Variational Autoencoder-based Hybrid Recommendation with Poisson Factorization for Modeling Implicit Feedback

IWAO TANUMA¹, TOMOKO MATSUI.² (Senior Member, IEEE)

¹Department of Statistical Science School of Multidisciplinary Sciences, The Graduate University for Advanced Studies, Tokyo, 190-8562, Japan
²Department of Statistical Modeling, The Institute of Statistical Mathematics, Tokyo 190-8562, Japan

Corresponding author: Iwao Tanuma (e-mail: tanuma@ism.ac.jp).

ABSTRACT Hybrid recommendation, which is based on collaborative filtering and supplemented with auxiliary content information, is being actively researched due to its ability to overcome the cold-start problem. Many proposed hybrid methods make recommendations using Gaussian distribution-based collaborative filtering even though they handle variables that tend to be non-Gaussian, such as the number of interactions. We present a method that uses a hybrid recommendation framework based on collaborative filtering that models the number of interactions as a Poisson-distributed and variational autoencoder-based content information generation process that shares latent variables with collaborative filtering. As a prior for the shared latent variables, we use a gamma distribution, which is a conjugate prior of a Poisson distribution. An implicit-derivative-based reparameterization trick enables the use of a gamma distribution in a variational autoencoder. The latent variables in the generative model are inferred using the stochastic gradient variational Bayes approach, taking the number of interactions corresponding to users and items and content information as input. In accordance with the inference, unobserved interactions between users and items are predicted for recommendation. The use of a neural-network-based generative model for content information enables the framework to handle various types of content information. Experimental results show that the proposed method utilizes content information effectively for predicting the number of interactions and that it should aid in overcoming the cold-start problem.

INDEX TERMS Collaborative filtering, Implicit feedback, Matrix factorization, Neural networks, Recommendation system

I. INTRODUCTION

The growth of web services and applications has enabled access to a huge variety of content such as articles, music, movies, and games. Recommendation systems for finding items of interest for users from the huge amount of information have become an essential technology for various services. In addition, the number of interactions between users and items, such as the number of views and play count, is becoming increasingly important as a key performance indicator for not only service providers but also content creators, especially for advertising and music/video streaming services. The number of interactions with an item is strongly related to the interest in the item itself. Hence, predicting the number of interactions with an unknown item by a user is closely related to recommendation problems.

One of the most representative approaches for recommendation systems is based on collaborative filtering (CF) [1]–[6]. CF-based recommendation systems typically use a user-item matrix composed of feedback from each user for each item. The feedback can be explicit feedback (e.g., rating) or implicit feedback (e.g., number of interactions) [7]. Explicit feedback requires the user to provide additional input while implicit feedback requires only that the service be used. In either case, a matrix decomposition-based technique is usually used to fill the matrix elements with low-dimensional variables describing the characteristics of the users and items.

An alternative approach is the content-based (CB) approach. CB-based recommendation systems predict matrix elements from content information such as a description of the item itself or its metadata [8]. Generally, if each item has
been interacted with by a certain number of users, the CF approach works well [9]. However, for an item that has not received any user feedback, the CF approach does not work well due to the lack of observations (the “cold-start problem” [10]). Hybrid approaches combining CF and CB, which can overcome this weakness, are thus being actively researched [11]–[15]. By learning the correspondence between the latent expression of items and the content information, the system can acquire a latent representation even for an item that has never been selected.

As noted above, the number of interactions is an example implicit feedback variable. It is an element of the matrix used for CF input and is often modeled as a Poisson or multinomial distribution in statistics. However, the major approaches used for deriving implicit feedback treat interactions as if they are Gaussian distributed, in other words, deriving them by least squares error approximation [2], [13], [14].

In this paper, we present a hybrid recommendation method based on building a CF model that models the number of interactions as Poisson distributed. The use of a neural network-based generative model for content information enables the method to handle various types of content information.

A list of the main contribution of this paper is summarized as follows:

- With content generation model using VAE based on a gamma distribution, which is a conjugate prior distribution of Poisson distribution, we extend of Poisson distributed collaborative filtering to hybrid model.
- Experimental results show effectiveness of our proposed method for implicit feedback.
- We show that the framework comprising the proposed method is applicable to a broader range of collaborative filtering, including state-of-the-art methods, and can be extended to a hybrid recommendation algorithm that handles the number of interactions.

The rest of this paper is structured as follows. First, related works are introduced in section 2, then section 3 provides a detailed description and discussion of the extension of the proposed method, and the experimental results are discussed in section 4. Finally, section 5 provides concluding remarks.
using the standard variational Bayesian method [26] without losing conjugacy.

Also being actively studied is the use of information incorporation combined with deep learning for supporting various types of content information such as images and acoustic features (Fig. 1). For example, a collaborative variational autoencoder (CVAE) [12], a generative model for hybrid recommendation, shares latent variables between content corresponding to each item itself and its components for CF. Similar to the operation of a variational autoencoder (VAE) [27], a CVAE generates content information from latent variables and simultaneously biases the item components used for CF by adding latent variables.

