Aalto-yliopisto
Aalto-universitetet
Aalto University

Master’s Programme in ICT Innovation

Improving Recommender System Diversity using Variational Autoencoders

Sheetal Borar

Master’s Thesis
2022

Aalto-yliopisto
Aalto-universitetet
Aalto University

Author Sheetal Borar
Title of thesis Improving Diversity in Recommender System using Variational Autoencoders

Programme MSc. in ICT Innovation
Major SCI3115 Data Science

Thesis supervisor Prof. Dr. Mykola Pechenizkiy, Prof. Wilhelmiina Hämaläinen
Thesis advisor(s) Binyam Gebre (PhD), Hilde Weerts (MSc.)

Collaborative partner Bol.com
Date 22.08.2022 **Number of pages** 54+8 **Language** English

Abstract

Recommender systems have been widely adopted in many use cases to help customers find relevant items in eCommerce and content recommendation platforms. Collaborative filtering algorithms are often trained to optimize accuracy, but recent user research has shown that other system features, including diversity, are also crucial to facilitate a good user experience. In this work, we aim to assess and improve the diversity of recommendations in the context of large eCommerce platforms. This research has been done in collaboration with Bol.com, the largest e-commerce retailer in the Benelux region.

Diversity of recommendations has been defined in numerous ways in the literature. However, these definitions have not been evaluated with the limitations of a real-world recommender system like scalability constraints. Therefore, we first evaluate how diversity should be measured holistically and feasibly in real-world recommender systems.

Second, we achieve diversity improvements in recommender systems by using variational autoencoders. These models have previously been used in natural language tasks for improving diversity, but not in recommender systems. In this study, we have used the generative nature of variational autoencoders to generate a distribution from which we sample multiple user profiles that are used to generate recommendations with higher user and item- level diversity.

Through empirical analysis over benchmark and real-world datasets, we show that our approach produces recommendations that are more diverse in several ways. First, a single recommendation list for a user is more diverse. Second, recommendations generated for each user over time are more diverse. We propose a novel metric called temporal inter-list diversity to measure this effect. Third, the total number of items exposed to the users increased as well. Additionally, we have done a parameter sensitivity analysis to verify to what extent the results depend on the parameter settings and help practitioners identify how to tune the different parameters in the system to achieve the desired accuracy and diversity in recommendations.

We think that the proposed method and evaluations can help improve customer satisfaction and vendor exposure in recommender systems.

Keywords Recommender Systems, Diversity, Variational Autoencoders

Preface

In 2017, I implemented my first machine learning model to detect whether a person was being deceptive. Since then, I have found the power of machine learning in doing complex tasks like cancer and fraud detection fascinating. Uncle Ben has wisely said - "With great power comes great responsibility". While working with machine learning models in the industry and during my masters, I was repeatedly reminded that they rely heavily on data and can inherit the biases found in the data. This research has allowed me to explore some common biases found in recommender systems that are highly pervasive in an ordinary person's life today. It has also enabled me to propose a method to correct them.

Firstly, I would like to take this opportunity to thank my supervisor Mykola Pechenizkiy for his guidance on how to conduct good research and for supporting and challenging me during the thesis. Your feedback was critical in helping me improve my work.

Secondly, I would like to extend gratitude to Hilde Weerts, my regular supervisor, for her support throughout the process. You consistently guided me in the right direction and helped me make critical choices during the thesis journey. I would also like to thank you for the emotional support and mental strength you gave me to pursue this project. Your critical thinking and approach to solving problems were very valuable as well.

Thirdly, I would like to thank Binyam Gebre, the recommendations data scientist at Bol.com and my direct supervisor in the company, for always guiding me technically and helping me direct my energy in the right direction. Your explanations about many of the critical concepts in this thesis to me are greatly appreciated. I would also like to thank the company for their support in pursuing this project using their data and resources.

Lastly, but most importantly, I would like to thank my family and friends for their continuous support during this journey. For listening to my frustrations with models not working and calmly pushing me to continue. Specifically, I would like to thank my parents and my friends, Vaishali, Enrico, and Dipen who helped me stay optimistic during this process. I'm also very grateful to Wafa, Keerthana, Mayank, Aditya and Onno for their thorough feedback on the thesis which helped me improve it.

I'm grateful to have gone on this journey and to end my master's education with this thesis. I have learned a lot in this process, and I believe the time I have invested in learning how to conduct good research will guide my choices in my future projects. Lastly, I hope that the results of this thesis can be used by industry and academia to improve the experience of millions of users worldwide.

Contents

Chapter 1

Introduction

Recommender systems (RSs) have become pervasive in our daily lives, whether searching for a movie to watch on Netflix or exploring top deals on Amazon. They were introduced to personalize the experience and help users find relevant items more effectively as the number of items on online platforms exploded. The ultimate goal of a RS is to provide a pleasant user experience and to increase sales. Most RSs are optimized for accuracy, which results in overly specific recommendations [15]. This may lead to a poor user experience and dissatisfied users, and limit the opportunities different vendors get on the platform. In this thesis, we will focus on the diversity aspect of RSs.

This chapter will describe the research problem we aim to address in this study, the experiments conducted to answer the research questions (RQs), highlight our contributions and provide an outline for the rest of the thesis.

1.1 Research Problem

In a real-world RS, user satisfaction is the highest priority and it has been linked with multiple RS features apart from accuracy. *Diversity* is defined as the condition of having many different elements. It is considered one of the crucial features of a useful RS. Studies have shown that user satisfaction improves with high diversity because users perceive it as high accuracy [22]. Moreover, higher item-level diversity ensures that a larger proportion of the item catalog gets user exposure. This is useful for all the stakeholders in the system - the vendors (whose items get more exposure), the users (who get more product options), and the company (which has increased product sales [47]).

The goal of this study is to increase the diversity of recommendations at the largest online retail platform in the Benelux region while maintaining a reasonable level of accuracy. In recent years, several techniques have been proposed to improve RS diversity with a slight decrease in relevance. However, these techniques often intervene at the later stages of the recommendation process, e.g., by post-processing the list of recommendations. We aim to intervene at the first stage of the process i.e. user profile generation. Recent work has shown that multiple user profiles can be more effective at capturing users' varied interests compared to a single user profile [17]. Motivated by this finding, we propose an approach to increase diversity through richer user representations. In particular, our method relies on Variational AutoEncoders (VAEs) to improve the diversity of recommendations. The generative nature of VAEs has been shown to improve diversity in natural language processing tasks like label and caption generation [14] [44] [19]. However, it has not yet been explored in the context of diversity in RSs.

Our work is divided into three parts. Firstly, diversity has been operationalized in many different ways. Therefore, we start with an assessment of different diversity metrics based on the requirements of the real-world RS at Bol.com. Secondly, we propose a VAE-based user profile generation method to improve diversity in RSs. Thirdly, we perform several experiments to evaluate the efficacy of this approach for increasing diversity while maintaining an acceptable level of relevance. We have also provided a parameter sensitivity analysis which can be used to understand the robustness or sensitivity of the method to different parameters in the system.

1.2 How to measure diversity holistically in a real-world RS?

We have provided a literature review of existing definitions of diversity. We have analyzed these definitions based on the system behavior they measure and whether this behavior is linked to the expectations that users or vendors have from a RS.

Although the literature provides many diversity definitions and their comparisons [24], these studies do not consider the scalability of the measures, which is an important constraint of a real-world RS meant to serve millions of users and vendors [42]. Moreover, RSs are utilized by users over a long period so the temporal aspect of diversity is also an important consideration. There are existing definitions to measure temporal diversity in RSs but they do not measure diversity by item representations [25]. To address this gap, we have proposed a novel metric called "Temporal Inter-list Diversity"

1.3 VAE-Based User Profile Generation

VAEs can be used to learn a distribution over user profiles. We propose a method to improve diversity at the user representation level by sampling multiple random user profiles from the learned distribution and combining the recommendations generated by these user embeddings. The balance between the generative and discriminative nature of VAEs results in user profiles that produce relevant recommendations. The sampled user profiles are also different enough to yield diverse results. To our knowledge, we are the first to use VAE-based multiple user profile generation to improve diversity in RSs while maintaining an adequate level of relevance.

1.4 Comparative study between representing user as a point estimate vs a distribution

Methodology

We evaluate our proposed method in an experiment. We use a vanilla autoencoder as a baseline, where a user is represented by a single latent vector. Such models are popularly used in real-world RS. We compared the recommendations generated by our approach to the ones generated by the vanilla autoencoder on benchmark and real-world datasets. These recommendations were evaluated on the selected diversity and relevance metrics. Additionally, we evaluate the sensitivity of the method towards changes in certain parameters and how this affects the potential trade-off between diversity and relevance.

Results

We found that our method improved the diversity of recommendations based on the selected metrics for both datasets, at a minimal cost in relevance.

Through the parameter sensitivity analysis, we found that the method was most sensitive to changes in dropout, followed by the number of vectors sampled from the distribution and the size of the user profile. We also found that the method's sensitivity depends on the dataset properties like sparsity. Our results validated that a diversity-relevance tradeoff exists in terms of individual measures of diversity. However, on some occasions, aggregate diversity decreases along with accuracy. This could happen if the RS recommends safe items which are popular but not personalized.

1.4.1 Academic Contribution

The academic contributions of this study are as follows -

- A literature review to evaluate how diversity should be measured holistically in real-world RSs. Unlike existing research, we have evaluated existing metrics on scalability, which is one of the major constraints in real-world RSs. We have also chosen measures to address the needs of different RS stakeholders.

- A new metric to evaluate the representational difference between items in recommendation lists generated for a single user over multiple timestamps. Unlike existing metrics that only measure the total number of items that are different over time, our metric measures the representation distance between item lists, using embeddings from pre-trained models.

- A novel method to generate multiple user profiles by sampling from a distribution learned from a VAE. To our knowledge, no other method has been proposed in the literature to generate multiple user profiles which are meaningful and can be combined to improve RS outcomes. We have also done a parameter sensitivity analysis that reports the impact of changing different parameters on the accuracy and diversity of the recommendations generated by our proposed method.

- A novel experimental finding that representing users as a distribution rather than a point estimate can help improve user and item level diversity. Different from existing research which shows theoretically that multiple vectors can decrease stereotyping in RSs, we empirically show that a distribution-based user profile could improve recommendations on multiple dimensions of diversity while maintaining an adequate level of accuracy.

1.5 Outline

This thesis is structured as follows - Chapter 2 provides background information about the concepts used in this study like RSs and Collaborative Filtering techniques like matrix factorization and neural collaborative filtering. Chapter 3 describes the research problem we are tackling, defines the RQs being answered in this thesis, and the research framework used to answer these questions. Chapter 4 motivates the need for diversity in RSs and highlights the reason behind the lack of diversity in RSs. This section also answers the first RQ of this study by evaluating existing diversity measures based on their feasibility for a real-world RS and proposing a new metric to evaluate diversity temporally. Chapter 5 describes different methods for improving diversity in RSs, followed by a motivation for using VAEs for improving diversity. It also describes the background of VAEs and the novel method proposed in this thesis to improve recommendation diversity using VAE-based user profile generation. Finally, Chapters 6 and 7 describe the two experiments conducted to answer the remaining RQs mentioned in Chapter 3, along with the results of these experiments. The thesis is concluded with the final Chapter 8, which provides a summary and the limitations of this study.

Chapter 2

Background

This chapter will introduce the background knowledge about RSs and the conventional algorithms used to create these systems. Section 2.1 introduces RSs, followed by Section 2.2 which describes collaborative filtering algorithms including matrix factorization and neural collaborative filtering.

2.1 Recommender Systems

RSs were first proposed in the mid-1990s to help online consumers deal with an exponential increase in choices and to improve their experience when browsing for books, movies, or news on the internet [35]. They can be divided into three categories - Content-based filtering, collaborative filtering, and hybrid approaches. *Content-based filtering* approaches utilize the user's past item preferences to make recommendations in the future. In contrast, *collaborative filtering (CF)* approaches make item recommendations based on similar users' preferences. Content-based models suffer from a lack of item diversity as the recommendations are only based on the current interests of the user [9]. CF algorithms, on the other hand, recommend items to users based on preferences by similar users and hence can improve a user's experience by providing them a wider variety of recommendations [39]. These approaches have been widely adopted in the industry because of their advantages over content-based filtering. In this research, we will focus on the CF approach.

2.2 Collaborative Filtering

Collaborative filtering methods can be divided into the following categories - Memory-based and model-based methods [39]. *Memory-based* methods rely on past interaction data to recommend new items to users. In contrast, *Model-based* methods use past ratings to learn a user representation and make recommendations using these representations. Model-based approaches are more scalable than memory-based approaches [8], which is a crucial consideration in RSs due to the huge size of the datasets [42]. Hence, model-based methods have been more widely adopted in the industry [10]. Several implementations of model-based methods have been proposed in the research using approaches like matrix factorization and neural networks [39].

2.2.1 Matrix Factorization

Matrix factorization is a fundamental technique used for CF. It falls in the category of latent factor models where the items and users are characterized by vectors of common factors inferred from the rating information [23]. These models map the user and the item to a joint latent factor space of a certain dimensionality such that the user-item interactions are the inner products of the latent vectors in that space. Item i can be represented by vector p_i and user u can be represented by vector q_u. p_i represents item i's association with a certain feature. q_u denotes user u's association to a certain feature. Figure 2.1 provides a simple example of how users and items are mapped to a common latent space. The dot product of $q_i^T p_u$ expresses the interaction between users and items and captures the user's interest in certain item characteristics. This value is denoted by $\hat{r_{iu}}$ which can be estimated by equation 2.1.

$$\hat{r_{ui}} = q_i^T p_u \tag{2.1}$$

Earlier systems relied on filling the missing values in the rating matrix to retrieve latent factors from the rating data using singular value decomposition. More recently, systems learn latent factors by p_i and q_u directly by modeling the observed rating and optimizing the regularized mean squared error objective described in equation 2.2. In this equation, N represents the non-blank elements in the user-item matrix.

$$\frac{1}{N}\sum_{u,i}(r_{ui} - \hat{r_{ui}})^2 + \lambda(||q_i||^2 + ||p_u||^2) \tag{2.2}$$

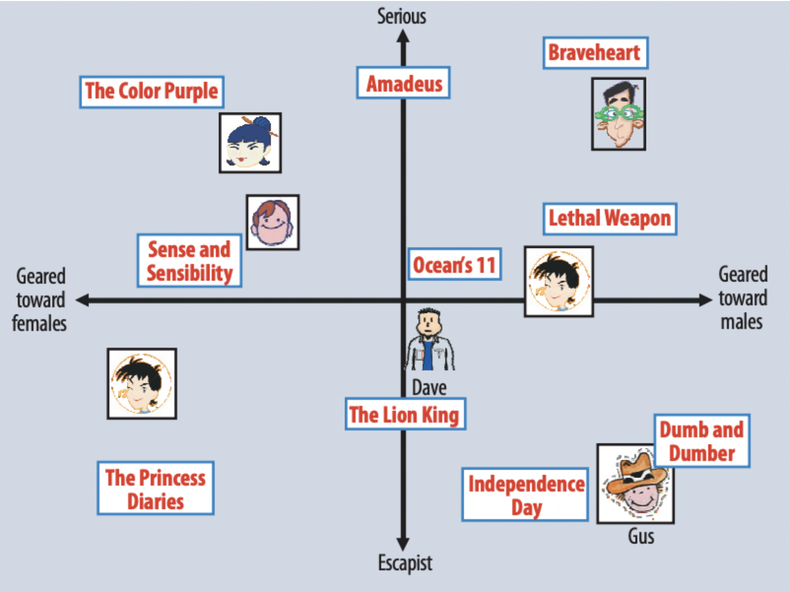


Figure 2.1: A simple example of how latent factor models work. in this example both users and items have been mapped to two features - serious or escapist, males or females [23]

2.2.2 Neural Collaborative Filtering

Matrix factorization techniques rely on an inner product of the latent factors to generate the rating matrix. But, an inner product can only capture the linear relationships. *Neural collaborative filtering* (NCF) algorithm was proposed to enhance the capability of matrix factorization techniques. These algorithms replace the inner product with a neural network to learn the interactions between users and items [23]. NCF provides a generalized framework that can model linear and non-linear relationships using a multi-layer perceptron. NCF is a fundamental algorithm based on which many deep learning-based CF algorithms were proposed.

