HAES: A New Hybrid Approach for Movie Recommendation with Elastic Serendipity

Xueqi Li
lee_xq@hnu.edu.cn
Hunan University

Wenjun Jiang∗
jiangwenjun@hnu.edu.cn
Hunan University

Weiguang Chen
cwg@hnu.edu.cn
Hunan University

Jie Wu
jiewu@temple.edu
Temple University

Guojun Wang
csgwang@gzhu.edu.cn
Guangzhou University

ABSTRACT

Recommendation systems provide good guidance for users to find their favorite movies from an overwhelming amount of options. However, most systems excessively pursue the recommendation accuracy and give rise to over-specialization, which triggers the emergence of serendipity. Hence, serendipity recommendation has received more attention in recent years, facing three key challenges: subjectivity in the definition, the lack of data, and users’ floating demands for serendipity. To address these challenges, we introduce a new model called HAES, a Hybrid Approach for movie recommendation with Elastic Serendipity, to recommend serendipitous movies. Specifically, we (1) propose a more objective definition of serendipity, content difference and genre accuracy, according to the analysis on a real dataset, (2) propose a new algorithm named JohnsonMax to mitigate the data sparsity and build weak ties beneficial to finding serendipitous movies, and (3) define a novel concept of elasticity in the recommendation, to adjust the level of serendipity flexibly and reach a trade-off between accuracy and serendipity. Extensive experiments on real-world datasets show that HAES enhances the serendipity of recommendations while preserving recommendation quality, compared to several widely used methods.

CSC Concepts

• Information systems → Recommender systems.

KEYWORDS

Movie Recommendation, Serendipity, Elasticity

ACM Reference Format:

∗Wenjun Jiang is the corresponding author.

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.

CIKM ’19, November 3–7, 2019, Beijing, China
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6976-3/19/11...
https://doi.org/10.1145/3357384.3357868

1 INTRODUCTION

With the development of the Internet, it becomes more and more flexible for users to choose where, when and how to watch movies, leading to the rapid growth in the number and genre of movies available. However, the delightful variation also brings users a huge trouble in finding movies they potentially like. Recommendation systems play an indispensable role in mitigating this problem, by providing interesting options in line with the users’ profile [26]. For catering to users’ preferences, the majority of methods analyze their past behaviors and adopt collaborative filtering to generate the corresponding recommendations [19, 32, 38]. As a result, over-specialization and the long tail effect grow increasingly obvious, calling for more researches on serendipity-oriented recommendation. Although some work has been done, they usually assume that there is a fixed level of serendipity suitable for all users, which results in unrelated recommendations. In this paper, we strive to develop a deep study of elastic serendipity and its application in movie recommendation. To be specific, we propose the concepts of user elasticity and movie elasticity, and put forward a hybrid approach for movie recommendation with elastic serendipity (HAES).

Table 1: Statistics of recommendations in Figure 1

<table>
<thead>
<tr>
<th># of subfigure</th>
<th>ACC</th>
<th>SER</th>
<th>HAES</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>$m_2$, $m_4$</td>
<td>$m_1$, $m_3$</td>
<td>$m_1$, $m_3$, $m_4$</td>
</tr>
<tr>
<td>(b)</td>
<td>$m_2$, $m_4$</td>
<td>$m_1$, $m_3$</td>
<td>$m_1$, $m_2$, $m_4$</td>
</tr>
</tbody>
</table>

Figure 1: An example to illustrate the difference among accuracy-oriented method (ACC), recommendation with fixed serendipity (SER), and our method (HAES).
We provide an example in Figure 1 to show the effect our approach tries to reach, where we compare our method with two other commonly used approaches: ACC (accuracy-based method) and SER (recommendation approach with fixed serendipity). In the figure, the solid circles denote the elasticity (defined in Section 3.1) of users (e.g., \( u_i \), \( u_j \)) or movies (e.g., \( m_1 \), \( m_2 \)). The larger a circle is, the stronger is the user’s ability to accept different movies (the bigger is the possibility of movie being adopted). The dashed circles represent the ranges of recommendation, e.g., \( l_{ACC} \) denotes the recommendation range of ACC, \( l_{SER} \) is that of SER, and \( l_{HAES} \) is that of HAES. We would recommend a movie if the edge closest to the user (e.g., \( \alpha_i \) for \( m_1 \) in Figure 1(a), \( \alpha_j \) for \( m_1 \) in (b)) is within our recommendation range, while the other two methods generate recommendations depending on whether the center of the movie is within their recommendation ranges. The difference indicates the sensitivity of HAES to movie elasticity. In addition, our approach is also aware of user elasticity, which is reflected in the comparison between (a) and (b) in Figure 1. The user in (a), \( u_i \), has the same associations with five movies \( (m_1, ..., m_5) \) as \( u_j \) in (b), but the elasticity of \( u_i \) is larger than that of \( u_j \). Among three methods, only our approach recommends different movies for \( u_i \) and \( u_j \) (see Table 1).

For example, we recommend \( m_3 \) only to \( u_i \), since the elasticity of \( u_i \) is too small to accept it. Considering \( u_i \)’s elasticity is bigger, it’s reasonable to recommend movies with weak relevance \( (m_2 \) is not a suitable option) to improve recommendation serendipity.

Existing approaches primarily aim at increasing the accuracy of recommendations [4, 7, 37], achieving great success with the powerful aid of deep learning [2, 27, 35]. However, user satisfaction doesn’t continually increase with the improvement of recommendation accuracy. On the contrary, it shows a correlation with the serendipity of recommendations [6, 20]. Consequently, it draws increasing attention to researches on serendipity-oriented recommendation. Researchers in the area generally split serendipity into related attributes that are easy to measure [28], such as the relevance, novelty and unexpectedness in [15]. Then, they recommend movies in accordance with these attributes, alleviating over-specialization. However, most researchers define serendipity based on some subjective views and generate recommendations with a fixed degree of serendipity for different users, causing the risk of irrelevant recommendations and decreasing user satisfaction [6, 22]. It’s not an easy task to enhance serendipity without lowering the quality of recommendations. There are three major challenges.

