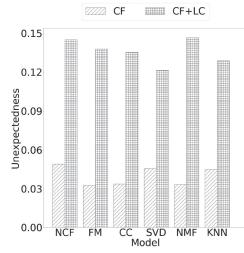
(a) RMSE



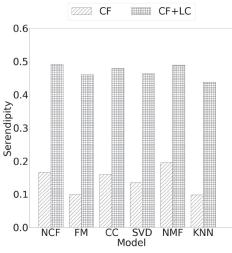
(b) MAE



(c) Unexpectedness



(d) Serendipity



The asterisk denotes 95% statistical significance; we observe significant improvements in unexpectedness measures in (c) and (d), while no significant change in accuracy measures in (a) and (b) at the same time.

the superiority of latent unexpectedness is robust to the specific selection of collaborative filtering algorithms, as we obtain significant increase of performance measures in all six algorithms and do not observe any significant difference in unexpectedness metric within these methods.

6.2 Improving Unexpectedness while Keeping Accuracy

As we discuss in the previous section, an important problem with incorporating unexpectedness into recommendations is the tradeoff between accuracy and novelty measures [76, 78], which is crucial to the practical use of unexpected recommendations. In this section, we compare the unexpected recommendation performance using hybrid utility functions with those classical recommender systems that provide recommendations based on estimated ratings only.

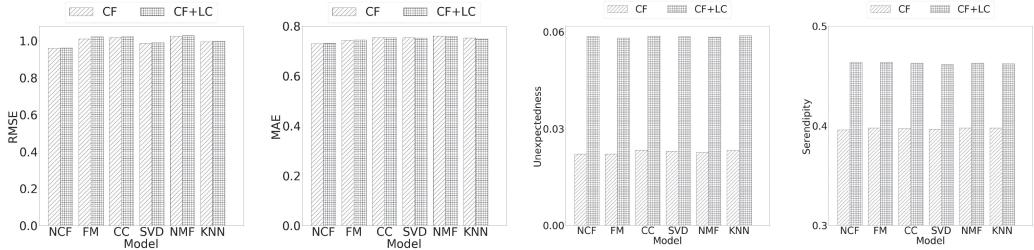
As shown in Tables 7, 8, and 9 and the corresponding plots, when including unexpectedness in the recommendation process, we consistently obtain significant improvements in terms of unexpectedness, serendipity, and diversity measures, while we do not witness any loss in the accuracy measures. Therefore, we show that it is indeed the proposed latent closure approach that enables us to provide useful and unexpected recommendations simultaneously. It is crucial for the successful deployment of unexpected recommendation models in the industrial applications.

In addition, we study the impact of the hyperparameter α in Equation (5) that controls for the degree of unexpectedness and usefulness in the hybrid utility function. Typically, a higher value of α indicates that the recommendation model is in favor of unexpected recommendations over useful

Table 8. Comparison of Recommendation Performance with and without Unexpectedness in the TripAdvisor Dataset

Model	RMSE	MAE	Pre@5	Rec@5	Unexp	Ser	Div
NCF	0.9588	0.7291	0.7230	0.9815	0.0222	0.3960	0.0010
NCF+LC	0.9624	0.7310	0.7201	0.9810	0.0586*	0.4635*	0.0472*
FM	1.0105	0.7440	0.7068	0.9590	0.0222	0.3979	0.0017
FM+LC	1.0230	0.7450	0.7031	0.9638	0.0581*	0.4637*	0.0388*
CC	1.0178	0.7543	0.6845	0.9732	0.0234	0.3973	0.0015
CC+LC	1.0230	0.7539	0.6887	0.9754	0.0587*	0.4629*	0.0491*
SVD	0.9868	0.7533	0.7010	0.9565	0.0231	0.3967	0.0006
SVD+LC	0.9908	0.7519	0.7093	0.9569	0.0585*	0.4614*	0.0477*
NMF	1.0241	0.7609	0.6850	0.9681	0.0227	0.3979	0.0010
NMF+LC	1.0280	0.7594	0.6864	0.9735	0.0584*	0.4629*	0.0488*
KNN	0.9940	0.7531	0.6969	0.9689	0.0233	0.3979	0.0019
KNN+LC	0.9981	0.7493	0.6909	0.9743	0.0588*	0.4625*	0.0488*

(a) RMSE (b) MAE (c) Unexpectedness (d) Serendipity



The asterisk denotes 95% statistical significance; we observe significant improvements in unexpectedness measures in (c) and (d), while no significant change in accuracy measures in (a) and (b) at the same time.

