Latent User Intent Modeling for Sequential Recommenders

Bo Chang, Alexandros Karatzoglou, Yuyan Wang, Can Xu, Ed H. Chi, Minmin Chen
Google, Inc.
Mountain View, CA, USA
{bochang, alexkz, yuyanw, canxu, edchi, minminc}@google.com

ABSTRACT

Sequential recommender models are essential components of modern industrial recommender systems. These models learn to predict the next items a user is likely to interact with based on his/her interaction history on the platform. Most sequential recommenders however lack a higher-level understanding of user intents, which often drive user behaviors online. Intent modeling is thus critical for understanding users and optimizing long-term user experience. We propose a probabilistic modeling approach and formulate user intent as latent variables, which are inferred based on user behavior signals using variational autoencoders (VAE). The recommendation policy is then adjusted accordingly given the inferred user intent. We demonstrate the effectiveness of the latent user intent modeling via offline analyses as well as live experiments on a large-scale industrial recommendation platform.

KEYWORDS

recommender systems, intent modeling, variational autoencoder

1 INTRODUCTION

Modern recommender systems are powering many online platforms. Sequential recommendation models that consider the order of user-item interactions are becoming increasingly popular [7, 18, 34, 38]. Most existing sequential recommender models mainly rely on the item-level interaction history of a user to capture his/her *topical interests*. These models often lack a higher-level understanding of *user intent*, i.e., what a user wants from the platform at request time, for example exploring new contents, continuing with contents from the last session, playing background music, etc. User intents often span across sessions, thus intent understanding is crucial to optimizing long-term user experience.

User intent can be explicitly defined. For example, search intents are classified into navigational, informational, and transactional [5, 16, 32]. With this approach, we can annotate the training data with explicitly defined user intent and cast it as a supervised learning

task to predict user intent. This formulation has clear advantages: it allows for good interpretability and reliable evaluation of model prediction. It however also comes with an obvious downside, i.e., requiring expert knowledge to manually define and enumerate user intent. Compared with the search use case, user intent in organic recommendation is often multifaceted and subconscious, which are much more challenging to manually define [17].

Rather than explicitly defining user intent, we propose to model user intent as latent variables using *latent variable models*, which have been extensively studied and applied across various fields including statistics [11], machine learning [4], and econometrics [1]. It is a probabilistic model relating observed variables or evidence to latent variables. Specifically, it defines a joint distribution over the observed and latent variables. One can then obtain the corresponding distribution of the observed variables through marginalization.

User behavior signals on an online platform—for example, search, browse, click, and consumption—are often good indicators of the underlying user intent. Therefore, they can be used as observed variables in the latent variable model. In other words, we formulate user intent as latent variables and jointly model them with user behavior signals, avoiding the need to manually define intent.

Together we make the following contributions:

- Propose a probabilistic model to formulate the user intent as latent variables and relate it to behavior and context signals.
- Apply variational inference techniques for efficient and scalable inference.
- Shed light on training stability of probabilistic models for industrial use cases.
- Conduct an analysis to gain insights into the semantics of the latent space.
- Demonstrate the benefits of the proposed technique in large-scale live experiments on a commercial recommendation platform serving billions of users and millions of items.

2 RELATED WORK

Sequential recommendation systems. Sequential recommenders are a class of recommender systems, which are based on modeling the order of the user-item interactions [38]. Various architectures have been applied to capture long-term and complex dependencies in user-item interactions, for example, recurrent neural networks (RNN) [3, 7, 13, 42], convolutional neural networks (CNN) [34, 35, 43], and self-attentive models [18]. However, most methods mainly focus on the item-level interaction history and often lack a higher-level understanding of user intent.

User intent modeling. Multiple approaches have been proposed to model user intent for improving recommender systems [6, 8, 21, 22, 28, 31, 36]; in particular, the implicit user intent approach has

gained popularity. Wang et al. [39] use the mixture-channel purpose routing network (MCPRN) to learn users' different purchase purposes of each item. Li et al. [24] propose methods to discover new intents based on existing explicitly-defined ones. Implicitly deduced intent from user interactions has also been used to understand user satisfaction in music recommendation [30]. As far as we know, our proposal to bring latent variable models into intent modeling and leverage behavior signals is a novel contribution.