2.3 Summary

In this section, we have described RSs and, more specifically, CF approaches. We have expanded on the principle behind CF algorithms and described the working of two popular CF algorithms.

Chapter 3

Research problem

This chapter will describe the context in which the problem occurs by explaining the application setting and problem relevance. It will expand on the process of building an RS and the requirements of a real-world RS. It concludes with the problem definition, RQs, and the research framework that has been adopted to answer the RQs.

3.1 Problem Context

This study proposes and evaluates a method that can help improve the diversity of RSs. This project is done in collaboration with Bol.com, the largest online retail platform in the Benelux region. The goal of the project is to improve diversity in RS, while maintaining relevance, and create a fairer marketplace for both customers and vendors while improving business outcomes.

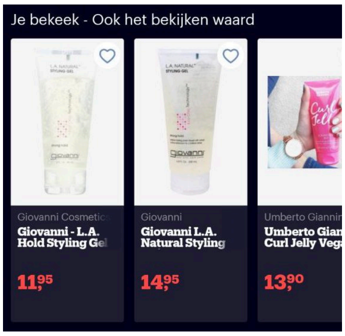
3.1.1 Application Setting

Bol.com is an eCommerce platform supporting millions of customers, and thousands of vendors [30]. The platform has thousands of customer journeys every single day. Different customers want a different experience from their online retailer; some know what they want and start their journey by searching for a specific product on a search engine which leads them to a specific product page. While, others have a general idea about what they want, like a gift for a birthday party, or they are trying to find the best deals on the products of their interest. The application's home page is catered toward helping customers in an exploratory phase of their customer journey. This page aims to ensure that these customers get recommendations that can be relevant to their interests while helping them discover new product categories that could be meaningful to them.

On this journey, following are some of the commonly raised customer complaints -

- "Why are the items in the recommendation list so similar to each other?"
- "I bought this once, but why is the same thing recommended to me every time I visit?"

The first complaint is about the items in each recommendation list being too similar, as shown in Figure 3.1. The second complaint is about the items being too similar between two recommendation lists generated over different visits for the same user. This issue can be reflected in Figure 3.3. The recommendations in both the user sessions are from the same category of hair products, and within each session, the products are not only from the same category but also the same brand. These examples show a lack of diversity in recommendations served by the Bol.com mobile application. A large number of near-duplicates in the product catalog and the focus on relevance while training reduces diversity in RSs.



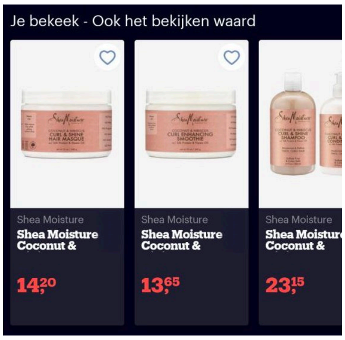
Je bekeek - Ook het bekijken waard

Giovanni Cosmetic
Giovanni - L.A. Hold Styling Gel
11,95

Giovanni
Giovanni L.A. Natural Styling
14,95

Umberto Giannin
Umberto Gian Curl Jelly Veg
13,90

Figure 3.1: Recommendations at time t



Je bekeek - Ook het bekijken waard

Shea Moisture
Shea Moisture Coconut &
14,20

Shea Moisture
Shea Moisture Coconut &
13,65

Shea Moisture
Shea Moisture Coconut &
23,15

Figure 3.2: Recommendations at time t+1

Figure 3.3: 1) These images show the recommendations provided to the author in the Bol.com mobile application over two user sessions. 2) The items in both sessions fall in the hair products category. 3) The items in each session are not only from the same category but also the same brand - 'Shea moisture' for recommendations at time t and 'Giovanni' for time t+1. Images by the author.

3.1.2 Problem Relevance

Each user session is an opportunity to help the user find a relevant set of items. A user sees a list as a whole and showing recommendations that are too similar to each other limits the user experience and is a lost business opportunity as well. Therefore, it is important to evaluate RS's performance on diversity within each session.

A crucial consideration often missed in RS research is that users use and evaluate these systems over time. If a user sees some recommendations in one session and similar recommendations are repeated in the next session, this reflects a lack of temporal diversity in RS. This behavior has been shown to reduce users' trust in the system due to decreased utility and dampen their experience [25]. Hence, it's also vital to provide diverse recommendations over multiple sessions.

Apart from customers, the vendors also suffer from this lack of diversity in recommendations. Popular items by more prominent vendors have more interaction data associated with them. This leads to smaller vendors and niche products getting recommended to fewer customers [2] [3]. For example - movies by bigger production houses like Walt Disney are more popular. They have a lot of interaction data associated with them compared to smaller production houses like Spyglass Entertainment. This creates and, over time, exacerbates the feedback loop and long tail problem in RSs [28]. Providing user exposure to a large number of items will provide opportunities to more vendors and might also positively impact sales [47]

Lack of diversity affects different stakeholders of a RS differently and hence in this thesis, we aim to address all these issues associated with lack of diversity in RSs and evaluate the proposed method on these criteria.

3.1.3 Recommendation Process

RSs have the following three stages (visualized in Figure 3.4) -

- **User profile generation** - In this stage, users are represented in a vectorized format so that the representations of similar users have a shorter distance between them. Distance can be measured by different metrics like cosine or hamming distance. This representation can be created with user features like -

  - Demographic information: Age, location, gender, etc.
  - Usage history: User clicks, purchases, the time spent on a product page
  - User feedback: Ratings, reviews, link copying, and sharing as described in [24]

  Lately, deep neural networks are being used to generate generalized user representations that capture the deeper relationships between individual features. Unsupervised or self-supervised methods can be used to train these representations. Richer representation of the user can help improve the quality of recommendations. Numerous questions can help us understand whether a representation is richer, for example -

  - Is it capturing more of the users' interests?
  - Is it able to capture the characteristics of the user? For example - is the user interested in top deals?
  - Does it capture the level of clarity the user has? For example, the user is looking for a particular book compared to a science fiction book.

  User profile generation is a time-consuming step and is often performed offline. The representations are generated and stored in an index to be used efficiently in real-time systems.

- **Recommendation generation** - This stage would generally include the following steps -

  - **Candidate Generation and Ranking** - The goal of this step is to generate a list of candidate items that could be relevant to the user and rank them. This step could also be combined with the user profile generation step. If performed separately, a user profile is retrieved to generate recommendations for a user, and items are ranked based on their similarity to the user profile and many other features (here ranking algorithms are also used to re-rank the generated candidates). A smaller list of K items is generated from the ranked list. The value of K depends on the use case, but it lies in the range of 5 - 50 items depending on the page size, structure, and the exact use case for the recommendations.
  - **Post-processing** - Post-processing is generally used to optimize for objectives like diversity or novelty. Techniques like Maximum Marginal Relevance (MMR) [12] or selecting items from unique genres make lists more diverse. This step adds to the time complexity of the recommendation generation stage.
  - **Applying Business Rules** - Apart from generating good recommendations, certain business rules must be applied to ensure the lists adhere to the business policies. For example - Products that the company does not support like firearms, and the ones that are no longer in stock have to be filtered out.

  This stage is also time-consuming because it involves comparing the user representation to all the item representations for candidate generation. User and item profiles are stored in an index to make the comparison more efficient.

- **Feedback Collection** - This is the final and most crucial stage of a RS because the algorithms rely heavily on the feedback the users provide. The feedback can be explicit or implicit. Explicit feedback includes users rating the items or leaving reviews about their experience with the product. Implicit feedback is more subtle and is collected based on the user's behavior on the website. For example - if a user purchased an item or clicked on a product, it could be implicitly assumed that the item was relevant to them. Although explicit feedback is generally more accurate, implicit feedback is not obstructive to the user's experience and hence, is more widely adopted.

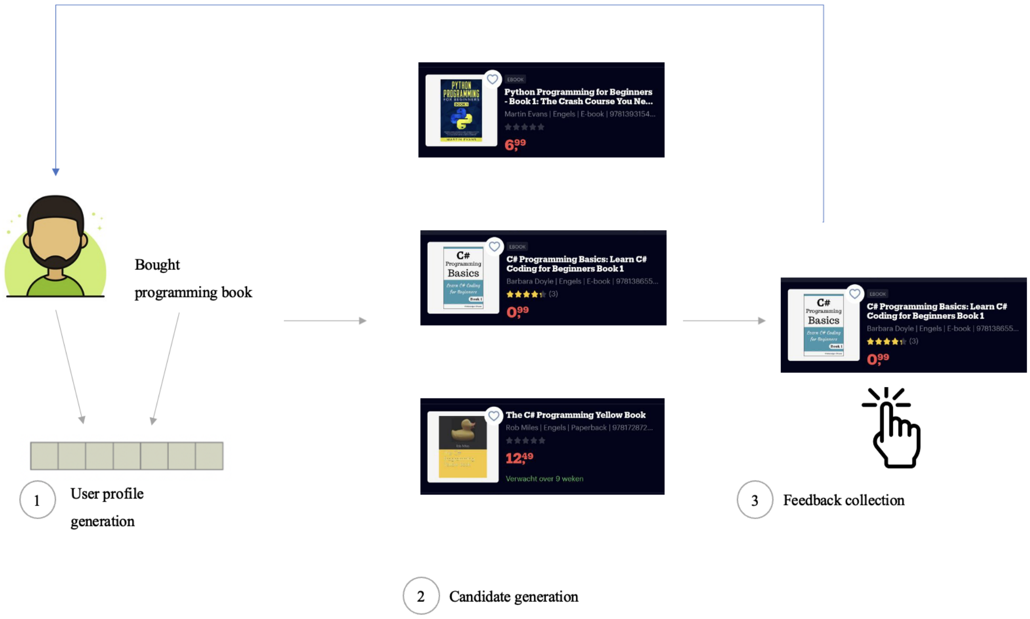

EBOOK
Python Programming for Beginners - Book 1: The Crash Course You Ne...
Martin Evans | Engels | E-book | 9781393154...
6,99
Bought
programming book
EBOOK
C# Programming Basics: Learn C# Coding for Beginners Book 1
Barbara Doyle | Engels | E-book | 978138655...
(3)
0,99
EBOOK
C# Programming Basics: Learn C# Coding for Beginners Book 1
Barbara Doyle | Engels | E-book | 978138655...
(3)
0,99
The C# Programming Yellow Book
Rob Miles | Engels | Paperback | 978172872...
12,49
Verwacht over 9 weken
1 User profile generation
3 Feedback collection
2 Candidate generation

Figure 3.4: An example of the three stages of recommendation systems - 1) User profile generation - A user purchased a programming book in the past; this information is used to generate his profile, 2) Candidate generation and ranking (learning-to-rank) - Based on his profile, a ranked list of items is generated based on the similarity with the user profile, 3) Feedback collection - The user clicks on the first item, and this action is recorded as feedback by the system. Image by the author.

3.1.4 Requirements

Within the problem context, the proposed method should adhere to the following requirements -

- *Relevant* - The recommendations made by the system should be relevant to the user's interests. For example - if the user is interested in sports, recommending a basketball is relevant to them.
- *Diverse* - There are many ways of defining diversity, and we aim to improve diversity from user and item perspectives using the metrics defined in Section 4.3.1. The goal is to improve the business objectives of selling more products and enhancing the user experience.
- *Scalable* - Given that Bol.com had 12.83 million active customers and 38.20 million products [30] in 2021, both of which are continuously growing, the scalability of the method is critical to ensure that it is practical and can be adopted in real life. Scalability is congruous to low time complexity in offline and online steps. It should be an important consideration not only for the algorithm but also for the metrics to be used to measure relevance and diversity in RSs to ensure that it is feasible to evaluate these systems at scale.

These three features ensure that the proposed method can provide a great user experience and can be implemented feasibly within a real-world RS.

3.2 Problem definition and Research Questions

Problem: Improve user and item level diversity in RS by making changes at the user representation level, while maintaining an adequate level of relevance.

We aim to propose a method to achieve this goal and to evaluate if our method of generating user profiles makes a difference by answering the following research question -

"Can representing users as a distribution rather than a point estimate improve the user and item level diversity in RS while maintaining an acceptable level of relevance?"

We have formulated five RQs to answer the main RQ of this study and described the corresponding research frameworks that will be used to answer each question.

Firstly, we need to understand how diversity should be measured in an RS. As described in the problem context, diversity can be operationalized in different ways when viewed from the perspective of distinct stakeholders. Several summaries and evaluations of diversity metrics have been provided in the literature, but it is still unclear how diversity can be measured holistically to address all the stakeholders and which measures are feasible in a real-world RS evaluation. This motivates the first question -

RQ1 - *"How should diversity be measured holistically in a real-world RSs?"*

This question has been answered by reviewing existing diversity metrics in the literature and evaluating them against the expectations (described in Section 3.1) of the different RS stakeholders - users and vendors. In contrast to other studies, we have also evaluated these metric definitions on scalability as it is a major constraint in real-world RS. The advantages and disadvantages of different metrics have been discussed to motivate whether they should be used in this research.