1. The definitions of serendipity lack objective evidences. (2) There is very limited labeled data available in serendipity-oriented recommendation. (3) It’s difficult to judge what level of serendipity is the most suitable for different movies and users.

Our Motivations. We have three motivations for addressing the above challenges in this paper. (1) providing a more objective definition on serendipity. We strive to define serendipity according to the analysis on real-world datasets instead of subjective ideas. (2) mitigating the lack of data. We try to dig out attributes related to serendipity and build weak ties among users and movies for alleviating the data sparsity. (3) recommending movies with elastic serendipity. We manage to adjust the degree of recommendations’ serendipity flexibly for different users and movies.

We try to recommend movies with an adaptive level of serendipity for different users, so as to mitigate the over-specialization, reduce the risk of unrelated recommendations, and make a balance between serendipity and accuracy. To be specific, our objective mainly consists of two parts: digging out user-movie elastic associations and predicting genres with high accuracy. Combining these two parts, we propose a new serendipity-based algorithm named HAES and verify its effectiveness on real-world datasets. Our contributions are fivefold, as follows:

- On the basis of a deep analysis on a real-world dataset, Serendipity-2018 [14], we develop a more objective definition of serendipity by decomposing it into two attributes, content difference and genre accuracy. (Section 3.1)
- We propose the concepts of user elasticity and movie elasticity, dynamically balancing serendipity and accuracy of recommended lists. (Sections 3.1 and 4.1)
- We propose JohnsonMax algorithm to optimize user-movie associations, alleviating the data sparsity. (Section 4.2)
- We predict user preferences in movie genre with Recurrent Neural Network (RNN) [21] and achieve significant performance improvements. To the best of our knowledge, this is the first work to apply RNN models into serendipity-oriented recommendation methods. (Section 4.3)
- We conduct comprehensive experiments on MovieLens-1m [8] and MovieLens-latest-small [8], to compare our method with widely used methods. Experimental results demonstrate that HAES provides significantly better guide for users to find serendipitous movies while preserving high genre accuracy. For example, HAES improves micro_F1 by 53.93% and increases 2.3 times on difference compared with the accuracy-oriented method on MovieLens-1m. (Section 5)

2 RELATED WORK

There is a long history of accuracy recommendation systems [1, 3, 25], where the more accurate they are, the more obvious over-specialization becomes, which accelerates the development of serendipity recommendation. In this section, we would briefly review related works on definitions, methods and metrics of serendipity.

Research on definitions of serendipity. The definition of serendipity stems from some abstract descriptions in literature. For example, Walpole describes it as the experience of discovering interesting things by accident [29]. Some researchers crystalize it with some related attributes, such as unexpectedness and usefulness in [6], novelty, relevance and unexpectedness in [17].

Research on methods of serendipity. Some proposed approaches are based on accuracy recommendation methods, and the others are novel, of which the most popular ones are reviewed here. Kotkov et al. [16] apply a greedy strategy to improve recommendation serendipity through resorting the recommended lists. Karpus et al. [13] introduce ontology-based contextual pre-filtering to remove movies particularly familiar to users. Zhang et al. [36] implement three basic approaches to generate corresponding lists respectively. Aides from those accuracy based recommendations, there are also many novel approaches. Niu [23] sets up a framework with the components of unexpectedness, value and learning to recommend

---

2. https://grouplens.org/datasets movielens/1m/
3. https://grouplens.org/datasets movielens/
serendipitous items. Serendipitous recommendations should meet the requirements of low pre-interest and high post-satisfaction in the views of Yang et al. [34]. Transfer learning is adopted by Pandee et al. [24], and Nguyen et al. [22] pose that users are keen on diversity, popularity and serendipity to different extent.

Among existing works, SIRUP [18] is the most similar one to ours, taking into account users’ ability to accept serendipity. There are two significant differences between SIRUP and ours: (1) the possibility of movies being adopted is only considered by us; (2) we propose elasticity to quantify the acceptance ability rather than divide users into two groups, which is more flexible and fine-grained.

Research on metrics of serendipity. Here we divide evaluation strategies into two types: online and offline. For online evaluations, most researchers evaluate the performance of recommendation systems with questionnaires [14, 16], while the others take advantage of users’ implicit feedback. For example, Ge et al. [6] use facial expression recognition for assessment on the serendipity of recommendations. For offline evaluations, it’s a common choice to measure serendipity with related attributes proposed in the definition [18, 36], and we employ this manner in our experiments.

3 PROBLEM STATEMENT AND MODEL OVERVIEW
We illustrate the system settings and concepts, define the problem we solve, and give a brief overview of our solution. Notations are described in Table 2.

3.1 System Settings
We consider the scenario in a movie recommendation system (e.g., MovieLens), where there are users, movies, and ratings. These entities constitute the input space, $I(U, M, S)$, of our work.

3.1.1 Data Analysis. Due to the subjectivity of serendipity and the lack of large-scale datasets, it’s almost impossible to recommend movies according to the value of serendipity directly. Motivated by the great progress achieved by methods based on serendipity decomposition, we employ a similar approach to define serendipity. Our difference lies in that we exploit the objective analysis rather than subjective ideas as the foundation for defining serendipity.