Except for CTPF, which can only handle simpler models for content information, collaborative filtering in most hybrid recommendations consists of Gaussian-based objective functions. To the best of our knowledge, the proposed method is the first hybrid recommendation method that uses Poisson distribution for implicit feedback and neural networks for handling diverse content information.

III. PROPOSED METHOD

For a more accurate fitting to the number of interactions, we use PF for CF and construct a hybrid recommendation framework with a probabilistic formulation that, like a CVAE, jointly describes a VAE-based content information generation process that shares latent variables with CF. As a prior for the shared latent variables, we use a gamma distribution, which is a conjugate prior of a Poisson distribution. The use of an implicit-derivative-based reparameterization trick [28], [29] enables the use of a gamma distribution. The latent variables in the generative model are inferred using the stochastic gradient variational Bayes (SGVB) method [27], taking the number of interactions corresponding to users and items and content information as input.

An overview of the proposed method is shown in Fig. 2. Next, we describe the problem setup, generative model, and algorithm used for inference and estimation.

A. PROBLEM SETUP

We formulate a recommendation problem for implicit feedback. The user index and item index are denoted as \( i \in \{0, \ldots, I - 1\} \) and \( j \in \{0, \ldots, J - 1\} \), respectively. The implicit feedback matrix is denoted as \( \mathbf{Y} \in \mathbb{N}^{I \times J} \), element \( y_{i,j} \) corresponds to the number of interactions between user \( i \) and item \( j \). The content information corresponding to item \( j \) is denoted as \( \mathbf{x}_j \). The recommendation problem is defined as how to acquire item subset \( J_i' \subseteq \{0, \ldots, J - 1\} \), which includes items with which user \( i \) is likely to interact, given implicit feedback matrix \( \mathbf{Y} \) and content information \( \mathbf{X} = \{\mathbf{x}_0, \ldots, \mathbf{x}_{J-1}\} \). The notations used in the proposed method are shown in Table 1.

B. GENERATIVE MODEL

We explain the processes for generating implicit feedback and generating content information, starting with their latent factors.

For each user \( i \), each element of the \( K \)-dimensional latent variable \( \mathbf{u}_i \) is drawn from a gamma distribution,

\[
\mathbf{u}_{i,k} \sim \text{Gamma}(a_{shp}^u, b_{rte}^u), \quad (1)
\]

where \( a_{shp}^u \) and \( b_{rte}^u \) are hyperparameters, and \( \text{Gamma}(a, b) \) is the gamma distribution characterized by shape parameter \( a \) and rate parameter \( b \). For each item \( j \), each element of the \( K \)-dimensional latent variables \( \mathbf{v}_j \) and \( \mathbf{z}_j \) is drawn from a corresponding gamma distribution,

\[
\mathbf{v}_{j,k} \sim \text{Gamma}(a_{shp}^v, b_{rte}^v), \quad (2)
\]

\[
\mathbf{z}_{j,k} \sim \text{Gamma}(a_{shp}^z, b_{rte}^z), \quad (3)
\]

where \( a_{shp}^v, b_{rte}^v, a_{shp}^z, b_{rte}^z \) are hyperparameters.

Latent variable mapping to content information is represented by \( \mathbf{z}_j \) and is used to generate content information. In other words, content information \( \mathbf{x}_j \) is drawn from probabilistic distribution \( p(\mathbf{x}_j | \mathbf{z}_j) \), which is parameterized by the outputs of nonlinear function \( f_{\phi_{dec}}(\mathbf{z}_j) \) (e.g., a neural network), where \( \phi_{dec} \) is the set of parameters of \( f_{\phi_{dec}}(\cdot) \).

Such a setup is often used in VAEs; the form of \( p(\mathbf{x}_j | \mathbf{z}_j) \) depends on the nature of the content information. Here, we discuss a specific case, music piece recommendation, in which the play count as implicit feedback and the content features represented as a fixed-length vector are provided for each piece. Our method is applicable not only to a fixed-length feature vector but also to images, texts, and their combinations by using a VAE as the generative model for \( \mathbf{x}_j \). The generative process for \( F \)-dimensional feature \( \mathbf{x}_j \) and its parameters is represented as

\[
\mu_j = f_{\phi_{dec}}(\mathbf{z}_j) \quad (4)
\]

\[
\mathbf{x}_j \sim N(\mu_j, \mathbf{I}_F), \quad (5)
\]

where \( \mathbf{x}_j \in \mathbb{R}^F \), and \( \mu_j \in \mathbb{R}^F \).
Next, we describe the generative process for implicit feedback, namely the count of user-item interactions. As mentioned above, there are two latent variables corresponding to items, \( z_j \) and \( v_j \). The item component is represented by their sum, \( z_j + v_j \). The feedback matrix is thus drawn from a Poisson distribution characterized by the inner product \( \sum_{i,j} y_{i,j} \), similar to that in collaborative filtering methods. Maximizing the evidence lower bound (ELBO) is equivalent to minimizing the Kullback–Leibler divergence between the prior \( q(z_j) \) and the posterior distribution \( q(z_j|x_j) \) of \( z_j \), where \( z_j \) is the sample index \( (l = 1, ..., L) \). In the SGVB method, the derivative of the objective function with respect to \( \phi \) and \( \phi^{dec} \) can be calculated in order to minimize the objective function with respect to \( \phi \) and \( \phi^{dec} \).