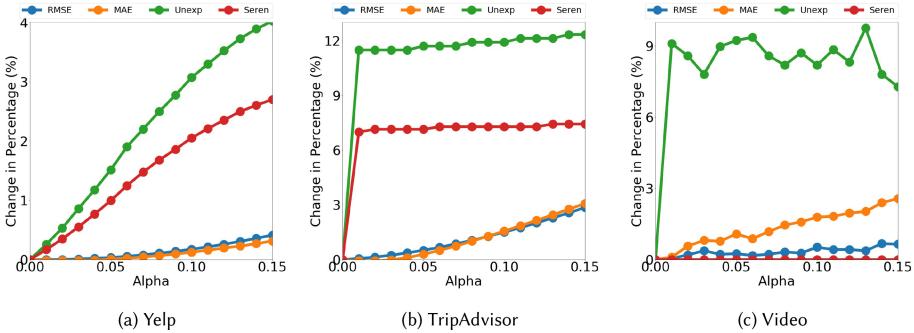


Fig. 5. Comparison of accuracy–novelty tradeoff.

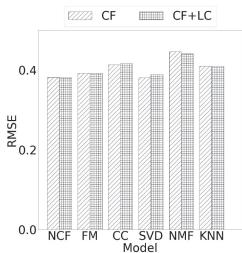
recommendations, while a lower value of α tends to recommend more useful items as opposed to unexpected items.

We plot the change of accuracy and novelty measures with respect to different α value in Figure 5. This figure illustrates that when we select relatively small value of α (e.g., $\alpha = 0.03$), we can obtain significant amounts of increase in unexpectedness (8.40%, 10.81%, and 9.10% respectively in three datasets) while the decrease of accuracy performance is not statistically significant for all three datasets. It is also worth noting that if we select a large value of α , then we might risk deteriorating the accuracy performance of recommendations significantly.

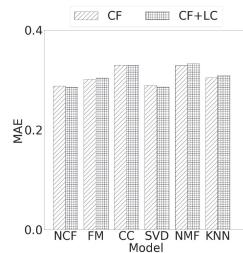
Table 9. Comparison of Recommendation Performance with and without Unexpectedness in the Video Dataset

Model	RMSE	MAE	Pre@5	Rec@5	Unexp	Ser	Div
NCF	0.3815	0.2877	0.2544	0.3630	0.6402	0.8678	0.2317
NCF+LC	0.3810	0.2854	0.2560	0.3615	0.7070*	0.9830*	0.2538*
FM	0.3920	0.3013	0.2472	0.3280	0.6398	0.8552	0.2396
FM+LC	0.3924	0.3044	0.2498	0.3265	0.7096*	0.9833*	0.2510*
CC	0.4129	0.3302	0.2560	0.3646	0.6479	0.8382	0.2362
CC+LC	0.4167	0.3296	0.2569	0.3676	0.7053*	0.9815*	0.2519*
SVD	0.3806	0.2895	0.2392	0.3232	0.6495	0.8480	0.2346
SVD+LC	0.3888	0.2862	0.2455	0.3253	0.7018*	0.9810*	0.2412*
NMF	0.4462	0.3285	0.2480	0.3391	0.6548	0.8655	0.2385
NMF+LC	0.4405	0.3330	0.2494	0.3439	0.6999	0.9792*	0.2450*
KNN	0.4103	0.3048	0.2531	0.3173	0.6416	0.8632	0.2385
KNN+LC	0.4088	0.3091	0.2608	0.3212	0.7014*	0.9814*	0.2558*

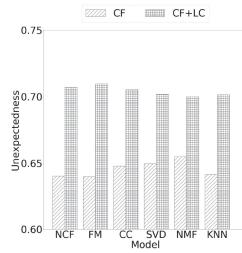
(a) RMSE



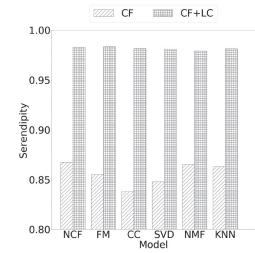
(b) MAE



(c) Unexpectedness



(d) Serendipity



The asterisk denotes 95% statistical significance; we observe significant improvements in unexpectedness measures in (c) and (d), while no significant change in accuracy measures in (a) and (b) at the same time.

6.3 Robustness Analysis

In this article, we show that the proposed latent modeling of unexpectedness significantly improves recommendation performance and provides indeed unexpected recommendations. In total, we conduct the experiments on 3 different datasets using 3 different latent embedding approaches, 6 different collaborative filtering algorithms, 7 different evaluation metrics and 3 different geometric structures for modeling unexpectedness, resulting in 378 experimental settings, where all 378 results are in supportive of our claims. We observe significant improvements in unexpectedness, serendipity, and diversity measures, while we do not witness any loss in accuracy measures compared to plain collaborative filtering algorithms that do not include unexpectedness during the recommendation process. In addition, when compared to baseline unexpected recommendation models, our model significantly outperforms them in both accuracy and unexpectedness measures.