We analyze a real-world dataset, Serendipity-2018 [14] with 2150 records, where 481 users rate 1678 movies in eight statements (see Table 3). In these statements, the objectives of serendipity recommendation are $s7$ and $s8$, and the others are attributes related to the serendipity. We adopt linear regression to acquire the associations $w$ between the objective and other attributes as follows:

$$s7 + \theta * s8 = w * [s1, s2, s3, s4, s5, s6] + b,$$

where $\theta$ indicates the contribution of $s8$ (broadening user horizons), we use $\theta = 1$ in this paper, $b$ is the random error. We learn the value of $w$ using the dataset Serendipity-2018 and display the weights of related attributes in Figure 2. It indicates that there are stronger associations between the serendipity and $s3$, $s4$, $s5$ (interesting to users, beyond their ability to discover, and different from history).

3.1.2 The Key Concepts. Based on the above observation, we conclude that serendipitous movies should meet the requirements of being different from users’ past behaviors and relevant to their interests. Here, we embody the difference as content difference and the relevance as genre accuracy, and we would verify its justifiability in the experiments (Section 5.1). Consequently, we propose the definitions of content difference, genre accuracy, and serendipity.

**Definition 1: Content Difference.** A movie with content difference is different from the history of the target user $u_t$ (i.e., a movie with weak similarity to the movies $u_t$ has rated).

**Definition 2: Genre Accuracy.** A movie with genre accuracy is the one that meets $u_t$’s short-term preferences in movie genre.

<table>
<thead>
<tr>
<th>No.</th>
<th>Description</th>
<th>Abbreviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>s1</td>
<td>The first time I heard of this movie was when MovieLens suggested it to me.</td>
<td>unknown</td>
</tr>
<tr>
<td>s2</td>
<td>MovieLens influenced my decision.</td>
<td>influence</td>
</tr>
<tr>
<td>s3</td>
<td>I expected to enjoy this movie before watching it for the first time.</td>
<td>interest</td>
</tr>
<tr>
<td>s4</td>
<td>This is the type of movie I would not normally discover on my own; I need a recommender system like MovieLens.</td>
<td>beyond users’ ability to discovery</td>
</tr>
<tr>
<td>s5</td>
<td>This movie is different from the movies I usually watch.</td>
<td>different</td>
</tr>
<tr>
<td>s6</td>
<td>I was surprised that MovieLens picked this movie to recommend to me.</td>
<td>surprise</td>
</tr>
<tr>
<td>s7</td>
<td>I am glad I watched this movie.</td>
<td>satisfaction</td>
</tr>
<tr>
<td>s8</td>
<td>Watching this movie broadened my preferences. Now I am interested in a wider selection of movies.</td>
<td>broaden users’ preferences</td>
</tr>
</tbody>
</table>

Table 2: Notations

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U$</td>
<td>users ${u_0, u_1, \ldots, u_n}$</td>
</tr>
<tr>
<td>$M$</td>
<td>movies ${m_0, m_1, \ldots, m_k}$</td>
</tr>
<tr>
<td>$G$</td>
<td>movie genres ${G_0, G_1, \ldots, G_k}$</td>
</tr>
<tr>
<td>$V$</td>
<td>nodes in relevance network, $V = U \cup M$</td>
</tr>
<tr>
<td>$S$</td>
<td>rating matrix</td>
</tr>
<tr>
<td>$I$</td>
<td>the input space, $I = {U, M, S}$</td>
</tr>
<tr>
<td>$R_{ij}$</td>
<td>the relevance factor from node $i$ to $j$</td>
</tr>
<tr>
<td>$E_{ij}$</td>
<td>the elasticity factor between $u_i$ and $m_j$</td>
</tr>
<tr>
<td>$N$</td>
<td>the relevance network, $N = (V, R)$</td>
</tr>
</tbody>
</table>

Table 3: Serendipity-2018 Statements

Figure 2: Weights of each related features.
We give a brief overview of the proposed (Section 3.1.1), we define it as $I$ to create a balance between serendipity and accuracy.

Then we propose the JohnsonMax algorithm to update it. (3) Genres. In order to acquire latent associations among users and movies, we introduce an asymmetric metric to build the recommendation network, and then propose the JohnsonMax algorithm to update it. (3) Genres. In order to preserve recommendation quality, we apply GRU to predict users’ short-term preferences in genre. (4) Recommendation. We filter the candidates with genres predicted by genre component and then recommend movies with elasticity and relevance, expecting to create a balance between serendipity and accuracy.

4 HAES: ALGORITHM DETAILS

In this section, we present the technical details of components in HAES: quantifying elasticity, building relevance network, predicting genres, and recommendation with elastic serendipity.

4.1 Quantifying Elasticity

We make a quantification of the elasticity to generate recommendations with elastic serendipity for users, so as to dynamically meet their varying demands for serendipity and accuracy [22].

**User Elasticity.** We measure the user elasticity of $u_i$ with the diversity in movie genres related to $u_i$. As stated in [18], the more movie genres $u_i$ has, the stronger is his ability to accept different movies. Suppose $G(u_i)$ is a genre set related to $u_i$, $G(U) = \{G(u_0), G(u_1), ..., G(u_n)\}$, and $G_{max}(U)$ is the element of $G(U)$ containing the most genres, the elasticity of $u_i$, $E(u_i)$, is calculated as follows:

$$E(u_i) = \frac{|G(u_i)|}{|G_{max}(U)|}.$$  

(Movie Elasticity. We regard the diversity of corresponding user (who has rated the movie) groups as a reference to measure the movie elasticity, considering users’ age, occupation, and the group size. To be specific, we suppose $A(m_i)$ denotes the age set of users associated to $m_i$ and $O(m_i)$ is about occupation; $\alpha$ and $\beta$ indicate the contributions of age and occupation, respectively; $U(m_i)$ is a set of users related to $m_i$. We calculate the diversity factor of $m_i$, $D(m_i)$, as follows:

$$D(m_i) = \frac{\alpha \cdot |A(m_i)| + \beta \cdot |O(m_i)| + |U(m_i)|}{\alpha + \beta + 1}.$$  

Suppose $D_{max}(M)$ is the maximum $D(m_j)$, where $m_j \in M$, the elasticity of $m_i$, $E(m_i)$, is calculated as in the following:

$$E(m_i) = \frac{D(m_i)}{D_{max}(M)}.$$  

User-Movie Elasticity. We propose the user-movie elasticity $E(u_i, m_j)$, to indicate the possibility that $u_i$ accepts $m_j$ considering only the elasticity of $u_i$ and $m_j$ (not considering the relevance between them). It is calculated with $E(u_i)$ and $E(m_j)$, as follows:

$$E(u_i, m_j) = \frac{E(u_i) + \delta \cdot E(m_j)}{1 + \delta},$$  

where $\delta$ denotes the contribution of $E(m_j)$.