Although a conventional VAE obtains derivatives of samples from Gaussian distributions by using a reparameterization trick, various methods have been developed to calculate derivatives for samples from various distributions in order to accommodate different applications [28], [30]. For a gamma distribution, an implicit reparameterization gradient [28] can be used. Using implicit differentiation with a cumulative distribution function enables an unbiased estimator of the derivative of the samples to be calculated for any continuous distribution by using a cumulative distribution function that can be calculated numerically.
method is necessary only for $z$. The objective function for MAP estimation is
\[
\mathcal{L}_{MAP}(\mathbf{Y}, \mathbf{X}, \mathbf{u}, \mathbf{v}, z) = \sum_{i,j} \log p(y_{i,j}|u_i, v_j, E_q[z_{j,k}]) + \sum_j \log p(u_j) + \text{const},
\]
(12)
where $a^u_{shp}$ and $b^u_{te}$ and the update equations for $u$ and $v$ are derived using the auxiliary function method [31]:
\[
u_{i,k} \leftarrow a^u_{shp} \cdot u_{i,k} + \sum_j C_{i,j} v_{j,k} + \sum_j E_q[z_{j,k}],
\]
(13)
\[
v_{j,k} \leftarrow a^v_{shp} \cdot v_{j,k} + \sum_i C_{i,j} u_{i,k} + \sum_i E_q[z_{i,j}].
\]
(14)

We note that the hybridization does not increase the computational complexity in the collaborative filtering. Comparing the update equations of the proposed method with those of conventional Poisson factorization, the only difference is the addition of the term $E_q[z_{j,k}]$. Therefore, the computational complexity of each update equation is the same with Poisson factorization, $O(IK)$ for (13) and $O(IKj)$ for (14), and since they are updated $IK$ and $IKj$ times for $u_{i,k}$ and $v_{j,k}$ respectively, the computational complexity per iteration is $O(IKj^2)$.

Although MAP estimation was used for fair comparison with prior research and simplification of the model, variational Bayesian iteration equations can also be derived [25], [32].

As $z_{j,k}, u_i$, and $v_j$ are mutually dependent, inference and estimation are performed by updating SGVB method to decrease (12) and by iteration equations (13) and (14) in turn (Algorithm 1). After convergence, the estimation of $y_{i,j}$ can be calculated using
\[
y_{i,j} = u_i(v_j + E_q[z_{j,k}]),
\]
(15)
from $y^*_{i,j}$, the count of user-item interactions is predicted, and the Top-$N$ items for each user is presented.

**D. VAE ARCHITECTURE FOR GAMMA DISTRIBUTION**
In the VAE architecture of the proposed method, $f_{\phi^{dec}}(\cdot)$, $a_{\phi}(\cdot)$, and $b_{\phi}(\cdot)$ are composed by a neural network such as a multilayer perceptron network, and the structure can be flexibly selected in accordance with the content information, $z_{j,k}$. For example, convolutional networks [33] and recurrent networks [34]–[36] are respectively used for image data and series data such as text [37]–[39].

Since $a_{\phi}(\cdot)$ and $b_{\phi}(\cdot)$ are parameters of a gamma distribution, they have to have positive values. We meet this requirement by applying an exponential function to the outputs of the neural network corresponding to the two parameters. This is similar to the procedure used for adjusting the variance parameter of a VAE with a Gaussian distribution.

In contrast, there is no restriction on $f_{\phi^{dec}}(\cdot)$. However, since its input, a posterior sample of $z_{j,k}$, is a component of the number of interactions, and since it is a sample from a gamma distribution, it has positive values and thus can have non-Gaussian properties. In general, preprocessing input data to make them Gaussian is beneficial in terms of stability and performance. Therefore, instead of using $z_{j,k}$, we used $\log z_{j,k}$ as input for $f_{\phi^{dec}}(\cdot)$. This can be interpreted as a special case of the Box-Cox transformation [40] in which $\log z_{j,k}$ is transformed into data that are more Gaussian.

**E. THE GENERAL FORM FOR GAMMA-POISSON HYBRID RECOMMENDATION ARCHITECTURE**
We have described the proposed method using a Poisson factorization-based CF up to this point. In this subsection, we discuss its extension to a broader class of CF. For simplicity, we do not consider regularization terms for $u_i$ and $v_j$ such as prior distributions. We assume that all functions treated in this subsection are differentiable.