Latent variable models and variational autoencoders. Latent variable models are a class of probabilistic models that assume the observed variables are generated by a set of unobserved or latent variables. The models learn the joint distribution of observed and latent variables, then the distribution of observed ones is obtained by marginalization [1, 4, 11]. Variational autoencoders (VAE) provide a principled framework for learning deep latent variable models and corresponding inference models [19, 20]. Several extensions of VAE are later proposed, including conditional VAE that models the conditional distributions [33] and variational RNN for modeling sequential data [10]. Applications of VAE to recommender systems have also been an active research topic [23, 25–27]. In this work, we propose a novel application of latent variable models for user intent modeling.

3 LATENT INTENT MODELING

In this section, we provide a detailed exposition of the proposed latent intent modeling technique. Section 3.1 discusses the model assumptions in the form of a probabilistic graphical model. Section 3.2 describes the model architecture and training objective of the latent intent module. Finally, Section 3.3 summarizes how the module is incorporated into a sequential recommender model.

3.1 Probabilistic Model

We start with a directed probabilistic graphical model (PGM) or Bayesian network to factorize the joint distribution of the random variables of interest. The high-level structural assumption of the model is that past user behavior and context information before the user request indicates a user's current intent, which is further manifested in future user behavior. Figure 1 provides a graphical representation of the model. Following the convention of visual representations of probabilistic graphical models, we use shaded nodes to denote observed variables and transparent nodes for latent ones. The nodes in the graph represent the following variables:

- Node x: past user behavior (e.g., number of clicks/searches in the past 15 minutes) and context (e.g., time of day, device);
- Node y: future user behavior (e.g., number of clicks/searches in the next 15 minutes);
- Node z: latent variable (i.e., user intent).

Note that x and y are assumed to be conditionally independent given z. In other words, the link between past user behavior and context x and future behavior y are mediated by the latent intent z.

3.2 Inference with Variational Autoencoders

From a probabilistic modeling perspective, the goal of the latent intent model is to capture the conditional distribution $p(y|x)$, where
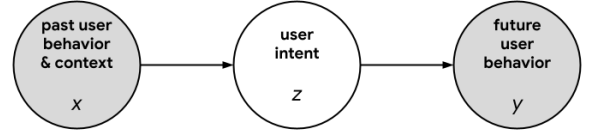

Figure 1: Probabilistic graphical model. The connection from past user behavior and context x to future user behavior y is mediated by the user intent z. Note that x and y are observed variables whereas z is latent.

the link between x and y is mediated by the latent variable z. Latent variable models are known to be difficult to learn and infer from, due to intractable posterior distributions. In order to make learning and inference tractable, we adopt a variational inference algorithm, i.e., the conditional VAE [19, 33] to efficiently scale to large datasets. The high-level idea is to introduce a variational distribution $q(z|x, y)$ that approximates the true posterior distribution and maximizes a lower bound of the log-likelihood: $\log p(y|x)$. It casts inference as an optimization problem. We refer interested readers to Kingma and Welling [20] and references therein for further details about variational inference and VAE.

The model is composed of the following networks/distributions:

- Prior: $p(z|x) \sim \mathcal{N}(\mu_\psi, \Sigma_\psi)$;
- Decoder/likelihood: $p(y|z) \sim \mathcal{N}(\mu_\theta, \Sigma_\theta)$;
- Encoder/variational distribution: $q(z|x, y) \sim \mathcal{N}(\mu_\phi, \Sigma_\phi)$.

All distributions are parameterized as multivariate Gaussian with diagonal covariance matrices; the mean and log-variance are the output of multi-layer perceptrons (MLP), with ReLU as the activation function.