To sum up, the superiority of latent modeling of unexpectedness is robust to the following:

- **Various Datasets.** We conduct the experiments in three different datasets: Yelp dataset, TripAdvisor dataset, and Video dataset, where we obtain consistent improvements in all three datasets.
- **Multiple Latent Embedding Approaches.** To construct the unexpectedness in the latent space, we utilize three state-of-the-art latent embedding approaches (HINE, AF, and ME) and obtain similar superior recommendation performance over baseline models.

Table 10. Comparison of Unexpected Recommendations in the Yelp Dataset
Using Different Geometric Structures

Model	RMSE	MAE	Pre@5	Rec@5	Unexp	Ser	Div
NCF+LCH	0.9158	0.7076	0.7798	0.6308	0.1478	0.4889	0.4170
NCF+LHS	0.9169	0.7078	0.7783	0.6291	0.1450	0.4905	0.4178
NCF+LHC	0.9180	0.7013	0.7725	0.6270	0.1478	0.4930	0.4178
FM+LCH	0.9178	0.6820	0.7700	0.6123	0.1422	0.4593	0.4198
FM+LHS	0.9180	0.6888	0.7704	0.6278	0.1378	0.4603	0.4164
FM+LHC	0.9162	0.6798	0.7698	0.6195	0.1402	0.4608	0.4198
CC+LCH	0.9504	0.7038	0.7596	0.5864	0.1400	0.4660	0.3869
CC+LHS	0.9514	0.7007	0.7626	0.5926	0.1355	0.4793	0.3961
CC+LHC	0.9501	0.7072	0.7645	0.5774	0.1349	0.4644	0.3847
SVD+LCH	0.9134	0.7076	0.7701	0.6175	0.1240	0.4569	0.3524
SVD+LHS	0.9136	0.7039	0.7722	0.6212	0.1214	0.4630	0.3511
SVD+LHC	0.9126	0.7081	0.7720	0.6133	0.1192	0.4534	0.3602
NMF+LCH	0.9522	0.7054	0.7722	0.6233	0.1390	0.4869	0.4030
NMF+LHS	0.9522	0.7026	0.7781	0.6238	0.1466	0.4894	0.4045
NMF+LHC	0.9558	0.7013	0.7692	0.6260	0.1471	0.4852	0.4012
KNN+LCH	0.9128	0.7751	0.7659	0.6273	0.1220	0.4365	0.3259
KNN+LHS	0.9133	0.7715	0.7674	0.6287	0.1288	0.4380	0.3388
KNN+LHC	0.9117	0.7753	0.7662	0.6272	0.1327	0.4421	0.3427

The asterisk denotes 95% statistical significance.

- **Specific Collaborative Filtering Algorithms.** We select six representative collaborative filtering algorithms to estimate user ratings and form the hybrid utility function accordingly. These methods include the deep-learning based approach NCF and five other popular models: FM, CC, SVD, NMF, and KNN. The latent modeling of unexpectedness enables each collaborative filtering algorithm to provide more unexpected recommendations without losing any accuracy measure.
- **Selective Evaluation Metrics.** We evaluate the recommendation performance using accuracy measures RMSE, MAE, Precision, and Recall and unexpectedness measures Unexpectedness, Serendipity, Diversity. The proposed model significantly outperforms baseline unexpected recommendation models in all these seven metrics.
- **Different Geometric Shapes of Latent Closures.** As discussed in Section 3.2, there are three common geometric structures in high-dimensional latent space that are suitable for modeling the closure of latent embeddings: Latent HyperSphere (LHS), Latent Hyper-Cube (LHC), and Latent Convex Hull (LCH). We calculate unexpectedness using the three structures separately and provide unexpected recommendations accordingly. As shown in Tables 10, 11, and 12, the specific selection of geometric structure does not influence the recommendation performance, as we get similar results and neither approach dominates the other two. Instead, it is really the latent modeling of unexpectedness that contributes to the significant improvements of recommendation performance.

6.4 Visualization of Latent Embeddings

Finally, we conduct case study to reveal the effectiveness of modeling unexpectedness through latent embedding approaches. Specifically, we visualize the learned embedding vectors to provide insights of their semantic information in the latent space. Taking the Yelp dataset as an example,

Table 11. Comparison of Unexpected Recommendations in the TripAdvisor Dataset
Using Different Geometric Structures