First, we consider a general form of objective function of CF based on matrix factorization. It is generally described as follows:
\[
\sum_{i,j} d(y_{i,j}, \alpha_{\Omega_{i,j}}(u_i, v_j)),
\]
(16)
where $d(\cdot, \cdot)$ is a metric function and $\alpha_{\Omega_{i,j}}(\cdot, \cdot)$ is a function that is parameterized by elements of a parameter set $\Omega_{i,j}$ and linear for each input. In the case of Gaussian-based matrix factorization for example, $d(\cdot, \cdot)$ is a squared error, $\alpha_{\Omega_{i,j}}(\cdot, \cdot)$ is inner product, and $\Omega_{i,j} = \emptyset$ for all pairs of $(i,j)$. In addition, many of the matrix factorization-based CF with such technique as weight reduction for unobserved elements [11] that are used in (6) and removal of exposure bias [17], [20] can be represented in this form.

Second, we consider CF using neural networks. To begin from a simple extension, we consider that the latent variables $u_i$ and $v_j$ are the outputs of neural networks with the $i$-th row and $j$-th column of $\mathbf{Y}$, represented as $u_{i,:}$ and $v_{j,:}$, as inputs. Namely, latent variables $u_i$ and $v_j$ in (16) are replaced with $u(y_{i,:})$ and $v(y_{j,:})$ respectively, i.e.,
\[
\sum_{i,j} d(y_{i,j}, \alpha_{\Omega_{i,j}}(u(y_{i,:}), v(y_{j,:}))),
\]
(17)
where $u(\cdot)$ and $v(\cdot)$ are nonlinear functions (e.g., neural network). This is the form of deep matrix factorization [4].

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
In addition, another common architecture for collaborative filtering using neural networks is by using only \( y_{i,:} \) or \( y_{:,j} \) as input. It can be considered as a special case of (17), where \( \alpha_{\Omega_{i,j}}(u_{i,:}, v_{j,:}) \) is regarded as a function that takes only \( u_{i,:} \) or \( v_{j,:} \) as input. Many of neural network-based collaborative filtering takes this form [3]–[5], [18], [19].

Finally, we extend the CF discussed above to a hybrid version with our framework. We add the latent variable of content \( z_j \) generated from a gamma distribution as in (3) as follows:

\[
\sum_{i,j} d(y_{i,j}, \alpha_{\Omega_{i,j}}(u_{i,:}, z_j + v_{j,:})), \tag{18}
\]
\[
\sum_{i,j} d(y_{i,j}, \alpha_{\Omega_{i,j}}(u_{i,:}, z_j + v_{j,:})), \tag{19}
\]

For example, let us assume \( \Omega_{i,j} = \{C_{i,j}\} \), \( \alpha_{\Omega_{i,j}}(u_{i,:}, z_j + v_{j,:}) = C_{i,j} u_{i,:} (z_j + v_{j,:})^T \), and

\[
d(s_1, s_2) = s_1 \log \frac{s_1}{s_2} - s_1 + s_2, \tag{20}
\]

where \( s_1 \) and \( s_2 \) are input variables for \( d(\cdot, \cdot) \). As (20) is called \( I \)-divergence and its minimization is known to be equivalent to Poisson likelihood maximization [41], under these conditions, (18) is equivalent to the proposed method described in this section.

As mentioned above, there are CFs that do not take \( v(y_{:,j}) \) as input, which means that there is no term to add \( z_j \) in (19). In such cases, i.e., methods that take \( u(y_{i,:}) \) as input can be regarded as methods that take \( v(y_{:,j}) \) as input due to the symmetry of \( y_{:,j} \) and \( y_{i,:} \) in most cases, and thus our framework is applicable.

Using \( I \)-divergence for \( d(\cdot, \cdot) \) and adding \( z_j \) drawn from gamma distribution-based VAE, the gamma-Poisson hybrid recommendation architecture can be applied to various CF, not limited to those based on matrix factorization.

IV. EVALUATION

A. DATASET SPECIFICATIONS

We evaluated our proposed method using both simulation and real-world data.

1) Simulation data

To obtain simulation data, we simulated the number of user-item interactions for 1000 users and both 1500 and 1000 items. The variation in the number of items was intended to evaluate the influence of the number of items for learning the correspondence between content information and latent variables. We represented the user factor as \( K \)-vector \( u'_i \) for each user, the item factor as \( K \)-vector \( v'_j \) for each item, and the content factor as \( K \)-vector \( z'_j \) for each item.

\[
u'_i,k \sim Gamma(0.2, 0.5), \tag{21}
\vspace{-20pt}
\]
\[
v'_j,k \sim Gamma(0.2, 0.5), \tag{22}
\vspace{-20pt}
\]
\[
z'_j,k \sim Gamma(0.2, 0.5). \tag{23}
\vspace{-20pt}
\]

The content factor is observed and utilized as the content information. In addition, for each user and item, the exposure variables were represented as

\[
A_{i,j} \sim Bernoulli(p), \tag{24}
\]
where $p$ is a parameter, and $Bernoulli(p)$ is the Bernoulli distribution characterized by $p$.