4.2 Building Relevance Network

In order to make personalized recommendations, it is essential to obtain relatively accurate relationships among users and movies. We adopt two steps, building and updating the relevance network, to capture relevance among nodes (i.e., users and movies).

Building the Relevance Network. We propose new asymmetric metrics on the basis of Jaccard index [10] instead of symmetric ones common in prior works, so as to reveal relationships among users and movies in a more proper manner. We illustrate our idea with 4(a), where $u_i$ is related to movies in genres of action, romance, and comedy, and $u_i$ is related to all genres in the figure. Then, it’s reasonable to recommend $u_i$’s relevant movies to $u_i$, but the opposite may generate unrelated recommendations for $u_i$. It indicates the necessity of asymmetric measures. For associations between movies, let $U(m_i)$ denote users related to $m_i$, researchers usually...
propose to update the relevance from node $v_i$ to $v_j$, $w(u_i, v_j)$, using the potential relevance $w'(v_i, v_j)$ generated by the influence propagation theory:

$$w'(v_i, v_j) = (1 - \Gamma) \cdot w(v_i, v_p) + w(v_p, v_j).$$

(11)

where $\Gamma$ is a loss factor, $\Gamma \in [0, 1]$, and $v_p \in \{V - v_i - v_j\}$, an intermediate node from $v_i$ to $v_j$.

$$w(v_i, v_j) = \begin{cases} w'(v_i, v_j) & w'(v_i, v_j) > w(v_i, v_j) \\ w(v_i, v_j) & \text{otherwise} \end{cases}.$$  

(12)

To more accurately capture the potential relevance and identify all possible weak ties, we need to find all the pairs with the strongest relevance by means of Equation 11 and 12. To be specific, in order to find the maximum $w(v_i, v_j)$, we calculate $w'(v_i, v_j)$ for each intermediate node in the relevance network. It is similar to finding the shortest path among all paths existing in the graph. We exploit Johnson algorithm, which consists of multiple Dijkstra algorithms. It’s particularly efficient for sparse graphs, and is easy to parallelize. Based on Johnson algorithm, we put forward the JohnsonMax algorithm to update the relevance network, and the details are shown in Algorithm 1. Lines 3 to 15 (Function Dijkstra update each row of adjacency matrix $mat$, which can be executed by multiple parallel processes. Note that lines 14 and 15 capture the potential associations between nodes, mitigating the data sparsity and discovering weak ties. Lines 16 and 17 update the whole relevance network.

The time complexity of the proposed JohnsonMax algorithm is $O(|V||R| + |V|^2 \log |R|)$, which is efficient in sparse graphs such as the relevance network in our work.

Algorithm 1 JohnsonMax to update the relevance network

Input: $N$, the original relevance network

Output: $N'$, the updated relevance network with more weak ties

1. let $mat$ be the adjacency matrix of $N$
2. //update the weights of the i-th row
3. function DijkstraMax($mat$)
4. stack $\leftarrow [1, 2, \ldots |V|]$ //the stack of index
5. stack.pop(i)
6. while stack is not empty do
7. //get the max weight from i to each node in stack
8. $maxW, maxW\_index \leftarrow \max(mat(i, V(stack)))$
9. if $maxW == 0$ then
10. break
11. stack.pop($maxW\_index$)
12. for $j$ in stack do
13. $w_{new} \leftarrow (1 - \Gamma) \cdot maxW \cdot mat(maxW\_index, j)$
14. if $w_{new} > mat(i, j)$ then
15. $mat(i, j) \leftarrow w_{new}$
16. for $i$ in $|V|$ do //N', the node set of user-movie graph
17. $mat(i) \leftarrow$ Dijkstra(Max($mat$))

$R(u_i, u_j) = \frac{M(u_i) \cap M(u_j)}{M(u_j)}$,  \hspace{1cm} (7)

$R(m_i, m_j) = \frac{U(m_i) \cap U(m_j)}{U(m_j)}$,  \hspace{1cm} (8)

$R(u_i, m_j) = \frac{S(u_i, m_j)}{Asgn(m_j)}$,  \hspace{1cm} (9)

$R(m_i, u_j) = \frac{S(u_i, m_j)}{Asgn(m_i)}$.  \hspace{1cm} (10)

Figure 4: Examples in building the relevance network. (a) shows the necessity of asymmetrical measures, and (b) indicates the necessity of JohnsonMax.
where GRU leverage the update gate vector representing users’ short-term preferences. We apply GRU instead of LSTM in generating prediction, because GRU reaches the same performance with LSTM at a smaller time cost.