The parameters of the networks are trained to maximize the evidence lower bound (ELBO):

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{q(z|x,y)}[\log p(y|z)] - D_{\text{KL}}(q(z|x,y)\|p(z|x)), \quad (1)$$

where $D_{\text{KL}}(\cdot\|\cdot)$ denotes the Kullback–Leibler (KL) divergence. It can be shown that the ELBO is a lower bound of the conditional likelihood: $\mathcal{L}_{\text{ELBO}} \leq \log p(y|x)$, which is the quantity we are interested in maximizing [33].

The first term of the ELBO in Equation 1 can be regarded as a *reconstruction loss*. Given a pair of past user behavior and context x and future user behavior y, if we pass it through the encoder and sample a latent intent z from $q(z|x, y)$, then z should contain relevant information about y and the model will be able to truthfully reconstruct y. The dimensionality of z is often chosen to be smaller than that of y so that it can act as an information bottleneck [2]. The second term of the ELBO is a *regularization loss*; it encourages the approximate posterior distribution $q(z|x, y)$ to not deviate too much from the prior distribution $p(z|x)$.

Figure 2 shows a diagram of the latent intent module. Note that one adaptation from the original conditional VAE model is that the decoder does not take x as input, i.e., $p(y|z)$ instead of $p(y|x, z)$. This is to be consistent with the probabilistic graphical model defined in Figure 1; that is, y is conditionally independent of x given z.

Training Stability. In practice, we observed training instability issues of the latent intent module, due to an exploding KL-divergence
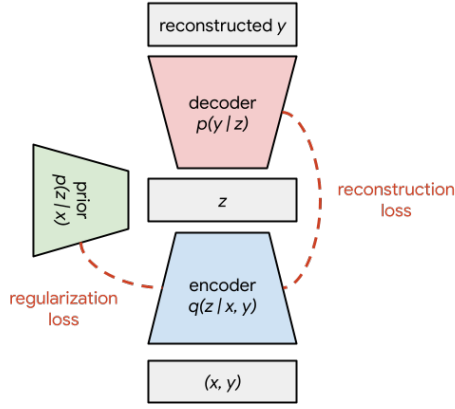
Figure 2: The latent intent module consists of three networks: prior, decoder; and encoder. The loss can be written as two terms: reconstruction loss and regularization loss.

term, either at the very beginning or in the middle of model training. Two techniques are found effective in mitigating the issues. The first one is the *proper initialization* of the trainable parameters. We initialize the weights by a uniform distribution around zero $\mathcal{U}(-\varepsilon, \varepsilon)$, with ε being a small positive number, and initialize the biases to zero. As a result, both $q(z|x, y)$ and $p(z|x)$ are close to $\mathcal{N}(0, I)$ at initialization, thus the KL-divergence is near zero. Secondly, *soft-clipping the log-variance* throughout training can prevent KL-divergence from exploding [9]. In particular, we use the softplus function to clip the log-variance to the interval (a, b):

$$f(x) = x - \log(1 + \exp(x - b)) + \log(1 + \exp(a - x)). \quad (2)$$

Combining the two techniques together, we observe that the KL-divergence term remains bounded throughout model training.

3.3 Incorporation into Sequential Recommenders

The latent intent model introduced in Section 3.2 is a standalone module with its own training objective. In this section, we describe how it is incorporated into the main recommendation model, in particular, a REINFORCE sequential recommendation model [7]. Note that the proposed technique can be easily applied to other types of sequential recommendation models, for example, self-attentive ones [18].

We briefly describe components of the recommendation model that are pertinent to this work and refer interested readers to Chen et al. [7] for more details. The sequential recommendation model uses an RNN to summarize a user's interaction history on the platform. The final RNN hidden state is concatenated with other context information and passed through the post-fusion layers (an MLP with ReLU activations), the output of which is used as the user representation. A softmax policy is defined on top of that, together with item representations. The model is trained using the REINFORCE algorithm [41] to optimize long-term user satisfaction; we denote the loss function by $\mathcal{L}_{\text{rec}}$.