Based on Section 3.1, we have chosen to evaluate RS diversity in the following ways - 1) How diverse are the recommendations within each list? 2) How diverse are the recommendations over time? and 3) What is the ratio of total recommended items to the entire item catalog?

After deciding how diversity should be defined in RSs, we describe our method for improving diversity in RSs. Our method is focused on the user profile generation stage. It is one of the most critical steps of the recommendation system. If the profile fails to capture sufficient information about the user, it heavily limits the ability of RS to produce a diverse product list that represents users' multiple interests in the following stages of the process. Different from existing RS diversity research which focuses on other stages of an RS, this research aims to study if an unconventional user profile generated by our method can improve item and user-level diversity in RS. Questions 2,3, and 4 evaluate whether our approach can holistically improve diversity (over the metrics defined in the previous step) over a conventional RS user profile represented by a single vector.

Diversity within each recommendation list helps optimize the space on the website and contributes to a good user experience in every user visit. In a conventional CF algorithm, a single user profile is used to create the entire recommendation list leading to the user complaints mentioned in Section 3.1. This motivates us to the second question -

RQ2 - *"Can representing users as a distribution rather than a point estimate improve user-level diversity of recommendations generated within a single session while maintaining an acceptable level of relevance?"*

As described in the recommendation process, user profiles are generated in longer intervals as it is a time-consuming step. The generated user profile is used repeatedly in intervals (possibly a week). This results in a static user profile being used during the interval to generate recommendations, making users feel that the recommendations are stale or boring. This frustration is reflected in the user comments mentioned in Section 3.1. This motivates the third question -

RQ3 - *"Can representing users as a distribution rather than a point estimate improve user-level diversity of recommendations generated over multiple sessions while maintaining an acceptable* *level of relevance?"*

Improving diversity from the user's perspective while sufficiently maintaining relevance should also help alleviate the long-tail problem in RSs [13] by exposing more products to the users overall. This objective has a two-fold impact. It makes the system fairer for vendors. It optimizes a business objective of recommending more items overall as that has been linked to higher sales. This brings us to the next question -

RQ4 - *"Can representing users as a distribution rather than a point estimate increase the total number of products recommended to all users while maintaining an acceptable level of relevance?"*

We aim to answer questions 2,3 and 4 to evaluate the performance of our method. We conducted a comparative study between recommendations generated by a single user profile and multiple user profiles sampled from a distribution generated by the VAE. We evaluated the recommendations by the chosen metrics to determine whether our method can improve user and item level diversity, while maintaining an adequate level of relevance, for benchmark and real-world datasets. We also evaluated the results for different list sizes to analyze the impact of list size on the results.

The method contains several parameters like user profile size, the number of vectors sampled from the distribution, and dropout rate. As diversity and relevance are both important requirements for the system, it would be useful to evaluate the impact of tuning different parameters on the potential diversity-relevance tradeoff. This analysis can help practitioners tune the parameters according to their requirements. This motivates the final question -

RQ5 - *"How does changing different parameters in the system impact the potential diversity-relevance tradeoff?"*

This question has been answered by observing and reporting the change in the diversity and relevance metrics as we change different parameters like the size of the latent vector, the number of vectors sampled from the distribution, and the regularization parameter. This study can help practitioners tune the system parameters based on their need for diversity vs. relevance.

3.3 Summary

This chapter described the problem context and defined the research framework we have used to address the problem. Section 3.1 explained the application setting that the problem is focused in and the relevance of this problem. Furthermore, it elucidated the recommendation generation process and concluded by providing the requirements of the proposed solution. Section 3.2 defined that the scope of this thesis is to improve user and item level diversity in RSs at a user representation level. It established the research questions that will be answered by this thesis and the research framework that has been used to answer those questions.

Chapter 4

Diversity in Recommender systems

RSs emerged as a way to help users find relevant information as online item catalogs increased in size. The role of RSs in online decision-making has evolved over time. Nowadays, users view RSs as more than a way to find relevant items from a large collection. They expect these systems to adapt and evolve as they do, and to broaden their interests. This can be achieved through diversity in RSs. Diversity is defined as the condition of having or being composed of differing elements. Users of existing RSs often find the recommendations to be too similar to their historical purchases or too homogeneous within each list. We can observe this from user complaints describing these systems as being “too naive” or “not working” [25].

4.1 Motivation for diversity in RS

4.1.1 Why do users prefer higher diversity in RS?

Knijnenburg et al. [22] describe the relationship between factors like relevance and diversity and how they fit into the overall user experience for a recommendation system. Their experiment verifies that there is a positive relationship between user satisfaction and diversity, which is due to lower choice difficulty and greater perceptions of system effectiveness. The perceived relevance is generally higher in diverse lists because users evaluate them as a whole, not individually, and these diverse recommendations better represent their diverse interests.

These results were shown for individual lists. But, it can be assumed that user satisfaction would be related to diversity similarly over time as well. The surveys conducted in [25] show that 86% of users want recommendations to change over time. Temporal diversity is essential because it gives a perception to users that the system is evolving and understanding their interests better over time.

4.1.2 How does item diversity affect sellers in RS?

In addition to users, increased item diversity also benefits the platform and the sellers on it. Aggregate diversity is a system-level diversity measure described as the proportion of the total items recommended to all users with respect to the entire item set. Lower aggregate diversity means that the system recommends only a small percentage of the total item list to the users. Thompson and Clive [43] provide evidence that RS algorithms often try to find popular items purchased by many users. Hence, they avoid extremes and recommend popular and safe items to users. This leads to a long tail problem, where most items are not exposed to users, or a considerable proportion of sales come from some selected items [13]. Exposing users to more items can be beneficial for increasing sales [2]. Brynjolfsson et al. estimate that 36.9% of Amazon's sales come from titles outside the top 100000 [11].

Recommending less popular items allows smaller vendors or producers to attract new customers. "Dancing from Danang" was an unpopular movie, which rose to the list of top 15 Netflix documentaries because of the algorithm [3]. This shows how RSs can help obscure items reach broad audiences by recommending from the long tail. Similarly, festival films that often have a smaller audience than blockbusters can also find an audience by improving the aggregate diversity of recommendations. Users interacting with obscure items can lead them to more obscure items. This effect has been illustrated by another example in [3]. Rhapsody (music RS) features Britney Spears on the main screen. Users who listen to "Britney Spears" are recommended similar artists like "Pink". If they listen to "Pink" and enjoy it, they might also like "No Doubt". This might lead them to "Selecter", a 1980 Coventry ska band. Users can reach from chart labels to albums that cannot even be found in record stores in a few clicks. This shows how a slight increase in aggregate diversity can snowball, leading to more opportunities for vendors and users looking for niche content. High temporal diversity can also help to spread user interactions across more products. If users are exposed to different items in each user session and interact with more items overall, it will help the system gain more information about the users over time, leading to richer user representations.

4.2 Why is there a lack of diversity in RS?

There are two possible reasons behind the lack of diversity in RS - focus on accuracy in algorithms and skewed underlying data causing feedback loops which further exacerbates the bias.

4.2.1 Focus on accuracy leads to lack of diversity in RS

Most CF-based algorithms are optimized for relevance. The experiments conducted by Fleder and Hosanagar showed that RSs reduce diversity because they are focused on improving the accuracy [15]. Moreover, a real-world RS contains a lot of near duplicates and hence, a list of highly relevant items can have many redundant items. Low diversity in RSs can lead to low user satisfaction [22]. Users may feel stereotyped by getting similar recommendations again and again [17].

4.2.2 Feedback Loops caused by skewed data result in lack of diversity

Recommendations generated by CF algorithms often suffer from biases that may stem from biases in the underlying data. Figure 4.1 shows that most of the item interaction data in MovieLens (a popular benchmark dataset) is concentrated among a small percentage of popular items [28]. These biases in data lead to a higher recommendation rate for more popular items, leading to more interaction data for these items within each user profile [13]. This creates a feedback loop between user profiles and recommendations. Chaney et. al. further show that the feedback loop causes a shift in consumption and homogenization of the user experience, making the system less beneficial for the users [13].

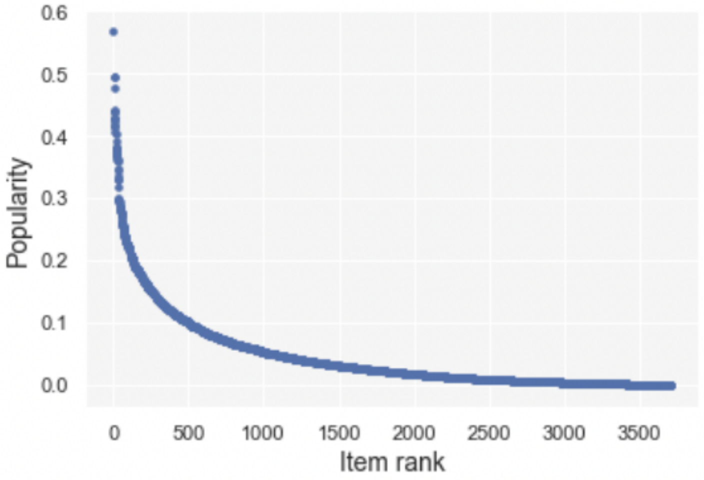


Figure 4.1: Item popularity in MovieLens [28]. Most of the interaction data is concentrated within a small percentage of items

4.3 How can diversity be measured in RS?

Defining diversity is a complex topic because diversity in RSs could mean different things for different people. Some people might consider Harry Potter and Percy Jackson to be very different books, while others might consider them to fall into one category - fantasy novels. Most of the research on diversity metrics can be divided into individual and aggregate level definitions of diversity. Individual definitions aim to provide a metric to quantify how diverse the recommendations are for each user. Alternatively, aggregate diversity definitions seek to quantify the total number of items that users are exposed to. Individual-level metrics focus on each user's satisfaction, and aggregate diversity metrics focus on vendor and business opportunities. Using multiple metrics can help us understand how the proposed method performs for multiple dimensions of diversity in RSs. Existing studies that compile different diversity measures do not assess them against the needs of the users or sellers or provide any guidance to practitioners about which subset of these measures can be used to holistically evaluate RSs. They also do not evaluate these measures against real-world constraints like scalability.

This section will be used to answer the first research question described in Chapter 3 -

RQ1 - *"How should diversity be measured holistically in real-world RSs?"*

4.3.1 Existing Diversity Metrics

The following subsections describe the diversity measures defined in existing studies.

Intra-list Diversity

Many individual-level definitions have been proposed in research. Most of these definitions build upon the seminal work by Bradley and Smyth [9]. They defined diversity as the opposite of similarity and measured it as the dissimilarity between all item pairs in the result set. This work was defined in the context of an information retrieval system. However, it can be applied to RSs, when considering user profiles to be analogous to queries in search engines. Bradley and Smyth, Zhang et al., and Ziegler et al. [36] [48] [50] have explored this perspective and provided a popular metric called intra-list diversity (ILD). ILD is the total distance between each pair in the recommended list averaged over all users. Distance can be measured by common distance (opposite of similarity) metrics like cosine distance and manhattan distance. Equation 4.1 defines ILD for one user where L is the recommended list for one user. This measure is normalized by the size of the list. It is not rank-aware and has a time complexity of $\mathcal{O}(k^2 \times U)$

$$ILD = \frac{1}{|L|(|L|-1)} \sum_{i,j \in L} dist(i,j) \tag{4.1}$$

Temporal Diversity

Lathia et al. considered another way of measuring individual diversity, which is over time. They introduced a measure called temporal diversity that could evaluate how the recommendations made to the users are changing over time [25]. The formal definition proposed in the research has been described in Equation 4.2 where $L1$ is the recommendation list generated at time t by the model for a user, and $L2$ is the recommendation list generated at time $t+1$ for the same user. This research raises the vital issue that RSs are dynamic and must evolve continuously for a pleasant user experience. They show through different studies that users prefer RSs with higher temporal diversity. $L2/L1$ in equation 4.2 is the set difference between the recommendation list generated between time $t+1$ and time t. k is the size of the recommendation list. This measure is not rank-aware, has been normalized by the recommendation list size, and has a time complexity of $\mathcal{O}(k \times U)$.

$$Temporal\ Diversity = |\frac{L2/L1}{k}| \tag{4.2}$$

Aggregate Diversity

Adomavicius et al. presented the idea of aggregate diversity, which measures the total number of items in the union of all the lists recommended to all users. Equation 4.3 shows the definition as described in [2] where $L(u)$ is a recommendation list for user u and U is the total list of users. Aggregate diversity might not increase when increasing individual diversity as the same five items recommended to one user might be recommended to all users. Increasing aggregate diversity can help tackle popularity bias and the feedback loop problem in RSs. This measure can be used in combination with individual diversity as one focuses on the business outcome of the number of products recommended. In contrast, the other focuses on individual users' experience [2]. This measure is also not rank-aware. It is also not normalized and has a time complexity of $\mathcal{O}(I)$

$$Aggregate\ Diversity = |\bigcup_{u \in U} L(u)| \tag{4.3}$$

Joint accessibility

Guo et al. took a different perspective on diversity and defined a concept of joint accessibility wherein they measured the proportion of item sets that can be jointly recommended to a user [17]. A RS is considered jointly accessible if, for all item sets of size k, there exists a user to whom that set can be recommended. This measure aims to understand whether users with niche interests that are uncommon among the majority will get recommendations that represent their interests. Joint accessibility has high complexity as we need to consider all subsets up to size K to measure it. To tackle the complexity issue, Guo et al. [17] evaluated joint accessibility only on the top 400 items and 79800 associated item pairs. Using a metric that is only evaluated on such a small sample of the whole population might lead to biased insights. This metric is not rank aware or normalized. It also has a high time complexity of $\mathcal{O}(2^k \times U)$.

Table 4.1 provides a comparison of the different diversity metrics. k is the size of the recommendation list, U is the total number of users, and I is the total number of items.