We then simulated count $y_{u,i}$ as

$$y_{u,i} \sim Poisson(A_{i,j} u_i^T (\lambda z_j^i + (1 - \lambda)v_j^i)),$$

where the constant $\lambda \in [0, 1]$ is used to control how much the content information contributes to the play count. Experiments were conducted for $K = 10$, $p = 0.25$, and $\lambda \in \{0.2, 0.4, 0.6, 0.8\}$.

2) Real-world data

To obtain real-world data, we used the MSD-A dataset [14], which includes users and songs that are a subset of those included in the Million Song Dataset [43], and the Echo Nest Taste Profile Subset, the official user dataset of the Million Song Dataset [44]. The MSD-A dataset provides richer information with 2048-dimensional features corresponding to songs than that for the entire dataset by embedding text information, including biographies of artists and acoustic features of audio samples obtained from the Web. We took information only for users who had listened to at least 200 songs and only for songs listened to by at least 1000 users. This resulted in a dataset comprising 881 users and 8866 songs.

**B. EXPERIMENTAL CONDITIONS**

We randomly split the data into training, validation, and testing sets at ratios of 70%, 5%, and 25% (dense case) and at ratios of 25%, 5%, and 70% (sparse case) to simulate the cold-start case (lack of observations).

As a baseline, we used the probabilistic matrix factorizations (PMFs) based on Gaussian likelihood [45] and based on Poisson likelihood [31] as objective functions and used the CVAE [12] as a hybrid recommendation method. The PMFs respectively correspond to the CVAE and the proposed method without utilizing content information. Hereafter, the PMF based on Gaussian likelihood will be referred to as “GF,” and the PMF based on Poisson likelihood will be referred to as “PF.”

The mean squared error (MSE) and the normalized discounted cumulative gain (NDCG) with log counts as the relevance were used as evaluation metrics. NDCG is the sum of the relevance measures discounted in accordance with their rankings (DCG), normalized by the DCG of the ideal recommendation result (IDCG). The NDCG takes values from 0 to 1, with higher values indicating more positive recommendations.

To utilize the MSD-A dataset, we changed the content generation loss of the variational autoencoder in the CVAE from binary cross entropy to Gaussian likelihood (corresponding to (5) in the proposed method). We did this because we used continuous features of the content information in our experiment, whereas bag-of-words features [47], which are discrete, were used in the original work by Li and She [12].

For fair comparison, the CVAE and the proposed method had a common neural network structure (Fig. 3). The encoders had three layers (respectively, 800, 400, and $k$), and the activations were ReLU [48], sigmoid, and linear in that order. The decoders also had three layers (units: 400, 800, $F$) and the same activations. Adam [49] was used as the optimization algorithm for the neural networks, with the learning rate set to 0.0001.

We performed a grid search for each hyperparameter with each method and compared the best results of each method. Early stopping with $\text{patient} = 10$ was used to determine convergence based on the likelihood of the validation set [50].

**C. RESULTS AND DISCUSSION**

First, the performance of each method on the simulation data with 1500 items, $J = 1500$, is shown in Fig. 4: NDCG (top) and MSE (bottom); the sparse case (left) and the dense case (right). The vertical axis represents performance, and the horizontal axis represents $\lambda$, the parameter used to control the contribution of content information.

We compare the results for ordinary collaborative filterings, i.e., GF and PF. PF was superior in terms of both NDCG and MSE in both the sparse case and the dense case. Although content information is not utilized in both methods, there was a slight difference in the metrics when $\lambda$ was changed. This difference, which is not the main point of the discussion, is attributed to the simulation data being expressed as a mixture of components that contribute and do not contribute to the content information. As $\lambda$ approached the center ($\lambda = 0.5$), the number of observations increased slightly. MSE improved as the number of observations increased whereas NDCG worsened as the number of items that made up the ranking increased. This trend was observed for both PF and GF.

We compare the results for the two hybrid recommendation methods, i.e., the proposed method and the CVAE. In the sparse case, both NDCG and MSE improved as $\lambda$ was increased for both methods. This means that the learning process was able to utilize the content information. It is also observed that the proposed method shows the best performance except for the MSE with $\lambda = 0.8$. In the dense case (right side of Fig. 4), the performances of both the proposed method with PF and CVAE with GF were almost the same for NDCG and MSE, regardless of the value of $\lambda$. This indicates that, in the dense case, there were enough observables to avert the need for content-derived supplementation. In addition, at least for this data set, using a Poisson distribution resulted in a better fitting than using a Gaussian one.