Genre vectors, the output (the last hidden state, \(h_L\)) of RNN, where we select 5) neighbors as an input sequence in Figure 5, and the next vector as a label. Here we introduce a RNN with Gated Recurrent Units (GRU) \([5]\) to predict user preferences in genre:

\[
z_t = \sigma(W_z x_t + U_z h_{t-1}), \quad r_t = \sigma(W_r x_t + U_r h_{t-1}), \quad \hat{h}_t = \text{tanh}(W_h x_t + U(h_{t-1})), \quad h_t = (1 - z_t) h_{t-1} + z_t \hat{h}_t, \tag{16}
\]

where GRU leverage the update gate \(z_t\) and the reset gate \(r_t\) to capture long-term dependencies on user behaviors; the inputs are genre vectors, the output (the last hidden state, \(h_{\text{end}}\)) is a genre vector representing users’ short-term preferences. We apply GRU instead of LSTM into genre prediction, because GRU reach the same performance with LSTM at a smaller time cost.

We train GRU on L samples, each of which is a \(g\)-dimension genre vector. Our goal is to predict higher scores for the true genres and lower scores for the false ones. Hence, we adopt binary cross entropy \([31]\) as the loss function:

\[
\text{Loss}(\hat{Y}, Y) = \sum_{i=1}^{L} \sum_{j=1}^{g} -y \log \hat{y} + (y - 1) \log (1 - \hat{y}). \tag{17}
\]

4.3 Predicting Genres

In serendipity-oriented recommendation systems, there is usually a risk of recommending unrelated movies accompanied with broadening user horizons. To address this issue, we propose to predict genres in line with users’ short-term preferences with RNN.

It’s necessary to make a pre-process for transforming data from original ratings to the input of RNN. First of all, we group movies by users, sort them within group according to rating timestamps, and convert each item of sequences into a genre vector (see Figure 5). Then, we adopt sliding windows to split these Boolean vectors and convert each item of sequences into a genre vector (see Figure 5, \(w = 3\), and the next vector as a label.

Here we introduce a RNN with Gated Recurrent Units (GRU) \([5]\) to predict user preferences in genre:

\[
z_t = \sigma(W_z x_t + U_z h_{t-1}), \quad r_t = \sigma(W_r x_t + U_r h_{t-1}), \quad \hat{h}_t = \text{tanh}(W_h x_t + U(h_{t-1})), \quad h_t = (1 - z_t) h_{t-1} + z_t \hat{h}_t, \tag{16}
\]

where GRU leverage the update gate \(z_t\) and the reset gate \(r_t\) to capture long-term dependencies on user behaviors; the inputs are genre vectors, the output (the last hidden state, \(h_{\text{end}}\)) is a genre vector representing users’ short-term preferences. We apply GRU instead of LSTM into genre prediction, because GRU reach the same performance with LSTM at a smaller time cost.

We train GRU on L samples, each of which is a \(g\)-dimension genre vector. Our goal is to predict higher scores for the true genres and lower scores for the false ones. Hence, we adopt binary cross entropy \([31]\) as the loss function:

\[
\text{Loss}(\hat{Y}, Y) = \sum_{i=1}^{L} \sum_{j=1}^{g} -y \log \hat{y} + (y - 1) \log (1 - \hat{y}). \tag{17}
\]

4.4 Recommendation with Elastic Serendipity

In the above subsections we gain the elasticity of users and movies, build the relevance network, and predict potential movie genres. Based on them, we will generate final recommendations.

We first filter options, keeping only those movies in the genres that GRU predict. Then, we calculate user-movie elasticity through Equation 6 and the relevance by Equation 9. Next, we combine elasticity and relevance to gain the elastic relevance between \(u_i\) and \(m_j\), \(RE(u_i, m_j)\), as follows:

\[
RE(u_i, m_j) = \frac{R(u_i, m_j) + \lambda * E(u_i, m_j)}{1 + \lambda}, \tag{18}
\]

where \(\lambda\) denotes the contribution of the elasticity \(E(u_i, m_j)\).

For broadening user horizons, we try to recommend movies relatively different from their history. In addition, we also expect to decrease unacceptable recommendations, so as to increase user satisfaction. Hence, movies with median elastic relevance factors are suitable ones for recommendation, while the smaller are unacceptable for users and the bigger may cause over-specialization. We propose a new concept of the recommendation factor, \(R_fact(u_i, m_j)\),

\[
R_fact(u_i, m_j) = |RE(u_i, m_j) - \text{Avg}(RE(U, M))|, \tag{19}
\]

where \(\text{Avg}(RE(U, M))\) is the average value of elastic relevance between all users and movies. Finally we recommend movies in the ascending order of the recommendation factor.

The intuition of Equation 19 is that when there is a strong elasticity between \(u_i\) and \(m_j\), it’s desirable to recommend ones with weak relevance to \(u_i\). There is a complementary relationship between the relevance and the elasticity in the serendipity-oriented recommendation \([18]\). Taking Figure 1(b) for instance, \(m_1\) is further away from \(u_i\) (indicating weaker relevance), than the other two recommendations, \(m_2\) and \(m_4\). However, the elasticity of \(m_1\) is bigger, indicating \(m_1\) is easier to accept for users (even \(m_1\) is somewhat unfamiliar to them). Thus, \(m_1\) also is a serendipitous recommendation for \(u_i\).

5 EXPERIMENTS

We evaluate our method based on two large-scale real-world datasets, MovieLens-1m (ml-1m) \([8]\) and MovieLens-latest-small (ml-latest-small) \([8]\), where we take the top 80% ratings as the training set and the rest as the testing set in chronological order. The statistics of datasets are shown in Table 4. Our experiments focus on the following issues:

<table>
<thead>
<tr>
<th>Item</th>
<th>ml-1m</th>
<th>ml-latest-small</th>
</tr>
</thead>
<tbody>
<tr>
<td># users</td>
<td>6040</td>
<td>610</td>
</tr>
<tr>
<td># movies</td>
<td>3900</td>
<td>9742</td>
</tr>
<tr>
<td># ratings</td>
<td>100209</td>
<td>100836</td>
</tr>
<tr>
<td>movie release time (year)</td>
<td>1919-2000</td>
<td>1902-2018</td>
</tr>
<tr>
<td>rating time (year)</td>
<td>2000-2003</td>
<td>1996-2018</td>
</tr>
</tbody>
</table>

Table 4: Statistics of datasets.
Problem 1. In this paper, we measure the serendipity with content difference and genre accuracy. Then, is it necessary to keep accuracy in genre, and how can we achieve it?