Metric / Criteria	**Explanation**	**Normalization**	**Time Complexity**
ILD	Dissimilarity between all item pairs in a recommendation list	Yes, by the size of the recommendation list	$\mathcal{O}(k^2 \times U)$
Temporal Diversity	Proportion of different items recommended over time wrt to the list size	Yes, by the size of the recommendation list	$\mathcal{O}(k \times U)$
Aggregate Diveristy	Total items recommended to all users	No	$\mathcal{O}(I)$
Joint Accessibility	Fraction of item sets that can be recommended together out of all possible subsets of size k	Yes, by possible subsets of size k	$\mathcal{O}(2^k \times U)$

Table 4.1: Comparison between existing diversity metrics

4.3.2 New diversity metric: Temporal inter-list diversity

To measure whether the list is diverse over multiple sessions, we need to evaluate the diversity between lists generated in two user sessions. Temporal diversity only calculates the number of items that are different between sessions. Hence, it does not help us identify whether the items are diverse in terms of representations or if they are just near duplicates. We propose a new metric called *Temporal Inter-list Diversity* (TILD), which is the total pairwise distance between items of two different recommendation lists ($L1$: recommendation list at time t=0 and $L2$: recommendation list at time t=1 of the same size) generated in separate sessions/timestamps. Equation 4.4 describes the formula of TILD for one user where *dist* could be any appropriate distance measure like cosine or hamming distance. This metric is normalized by the list size and has a time complexity of $\mathcal{O}(k^2 \times U)$.

$$TILD = \frac{1}{|L1|(|L1|-1)} \sum_{i \in L1, j \in L2} dist(i,j) \tag{4.4}$$

4.3.3 Selected diversity measures

To measure diversity holistically, we have decided to consider both the user and vendor perspectives. From the user perspective, we have chosen to evaluate the RS on how diverse the recommendations are in a single list and over time as these are the diversity expectations users have from RSs. From the vendor perspective, we have chosen to evaluate how many of the total items are getting user exposure from the entire item catalog. Evaluating these dimensions of diversity helps us understand if the RS is providing a positive user experience but also quantifies how fair the system is for the vendors. This creates a holistic evaluation of the system from the perspective of all stakeholders.

The user-level diversity metrics that we have chosen to focus on are ILD (measures representational diversity within items in a single list) and TILD (measures representational diversity of items in lists over time). These metrics are directly related to the user frustrations mentioned in Chapter 3. Regarding item-level diversity, aggregate diversity metric proposed by Adomavicius et al. [2] is the right fit as it helps us understand the proportion of items that get user exposure in each run of the model. All these measures are also scalable compared to metrics like joint accessibility, which has a time complexity of $\mathcal{O}(2^k \times U)$. ILD and TILD have a time complexity of $\mathcal{O}(k^2 \times U)$ and aggregate diversity has a time complexity of $\mathcal{O}(I)$. This allows us to use these measures over large datasets as well.

4.4 Summary

This chapter tackled the topic of diversity in RSs. Firstly, we motivated higher diversity in RSs and described how it affects the different stakeholders. Secondly, we explained the reasons behind the lack of diversity in conventional RSs algorithms. Finally, we concluded the chapter by describing the different ways diversity can be measured in RSs and proposed a new metric to measure temporal diversity representationally.

Chapter 5

VAE for Diversity in Recommendation Systems

User profile generation is a crucial stage of the recommendation process. Without sufficient information about the user in the profile, the following stages would fail to produce relevant and diverse recommendations. Studies have theoretically proven that representing a user by multiple user profiles can represent their varied interests better than a single user profile [17]. In this chapter, we will propose a method to generate multiple user profiles representing a user by employing VAEs to improve diversity in RSs.

This chapter will introduce different techniques for improving diversity in RSs in Section 5.1, followed by Section 5.2 listing case studies where VAEs have been used to enhance diversity in natural language processing tasks. Section 5.3 describes the theoretical underpinnings of VAEs. Finally, Section 5.4 proposes our novel method to improve diversity in RSs using VAEs.

5.1 Techniques to improve diversity in RSs

Several approaches have been introduced to improve diversity in RSs while limiting the decrease in relevance. Section 4.1 describes why this is a desirable outcome for RSs. The approaches can be divided into post-processing and algorithmic techniques.

5.1.1 Post-processing techniques

Post-processing techniques are employed after the candidate selection stage in RSs. Bradley & Smyth proposed a post-processing technique where they used a generic recommendation process to obtain $b \times k$ candidates, where k is the size of the final recommendation list and b is a constant [9]. Then, they selected k items from this list based on diversity ranking. Ziegler et al. added candidate items to the recommendation list based on a dual objective. They defined a dissimilarity rank for each item in the recommendation list with respect to other items and combined it with a relevance rank using a weight parameter to include items in the recommendation list according to diversity and relevance [51]. These works have been inspired by Maximum Marginal Relevance (MMR) introduced in [12]. This approach ranks items according to relevance and adds them to the list based on the diversity with respect to the items already present in the recommendation list. Adomavicius & Kwon also proposed a similar re-ranking approach where candidate items are selected based on a rating threshold and added to the recommendation list based on a rank derived by considering diversity and relevance. Aytekin and Karakaya proposed an approach called ClusDiv, where items are clustered into groups, and cluster weights are learned for each user. Finally, recommendations are generated from different clusters to increase the diversity [7]. These approaches fall into the category of post-processing techniques. The advantage of these approaches is that they are independent of the algorithm used to generate the recommendation list. Therefore, they can be used flexibly with different candidate generation algorithms. The disadvantage is that diversification occurs after the prediction of items. If the predicted items were not very diverse, to begin with, the final list would not be very diverse either [24].

5.1.2 Algorithmic techniques

Hurley and Wasilewski proposed using regularization techniques in matrix factorization-based CF algorithms to optimize diversity in CF RSs. The regularization term was dependent on the item distances in the recommendation list. A differentiable version of a diversity metric is added as the objective to be optimized [45]. Liu, Shi, and Guo propose a random-walk-based CF algorithm that enhances the opinions of small-degree users in a user-user similarity graph. This approach reduces the dependence of CF approaches on high-degree users leading to an increase in diversity [27]. The advantage of these techniques is that diversification is a part of the recommendation generation algorithm. This ensures that the final list produced will be diverse. It is also possible to apply the above-mentioned post-processing techniques for further improving diversification. The disadvantage is that these techniques have a specific architecture or model and cannot be generalized across other methods.

In our method, we use both algorithmic and post-processing techniques to generate a diverse RS. In contrast to existing research, we address the problem at the earliest stage of the process - the user profile generation stage.

5.2 Motivation for using VAEs to improve diversity

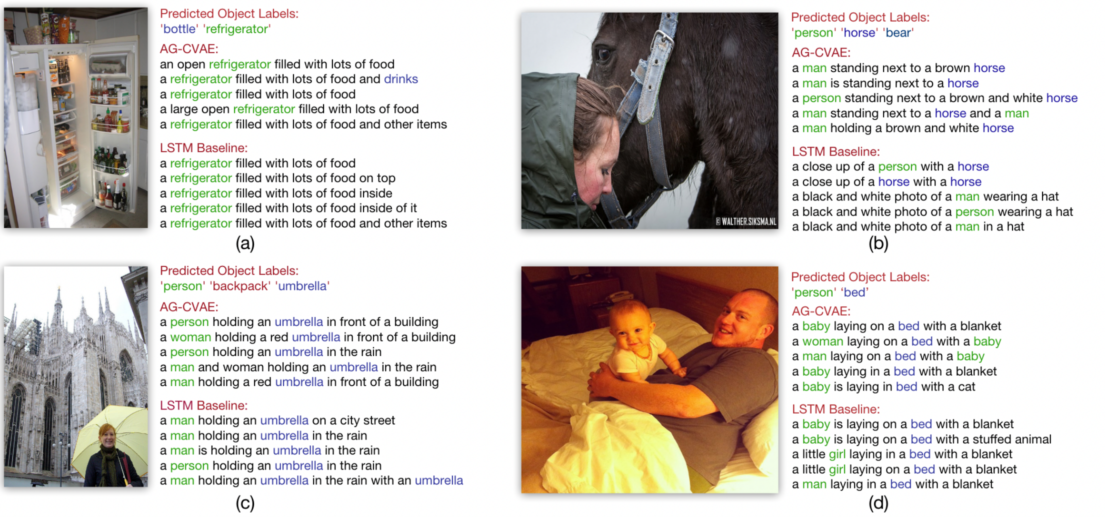
Significant strides have been made in research to represent data effectively (dense representations), and deep generative models are at the forefront of that research. VAEs are from a class of neural networks called *Autoencoders* (AEs) that learn dense data representations from the input in an unsupervised manner. In contrast, VAEs learn to map input data points to a latent distribution rather than a single point. They also belong to the class of generative models. Owing to their generative nature, these models have been used to improve diversity in natural language processing tasks (NLP) like caption generation and visual question generation. The following subsections describe different attempts at using VAEs to improve diversity.

5.2.1 VAEs to improve diversity in caption generation

Caption generation is a task with a high level of ambiguity as images can have various meanings, and multiple captions could fit any image. Hence, generating diverse captions can help encapsulate more information from an image for a highly descriptive result. Wang et al. [44] argued and proved that generating stochastic image representations using a generative model compared to a point estimate representation can help improve the diversity of captions generated. Figure 5.1 shows the difference between the captions generated by a VAE-based model and a non-generative LSTM-based encoder-decoder model. Using generative methods, Wang et al. could approximately double the total number of captions with respect to the existing ones in the training dataset.

5.2.2 VAEs to improve diversity in visual question generation

Jain et al. [19] have used a similar approach in visual question generation. Retaining the user's interest is essential for a question-answering task, which can only be achieved by continuously exposing users to varied questions. Visual question generation has similar attributes to recommendation generation as both tasks require a balance between relevance and diversity to keep the user engaged. To show that generative models can produce diverse yet accurate questions that capture the ambiguity in images, Jain et al. evaluated whether VAE could improve diversity in the question generation task over a non-generative model. The model was evaluated on traditional relevance metrics like BLEU and METEOR, along with intuitive diversity metrics like the average



Predicted Object Labels:
'bottle' 'refrigerator'

AG-CVAE:
an open refrigerator filled with lots of food
a refrigerator filled with lots of food and drinks
a refrigerator filled with lots of food
a large open refrigerator filled with lots of food
a refrigerator filled with lots of food and other items

LSTM Baseline:
a refrigerator filled with lots of food
a refrigerator filled with lots of food on top
a refrigerator filled with lots of food inside
a refrigerator filled with lots of food inside of it
a refrigerator filled with lots of food and other items

(a)

Predicted Object Labels:
'person' 'horse' 'bear'

AG-CVAE:
a man standing next to a brown horse
a man is standing next to a horse
a person standing next to a brown and white horse
a man standing next to a horse and a man
a man holding a brown and white horse

LSTM Baseline:
a close up of a person with a horse
a close up of a horse with a horse
a black and white photo of a man wearing a hat
a black and white photo of a person wearing a hat
a black and white photo of a man in a hat

(b)

Predicted Object Labels:
'person' 'backpack' 'umbrella'

AG-CVAE:
a person holding an umbrella in front of a building
a woman holding a red umbrella in front of a building
a person holding an umbrella in the rain
a man and woman holding an umbrella in the rain
a man holding a red umbrella in front of a building

LSTM Baseline:
a man holding an umbrella on a city street
a man holding an umbrella in the rain
a man is holding an umbrella in the rain
a person holding an umbrella in the rain
a man holding an umbrella in the rain with an umbrella

(c)

Predicted Object Labels:
'person' 'bed'

AG-CVAE:
a baby laying on a bed with a blanket
a woman laying on a bed with a baby
a man laying on a bed with a baby
a baby laying in a bed with a blanket
a baby is laying in bed with a cat

LSTM Baseline:
a baby is laying on a bed with a blanket
a baby is laying on a bed with a stuffed animal
a little girl laying in a bed with a blanket
a little girl laying on a bed with a blanket
a man laying in a bed with a blanket

(d)

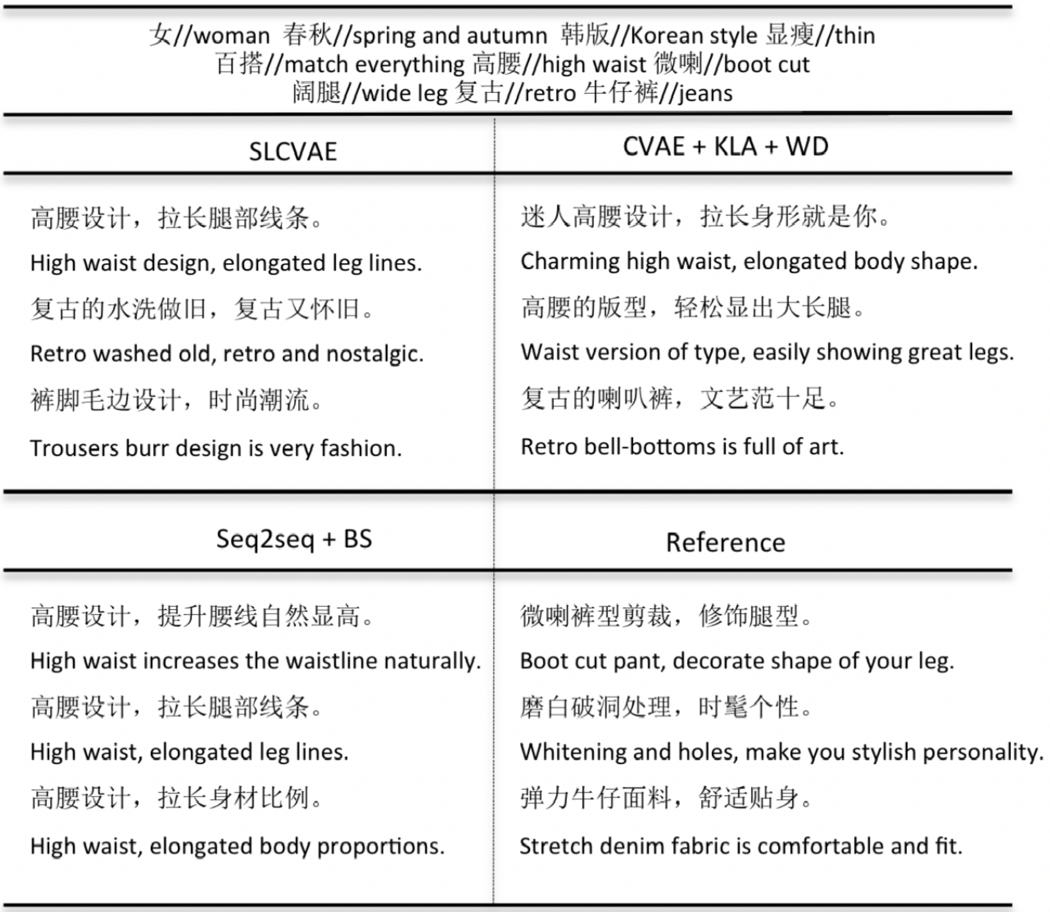
Figure 5.1: Caption Generation Results comparison between LSTM-based encoder-decoder model and VAE [44]

number of unique questions generated for an image and the percentage of questions never seen at training time. On average, the method generated 63.83 unique questions per image on the Bing dataset, out of which 36.92% had not been seen in the training dataset.