Next, cases with fewer item observations, The performance of each method on the simulation data with 1000 items, $J = 1000$, is shown in Fig. 5: Same as Fig. 4, NDCG (top) and MSE (bottom); the sparse case (left) and the dense case (right). The vertical axis represents performance, and the horizontal axis represents $\lambda$, the parameter used to control the contribution of content information. Since the trend in the dense case was not much different from that in the $J = 1500$ case, we put the scope on the sparse case.
In the sparse case, one of the trends that differs from the case of \( J = 1500 \) is that GF was superior in terms of both NDCG and MSE. In addition, although NDCG and MSE improved as increasing \( \lambda \) for both methods, the trend when \( \lambda \) was smaller was different between the Poisson (Proposed) and Gaussian (CVAE) approaches. The performance with the CVAE approach was worse than that with the GF approach when \( \lambda = 0.2 \), and the performance was almost the same when \( \lambda = 0.4 \). In contrast, the performance of the proposed method was almost the same as that of PF at \( \lambda = 0.2 \) and much higher at \( \lambda = 0.4 \). Going back to the sparse case in \( J = 1500 \) again (left side of Fig. 4), we can observe that there is a slight spread in the performance of the proposed method and CVAE when \( \lambda \) is small. This indicates that the content information can be better utilized by using a Poisson distribution.

The utilization of content information could have been affected by the scale invariance property of the Poisson distribution model. Fig. 6 shows plots of the training data with the target variable (play count) on the x-axis and the estimated variable on the y-axis for the CVAE (left) and the proposed method (right). The dotted lines represent the 95% confidence interval for each of the metrics (Gaussian distribution for CVAE; Poisson distribution for proposed method).

A Gaussian distribution has a constant variance regardless of its scale, whereas the variance of a Poisson distribution depends on its scale. We can observe these trends in the learning outcomes. That the variance is constant means that, for hybrid recommendation using a Gaussian distribution, the latent variable estimated from the content information must contribute to predicting values very strictly, even for relatively large values of target variables. This may have caused overfitting in learning the relationship between the content information and the target variables. For \( \lambda = 0.6 \) and 0.8, the proposed method performed better in terms of NDCG, and the CVAE performed better in terms of MSE. Note that, related to these trends, assuming a Gaussian distribution is equivalent to minimizing the MSE. This can be interpreted to mean that using a Gaussian distribution minimizes the MSE without allowing for large variance for large values of the latent variable, whereas a Poisson distribution allows for a large variance for large values and renders other values as important as well, resulting in a better performance.

Finally, we discuss application to a real-world dataset, MSD-A. As shown in Table 2, in both the sparse and the
FIGURE 5. Performance of each method on simulation data ($J = 1000$), NDCG (top) and MSE (bottom), in sparse case (left) and dense case (right). Vertical axis represents performance, and horizontal axis represents contribution of content information, which is controlled by $\lambda$.

TABLE 2. Results of application to real-world dataset, MSD-A: mean square error and normalized discounted cumulative gain.

| Sparse NDCG | Sparse MSE | Dense NDCG | Dense MSE |
|-------------|------------|------------|-----------|
| GF (CF)     | 0.61       | 17.82      | 0.64      | 14.48     |
| PF (CF)     | 0.60       | 17.38      | 0.60      | 16.80     |
| CVAE (Hybrid) | 0.61      | 18.83      | 0.64      | 14.92     |
| Proposed (Hybrid) | 0.62  | 16.97      | 0.64      | 13.28     |

dense cases, the performance of CVAE with content information was almost equal to or lower than that of GF, which is the same trend as that with a low content information contribution rate in the simulation data. In the real-world case, the play count of a music piece is affected not only by the content but also by many other factors such as marketing (e.g., advertisements) or the period of its availability. This means that the contribution of content information to the play count is not so strong. Even in such a case with weak contributing content features, the proposed recommendation method, a hybrid method using a Poisson distribution, extracted beneficial information from the content information and performed well, except for the dense case in which NDCG was almost equal to that of GF and CVAE.

These results suggest that the proposed hybrid recommendation method utilizes content information effectively for predicting the number of interactions and is thus able to aid in overcoming the cold-start problem.

V. CONCLUSION

Our proposed hybrid recommendation framework uses Poisson factorization for collaborative filtering and jointly describes a generative model for content information that shares latent variables with collaborative filtering. The processes for generating content information and for generating implicit feedback are described in a probabilistic manner. The algorithm for inference and estimation is derived using the stochastic gradient variational Bayes method. Furthermore, we present that the framework comprising the proposed method is applicable to a broader range of collaborative filtering, including state-of-the-art methods, and can be extended to a hybrid recommendation algorithm that handles the number of interactions. Experimental results show that the proposed method utilizes content information effectively for predicting the number of interactions and should thus aid in overcoming the cold-start problem.
REFERENCES

[1] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, “Using collaborative filtering to weave an information tapestry,” Communications of the ACM, vol. 35, no. 12, pp. 61–70, 1992.