Model	RMSE	MAE	Pre@5	Rec@5	Unexp	Ser	Div
NCF+LCH	0.9635	0.7317	0.7210	0.9795	0.0579	0.4622	0.0478
NCF+LHS	0.9624	0.7310	0.7201	0.9810	0.0586	0.4635	0.0472
NCF+LHC	0.9652	0.7305	0.7214	0.9814	0.0593	0.4647	0.0469
FM+LCH	1.0275	0.7445	0.7040	0.9656	0.0543	0.4631	0.0393
FM+LHS	1.0230	0.7450	0.7031	0.9638	0.0581	0.4637	0.0388
FM+LHC	1.0218	0.7472	0.7020	0.9632	0.0561	0.4607	0.0407
CC+LCH	1.0285	0.7541	0.6865	0.9703	0.0552	0.4619	0.0471
CC+LHS	1.0230	0.7539	0.6887	0.9754	0.0587	0.4629	0.0491
CC+LHC	1.0200	0.7539	0.6864	0.9730	0.0562	0.4667	0.0498
SVD+LCH	0.9937	0.7517	0.7085	0.9594	0.0544	0.4621	0.0499
SVD+LHS	0.9908	0.7519	0.7093	0.9569	0.0585	0.4614	0.0477
SVD+LHC	0.9884	0.7541	0.7091	0.9474	0.0562	0.4654	0.0485
NMF+LCH	1.0262	0.7533	0.6881	0.9775	0.0544	0.4627	0.0499
NMF+LHS	1.0280	0.7594	0.6864	0.9735	0.0584	0.4629	0.0488
NMF+LHC	1.0265	0.7600	0.6853	0.9711	0.0559	0.4677	0.0504
KNN+LCH	1.0001	0.7483	0.6907	0.9763	0.0543	0.4631	0.0492
KNN+LHS	0.9981	0.7493	0.6909	0.9743	0.0588	0.4625	0.0488
KNN+LHC	0.9950	0.7524	0.6927	0.9701	0.0564	0.4671	0.0500

The asterisk denotes 95% statistical significance.

Table 12. Comparison of Unexpected Recommendations in the Video Dataset
Using Different Geometric Structures

Model	RMSE	MAE	Pre@5	Rec@5	Unexp	Ser	Div
NCF+LCH	0.3799	0.2870	0.2572	0.3638	0.7049	0.9819	0.2538
NCF+LHS	0.3810	0.2854	0.2560	0.3615	0.7070	0.9830	0.2538
NCF+LHC	0.3817	0.2846	0.2549	0.3632	0.7101	0.9852	0.2536
FM+LCH	0.3906	0.2998	0.2510	0.3278	0.7112	0.9840	0.2518
FM+LHS	0.3924	0.3044	0.2498	0.3265	0.7096	0.9833	0.2510
FM+LHC	0.3940	0.3056	0.2506	0.3302	0.7177	0.9833	0.2518
CC+LCH	0.4157	0.3240	0.2564	0.3624	0.7101	0.9817	0.2512
CC+LHS	0.4167	0.3296	0.2569	0.3676	0.7053	0.9815	0.2519
CC+LHC	0.4151	0.3358	0.2553	0.3659	0.7065	0.9830	0.2508
SVD+LCH	0.3841	0.2925	0.2400	0.3277	0.7010	0.9844	0.2408
SVD+LHS	0.3888	0.2862	0.2455	0.3253	0.7018	0.9810	0.2412
SVD+LHC	0.3836	0.2841	0.2433	0.3271	0.7007	0.9812	0.2454
NMF+LCH	0.4423	0.3306	0.2380	0.3491	0.7008	0.9799	0.2488
NMF+LHS	0.4405	0.3330	0.2494	0.3439	0.6999	0.9792	0.2450
NMF+LHC	0.4433	0.3387	0.2420	0.3459	0.6961	0.9803	0.2438
KNN+LCH	0.4106	0.3107	0.2584	0.3175	0.7007	0.9817	0.2558
KNN+LHS	0.4088	0.3091	0.2608	0.3212	0.7014	0.9814	0.2558
KNN+LHC	0.4069	0.3099	0.2620	0.3248	0.7073	0.9830	0.2519

The asterisk denotes 95% statistical significance.

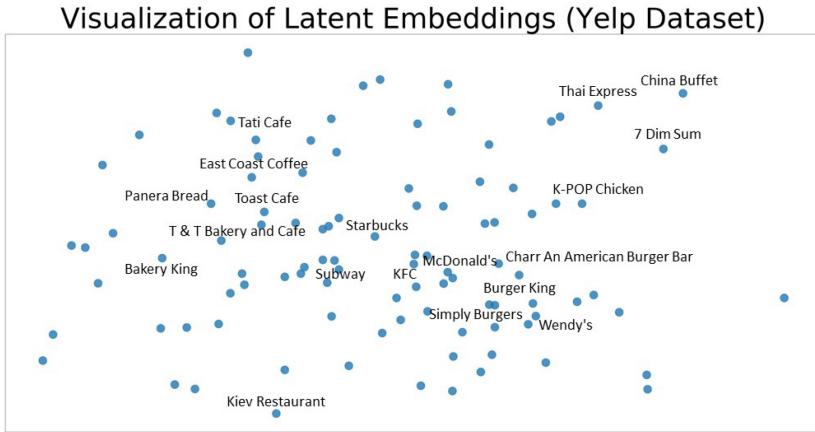


Fig. 6. t-SNE Visualization of Latent Embeddings.