Among the three components of the latent intent model (prior, encoder, and decoder), we only make the prior network directly interact with the main recommender model. Recall that the prior network $p(z|x)$ infers user intent z given past user behavior and context x. Both at training and serving time, we pass x through the prior network and draw a sample z from $p(z|x)$. In order to further decouple the latent intent module from the main recommendation model, we apply a stop gradient operation to z. Then the sampled user intent z is concatenated with the final RNN hidden state and then passed to the post-fusion layers. The output is the user representation augmented with the inferred user intent, which further conditions the recommendation policy. The training objective of the sequential recommendation model $\mathcal{L}_{\text{rec}}$ remains unchanged. The overall loss function[1] is $\mathcal{L} = \mathcal{L}_{\text{rec}} - \lambda \mathcal{L}_{\text{ELBO}}$, where $\lambda > 0$ is a hyperparameter controlling the relative strength of the ELBO loss. The overall model architecture is illustrated in Figure 3.
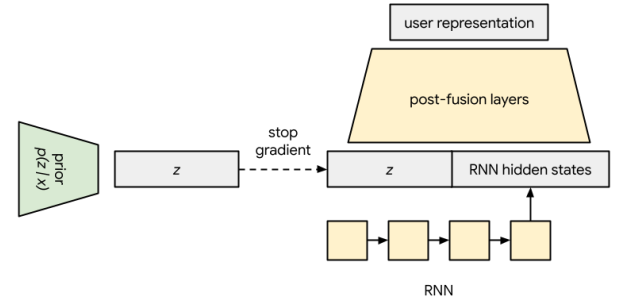

Figure 3: The overall architecture incorporating the latent intent module to the sequential recommendation model. Only the prior network is used and a stop gradient is applied to further decouple them.

4 LATENT SPACE ANALYSIS

One drawback of VAE is the limited interpretability of the latent space. In our use case, the inferred user intent z is a compressed representation of user intent in a continuous latent space. In order to gain more insights into what the latent space encodes, we adopt an analysis technique by Chung et al. [10].

Recall that the KL-divergence term in Equation 1 is computed between the approximate posterior $q(z|x, y)$ and prior distributions $p(z|x)$; in other words, it can be seen as the difference in user intent with and without the knowledge of future user behavior. Therefore, it can be regarded as a measure of *surprise*. In fact, the KL-divergence between posterior and prior distributions has long been used in neuroscience to quantify surprise [12, 14].

Intuitively, when a user consumes an item that is "new", the amount of surprise and KL-divergence should be higher than that of an "old" item. Therefore, we can compare the KL-divergence between new and old items consumed by users to corroborate the statement above. Two definitions of new items are considered: (1) item-level: an item that has not been consumed by the user before; (2) topic-level: an item that belongs to a topic cluster[2] that has not been consumed by the user before.

---

[1]Note that we want to maximize $\mathcal{L}_{\text{ELBO}}$, thus the minus sign before it in the loss function.

[2]The topic clusters are produced by: 1) taking the item co-occurrence matrix, where the (i, j)-th entry counts the number of times item i and j were interacted by the

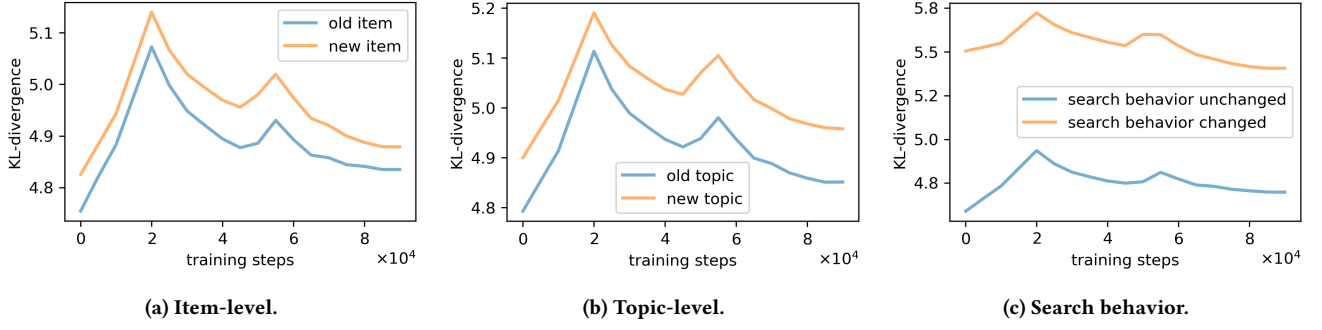(a) Item-level. (b) Topic-level. (c) Search behavior.