[2] Y. Hu, Y. Koren, and C. Volinsky, “Collaborative filtering for implicit feedback datasets,” in 2008 Eighth IEEE International Conference on Data Mining, Ieee, 2008, pp. 263–272.

[3] H. Liang and T. Baldwin, “A probabilistic rating auto-encoder for personalized recommender systems,” in Proceedings of the 24th ACM international conference on information and knowledge management, 2015, pp. 1863–1866.

[4] H.-J. Xue, X. Dai, J. Zhang, S. Huang, and J. Chen, “Deep matrix factorization models for recommender systems,” in IJCAI, vol. 17. Melbourne, Australia, 2017, pp. 3202–3209.

[5] D. Liang, R. G. Krishnan, M. D. Hoffman, and T. Jehbara, “Variational autoencoders for collaborative filtering,” in Proceedings of the 2018 World Wide Web Conference, 2018, pp. 689–698.

[6] A. K. Yengikand, M. Meghdadi, S. Ahmadian, S. M. J. Jalali, A. Khosravi, and S. Nahavandi, “Deep representation learning using multilayer perceptron and stacked autoencoder for recommendation systems,” in 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, 2021, pp. 2485–2491.

[7] D. Kelly and J. Terveen, “Implicit feedback for inferring user preference: a bibliography,” in Acm Sigir Forum, vol. 37, no. 2. ACM New York, NY, USA, 2003, pp. 18–28.

[8] R. J. Mooney and L. Roy, “Content-based book recommending using learning for text categorization,” in Proceedings of the fifth ACM conference on Digital libraries, 2000, pp. 195–204.

[9] M. Slaney, “Web-scale multimedia analysis: Does content matter?” IEEE Multimedia, vol. 18, no. 2, pp. 12–15, 2011.

[10] J. Bobadilla, F. Ortega, A. Hernando, and J. Bernal, “A collaborative filtering approach to mitigate the new user cold start problem,” Knowledge-based systems, vol. 26, pp. 225–238, 2012.

[11] C. Wang and D. M. Blei, “Collaborative topic modeling for recommending scientific articles,” in Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, 2011, pp. 448–456.

[12] X. Li and J. She, “Collaborative variational autoencoder for recommender systems,” in Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, 2017, pp. 305–314.

[13] A. Van den Oord, S. Dieleman, and B. Schrauwen, “Deep content-based music recommendation,” in Advances in neural information processing systems, 2013, pp. 2643–2651.

[14] S. Oramas, O. Nieto, M. Sordo, and X. Serra, “A deep multimodal approach for cold-start music recommendation,” in Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems, 2017, pp. 32–37.

[15] O. Barkan, N. Koenigstein, E. Yogev, and O. Katz, “Ch2cf: a neural multi-view content-to-collaborative filtering model for completely cold item recommendations,” in Proceedings of the 13th ACM Conference on Recommender Systems, 2019, pp. 228–236.

[16] S.-H. Yang, B. Long, A. J. Smola, H. Zha, and Z. Zheng, “Collaborative competitive filtering: learning recommender using context of user choice,” in Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval, 2011, pp. 295–304.

[17] T. Schnabel, A. Swaminathan, A. Singh, N. Chandak, and T. Joachims, “Recommendations as treatments: Debiasing learning and evaluation,” in International conference on machine learning. PMLR, 2016, pp. 1670–1679.

[18] Y. Wu, C. DuBois, A. X. Zheng, and M. Ester, “Collaborative denoising auto-encoders for top-n recommender systems,” in Proceedings of the ninth ACM international conference on web search and data mining, 2016, pp. 153–162.

[19] I. Shenbin, A. Alekseev, E. Tutubalina, V. Malykh, and S. I. Nikolenko, “Recvae: A new variational autoencoder for top-n recommendations with implicit feedback,” in Proceedings of the 13th International Conference on Web Search and Data Mining, 2020, pp. 528–536.

[20] Y. Wang, D. Liang, L. Charlin, and D. M. Blei, “Causal inference for recommender systems,” in Fourteenth ACM Conference on Recommender Systems, 2020, pp. 426–431.

[21] M. Ahmadian, M. Ahmadi, S. Ahmadian, S. M. J. Jalali, A. Khosravi, and S. Nahavandi, “Integration of deep sparse autoencoder and particle swarm optimization to develop a recommender system,” in 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC). IEEE, 2021, pp. 2524–2530.

[22] M. Ahmadian, M. Ahmadi, and S. Ahmadian, “A reliable deep representation learning to improve trust-aware recommendation systems,” Expert Systems with Applications, vol. 197, p. 116697, 2022.

[23] S. Ahmadian, M. Ahmadi, and M. Jalili, “A deep learning based trust-and-tug-aware recommender system,” Neurocomputing, vol. 488, pp. 557–571, 2022.

[24] S. Ahmadian, N. Joorabloo, M. Jalili, and M. Ahmadian, “Alleviating data sparsity problem in time-aware recommender systems using a reliable rating profile enrichment approach,” Expert Systems with Applications, vol. 187, p. 115849, 2022.