we randomly select 100 restaurants from the dataset and obtain their corresponding embeddings through the HINE method. In Figure 6, we show the visualization of those embeddings through t-SNE [40], in which similar restaurants are clustered close to each other. We could see that cafes and bakeries are clustered to the left side, whereas burger bars and fast food restaurants are clustered to the right side, and Asian restaurants are clustered to the far right in the latent space. Therefore, we show that the latent embedding approaches we use in this article are indeed capable of capturing latent relations among items and thus providing precise modeling of unexpectedness.

7 CONCLUSION

In this article, we propose novel latent modeling of unexpectedness that simultaneously provides unexpected and satisfying recommendations. Specifically, we define unexpectedness of a new item as the distance between the embedding of that item in the latent space and the closure of all the previously consumed item embeddings. This new definition enables us to capture latent, complex and heterogeneous relationships between users and items that significantly improves performance and practicability of unexpected recommendations. To achieve this, we design a hybrid utility function as the linear combination of estimated ratings and unexpectedness to optimize accuracy and unexpectedness objectives of recommendations simultaneously. Furthermore, we demonstrate that the proposed approach consistently and significantly outperforms all other baseline models in terms of unexpectedness, serendipity, and diversity measures without losing any accuracy performance.

The contributions of this article are threefold. First, we propose *latent* modeling of unexpectedness. Though it is a common idea to explore latent space for recommendations, it is not obvious how to do it for *unexpected* recommendations, as we have discussed in Section 3. Second, we construct the hybrid utility function that combines the proposed unexpectedness measure with the rating estimation value and provides unexpected recommendations based on the hybrid utility values. We demonstrate that this approach significantly outperforms all other unexpected recommendation baselines. Third, we conduct extensive experiments in multiple settings and show that it is indeed the latent modeling of unexpectedness that leads to the significant increase in unexpectedness measures without sacrificing any performance accuracy. Thus, the proposed approach helps users to break out of their filter bubbles.

As the future work, we plan to conduct live experiments within real business environments to further evaluate the effectiveness of unexpected recommendations and analyze both qualitative

and quantitative aspects in online retail settings through A/B tests. Specifically, we plan to launch our model in an industrial platform and measure its performance using business metrics, including CTR and GMV. Moreover, we will further explore the impact of unexpected recommendations on user satisfaction. Finally, we plan to design algorithms that automatically incorporate the concept of unexpectedness into the deep-learning recommendation framework that optimizes the recommendation performance and the construction of latent embeddings at the same time.