Figure 4: Latent space analysis. The KL-divergence between the approximate posterior $q(z|x, y)$ and prior $p(z|x)$ is plotted against the training steps. The two curves correspond to new and old items at both item- and topic-levels in (a) and (b), and changed/unchanged search behavior in (c).

We also consider scenarios when there are sudden changes in user search behavior, and the KL-divergence should be higher when such changes occur. In particular, we focus on the number of search queries a user issued/issues in the past 15 minutes s_{past} and in the next 15 minutes s_{future}. We say there is a change in user search behavior if a user searched in the past but not in the future ($s_{\text{past}} > 0$ and $s_{\text{future}} = 0$) or vice versa ($s_{\text{past}} = 0$ and $s_{\text{future}} > 0$); otherwise, user search behavior is deemed unchanged.

Figure 4 shows the average KL-divergence against the training steps. Figures 4a and 4b compare "new" and "old" items at the item- and topic-levels, respectively. When a user consumes a new item or topic, the KL-divergence is higher. Figure 4c shows the KL-divergence when user search behavior is changed/unchanged. When there is a change in user search behavior, the KL-divergence is higher. These results indicate that the latent space is able to capture salient information and detect transitions in user behavior.

5 LIVE EXPERIMENTS

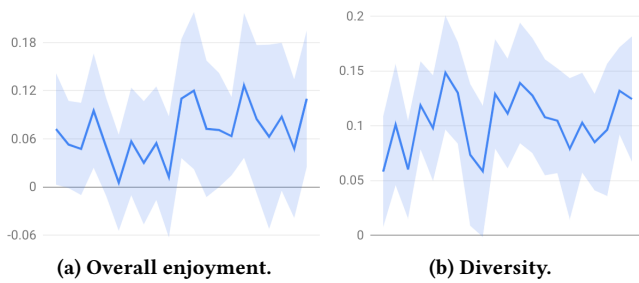(a) Overall enjoyment. (b) Diversity.

Figure 5: Live experiment results. On the x-axis is the date; on the y-axis is the relative difference in percentage between the experiment and control.

We conduct A/B experiments in a live system serving billions of users to measure the benefits of the proposed latent intent modeling technique. The sequential recommender system [7] is built to

same user consecutively; 2) performing matrix factorization to generate an embedding for each item; 3) using k-means to cluster the learned embeddings into 10,000 clusters; 4) assigning the nearest cluster to each item.

retrieve hundreds of candidates from a corpus of millions of items upon each user request. The retrieved candidates, along with those returned by other sources, are scored and ranked by a separate ranking system before showing the top results to the user.

Experiments are run for three weeks, during which both the control and experiment models are trained continuously with new interactions and feedback being used as training data. We focus on the following two metrics: (1) users' overall enjoyment of the platform; (2) diversity of user-item interactions, which represents the number of unique topic clusters users have interacted with. It has been shown that consumption diversity is an effective surrogate for long-term user experience [40].

The experiment and control models are the sequential recommender models with and without the latent intent module. Figure 5 summarizes the live experiment results. On the x-axis is the date, and on the y-axis is the relative difference of a metric in percentage between the experiment and control. We report the mean and 95% confidence intervals of the metrics. Relative to the control, the experiment model improves the overall enjoyment by +0.07% with a 95% confidence interval of $(+0.02\%, +0.11\%)$. Diversity of user-item interactions improves by +0.10% with a 95% confidence interval of $(+0.08\%, +0.13\%)$. Furthermore, there is an upward trend in the overall enjoyment metric, suggesting a user learning effect, i.e., user states change in response to the recommendation policy.