5.2.3 VAEs to improve diversity in label generation

Zhang et al. [49] argued that label generation in e-commerce is a one-to-many task often not modeled accordingly. Encoder-decoder-based Seq2Seq models have been widely adopted due to their relevance. But, these models struggle to generate diverse results as a single input is mapped to a single output. They used VAEs to model label generation as a one-to-many task and compared the results against the state-of-the-art Seq2Seq models. Figure 5.2 shows the difference in the labels generated between Seq2Seq models and VAE-based models. They observed that Seq2Seq models performed the most accurately, but their VAE-based generative models significantly outperformed for diversity metrics like BLEU-recall, distinct-1, and distinct-2.

These studies in various domains show that VAEs are an effective tool for improving diversity while producing accurate results. VAEs have not been used in a generative manner within RS to improve diversity. A user profile distribution can be generated by making the VAE-based CF approach [26] stochastic at inference. We aim to study if this approach can be useful in the RS domain as well.



女//woman 春秋//spring and autumn 韩版//Korean style 显瘦//thin
百搭//match everything 高腰//high waist 微喇//boot cut
阔腿//wide leg 复古//retro 牛仔裤//jeans

SLCVAE	CVAE + KLA + WD
高腰设计，拉长腿部线条。	迷人高腰设计，拉长身形就是你。
High waist design, elongated leg lines.	Charming high waist, elongated body shape.
复古的水洗做旧，复古又怀旧。	高腰的版型，轻松显出大长腿。
Retro washed old, retro and nostalgic.	Waist version of type, easily showing great legs.
裤脚毛边设计，时尚潮流。	复古的喇叭裤，文艺范十足。
Trousers burr design is very fashion.	Retro bell-bottoms is full of art.

Seq2seq + BS	Reference
高腰设计，提升腰线自然显高。	微喇裤型剪裁，修饰腿型。
High waist increases the waistline naturally.	Boot cut pant, decorate shape of your leg.
高腰设计，拉长腿部线条。	磨白破洞处理，时髦个性。
High waist, elongated leg lines.	Whitening and holes, make you stylish personality.
高腰设计，拉长身材比例。	弹力牛仔面料，舒适贴身。
High waist, elongated body proportions.	Stretch denim fabric is comfortable and fit.

Figure 5.2: Label Generation Results comparison between Seq2Seq model and VAE [49]

5.3 VAEs in RSs

In recent years, some researchers have explored the use of VAEs in RS. VAEs can be described as AEs that learn a latent distribution that can be used to generate the underlying data that it is trained on [33]. Ferrari et al. published a reproducibility report in which they reported that VAE-based CF approaches give a state-of-the-art performance in RSs when compared against other model classes like Top popular, SLIM, ItemKNN CF on relevance metrics like NDCG and recall [14].

The following sections describe the model classes VAEs fall into like probabilistic ML, generative models, and AEs, and build upon those concepts to describe the mechanisms behind VAEs.

Probabilistic Machine Learning

Machine learning is often used to learn probabilistic models of natural and artificial phenomena. *Probabilistic machine learning* models aim to approximate the unknown underlying process which generates the observed data x, by a model parameterized by θ. The values of θ are learned such that the $p_\theta(x)$ approximates the true distribution of the underlying data [21]. Function p_θ can be learned using neural networks, where θ represents the weights of the network.

Oftentimes, we need to learn a conditional probabilistic model $p_\theta(y|x)$ rather than an unconditional model of the underlying data $p_\theta(x)$ [21]. This is the model formulation used in most classification tasks wherein we are trying to determine the conditional probability of label y (for example - dog or cat) given the data point x (for example - an image of a dog or a cat).

Generative models

Machine learning models can be divided into discriminative and generative models. *Discriminative models* learn a mapping in the direction in which we aim to make future predictions. For example - learning to predict whether an image contains a cat or a dog is a task that a discriminative model can solve. They are often focused on learning a conditional probabilistic model of the data, as described in the previous section. *Generative models* aim to learn the underlying data distribution by simulating the data generation process. For example - Latent Dirichlet Allocation models assume that documents are generated by sampling words from different topics and the model simulates this process to generate documents or sentences. They learn the unconditional probabilistic model of the data as described in the previous section. There are four categories of deep generative models - generative adversarial networks (GANs), VAEs, flow-based deep generative models, and diffusion models. *GANs* learn to generate samples by trying to make them look like real data [16]. VAEs learn an approximation of the data distribution $p(x)$ by using variational inference, which would optimize the log-likelihood of seeing the data [20]. *Flow models* use the concept of normalizing flows to directly model the otherwise intractable data distribution $p(x)$. Thermodynamics concepts have been employed in *diffusion models* [32]. A Markov chain of diffusion steps is defined to add random noise to the data, and the model learns to reverse the diffusion process to generate data from noise [37]. Figure 5.3 shows the model architecture of different kinds of generative models.

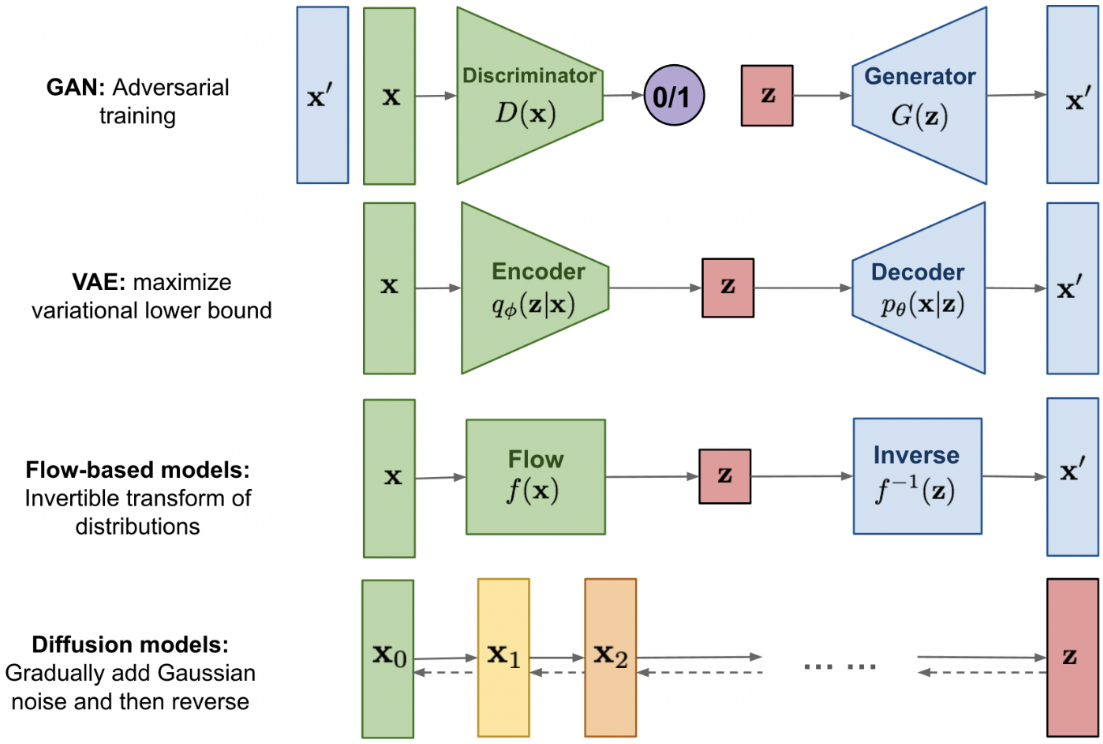

GAN: Adversarial training
x'
x
Discriminator
D(x)
0/1
z
Generator
G(z)
x'
VAE: maximize variational lower bound
x
Encoder
qϕ(z|x)
z
Decoder
pθ(x|z)
x'
Flow-based models: Invertible transform of distributions
x
Flow
f(x)
z
Inverse
f⁻¹(z)
x'
Diffusion models: Gradually add Gaussian noise and then reverse
x0
x1
x2
… …
z

Figure 5.3: Model architecture of different kinds of generative models - GANs, VAEs, Flow-based models and diffusion models [46]

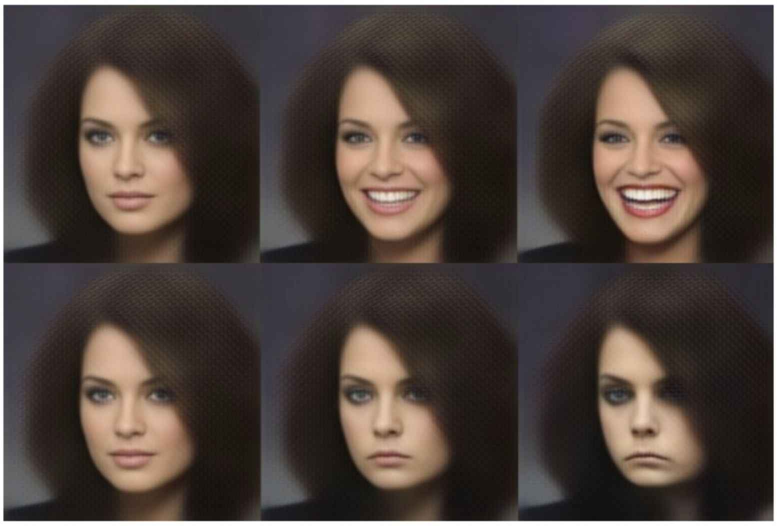


Figure 5.4: Images generated by VAEs [21]

AutoEncoders

AutoEncoders (AEs) were introduced to learn compressed representations of images which could then be used for downstream tasks like classification and regression [34]. AEs attempt to reconstruct higher dimensional data by learning a latent lower dimensional representation of the data. Figure 5.5 shows the model structure of an AutoEncoder network. The network contains an encoder e_θ that maps input x to a latent code z and a decoder d_ϕ that maps the latent code back to the input. Often, a bottleneck is introduced to help the model generate a non-trivial solution. For example - in denoising AEs, a slight perturbation is added to the input image, and the model needs to produce a noise-free image as an output. This challenges the model to remove the noise and helps the model learn non-trivial representations for the image.

Equation 5.1 shows what the network equation would look like.

$$\hat{x} = d_\phi(e_\theta(x)) \tag{5.1}$$

The goal of the loss function is to minimize the difference between the input and the regenerated input as shown in equation 5.2,

$$min_{\phi,\theta} \sum_{i=1}^{n} ||\hat{x_i} - x_i||^2 \tag{5.2}$$

where $\{x_i\}_{i=1...n}$ is the dataset.

Theory of VAEs

Variational Autoencoders attempt to produce embeddings that can reconstruct the input and arrange them in the latent space such that clusters are almost normally distributed, and interpolating between clusters gives meaningful results [20]. This is done by learning the mapping between the input space x, and the distribution parameters μ and σ from which we can sample a latent vector z, rather than a mapping between x and z [21]. VAEs are generative models as they can be used to generate new data points using μ and σ. Figure 5.4 shows the images of faces generated by VAEs. Figure 5.7 shows the difference between AE and VAE model architectures. VAE loss function consists of two components - *Reconstruction loss* and *Kullback–Leibler (KL) divergence*. These two components of the loss function must be balanced to reconstruct the input while producing a continuous latent space. Figure 5.10 shows the difference in the latent space generated from a VAE compared to an AE latent space.

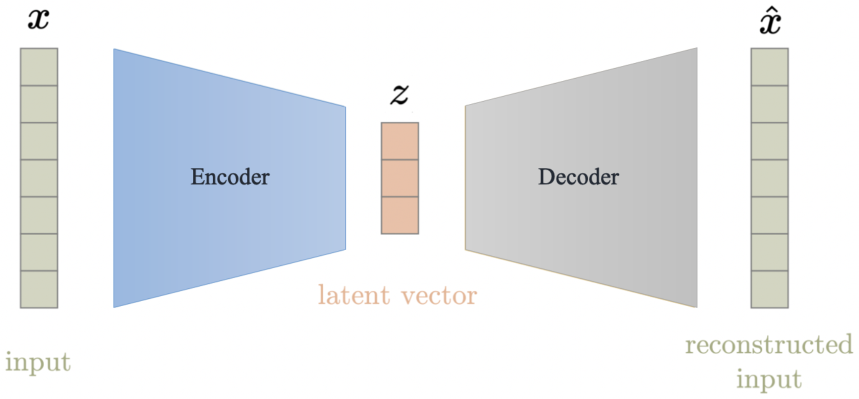


Figure 5.5: AE model structure

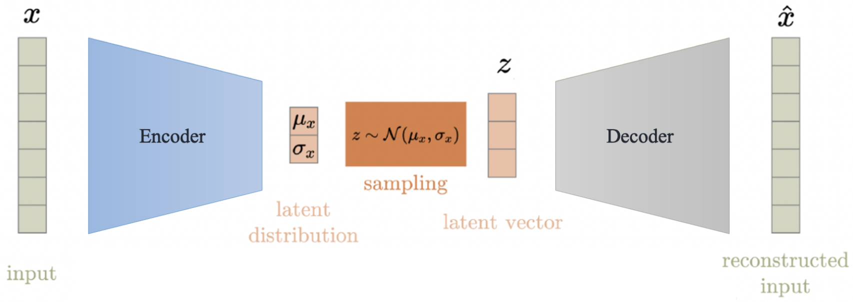


Figure 5.6: VAE model structure

Figure 5.7: Difference in the model structure between an AE and a VAE [4]

A generative model can be trained by maximizing equation 5.3, which is the likelihood of observing the input data through the model parameterized by θ.

$$\sum_{i=1}^{n} log\ P_\theta(x_i) \tag{5.3}$$

Vanilla AEs are not generative because the latent space they generate is very sparse and $log\ P_\theta(x_i) = 0$ for most samples. Sampling from that latent space will not result in a data point from the original distribution for most samples. To resolve this issue, the generation process can be modeled by a noisy observation model as shown in equation 5.4.

$$P_\theta(x|z) = \mathcal{N}(x, g_\theta, \epsilon) \tag{5.4}$$

In equation 5.4, the latent code z does not map to a single point in the input space x, but a distribution.

$log\ P_\theta(x_i)$ is intractable in most cases, so instead its lower bound is optimized[21].

$$log\ P_\theta(x) = \mathbb{E}_{z \backsim p(z|x)}\ log\ \frac{P_\theta(x,z)}{Q_\phi(z|x)} + \mathbb{E}_{z \backsim p(z|x)}\ log\ \frac{Q_\phi(z|x)}{P_\theta(z|x)} \tag{5.5}$$

The second term in equation 5.5 has the form of a KL divergence. KL divergence has a property that it is always greater than 0, and hence the first term becomes a lower bound for P_θ. This term is called the *Variational lower bound (VLB)* or *Evidence lower bound (ELBO)*, and maximizing it would maximize $logP_\theta(x)$ as well.