REFERENCES

- [1] Panagiotis Adamopoulos. 2014. On discovering non-obvious recommendations: Using unexpectedness and neighborhood selection methods in collaborative filtering systems. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*. ACM, 655–660.
- [2] Panagiotis Adamopoulos and Alexander Tuzhilin. 2014. On unexpectedness in recommender systems: Or how to better expect the unexpected. *ACM Trans. Intell. Syst. Technology (TIST)* 5, 4 (2014), 1–32.
- [3] Gediminas Adomavicius and YoungOk Kwon. 2011. Improving aggregate recommendation diversity using ranking-based techniques. *IEEE Trans. Knowl. Data Eng.* 24, 5 (2011), 896–911.
- [4] Gediminas Adomavicius and YoungOk Kwon. 2011. *Maximizing Aggregate Recommendation Diversity: A Graph-theoretic Approach*. Citeseer.
- [5] Takayuki Akiyama, Kiyohiro Obara, and Masaaki Tanizaki. 2010. Proposal and evaluation of serendipitous recommendation method using general unexpectedness. In *Proceedings of the ACM Conference on Practical Use of Recommender Systems, Algorithms, & Technology and ACM Conference on Recommender Systems (PRSAT@ RecSys’10)*. 3–10.
- [6] Naomi S. Altman. 1992. An introduction to kernel and nearest-neighbor nonparametric regression. In *The American Statistician*. Taylor & Francis Group, 175–185.
- [7] C. Bradford Barber, David P. Dobkin, and Hannu Huhdanpaa. 1996. The quickhull algorithm for convex hulls. *ACM Trans. Math. Softw.* 22, 4 (1996), 469–483.
- [8] Andrea Barraza-Urbina. 2017. The exploration-exploitation trade-off in interactive recommender systems. In *Proceedings of the 11th ACM Conference on Recommender Systems*. 431–435.
- [9] Li Chen, Yonghua Yang, Ningxia Wang, Keping Yang, and Quan Yuan. 2019. How serendipity improves user satisfaction with recommendations? a large-scale user evaluation. In *Proceedings of the World Wide Web Conference*. ACM, 240–250.
- [10] Laming Chen, Guoxin Zhang, and Eric Zhou. 2018. Fast greedy MAP inference for determinantal point process to improve recommendation diversity. In *Advances in Neural Information Processing Systems*. 5622–5633.
- [11] Yu-Shian Chiu, Kuei-Hong Lin, and Jia-Sin Chen. 2011. A social network-based serendipity recommender system. In *Proceedings of the 2011 International Symposium on Intelligent Signal Processing and Communications Systems (IS-PACS’11)*. IEEE, 1–5.
- [12] Yuxiao Dong, Nitesh V. Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 135–144.
- [13] Mike Gartrell, Ulrich Paquet, and Noam Koenigstein. 2017. Low-rank factorization of determinantal point processes. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*. 1912–1918.
- [14] Mouzhi Ge, Carla Delgado-Battenfeld, and Dietmar Jannach. 2010. Beyond accuracy: Evaluating recommender systems by coverage and serendipity. In *Proceedings of the 4th ACM Conference on Recommender Systems*. ACM, 257–260.
- [15] Thomas George and Srujana Merugu. 2005. A scalable collaborative filtering framework based on co-clustering. In *Proceedings of the 5th IEEE International Conference on Data Mining*. 625–628.
- [16] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 855–864.
- [17] 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.
- [18] Sigurdur Helgason. 2001. *Differential Geometry and Symmetric Spaces*. Vol. 341. American Mathematical Soc.
- [19] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. 2004. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.* 22, 1 (2004), 5–53.
- [20] Yoshinori Hijikata, Takuya Shimizu, and Shogo Nishida. 2009. Discovery-oriented collaborative filtering for improving user satisfaction. In *Proceedings of the 14th International Conference on Intelligent User Interfaces*. 67–76.
- [21] Geoffrey E. Hinton and Ruslan R. Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *Science* 313, 5786 (2006), 504–507.

- [22] Neil J. Hurley. 2013. Personalised ranking with diversity. In *Proceedings of the 7th ACM Conference on Recommender Systems*. 379–382.
- [23] Leo Iaquinta, Marco De Gemmis, Pasquale Lops, Giovanni Semeraro, Michele Filannino, and Piero Molino. 2008. Introducing serendipity in a content-based recommender system. In *Proceedings of the 8th International Conference on Hybrid Intelligent Systems (HIS'08)*. IEEE, 168–173.
- [24] Leo Iaquinta, Marco de Gemmis, Pasquale Lops, Giovanni Semeraro, and Piero Molino. 2010. Can a recommender system induce serendipitous encounters? In *E-commerce*. InTech.
- [25] Masayuki Ishikawa, Peter Geczy, Noriaki Izumi, and Takahira Yamaguchi. 2008. Long tail recommender utilizing information diffusion theory. In *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, Vol. 1. IEEE, 785–788.
- [26] Junzo Kamahara, Tomofumi Asakawa, Shinji Shimojo, and Hideo Miyahara. 2005. A community-based recommendation system to reveal unexpected interests. In *Proceedings of the 11th International Multimedia Modelling Conference*. IEEE, 433–438.
- [27] Marius Kaminskas and Derek Bridge. 2016. Diversity, serendipity, novelty, and coverage: A survey and empirical analysis of beyond-accuracy objectives in recommender systems. *ACM Trans. Interact. Intell. Syst.* 7, 1 (2016), 1–42.
- [28] Komal Kapoor, Vikas Kumar, Loren Terveen, Joseph A. Konstan, and Paul Schrater. 2015. I like to explore sometimes: Adapting to dynamic user novelty preferences. In *Proceedings of the 9th ACM Conference on Recommender Systems*. ACM, 19–26.
- [29] Komal Kapoor, Karthik Subbian, Jaideep Srivastava, and Paul Schrater. 2015. Just in time recommendations: Modeling the dynamics of boredom in activity streams. In *Proceedings of the 8th ACM International Conference on Web Search and Data Mining*. ACM, 233–242.
- [30] Noriaki Kawamae. 2010. Serendipitous recommendations via innovators. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 218–225.
- [31] Noriaki Kawamae, Hitoshi Sakano, and Takeshi Yamada. 2009. Personalized recommendation based on the personal innovator degree. In *Proceedings of the 3rd ACM Conference on Recommender Systems*. ACM, 329–332.
- [32] John Paul Kelly and Derek Bridge. 2006. Enhancing the diversity of conversational collaborative recommendations: A comparison. *Artif. Intell. Rev.* 25, 1–2 (2006), 79–95.
- [33] Daniel D. Lee and H. Sebastian Seung. 2001. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*. 556–562.
- [34] Kibeom Lee and Kyogu Lee. 2015. Escaping your comfort zone: A graph-based recommender system for finding novel recommendations among relevant items. *Exp. Syst. Appl.* 42, 10 (2015), 4851–4858.
- [35] Pan Li and Alexander Tuzhilin. 2019. Latent modeling of unexpectedness for recommendations. In *Proceedings of ACM RecSys 2019 Late-breaking Results* (2019), 7–10.
- [36] Pan Li and Alexander Tuzhilin. 2019. Latent multi-criteria ratings for recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 428–431.
- [37] Pan Li and Alexander Tuzhilin. 2020. DDTCRD: Deep dual transfer cross domain recommendation. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 331–339.
- [38] Yen-Yu Lin, Tyng-Luh Liu, and Hwann-Tzong Chen. 2005. Semantic manifold learning for image retrieval. In *Proceedings of the 13th Annual ACM International Conference on Multimedia*. 249–258.
- [39] Qiuxia Lu, Tianqi Chen, Weinan Zhang, Diyi Yang, and Yong Yu. 2012. Serendipitous personalized ranking for top-n recommendation. In *Proceedings of the 2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology (WI-IAT'12)*, Vol. 1. IEEE, 258–265.
- [40] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *J. Mach. Learn. Res.* 9, Nov. (2008), 2579–2605.
- [41] Sean M. McNee, John Riedl, and Joseph A. Konstan. 2006. Being accurate is not enough: How accuracy metrics have hurt recommender systems. In *CHI'06 Extended Abstracts on Human Factors in Computing Systems*. 1097–1101.
- [42] Tomoko Murakami, Koichiro Mori, and Ryoei Orihara. 2007. Metrics for evaluating the serendipity of recommendation lists. In *Proceedings of the Annual Conference of the Japanese Society for Artificial Intelligence*. Springer, 40–46.
- [43] Tien T. Nguyen, Pik-Mai Hui, F. Maxwell Harper, Loren Terveen, and Joseph A. Konstan. 2014. Exploring the filter bubble: the effect of using recommender systems on content diversity. In *Proceedings of the 23rd International Conference on World Wide Web*. ACM, 677–686.
- [44] Jinoh Oh, Sun Park, Hwan-Yu Yu, Min Song, and Seung-Taek Park. 2011. Novel recommendation based on personal popularity tendency. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining*. IEEE, 507–516.
- [45] Yingwei Pan, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui. 2016. Jointly modeling embedding and translation to bridge video and language. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4594–4602.