The proposed model has been deployed to the production system for more than two weeks. The way it is deployed is described at the beginning of this section.

6 CONCLUSION

In this work, we propose the latent user intent model, a probabilistic modeling approach to capturing user intent and complementing existing sequential recommenders. The variational inference technique we adopt is efficient and scalable, and techniques to stabilize and interpret the model are also studied. Finally, the effectiveness of the proposed method is validated in large-scale live experiments. One future research direction is to further improve the interpretability of latent user intent by designing a discrete latent space, using Gumbel softmax [15, 29] or vector quantized variational autoencoder (VQVAE) [37].

REFERENCES

[1] Dennis J Aigner, Cheng Hsiao, Arie Kapteyn, and Tom Wansbeek. 1984. Latent variable models in econometrics. *Handbook of econometrics* 2 (1984), 1321–1393.

[2] Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. 2017. Deep Variational Information Bottleneck. In *International Conference on Learning Representations*.

[3] Alex Beutel, Paul Covington, Sagar Jain, Can Xu, Jia Li, Vince Gatto, and Ed H Chi. 2018. Latent cross: Making use of context in recurrent recommender systems. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. 46–54.

[4] Christopher M Bishop. 1998. Latent variable models. In *Learning in graphical models*. Springer, 371–403.

[5] Andrei Broder. 2002. A taxonomy of web search. In *ACM Sigir forum*, Vol. 36. ACM New York, NY, USA, 3–10.

[6] Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. 2020. Controllable multi-interest framework for recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2942–2951.

[7] Minmin Chen, Alex Beutel, Paul Covington, Sagar Jain, Francois Belletti, and Ed H Chi. 2019. Top-k off-policy correction for a REINFORCE recommender system. In *WSDM*. 456–464.

[8] Yongjun Chen, Zhiwei Liu, Jia Li, Julian McAuley, and Caiming Xiong. 2022. Intent Contrastive Learning for Sequential Recommendation. In *Proceedings of the ACM Web Conference 2022*. 2172–2182.

[9] Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. 2018. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in neural information processing systems* 31 (2018).

[10] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. 2015. A recurrent latent variable model for sequential data. *Advances in neural information processing systems* 28 (2015).

[11] B Everett. 2013. *An introduction to latent variable models*. Springer Science & Business Media.

[12] Karl Friston. 2010. The free-energy principle: a unified brain theory? *Nature reviews neuroscience* 11, 2 (2010), 127–138.

[13] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. *International Conference of Learning Representations ICLR* (2016).

[14] Laurent Itti and Pierre Baldi. 2005. Bayesian surprise attracts human attention. *Advances in neural information processing systems* 18 (2005).

[15] Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical Reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations*.

[16] Bernard J Jansen, Danielle L Booth, and Amanda Spink. 2007. Determining the user intent of web search engine queries. In *Proceedings of the 16th international conference on World Wide Web*. 1149–1150.

[17] Fotis P Kalaganis, Kostas Georgiadis, Vangelis P Oikonomou, Nikos A Laskaris, Spiros Nikolopoulos, and Ioannis Kompatsiaris. 2021. Unlocking the Subconscious Consumer Bias: A Survey on the Past, Present, and Future of Hybrid EEG Schemes in Neuromarketing. *Frontiers in Neuroergonomics* 2 (2021), 672982.

[18] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*. IEEE, 197–206.

[19] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[20] Diederik P. Kingma and Max Welling. 2019. An Introduction to Variational Autoencoders. *Foundations and Trends® in Machine Learning* 12, 4 (2019), 307–392.

[21] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. 2019. Multi-interest network with dynamic routing for recommendation at Tmall. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2615–2623.