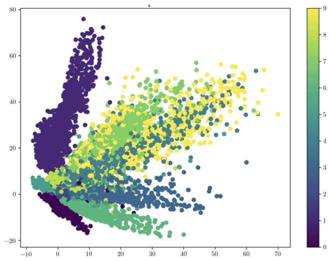


Figure 5.8: AE latent space visualization

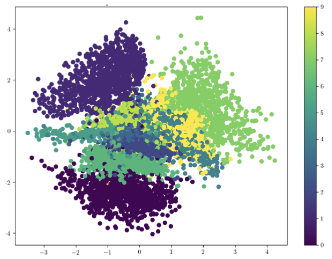


Figure 5.9: VAE latent space visualization

Figure 5.10: Difference in the latent space generated by an AE and a VAE for MNIST dataset [4]. The clusters in the VAE latent space are more spherical and resemble a normal distribution

Expanding the VLB will result in equation 5.6. The first term is equivalent to reconstruction loss which aims to make $Q_\phi(z)$ as similar of a point estimate as possible. The second term aims to reduce the KL divergence between $Q_\phi(z|x)$ and $P(z)$, where $P(z)$ is a predefined tractable distribution like a Gaussian distribution.

$$VLB = \mathbb{E}_{z \backsim p(z|x)}\ log\ P_\theta(x, z) + \mathbb{E}_{z \backsim p(z|x)}\ log\ \frac{P(z)}{Q_\phi(z|x)} \tag{5.6}$$

So the two-fold aim of optimizing VLB is -

- Roughly maximizes $log\ P_\theta(x_i)$
- Minimizes KL divergence between $Q_\phi(z|x)$ and $P(z)$, thereby increasing the probability of $Q_\phi(z|x)$ being similar to a Gaussian. This term has a regularizing effect on the model.

Stochastic gradient descent using Amortized Inference

Traditional variational inference methods do not share the variational parameters between data points. Computing gradient descent on each record would be computationally expensive and not feasible. Hence [21] has shown ELBO has a valuable property that allows for joint optimization for all parameters (ϕ and θ) via stochastic Gradient descent. The parameters are initialized randomly and can be stochastically optimized until they converge. Stochastic gradient descent optimizes the sum of the individual data point's ELBO. Individual data point's ELBO is intractable in most cases, and hence its unbiased estimate is used - a random sample from $Q_\phi(z|x)$. ELBO expectation is taken w.r.t. to $Q_\phi(z|x)$, which is a function of ϕ, rather than parameters θ as the latter is more difficult to obtain. *Amortized inference* uses the strategy of sharing variational parameters across data points rather than using different parameters for each data point. Amortized inference in VAEs helps to provide a warm start to new data points by leveraging the data that has been seen before. This property is well-aligned with the goal of CF, which relies on similar users' data to make recommendations.

Algorithm 1: Stochastic gradient descent for Amortized Inference in VAE [26]

```
Input: Bag-of-words click matrix X $\in \mathbb{R}^{U \times I}$
Random initialization of the neural network parameters $\phi$ and $\theta$
while not converged do
    Sample a batch of users U
    for  $u \in U$ do
        Sample $\epsilon \sim \mathbb{N}\{1, 0\}$
        Compute $z_u = \mu + \epsilon * \sigma$ according to reparametrization trick
        Compute gradient of $\Delta_\theta VLB$ and $\Delta_\phi VLB$ from equation 5.6 with $z_u$
    Update $\theta$ and $\phi$ by the average gradient across U
  Return $\theta$, $\phi$
```

Reparametrization trick

As shown in Figure 5.6, the VAE network has a sampling operation. As sampling is not a differentiable operation and it is impossible to backpropagate error through it, a *reparametrization trick* is used in VAE as shown in Figure 5.11. Instead of adding stochasticity by randomly sampling z from μ and σ, the source of randomness is separated from any of the quantities which will be differentiated. ϵ is made the source of randomness and combined with μ and σ to draw from the distribution. The model does not backpropagate through ϵ. This allows backpropagation of error through z.

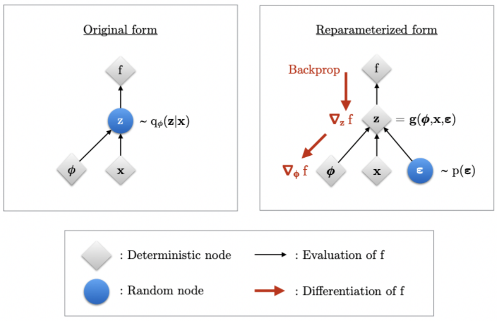


Figure 5.11: Reparametrization trick [21]

5.4 VAE-based User Profile Generation

5.4.1 Motivation

Guo et. al. [17] proved that CF algorithms would not recommend users a set of items that are anti-correlated to the entire population regardless of their preferences. They introduced a notion of joint accessibility, which describes the extent to which a set of items can be jointly recommended to the user. They assessed standard matrix-factorization-based CF models on joint accessibility and found that these models are not jointly accessible especially when they represent a user by a single vector. They theoretically proved that it is better to represent a user by multiple vectors to capture their diverse interests. However, they did not provide any practical method to produce multiple user profiles.

VAEs have been used in other domains to improve diversity and they have been used in RSs to give a state-of-the-art result with respect to relevance measures, but the generative nature of these models has not been exploited to improve diversity in RSs as yet. VAEs can be used to learn a distribution to represent a user. Oftentimes, only the mean of the VAE-generated latent distribution is considered during inference. But, if we keep the model stochastic during inference, we can sample from the latent distribution to obtain multiple user profile vectors. Combining recommendations from these user profiles could help improve diversity in RSs.

Apart from the ability to generate multiple user profiles, the implicit regularization of a VAE could potentially help improve diversity. We have seen several techniques in Section 5.1 where regularization has been used previously to improve diversity in RSs. KL divergence can be seen as a regularizing factor in the loss function as it aims to make the latent distribution as similar to a pre-determined tractable distribution (Gaussian distribution here) as possible [5].

5.4.2 Step-by-step Description

- **Use VAEs to generate a user profile distribution**: VAEs learn the parameters μ and σ of a continuous latent space that can represent the input with a tractable distribution like a Gaussian distribution. To sample different user profiles from this distribution, σ mustn't be a very small value. Otherwise, the latent distribution might collapse to a point. During training, σ might drop to 0 indicating that the model is very confident in its choice of μ. KL divergence serves as a regularizing factor that is designed to reduce this confidence by forcing the variance to be greater than 0 [6].

- **Sample from the generated distribution to obtain multiple user profiles**: Each user profile produces a set of recommended items. Each user profile generated by the VAE should have slight differences so that the corresponding recommendation lists would be slightly different.

- **Combine recommendations generated by different user profiles**: This step can ensure that we exploit the recommendations generated through the different user profiles to improve diversity. This can be illustrated with the following example - imagine a user purchased a diaper. One user profile connects it to baby products and generates a recommendation list with items like milk powder and toys. In contrast, another user profile relates it to hygiene products like cleaning wipes. Combining these recommendations from both lists can ensure that the final recommendation list is diverse. There are many ways to combine the results - 1) by optimizing the relevance of these recommendations, 2) randomly, or 3) by maximizing the diversity of the items. As this research focuses on producing diverse results, we will pick items from the different recommendation lists based on diversity. We will give a diversity ranking to each item based on how different it is from all the other items on the list. We will select the items with the highest diversity rank.

- **Rank them according to relevance**: The combined list is ranked according to the relevance score. As users focus more on the items at the top, this step ensures the combined list will also have high relevance.

We expect the recommendations generated by our method to be more diverse in a single session because of two reasons - 1) VAEs have a regularizing factor in the loss optimization and regularization has been used in other research to improve recommendation diversity, 2) We plan to use multiple profiles which would better reflect the user's varied interests better. 3) We plan to select items from these candidate lists based on diversity. We expect the recommendations to be different over time because there is randomness in the process and every time we sample we get a slightly different user profile. We expect the list to have higher aggregate diversity because our method should produce a personalized yet diverse list. This would lead to more niche items getting recommended to users and ensure that the model does not focus on popular items only which leads to low aggregate diversity.

VAEs have been used to improve relevance but to our knowledge, we are the first to employ them to improve diversity in RSs.

5.4.3 Evaluation over requirements

In this subsection, we will describe how the method adheres to the requirements described in chapter 3.

- *Relevance* - VAEs have been shown to provide state-of-the-art performance in RSs in terms of relevance. The model's loss function has the reconstruction term which focuses on reconstructing the input regardless of the shape of the latent distribution. This ensures that the model focuses on the relevance of the output with respect to the input data.
- *Diversity* - KL Divergence term in the loss function forces the value of σ to be greater than 0, ensuring that there is variance in the distribution. This variance ensures that the vectors sampled from the distribution are different and would lead to a diverse combined recommendation list.
- *Scalability* - A single vector cannot represent a user's varied interests [17]. As we mentioned in the recommendation process (Section 3.1), generating user profiles is a tedious step and is done at longer intervals. If we want to generate diverse recommendations over time using the current user representation (point estimate), we would need to store multiple user profiles for each user between the training intervals, e.g. if the profiles are trained once a week, we would need to store seven profiles to generate different recommendations each day. In contrast, our method allows us to produce new user profiles every day/in every run by storing only two vectors μ and σ and generating different recommendation lists in each run. In this way, we reduce the space complexity, while maximizing diversity.

5.5 Limitations

In terms of time complexity, the step for combining different recommendation lists adds latency in the order of $\mathcal{O}(k^2)$ to the model complexity. This additional time is not needed to generate recommendations with a single user profile. This method is still feasible as the value of k is generally very small. For large values of k, it might be useful to consider other ways of combining lists generated from multiple user profiles.

This method assumes that all the information about the user is captured in the user purchase history and hence only aims at making recommendations based on user history. These assumptions might not always be true and this is a limitation of this study.

Another limitation is that the user representations sampled from the distribution generated by the VAE could overlap. Learning a distribution that generates distinct user profiles distinct could potentially provide better results. VQ-VAE could be used to generate a discrete distribution rather than a continuous one, potentially ensuring that user representations sampled from the distribution are distinct. Future works could explore this approach in more detail.

5.6 Summary

Section 5.1 introduced different techniques to improve diversity in RSs that can be categorized into post-processing and algorithmic techniques. Section 5.2 motivated using VAEs for improving diversity by giving examples from other domains. Section 5.3 provided a theoretical background about VAEs and Section 5.4 described a novel VAE-based user profile generation method to enhance diversity in RSs.

Chapter 6

Comparative study between representing user as a point estimate vs a distribution

This chapter describes the experiment conducted to answer the second, third, and fourth RQs related in Chapter 3. Section 6.1 motivates this experiment and elucidates the value it adds to the RS research. Section 6.2 describes the experiment procedure and concludes the chapter by providing the results of the experiment.

6.1 Motivation

The goal of this experiment is to evaluate if the method described in the previous chapter for representing a user as a distribution (multiple profiles sampled from a distribution) rather than a point estimate can produce more diverse recommendations.

There are multiple stakeholders in the recommendation process, like users, vendors, and platform owners. We focus on diversity from the perspective of all of these stakeholders. Improving diversity from the user's perspective will enhance their experience and make them feel more represented [22]. While improving diversity from the vendor or business perspectives will increase vendor opportunities by exposing more products to users and boost business outcomes as well. RQ2 and RQ3 defined in chapter 3 are focused on evaluating the proposed method for diversity from the user's perspective, while RQ4 focuses on diversity from the vendor's perspective. In this experiment, we will try to answer RQ2, RQ3, and RQ4, which have been motivated in Chapter 3.

RQ2 - *"Can representing users as a distribution rather than a point estimate improve user-level diversity of recommendations generated within a single session while maintaining an acceptable level of relevance?"*

RQ3 - *"Can representing users as a distribution rather than a point estimate improve user-level diversity of recommendations generated over multiple sessions while maintaining an acceptable level of relevance?"*

RQ4 - *"Can representing users as a distribution rather than a point estimate increase the total number of products recommended to all users while maintaining an acceptable level of relevance?"*

Answering these questions can help us determine if our way of representing the user is better than the conventional way from the multiple perspectives of diversity.

6.2 Experiment

6.2.1 Experimental Procedure

We will use the VAE-based method described in the previous chapter to generate a user profile distribution, sample multiple profiles from it, and combine their results to generate a recommendation list. We will use a vanilla AE to generate recommendations from a single user profile and compare the relevance and diversity of the recommendations generated from both methods to answer RQ2, RQ3, and RQ4.

We have chosen these two models because they are similar in their architecture apart from the elements needed to generate the latent distribution vs. point estimate. This helps ensure that the differences can be attributed to the user profile representation. A similar approach has been used in the studies described in Chapter 5. Another reason for selecting these two models for this experiment is that both these models have been proven to produce state-of-the-art performance in terms of relevance in RSs. Hence, if our method improves diversity, this will give practitioners a method that can make highly accurate recommendations while improving diversity.

Data Pre-processing

As the data is generated by user interactions with products, there is a high likelihood of noise. Hence, it is essential to filter out items that have been viewed less than five times and users that have interacted with less than five items. The threshold for the number of users and items has been taken from [26]. The model's input and output is a one-hot encoded vector representation of the user's purchases where if the user has bought/rated an item, it is marked 1. Otherwise, it is marked 0. It has the same dimensionality as the item space $\mathbb{R}^i$

This preprocessing step allows us to predict users' interest in the entire item catalog through multi-class classification. Furthermore, this eliminates the need to do negative sampling, which is required when measuring the binomial likelihood of a user being interested in a single item. Negative sampling is a method by which we add records to the training data to indicate the items the user is not interested in, as the original data only includes positive samples. The difference in preprocessing between these two approaches can be described by the following example - If a dataset contains five items and $user_1$ liked $item_2$ and $item_3$, the data would be preprocessed in the following manner for binomial and multinomial likelihood estimation -

- Binomial likelihood - There would be two records in the training data - $user_1$, $item_2$ and $user_1$, $item_3$, for the positive samples, and some negative samples would be sampled from the dataset and added to the training data to create a balanced dataset.
- Multinomial likelihood - $user_1$ would be represented by [0,1,1,0,0], which eliminates the need for negative sampling.

The users were split into train/test/validation sets with an 80-10-10 split. During training, the models learn to reconstruct the entire interaction history of the train users, which is encoded as a bag of words one-hot-encoded item vector. During inference, only 80% of the items per user were included in the input, and the model is evaluated on whether the recommendations were present in the held-out set, which contains 20% of the items from that user's interaction history.

Implicit feedback has been used to model users' interest based on their ratings or past purchases. This means that movies rated 4/5 are treated similarly to movies rated 5/5 because we are only modeling whether the movie is rated highly.