- [46] Eli Pariser. 2011. *The Filter Bubble: How the New Personalized Web Is Changing What We Read and How We Think*. Penguin.
- [47] Yoon-Joo Park and Alexander Tuzhilin. 2008. The long tail of recommender systems and how to leverage it. In *Proceedings of the 2008 ACM Conference on Recommender Systems*. 11–18.
- [48] Steffen Rendle. 2010. Factorization machines. In *Proceedings of the 2010 IEEE 10th International Conference on Data Mining (ICDM'10)*. IEEE, 995–1000.
- [49] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1985. *Learning Internal Representations by Error Propagation*. Technical Report. University of California, San Diego, La Jolla Institute for Cognitive Science.
- [50] Alan Said, Benjamin Kille, Brijnesh J. Jain, and Sahin Albayrak. 2012. Increasing diversity through furthest neighbor-based recommendation. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM '12)*.
- [51] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2002. Incremental singular value decomposition algorithms for highly scalable recommender systems. In *Proceedings of the 5th International Conference on Computer and Information Science*. Citeseer.
- [52] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. 2015. Autorec: Autoencoders meet collaborative filtering. In *Proceedings of the 24th International Conference on World Wide Web*. ACM, 111–112.
- [53] Guy Shani and Asela Gunawardana. 2011. Evaluating recommendation systems. In *Recommender Systems Handbook*. Springer, 257–297.
- [54] Chuan Shi, Binbin Hu, Wayne Xin Zhao, and Philip S. Yu. 2018. Heterogeneous information network embedding for recommendation. *IEEE Trans. Knowl. Data Eng.* 31, 2 (2018), 357–370.
- [55] Chuan Shi, Yitong Li, Jiawei Zhang, Yizhou Sun, and S. Yu Philip. 2017. A survey of heterogeneous information network analysis. *IEEE Trans. Knowl. Data Eng.* 29, 1 (2017), 17–37.
- [56] Lei Shi. 2013. Trading-off among accuracy, similarity, diversity, and long-tail: A graph-based recommendation approach. In *Proceedings of the 7th ACM Conference on Recommender Systems*. 57–64.
- [57] Yue Shi, Xiaoxue Zhao, Jun Wang, Martha Larson, and Alan Hanjalic. 2012. Adaptive diversification of recommendation results via latent factor portfolio. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 175–184.
- [58] Morton Slater. 2014. Lagrange multipliers revisited. In *Traces and Emergence of Nonlinear Programming*. Springer, 293–306.
- [59] Barry Smyth and Paul McClave. 2001. Similarity vs. diversity. In *International Conference on Case-based Reasoning*. Springer, 347–361.
- [60] Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. 2012. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 2951–2959.
- [61] Ruilong Su, Li'Ang Yin, Kailong Chen, and Yong Yu. 2013. Set-oriented personalized ranking for diversified top-n recommendation. In *Proceedings of the 7th ACM Conference on Recommender Systems*. 415–418.
- [62] Yizhou Sun and Jiawei Han. 2013. Mining heterogeneous information networks: A structural analysis approach. *ACM SIGKDD Explor. Newslett.* 14, 2 (2013), 20–28.
- [63] Maria Taramigkou, Eftimios Bothos, Konstantinos Christidis, Dimitris Apostolou, and Gregoris Mentzas. 2013. Escape the bubble: Guided exploration of music preferences for serendipity and novelty. In *Proceedings of the 7th ACM Conference on Recommender Systems*. ACM, 335–338.
- [64] Saúl Vargas, Linas Baltrunas, Alexandros Karatzoglou, and Pablo Castells. 2014. Coverage, redundancy and size-awareness in genre diversity for recommender systems. In *Proceedings of the 8th ACM Conference on Recommender Systems*. 209–216.
- [65] Saúl Vargas and Pablo Castells. 2011. Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the 5th ACM Conference on Recommender Systems*. 109–116.
- [66] Jizhe Wang, Pipei Huang, Huan Zhao, Zhibo Zhang, Binqiang Zhao, and Dik Lun Lee. 2018. Billion-scale commodity embedding for e-commerce recommendation in alibaba. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 839–848.
- [67] Jacek Wasilewski and Neil Hurley. 2019. Bayesian personalized ranking for novelty enhancement. In *Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization*. 144–148.
- [68] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, Richang Hong, and Tat-Seng Chua. 2019. MMGCN: Multi-modal graph convolution network for personalized recommendation of micro-video. In *Proceedings of the 27th ACM International Conference on Multimedia*. 1437–1445.
- [69] Li-Tung Weng, Yue Xu, Yuefeng Li, and Richi Nayak. 2007. Improving recommendation novelty based on topic taxonomy. In *Proceedings of the 2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Workshops*. IEEE, 115–118.