Problem 2. We propose the JohnsonMax algorithm to update the relevance network based on the influence propagation theory. Then, what effect has been achieved by JohnsonMax in mitigating data sparsity and building weak ties?

Problem 3. How does the proposed approach perform compared with other methods?

5.1 Verification on Problem 1

For Problem 1: whether it is necessary to keep accuracy in genre, and how to achieve it, we conduct the following experiments to demonstrate the necessity of genre accuracy, and adopt GRU to accurately predict user short-term preferences in movie genres.

5.1.1 Checking the necessity of genre accuracy. We visualize the distribution of movie genres for several individuals in chronological order. In order to make the results of visualization more reliable, we select four representative types of users to analyze: (1) users with a large number of related movies and wide interests in genre; (2) users who rate lots of movies in relatively single genres; (3) users related to a limited number of movies but with wide interests in genre and (4) users with few related movies in limited genre. We choose a representative user for each type, as shown in Figure 6 (a) (b) (c) (d) respectively.

We gain two main findings: (1) local consistency. Although it’s almost impossible to capture overall preferences of users, we find it easy to predict genres of users’ interest in the given context (at a fixed value of the X axis). (2) global dynamics. The dynamics of genre preferences widely exist in all users, not depending on the number of relevant movies and the scope of users’ interest. Due to local consistency in users’ preference, it’s essential to guarantee the accuracy in genre to minimize unrelated recommendations. Considering global dynamics, the sequence model (e.g. GRU, LSTM) is a good choice to identify the movie genre that users may prefer.

5.1.2 Checking the effects of parameters. To improve the performance of genre prediction with GRU, we vary the length of inputs and the type of filters to check their effects. We adopt micro F1 [33], micro precision [33], and hamming distance [33] as evaluation metrics. The results are shown in Figure 7 and Figure 8, respectively.

For the length of sequences, we find that it’s hardly possible to train a good model for predicting genres based on very short sequences (e.g., the length is within 10). This is because user preferences have a long-term dependence on the past behaviors. However, the performance wouldn’t continue to be enhanced as the length of the sequence increases, due to the local consistency of user preferences. To make a trade-off, we use 20 as the length of input sequences, which is optimal with a comprehensive consideration of all metrics (i.e., micro F1, micro precision, and hamming distance).

We check the effects of filters, topK method and threshold filtering, in Figure 8. It indicates that the performance of genre prediction with threshold filtering fluctuates greatly, while topK is relatively stable. Possible reason for the fluctuation may be that the amount of positive predictions varies greatly in the prediction data; however, it is relatively stable in the label data. The fluctuation degree of threshold filtering performance is much stronger than that of topK, although the threshold filtering (t=0.3) has a slightly better performance than topK (k=2). Thus, we adopt topK method as the filter and use k=2 in the following experiments.

5.2 Verification on Problem 2

In this section, we verify Problem 2: what effect has been achieved by JohnsonMax in mitigating data sparsity and building weak ties.

We visualize the distribution of weights originally existing in relevance network and updated by JohnsonMax in Figure 9. We have two main findings: (1) the JohnsonMax algorithm increases graph density\(^4\) from 0.47 to 0.74, indicating its effectiveness in mitigating data sparsity; (2) the weights updated by JohnsonMax are distributed in (0, 0.3) and concentrated on (0, 0.1), indicating that it captures weak ties in the relevance network. Therefore, it meets the requirements of digging out movies related but not limited to users’ histories in serendipity-oriented recommendation. Moreover, it is simple to implement, without the aid of auxiliary information.

5.3 Verification on Problem 3

We compare HAES model with some other methods on content difference and genre accuracy to verify Problem 3.

5.3.1 Baselines. Since there is no agreement on the definition of serendipity, researchers propose different algorithms for various purposes in line their definitions, which restricts the comparison among different serendipity-oriented methods. In this subsection, we adopt some widely used methods as benchmarks:

\[^4\text{graph density} = \frac{|E|}{|V|^2}\]
5.3.2 Evaluation Metrics. From the definition of serendipity above, genre accuracy and content difference, we select four metrics: micro_F1 and average hamming distance (denoted as avg_hamming) as metrics for the genre accuracy; difference from histories (denoted as difference) and diversity in recommendation lists (denoted as diversity) as metrics for the content difference.

**micro_F1.** We adopt micro_F1, a metric in multi-label prediction [33], to measure the recall, micro_r, and precision, micro_p, of recommendations comprehensively. Suppose G is a genre vector, \( \sum(G_i) \) is the sum of all elements in \( G_i \), \( G \) is a label set of genre vectors, \( \hat{G} \) is the predicted one, we calculate micro_F1 on a samples:

\[
\text{micro}_r(\hat{G}, G) = \frac{1}{a} \sum_{i=1}^{a} \frac{\text{sum}(\hat{G}_i + G_i)}{\text{sum}(G_i)},
\]

\[
\text{micro}_p(\hat{G}, G) = \frac{1}{a} \sum_{i=1}^{a} \frac{\text{sum}(\hat{G}_i \cap G_i)}{\text{sum}(G_i)},
\]

\[
\text{micro}_F1(\hat{G}, G) = \frac{2 \times \text{micro}_r(\hat{G}, G) \times \text{micro}_p(\hat{G}, G)}{\text{micro}_r(\hat{G}, G) + \text{micro}_p(\hat{G}, G)}
\]

**avg_hamming.** avg_hamming considers not only positive samples but also negative ones. Suppose \( G_i \) is a q-dimension genre vector, we calculate avg_hamming on a samples as follows [33]:

\[
\text{avg}_\text{hamming}(\hat{G}, G) = \frac{1}{a + q} \sum_{i=1}^{a+q} \text{sum}(\hat{G}_i \oplus G_i).
\]

**difference.** Based on the associations between users and movies, we calculate the difference, difference(\( u_t, M_t \)), between the target
Table 5: Experimental results of different methods on content difference (MovieLens-1m). "HAES-NE" refers to our approach without the elasticity component, and "HAES-NG" is the one without genre prediction.