Training

To compare the difference between representing a user as a distribution vs. a point estimate, we used two models: a Vanilla AE (point estimate) and VAE (distribution) proposed in [26].

Following is the architecture of the VAE -

- Encoder
  - Dropout
  - Linear layer (item set size, 600)
  - Tanh activation layer
  - Linear layer (600, d * 2)
- Encoder model outputs μ and σ which models the data distribution p(z/x)
- Sample a user profile z from the distribution defined by μ and σ as described in Algorithm 1
- Decoder
  - Linear layer (600, d)
  - Tanh activation layer
  - Linear layer (600, item set size)

Following is the architecture of the Vanilla AE -

- Encoder
  - Dropout
  - Linear layer (item set size, 600)
  - Tanh activation layer
  - Linear layer (600, d)
- Encoder model outputs a point estimate for the user profile. Notice that the encoder output size here is half the size of the encoder output in VAE, because VAE generates two outputs μ and σ and samples a vector z from this distribution which has the same size as the latent vector of the vanilla AE.
- Decoder
  - Linear layer (600, d)
  - Linear layer (600, item set size)

The capacity and architecture of both models are as similar as possible. The difference between the two is that the VAE encoder outputs μ and σ, parameters of the distribution rather than a point estimate, and the input to the decoder is sampled from this distribution. Vanilla AE output is one vector of dimension d, whereas VAE outputs are μ and σ of dimension d.

Using multinomial likelihood decreases training time compared to binomial likelihood because the data grows linearly with the number of users compared to the number of items × users.

The models are trained using the stochastic gradient descent algorithm with amortized inference described in 5.3.

6.2.2 Loss function

- Vanilla AE: This model uses Binary Cross Entropy loss as reconstruction loss.
- VAE: As described in Section 5.3, VAE loss functions are made up of two components - KL divergence and reconstruction loss. This model also uses Binary Cross Entropy loss as reconstruction loss and the closed form formula of KL divergence as derived in [40].

Hyperparameters

Hyperparameter tuning is an instrumental step in optimizing the results of deep learning models. This is a trial-and-error process where a certain model is run on the validation set to find the parameters that can provide the best results. In this study, we want to optimize relevance and diversity. Hence, we seek to find the sweet spot for each parameter that balances relevance and diversity. Table 6.1 reports the parameter values that have been chosen after tuning. The hyperparameters have been kept constant between the vanilla AE and VAE to ensure a fair comparison.

VAEs could suffer from over-regularization if KL Divergence starts dominating the loss function. Liang et al. introduced a parameter β to control the effect of KL divergence in the loss function [26]. They observed that monotonically increasing β is an efficient way of finding the optimal value of β. They let the β value increase till the optimal value and then stopped increasing it. We follow their approach for KL annealing. Apart from over-regularization, over-fitting is also possible due to a sparse matrix; hence a 50% dropout has been applied to the vanilla AE and the VAE input to prevent over-fitting.

Generating item representations

Item representations need to be generated to calculate the distance of items for the following metrics - ILD and TILD, which compute the representation distance between the items. We have generated item embeddings by concatenating the words in the item metadata to form a sentence. This sentence is passed through pre-trained language models. We have used “bert-base-nli-mean-tokens” from HuggingFace [31] to generate item embeddings for MovieLens dataset. The Bol.com category information is in Dutch. Hence, we have used “paraphrase-multilingual-mpnet-base-v2” model from HuggingFace [31], which can generate sentence embeddings for 50+ languages including Dutch. Both models map sentences and paragraphs to a 768-dimensional dense representation that can be used for clustering or semantic search. The following metadata was used for each dataset -

- **MovieLens dataset** - Genre of each movie (for example - Adventure, Animation, Children, Comedy), tags users had assigned to each movie (for example - “life philosophy”, “great ending”, “brutality”, “visually appealing”)

- **Bol.com dataset** - Deepest category names in Dutch (e.g. Gezondheid, Persoonlijke verzorging, Koken & Tafelen)

Using the metadata to generate item embeddings allows the representations to be independent of the models.

Hyperparameters	Values
Dropout	0.5
Batch Size	500
Number of sampled vectors	2
KL Divergence Weight	Monotonically increasing from 0 to 0.2
Latent Vector Size	200
Encoder - Number of Layers	2
Decoder - Number of Layers	2
Learning Rate	0.0001
Momentum	0.9

Table 6.1: Hyperparameter Values

6.2.3 Experiment Details

Datasets

- **ML-20M** - Movielens [18] is the most commonly used benchmark dataset in RSs. It contains 20 million records of user movie ratings collected from a movie recommendation service. The data has been filtered to remove movies rated less than five times and users who have rated less than five movies. The data was further divided into train, test, and validation sets with the following split - 80-10-10. One-hot encoded vectors have been created for each user's item ratings to facilitate multinomial likelihood. We have also filtered out the ratings below 4/5 to reduce noise. This has resulted in a dataset with 138493 users and 26164 items, with a data sparsity of 0.541%.

- **Bol.com dataset** - We have taken a sample of a real-world dataset from Bol.com, which shows the categories (the most granular category level) users purchased items in a year. These were the users who were active on a given day. The same preprocessing steps, data split, and data preprocessing steps have been followed as the ML-20M dataset. This resulted in a dataset with 11547 categories, 6 million records, and 55 thousand users, with a data sparsity of 0.951%.

Baseline

We used a vanilla AE of the same capacity as the VAE as described in the experimental procedure. This model represents the user in the latent layer as a point estimate. This baseline has been selected because it has state-of-the-art RS relevance and has a similar architecture to the VAE [26] [14]. The only difference is in the latent layer. This allows us to attribute any difference in diversity to using a distribution to represent a user rather than a point estimate. Furthermore, we have not selected some other method for improving diversity as a baseline because we want to see if our approach can improve diversity by changing the representation of the user. Post-processing techniques can be applied to the results of our method to enhance diversity further.

Evaluation methods

Mean reciprocal rank (MRR), Mean average precision (MAP), and Normalized discounted cumulative gain (NDCG) are the most common relevance metrics in RSs [41]. To measure relevance in RSs, it is important to understand 1) how relevant is each item in the list and 2) if the items are arranged according to their relevance to the user. MRR only evaluates if the first item in the list is relevant. MAP only evaluates if the item is relevant or not, but does not take into account how relevant each item is. We have chosen NDCG to evaluate the relevance of recommendations in this experiment as it focuses on how relevant each item in the list is along with the order of the items in a recommendation list [41].

ILD@K, TILD@K, and Aggregate Diversity have been used to assess the recommendation lists on user and item-level diversity. The reasoning for using these diversity measures has been elaborated in Chapter 4.

- **NDCG@K** - Normalized discounted cumulative gain measures the relevance of items in the recommendation list of size K. Each item is discounted based on its position in the ground truth so that it can account for the order of the recommendation list as well. It is normalized by the IDCG (Ideal discounted cumulative gain), which is the DCG of the perfect recommendation list.

- **ILD@K** - Intra-list diversity measures how different the items are in a single recommendation list. It is normalized by the size of the list. It is a user-level metric and measures how diverse the recommendations are in a single recommendation list. We have used cosine distance between item representations to measure how different items are from each other.

- **TILD@K** - Temporal Inter-list Diversity measures how different the items are over time. The size of the list is used to normalize the metric. It is a user-level metric and measures how different the recommendation for each user is in separate sessions. We have used cosine distance between item representations to measure how different items are from each other.

- **Aggregate Diversity** - Aggregate diversity is a valuable metric to understand how much exposure items get on the platform. As the original definition described by equation 4.3 has not been normalized by the total number of items and would be difficult to compare over datasets with a different number of items. We have normalized it by the total number of items.

6.2.4 Results

Table 6.2 and 6.3 report the results obtained from the experiment for the MovieLens and Bol.com dataset for lists of size 5, 10 and 20. We have combined recommendations from the two user profiles sampled from the distribution generated by the VAE. We have only reported the deltas for the Bol.com dataset to maintain confidentiality.

The following results obtained from MovieLens and Bol.com datasets provide evidence to answer RQ2,RQ3, and RQ4 -

- **MovieLens dataset** - The recommendations generated by our method are 32% more diverse within a single list, 48% more diverse temporally, and exposed 48% more items to the users in comparison to the baseline. There was a drop of 25% drop in NDCG from the baseline for a list of size 5. For a list of size 10, the recommendations were 26% more diverse within a single list, 32% more diverse temporally, and exposed 44% more items to the users, while leading to a 24% drop in NDCG. For a list of size 20, the recommendations were 22% more diverse within a single list, 24% more diverse temporally, and showed 38% more items to the users, while leading to a 22% drop in NDCG.

- **Bol.com dataset** - The recommendations generated by our method are 9% more diverse within a single list, 18.99% more diverse temporally, and exposed 33% more items to the users in comparison to the baseline. There was a drop of 5% drop in NDCG from the baseline for a list of size 5. For a list of size 10, the recommendations were 9% more diverse within a single list, 13% more diverse temporally, and displayed 30% more items to the users while leading to a 3% drop in NDCG. For a list of size 20, the recommendations were 9% more diverse within a single list, 11% more diverse temporally, and exposed 20% more items to the users while leading to a 5% drop in NDCG.

We can answer the three research questions we stated in this experiment based on these results. Our method significantly improved diversity for both datasets in all three diversity metrics - ILD, TILD, and Aggregate diversity. This came at a cost of 25% decrease in relevance for the MovieLens dataset, while a smaller decrease of 5% for the Bol.com dataset for a list of size 5. This shows that the results of this method depend on the dataset. This could be because the Bol.com dataset is less sparse at the category level, and interaction data is available for more items (items are represented by their deepest categories making it more likely for the interaction data to be sparse). Hence, it is possible to capture more data while maintaining relevance. In comparison, the MovieLens dataset does not contain interaction information related to enough items, due to which increasing diversity leads to higher costs to relevance.

Appendix A shows how diversity is increased through our method for both datasets using a single user as an example. User profile coverage should be better through our method than a single user profile as we firstly try to capture different interests through multiple user profiles and secondly select the most diverse items while combining the lists from multiple user profiles. But, it is difficult to measure at which level is the user profile coverage improving as our definitions of diversity do not take category levels into account. This could be explored in future work.

Furthermore, the tradeoff between RS diversity and relevance depends on different system parameters. In this experiment, we created a recommendation list based on the results of two

Metrics	**Point estimate**	**Distribution (2 vectors)**	**Percent change**
NDCG@5	0.456 ± 0.0032	0.3413 ± 0.0028	-25.15%
NDCG@10	0.4322 ± 0.0027	0.3303 ± 0.0025	-23.58%
NDCG@20	0.3917 ± 0.0021	0.3039 ± 0.0023	-22.42%
ILD@5	0.0811 ± 0.0003	0.1067 ± 0.0004	31.57%
ILD@10	0.0849 ± 0.0003	0.107 ± 0.0003	26.03%
ILD@20	0.08949 ± 0.0002	0.109 ± 0.0003	21.80%
TILD@5	0.1622 ± 0.0006	0.2393 ± 0.0007	47.53%
TILD@10	0.1699 ± 0.0005	0.2246 ± 0.0006	32.20%
TILD@20	0.179 ± 0.0004	0.2226 ± 0.0005	24.36%
Aggregate Diversity@5	0.1076 ± 0.0000	0.1599 ± 0.0004	48.65%
Aggregate Diversity@10	0.1367 ± 0.0000	0.1973 ± 0.0009	44.34%
Aggregate Diversity@20	0.1707 ± 0.0000	0.2367 ± 0.0006	38.72%

Table 6.2: The mean and standard deviation of the NDCG, ILD, TILD, and AD for the MovieLens data set. The standard deviation of NDCG, ILD, and TILD has been calculated at a user level as these are user-level metrics. The standard deviation of aggregate diversity has been calculated at a session level

vectors sampled from the distribution, which impacts the tradeoff. The next chapter explores this tradeoff and shows how the number of vectors sampled from the distribution affects the results. We can also observe that the VAE's improvement in terms of diversity stagnates as the size of the list increases. Hence, it is important to evaluate the feasibility of this change in a RS depending on the use case.

Metrics	Percent change
NDCG@5	-5.00%
NDCG@10	-2.79%
NDCG@20	-5.11%
ILD@5	9.41%
ILD@10	9.07%
ILD@20	9.22%
TILD@5	18.99%
TILD@10	13.33%
TILD@20	11.17%
Aggregate Diversity@5	33.09%
Aggregate Diversity@10	29.71%
Aggregate Diversity@20	20.20%

Table 6.3: Bol.com results

6.3 Limitations

The first limitation of this experiment is that the sample taken from the real-world dataset includes a year of purchasing data for users who were active on a given day. Sampling users by time might result in more active users being selected, as the likelihood of those users being active on any given day might be more. The benchmark dataset was not sampled by users and as the results were consistent across both datasets, we can conclude the results to be valid.

The second limitation is that we have used item embedding generated by pre-trained text embedding models to measure ILD and TILD, and for combining the item lists based on diversity ranking. These text embedding models have been shown to have certain biases which could affect the result of this experiment [1]. Future work could explore this direction and evaluate how the results vary with different pre-trained models or a different way of generating item representations

The third limitation of this experiment is that we have only generated and sampled user vectors from a Gaussian distribution based on our method. Future studies could evaluate how representing a user by other distributions impacts the results. The results of this paper can likely be generalized over other distributions as other distributions can represent more complexity than a Gaussian distribution.

6.4 Summary

This chapter describes the first experiment to answer RQ2, RQ3 and RQ4 presented in Chapter 3. Based on the experiment, we can conclude that representing users as a distribution rather than a point estimate can improve user and item level diversity. Our method increases ILD, TILD, and aggregate diversity by 9%, 19%, and 33% respectively for a list of size 5 for the Bol.com dataset, with a small decrease of 5% in NDCG. The increase in diversity and decrease in relevance seems to be dependent on the dataset and its properties based on our experiments.

Chapter 7

Effect of different parameters on the diversity-relevance trade off

The goal of this experiment is to evaluate how changing certain experiment parameters impacts the diversity-relevance tradeoff in RSs. This would help practitioners understand how robust or sensitive the method is to different parameter settings. Section 7.1 describes the motivation behind doing this experiment. Section 7.2 explains the experiment procedure used to answer the RQ5 described in chapter 3. Section 7.3 reports the results and provides a ranking of parameters in the order of their influence on the model performance. Section 7.5 summarizes the experiment and concludes the chapter.