- [70] Le Wu, Qi Liu, Enhong Chen, Nicholas Jing Yuan, Guangming Guo, and Xing Xie. 2016. Relevance meets coverage: A unified framework to generate diversified recommendations. *ACM Trans. Intell. Sys. Technol.* 7, 3 (2016), 1–30.
- [71] Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norick, and Jiawei Han. 2014. Personalized entity recommendation: A heterogeneous information network approach. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*. ACM, 283–292.
- [72] Xiao Yu, Xiang Ren, Yizhou Sun, Bradley Sturt, Urvashi Khandelwal, Quanquan Gu, Brandon Norick, and Jiawei Han. 2013. Recommendation in heterogeneous information networks with implicit user feedback. In *Proceedings of the 7th ACM Conference on Recommender Systems*. ACM, 347–350.
- [73] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM Comput. Surv. (CSUR)* 52, 1 (2019), 1–38.
- [74] Yuan Cao Zhang, Diarmuid Ó Séaghdha, Daniele Quercia, and Tamas Jambor. 2012. Auralist: Introducing serendipity into music recommendation. In *Proceedings of the 5th ACM International Conference on Web Search and Data Mining*. ACM, 13–22.
- [75] Qianru Zheng, Chi-Kong Chan, and Horace H. S. Ip. 2015. An unexpectedness-augmented utility model for making serendipitous recommendation. In *Proceedings of the Industrial Conference on Data Mining*. Springer, 216–230.
- [76] Tao Zhou, Zoltán Kuscsik, Jian-Guo Liu, Matúš Medo, Joseph Rushton Wakeling, and Yi-Cheng Zhang. 2010. Solving the apparent diversity-accuracy dilemma of recommender systems. *Proc. Natl. Acad. Sci. U.S.A.* 107, 10 (2010), 4511–4515.
- [77] Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. 2005. Improving recommendation lists through topic diversification. In *Proceedings of the 14th International Conference on World Wide Web*. ACM, 22–32.
- [78] Zainab Zolaktaf, Reza Babanezhad, and Rachel Pottinger. 2018. A generic top-n recommendation framework for trading-off accuracy, novelty, and coverage. In *Proceedings of the 2018 IEEE 34th International Conference on Data Engineering (ICDE'18)*. IEEE, 149–160.

Received February 2020; revised April 2020; accepted June 2020