[25] P. K. Gopalan, L. Charlin, and D. Blei, “Content-based recommendations

FIGURE 6. Scatter plots of training data with target variable (play count) on x-axis and estimated variable on y-axis for CVAE (left) and proposed method (right). Dotted lines represent 95% confidence intervals for each metric, i.e., Gaussian distribution for CVAE and Poisson distribution for proposed method.
with Poisson factorization,” in Advances in Neural Information Processing Systems, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2003.

[26] M. J. Beal, Variational algorithms for approximate Bayesian inference, University of London, University College London (United Kingdom), 2003.

[27] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” arXiv preprint arXiv:1312.6114, 2013.

[28] M. Figurnov, S. Mohamed, and A. Mnih, “Implicit reparameterization gradients,” in Proceedings of the 32nd International Conference on Neural Information Processing Systems, 2018, pp. 439–450.

[29] M. Jankowiak and F. Obermeyer, “Pathwise derivatives beyond the reparameterization trick,” in International conference on machine learning, PMLR, 2018, pp. 2235–2244.

[30] E. Jang, S. Gu, and B. Poole, “Categorical reparameterization with gumbel-softmax,” arXiv preprint arXiv:1611.01144, 2016.

[31] D. Lee and H. S. Seung, “Algorithms for non-negative matrix factorization,” in Advances in Neural Information Processing Systems, T. Leen, T. Dietterich, and V. Tresp, Eds. MIT Press.

[32] P. Gopalan, J. M. Hofman, and D. M. Blei, “Scalable recommendation with hierarchical Poisson factorization,” in UAI, 2015, pp. 326–335.

[33] J. Schmidhuber, “Deep learning in neural networks: An overview,” Neural networks, vol. 61, pp. 85–117, 2015.

[34] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[35] H. Sak, A. Senior, and F. Beaufays, “Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition,” arXiv preprint arXiv:1402.1128, 2014.

[36] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rn encoder-decoder for statistical machine translation,” arXiv preprint arXiv:1406.1078, 2014.

[37] Y. Pu, Z. Gan, R. Henao, X. Yuan, C. Li, A. Stevens, and L. Carin, “Variational autoencoder for deep learning of images, labels and captions,” in Advances in Neural Information Processing Systems, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc.

[38] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio, “A recurrent latent variable model for sequential data,” in Advances in Neural Information Processing Systems, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc.

[39] J. Engel, L. Hantrakul, C. Gu, and A. Roberts, “DDSP: Differentiable digital signal processing,” arXiv preprint arXiv:2001.04643, 2020.

[40] G. E. P. Box and D. R. Cox, “An analysis of transformations,” Journal of the Royal Statistical Society. Series B (Methodological), vol. 26, no. 2, pp. 211–252, 1964.

[41] Z. Yang, H. Zhang, Z. Yuan, and E. Oja, “Kullback-leibler divergence for nonnegative matrix factorization,” in International Conference on Artificial Neural Networks. Springer, 2011, pp. 250–257.

[42] L. Bottou, “Online algorithms and stochastic approximations,” Online learning and neural networks, 1998.

[43] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, “The million song dataset,” 2011.

[44] B. McFee, T. Bertin-Mahieux, D. P. Ellis, and G. R. Lanckriet, “The Million Song Dataset Challenge,” in Proceedings of the 21st International Conference on World Wide Web, 2012, pp. 909–916.

[45] A. Mnih and R. R. Salakhutdinov, “Probabilistic matrix factorization,” in Advances in neural information processing systems, 2008, pp. 1257–1264.

[46] K. Ć. Järvelin and J. Kekäläinen, “Cumulated gain-based evaluation of IR techniques,” ACM Transactions on Information Systems (TOIS), vol. 20, no. 4, pp. 422–446, 2002.

[47] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” Journal of machine Learning research, vol. 3, no. Jan, pp. 993–1022, 2003.

[48] A. F. Agarap, “Deep learning using rectified linear units (ReLU),” CoRR, vol. abs/1803.08375, 2018.

[49] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv preprint arXiv:1412.6980, 2014.

[50] Y. Yao, L. Rosasco, and A. Caponnetto, “On early stopping in gradient descent learning,” Constructive Approximation, vol. 26, no. 2, pp. 289–315, 2007.

IWAO TANUMA received a B.S. degree in science from the Tokyo University of Science, Tokyo, Japan, in 2010 and an M.S. degree in information physics & computing from the University of Tokyo, Tokyo, Japan, in 2012. He worked as a researcher in several laboratory of Hitachi, Ltd. R&D Group and Hitachi America, Ltd. R&D from April 2012 to November 2021. He is currently working as a backend engineer at DWANGO Co., Ltd. as well as working on a Ph.D. in the Dept. of Statistical Science, Graduate University for Advanced Studies, Japan. His research interests include unsupervised and semi-supervised learning based on probabilistic latent factor models as well as potential applications.