<table>
<thead>
<tr>
<th>method</th>
<th>diff@5</th>
<th>diff@10</th>
<th>diff@15</th>
<th>div@5</th>
<th>div@10</th>
<th>div@15</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACC</td>
<td>0.2255</td>
<td>0.2511</td>
<td>0.2681</td>
<td>0.7037</td>
<td>0.714</td>
<td>0.71</td>
</tr>
<tr>
<td>NOV</td>
<td>0.7078</td>
<td>0.7289</td>
<td>0.7351</td>
<td>0.5995</td>
<td>0.6872</td>
<td>0.649</td>
</tr>
<tr>
<td>POP</td>
<td>0.5571</td>
<td>0.5571</td>
<td>0.5562</td>
<td>0.2118</td>
<td>0.2472</td>
<td>0.2376</td>
</tr>
<tr>
<td>RAND</td>
<td>0.7428</td>
<td>0.7435</td>
<td>0.7438</td>
<td>0.7229</td>
<td>0.7229</td>
<td>0.7229</td>
</tr>
<tr>
<td>HAES-NE</td>
<td>0.5108</td>
<td>0.5186</td>
<td>0.5243</td>
<td>0.5209</td>
<td>0.5294</td>
<td>0.535</td>
</tr>
<tr>
<td>HAES-NG</td>
<td>0.7973</td>
<td>0.8062</td>
<td>0.8098</td>
<td>0.7624</td>
<td>0.7479</td>
<td>0.7408</td>
</tr>
<tr>
<td>HAES</td>
<td>0.8134</td>
<td>0.82</td>
<td>0.8242</td>
<td>0.683</td>
<td>0.6701</td>
<td>0.6606</td>
</tr>
</tbody>
</table>

Table 6: Results on genre accuracy (MovieLens-1m).

<table>
<thead>
<tr>
<th>method</th>
<th>F1@5</th>
<th>F1@10</th>
<th>F1@15</th>
<th>h@5</th>
<th>h@10</th>
<th>h@15</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACC</td>
<td>0.2059</td>
<td>0.2036</td>
<td>0.2006</td>
<td>0.1577</td>
<td>0.1601</td>
<td>0.162</td>
</tr>
<tr>
<td>NOV</td>
<td>0.1964</td>
<td>0.1913</td>
<td>0.2127</td>
<td>0.1759</td>
<td>0.1684</td>
<td>0.1677</td>
</tr>
<tr>
<td>RAND</td>
<td>0.2065</td>
<td>0.2034</td>
<td>0.2011</td>
<td>0.2351</td>
<td>0.2221</td>
<td>0.2179</td>
</tr>
<tr>
<td>HAES-NE</td>
<td>0.3072</td>
<td>0.3035</td>
<td>0.3019</td>
<td>0.1604</td>
<td>0.1699</td>
<td>0.1614</td>
</tr>
<tr>
<td>HAES-NG</td>
<td>0.1887</td>
<td>0.1881</td>
<td>0.1811</td>
<td>0.1628</td>
<td>0.1644</td>
<td>0.1656</td>
</tr>
<tr>
<td>HAES</td>
<td>0.3149</td>
<td>0.3119</td>
<td>0.3122</td>
<td>0.1482</td>
<td>0.1506</td>
<td>0.1514</td>
</tr>
</tbody>
</table>

Table 7: Results on content difference (MovieLens-latest-small).

<table>
<thead>
<tr>
<th>method</th>
<th>diff@5</th>
<th>diff@10</th>
<th>diff@15</th>
<th>div@5</th>
<th>div@10</th>
<th>div@15</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACC</td>
<td>0.2917</td>
<td>0.3036</td>
<td>0.3131</td>
<td>0.2346</td>
<td>0.2414</td>
<td>0.2464</td>
</tr>
<tr>
<td>NOV</td>
<td>0.5803</td>
<td>0.7353</td>
<td>0.5916</td>
<td>0.722</td>
<td>0.4351</td>
<td>0.7307</td>
</tr>
<tr>
<td>RAND</td>
<td>0.1951</td>
<td>0.3131</td>
<td>0.2079</td>
<td>0.3045</td>
<td>0.1781</td>
<td>0.31</td>
</tr>
<tr>
<td>HAES-NE</td>
<td>0.6836</td>
<td>0.7028</td>
<td>0.6826</td>
<td>0.7005</td>
<td>0.6839</td>
<td>0.7035</td>
</tr>
<tr>
<td>HAES-NG</td>
<td>0.4831</td>
<td>0.5163</td>
<td>0.4917</td>
<td>0.5064</td>
<td>0.4644</td>
<td>0.512</td>
</tr>
<tr>
<td>HAES</td>
<td>0.6706</td>
<td>0.6652</td>
<td>0.6663</td>
<td>0.6654</td>
<td>0.6774</td>
<td>0.6653</td>
</tr>
</tbody>
</table>

5.3.3 Results. We compare HAES with baselines on micro_F1(F1), avg_hamming(h), diff( divorrence), and diversity(div). The results on MovieLens-latest-small are shown in Table 5 and Table 6, and those on MovieLens-latest-smallest are shown in Table 7 and Table 8.