[22] Haoyang Li, Xin Wang, Ziwei Zhang, Jianxin Ma, Peng Cui, and Wenwu Zhu. 2021. Intention-aware sequential recommendation with structured intent transition. *IEEE Transactions on Knowledge and Data Engineering* (2021).

[23] Xiaopeng Li and James She. 2017. Collaborative variational autoencoder for recommender systems. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 305–314.

[24] Yinfeng Li, Chen Gao, Xiaoyi Du, Huazhou Wei, Hengliang Luo, Depeng Jin, and Yong Li. 2022. Automatically Discovering User Consumption Intents in Meituan. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3259–3269.

[25] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 world wide web conference*. 689–698.

[26] Kai Luo, Hojin Yang, Ga Wu, and Scott Sanner. 2020. Deep critiquing for VAE-based recommender systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1269–1278.

[27] Chao Ma, Wenbo Gong, José Miguel Hernández-Lobato, Noam Koenigstein, Sebastian Nowozin, and Cheng Zhang. 2018. Partial VAE for hybrid recommender system. In *NIPS Workshop on Bayesian Deep Learning*, Vol. 2018.

[28] Jianxin Ma, Chang Zhou, Hongxia Yang, Peng Cui, Xin Wang, and Wenwu Zhu. 2020. Disentangled self-supervision in sequential recommenders. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 483–491.

[29] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. 2017. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. In *International Conference on Learning Representations*.

[30] Rishabh Mehrotra, Mounia Lalmas, Doug Kenney, Thomas Lim-Meng, and Golli Hashemian. 2019. Jointly leveraging intent and interaction signals to predict user satisfaction with slate recommendations. In *The World Wide Web Conference*. 1256–1267.

[31] Zhiqiang Pan, Fei Cai, Yanxiang Ling, and Maarten de Rijke. 2020. An intent-guided collaborative machine for session-based recommendation. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*. 1833–1836.

[32] Daniel E Rose and Danny Levinson. 2004. Understanding user goals in web search. In *Proceedings of the 13th international conference on World Wide Web*. 13–19.

[33] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. 2015. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems* 28 (2015).

[34] Jiaxi Tang, Francois Belletti, Sagar Jain, Minmin Chen, Alex Beutel, Can Xu, and Ed H. Chi. 2019. Towards neural mixture recommender for long range dependent user sequences. In *The World Wide Web Conference*. 1782–1793.

[35] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the eleventh ACM international conference on web search and data mining*. 565–573.

[36] Md Mehrab Tanjim, Congzhe Su, Ethan Benjamin, Diane Hu, Liangjie Hong, and Julian McAuley. 2020. Attentive sequential models of latent intent for next item recommendation. In *Proceedings of The Web Conference 2020*. 2528–2534.

[37] Aaron Van Den Oord, Oriol Vinyals, et al. 2017. Neural discrete representation learning. *Advances in neural information processing systems* 30 (2017).

[38] Shoujin Wang, Liang Hu, Yan Wang, Longbing Cao, Quan Z. Sheng, and Mehmet Orgun. 2019. Sequential Recommender Systems: Challenges, Progress and Prospects. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. 6332–6338.

[39] Shoujin Wang, Liang Hu, Yan Wang, Quan Z Sheng, Mehmet Orgun, and Long-bing Cao. 2019. Modeling multi-purpose sessions for next-item recommendations via mixture-channel purpose routing networks. In *International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence.

[40] Yuyan Wang, Mohit Sharma, Can Xu, Sriraj Badam, Qian Sun, Lee Richardson, Lisa Chung, Ed H Chi, and Minmin Chen. 2022. Surrogate for Long-Term User Experience in Recommender Systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4100–4109.

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

[42] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J Smola, and How Jing. 2017. Recurrent recommender networks. In *Proceedings of the tenth ACM international conference on web search and data mining*. 495–503.

[43] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M Jose, and Xiangnan He. 2019. A simple convolutional generative network for next item recommendation. In *Proceedings of the twelfth ACM international conference on web search and data mining*. 582–590.