Comparison on content difference. HAES achieves the best performance on MovieLens-1m (see Table 5), e.g., it even increases 2.3 times on difference compared with ACC. However, random-based (i.e., RAND) and novelty-based (i.e., NOV) method perform better on MovieLens-latest-small (see Table 7). The reason may be that the timespan of ratings on it is twenty-two years, which is too long to build weak ties for HAES. But the long span makes it easy to generate recommendations without relevance. To verify the above impact of the long timespan, we analyze all rating timestamps on MovieLens-latest-small in Figure 10. It indicates that most users provide ratings only within a limit timespan (the average span is 0.64 year), i.e., the longer the span is, the less the interaction of users is. Another finding is that HAES-NE, HAES without the elasticity component, performs poorer than HAES on both datasets, indicating the importance of the elasticity in broadening user horizons.

Comparison on genre accuracy. HAES and HAES-NE, our approaches with the genre prediction component, achieve significant improvements on genre accuracy (see Table 6 and Table 8). Particularly, HAES improves the micro_F1 by 59.93% compared with ACC on MovieLens-latest-small. It has a close association with the fact that we capture user short-term preferences with GRU, while the others either acquire user preferences as a whole or neglect user interests. In addition, we find that methods with consideration of global preferences (e.g., ACC) even are inferior to the random recommendation (RAND) on Movielens-latest-small (see Table 8). The reason may be that user global preferences can’t represent their short-term demands, while they also limit the range of potential recommendations.

5.4 Summary of Experiments

In summary, we have the following findings in the experiments. (1) The threshold filtering method would cause the fluctuation of prediction performance, because the amount of the positive predictions varies greatly. (2) On MovieLens-latest-small, random-based (RAND) and novelty-based (NOV) approaches have a better performance on genre accuracy, than the one based on accuracy (ACC). One possible reason is that global preferences of users can’t represent their short-term demands, while they also limit the range of potential recommendations. (3) Overall, our approach (HAES)
achieves significant improvements, e.g., it improves the micro\_F1 by 53.93% and increases 2.3 times on \textit{difference} compared with the accuracy-oriented method (ACC) on MovieLens-1m.

## 6 CONCLUSIONS AND FUTURE WORK

In this paper, we propose a hybrid approach for recommending movies with elastic serendipity, based on a more objective definition of serendipity. We propose and quantify the elasticity in recommendation system, for meeting users’ varying demands for serendipity and accuracy. We put forward asymmetric measures to more accurately capture associations among users and movies, and then present JohnsonMax to mitigate the data sparsity and build weak ties. We introduce GRU to acquire users’ short-term dynamics. In this paper, we propose a hybrid approach for recommending movies with elastic serendipity, based on a more objective definition of serendipity. We propose and quantify the elasticity in recommendation system, for meeting users’ varying demands for serendipity and accuracy. We put forward asymmetric measures to more accurately capture associations among users and movies, and then present JohnsonMax to mitigate the data sparsity and build weak ties. We introduce GRU to acquire users’ short-term dynamics.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we propose a hybrid approach for recommending movies with elastic serendipity, based on a more objective definition of serendipity. We propose and quantify the elasticity in recommendation system, for meeting users’ varying demands for serendipity and accuracy. We put forward asymmetric measures to more accurately capture associations among users and movies, and then present JohnsonMax to mitigate the data sparsity and build weak ties. We introduce GRU to acquire users’ short-term dynamics. In this paper, we propose a hybrid approach for recommending movies with elastic serendipity, based on a more objective definition of serendipity. We propose and quantify the elasticity in recommendation system, for meeting users’ varying demands for serendipity and accuracy. We put forward asymmetric measures to more accurately capture associations among users and movies, and then present JohnsonMax to mitigate the data sparsity and build weak ties. We introduce GRU to acquire users’ short-term dynamics.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we propose a hybrid approach for recommending movies with elastic serendipity, based on a more objective definition of serendipity. We propose and quantify the elasticity in recommendation system, for meeting users’ varying demands for serendipity and accuracy. We put forward asymmetric measures to more accurately capture associations among users and movies, and then present JohnsonMax to mitigate the data sparsity and build weak ties. We introduce GRU to acquire users’ short-term dynamics. In this paper, we propose a hybrid approach for recommending movies with elastic serendipity, based on a more objective definition of serendipity. We propose and quantify the elasticity in recommendation system, for meeting users’ varying demands for serendipity and accuracy. We put forward asymmetric measures to more accurately capture associations among users and movies, and then present JohnsonMax to mitigate the data sparsity and build weak ties. We introduce GRU to acquire users’ short-term dynamics.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we propose a hybrid approach for recommending movies with elastic serendipity, based on a more objective definition of serendipity. We propose and quantify the elasticity in recommendation system, for meeting users’ varying demands for serendipity and accuracy. We put forward asymmetric measures to more accurately capture associations among users and movies, and then present JohnsonMax to mitigate the data sparsity and build weak ties. We introduce GRU to acquire users’ short-term dynamics. In this paper, we propose a hybrid approach for recommending movies with elastic serendipity, based on a more objective definition of serendipity. We propose and quantify the elasticity in recommendation system, for meeting users’ varying demands for serendipity and accuracy. We put forward asymmetric measures to more accurately capture associations among users and movies, and then present JohnsonMax to mitigate the data sparsity and build weak ties. We introduce GRU to acquire users’ short-term dynamics. In this paper, we propose a hybrid approach for recommending movies with elastic serendipity, based on a more objective definition of serendipity. We propose and quantify the elasticity in recommendation system, for meeting users’ varying demands for serendipity and accuracy. We put forward asymmetric measures to more accurately capture associations among users and movies, and then present JohnsonMax to mitigate the data sparsity and build weak ties. We introduce GRU to acquire users’ short-term dynamics.