7.1 Motivation

Sensitivity analysis is a method to measure how the uncertainties in the input variables impact the output variables [29]. This analysis helps reduce the method's uncertainty by evaluating the robustness or sensitivity of the target for various input variables. It can help us identify input parameters that can cause significant uncertainty in the output and provide guidance about how these parameters should be selected. There are two kinds of sensitivity analyses - local and global. Local analysis determines how small perturbations to the input variable around a nominal target value affect the target. It is usually computed by calculating the partial derivative of the output for the input. One limitation of the local analysis is that the results can only be meaningful to the neighborhood of the nominal point. Global analysis aims to determine the changes in the output variable across the entire field of possible input variation. This is done to determine how much of the model uncertainty comes from changes in a particular input variable. One limitation of the global analysis is that it has high time complexity. In this study, we have performed a global sensitivity analysis for the VAE model described in Section 6.2 over both datasets. To limit the complexity, we have selected a few values for each input (as the input variables we have chosen are continuous) to show their impact on diversity and relevance.

The proposed method can be used in varied settings where the data has different properties, and the diversity-relevance expectations differ from our use case. It would be useful to study the effect each parameter has on this tradeoff. It would help the practitioners use this method and tune the different parameters according to their use cases. This motivates the RQ we aim to answer in this experiment -

RQ5 - *"How does changing different parameters in the system impact the potential diversity-relevance tradeoff?"*

In this chapter, we have first proposed the expected effect these parameters could have on RS relevance and diversity. After the experiment, we study the consequence to evaluate whether the model behaves as expected and explain why specific parameters affect diversity and relevance in a certain way. This analysis can help us find the balance between relevance and diversity, as these dual objectives often conflict.

7.2 Experiment

The overall design and details of the experiment are similar to chapter 6. In this experiment, we have changed the following system parameters one at a time -

- **Latent vector size** - Size of the latent vector z, which is sampled from the distribution generated by the encoder. This is the vector that encodes user profile information.
- **Number of vectors sampled from the distribution** - We can generate the final recommendation list by combining the results of multiple vectors from the distribution generated by the encoder.
- **Dropout Rate** - The rate of units dropped from input during training.

while keeping the rest of the experiment parameters constant. We have selected these parameters because we expect these parameters to impact the diversity and relevance of the model in a certain way. The model has been trained for 100 epochs with a certain set of parameters with the data of 80% of the users and it has been evaluated on the 10% held-out users. The results have been evaluated over the metrics outlined in Section 6.2.3 for a list of size 10.

Table 7.1 reports the standard parameter values used while changing the parameter in question. For example, if we are evaluating the impact of the size of the latent vector, we will set the list size to 10, the dropout rate to 0.5, and use two samples from the distribution. After that, we will run the model with different values of the latent vector size to isolate the impact of changing the latent vector size. The process has been repeated for all three parameters and the datasets mentioned in Experiment 1.

Parameter	**Value**
List Size	10
Dropout Rate	0.5
Number of samples used	2
Latent Vector Size	200

Table 7.1: Standard Parameters Values

7.3 Results

7.3.1 Effect of user profile size

The latent vector z sampled from the distribution generated by the encoder captures all the information about the user. This latent vector represents the user profile and is used to generate the recommendations. The size of the vector is generally smaller than the input size so that the model does not learn a trivial encoding of the input. Increasing the size of this latent vector could encapsulate more information about the user and help improve the diversity while having minimal impact on relevance. The VAE is evaluated with latent vector sizes of 100, 200, and 400 for both datasets. Figure 7.1 shows that the VAE model trained on the MovieLens dataset is not very sensitive to the latent vector size. In comparison, the VAE model trained on the Bol.com dataset is more sensitive to changes in the latent vector size, as shown in Figure 7.2. In both datasets, NDCG decreases with increasing vector size. This could be the result of overfitting in models with bigger latent vectors. For the Bol.com dataset, individual diversity measures (ILD

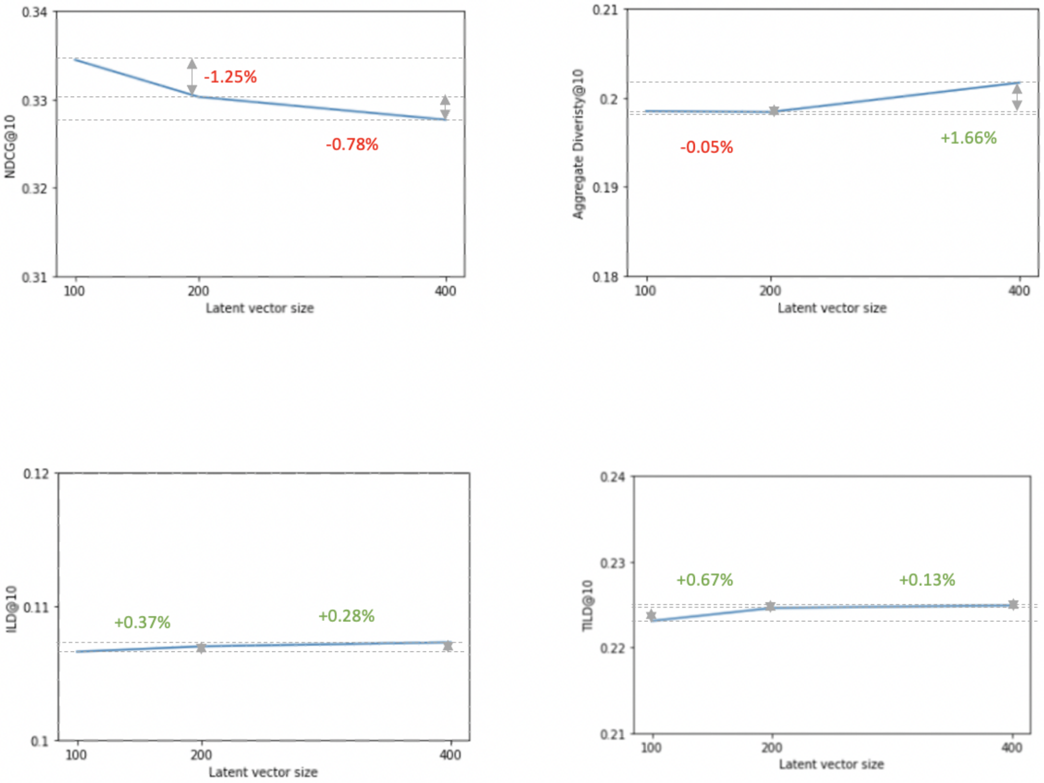


Figure 7.1: Effect of the latent vector on relevance and diversity metrics - MovieLens Dataset. The VAE model trained on the MovieLens dataset is robust to changes in latent vector size. NDCG reduces slightly with an increase in the size of vector size and aggregate diversity increases slightly. ILD and TILD are almost constant

and TILD) increase until the size of the latent vector is 200, and then it stabilizes. Aggregate diversity decreases with the increase in latent vector size for the Bol.com dataset. This could be due to common popular items being recommended to all users regardless of their preferences. For the MovieLens dataset, NDCG decreased by 2%, and Aggregate diversity, ILD, and TILD only increased by 1.6%, 0.6%, and 0.8%, respectively. For the Bol.com dataset, NDCG and Aggregate diversity decrease by 4.5% and 3.7%, respectively, when the vector size is increased from 100 to 300. At the same time, ILD and TILD increased by 1.7% and 2.5%, respectively. These results show that the model is not as sensitive to changes in the latent vector size, but the sensitivity to latent vector size is dependent on the dataset. In the current setting, increasing the latent vector size would reduce relevance and increase individual diversity measures.

7.3.2 Effect of dropout rate

Dropout [38] is a commonly used technique in machine learning models to tackle overfitting, where a percentage of input neurons are randomly removed to decrease the excessive reliance of the model on any particular feature. The dropout rate can fall between 0 to 1, where 0 denotes that none of the input units have been dropped and 1 denotes that all the input units have been dropped. Dropout is a parameter that needs to be tuned. Having a low dropout rate can cause overfitting while having a high dropout rate can cause underfitting, as too much of the input data has been removed. An optimized dropout rate can help the model improve its performance in terms of relevance while increasing its generalization capability. We evaluate the VAE using dropout rates

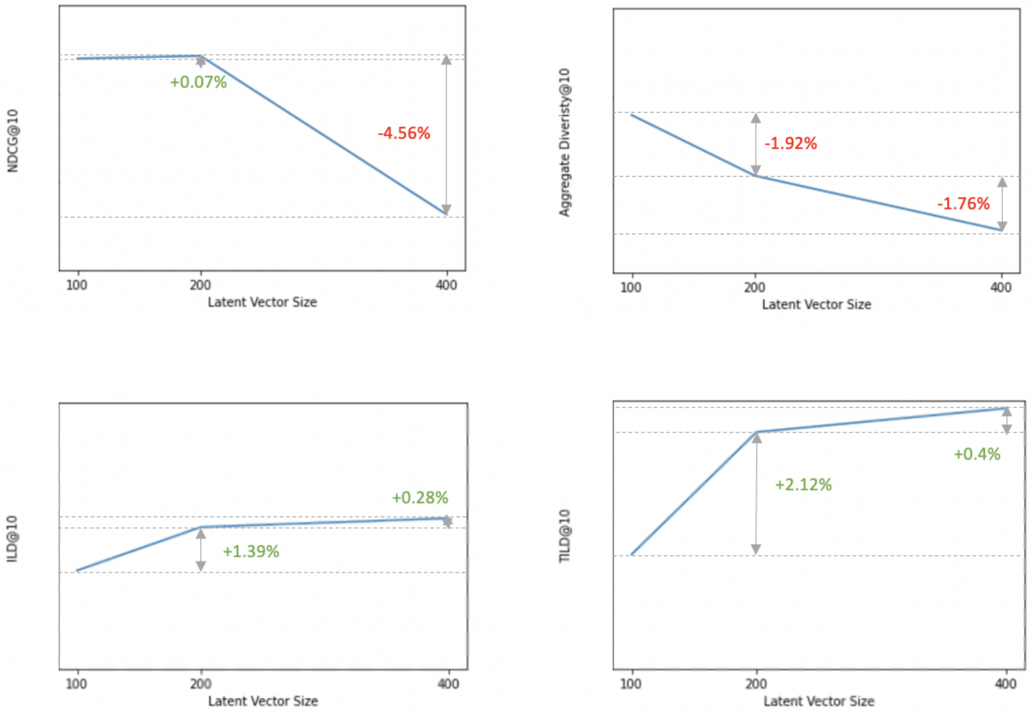


Figure 7.2: Effect of the latent vector on relevance and diversity metrics - Bol.com Dataset. The VAE model trained on the Bol.com dataset is sensitive to changes in latent vector size. NDCG and aggregate diversity reduce with an increase in the size of vector size, and ILD and TILD show a sharper increase up till the latent vector size is 200. Y-axis values have been hidden to protect company confidentiality

of 0.2,0.5, and 0.9 over both datasets. Dropout rate impacts both datasets differently as both datasets have different sparsity. The input is a one-hot encoded vector of all the items a user purchased marked as 1. If there are very few 1s, the chance of any 1's being removed is less, and hence dropout impacts sparse datasets less. Movielens is more sparse than the Bol.com dataset, and therefore, NDCG continues to increase when the dropout rate increases from 0.2 to 0.5 to 0.9, as shown in Figure 7.3. NDCG increases by 17.8% when the dropout rate increases from 0.2 to 0.9. This is due to the effective dropout rate being much smaller because of the sparse input. We can also observe in Figure 7.3 that the diversity measures plunge as the relevance increases with an increase in the dropout rate. Aggregate diversity, ILD, and TILD drop by 48%,19.9%, and 20.9%, respectively, between a dropout rate of 0.2 to 0.9. A dropout rate of 0.5 achieves a balance between NDCG and diversity measures, as the diversity is not compromised too much for increased relevance. In comparison, we can observe in Figure 7.4 that for the Bol.com dataset, the optimal dropout rate would be around 0.5, as a dropout rate of 0.9 would remove too much of the input data and would lead to underfitting. This can be observed by the NDCG plot in Figure 7.4. Increasing the dropout rate in the Bol.com dataset from 0.5 to 0.9 reduced the NDCG by 37.9% and aggregate diversity by 67.2% (fewer products were recommended to all the users). ILD and TILD are not impacted much by the change in the dropout rate. These results show that the VAE models are highly susceptible to the changes in dropout, and dropout needs to be tuned individually based on the dataset properties like sparsity.

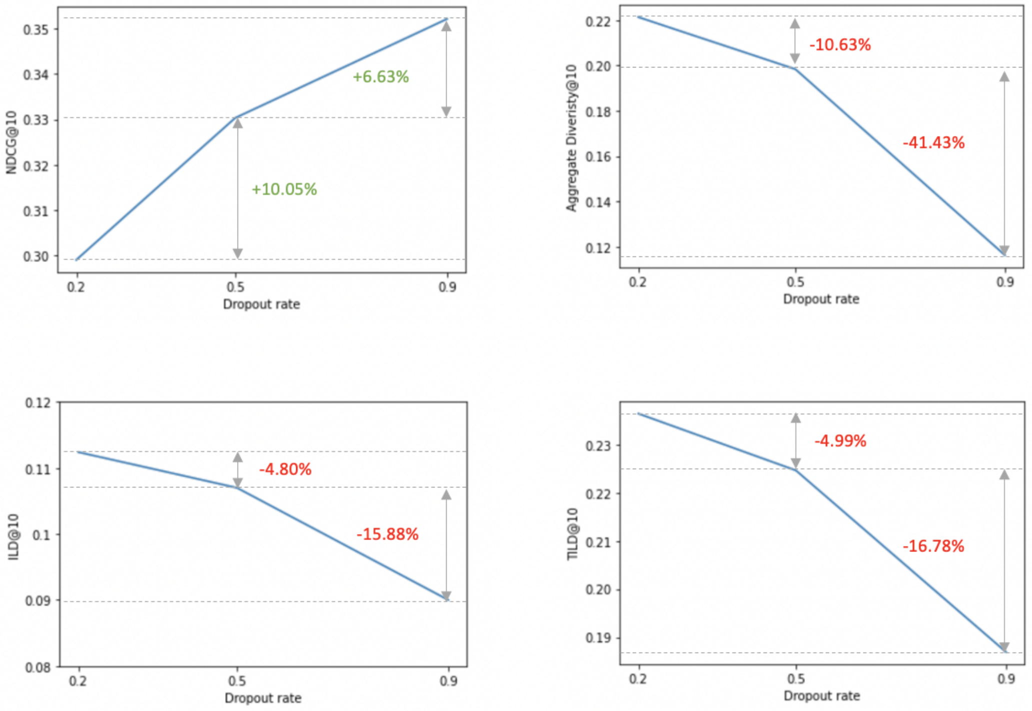


Figure 7.3: Effect of the dropout rate on relevance and diversity metrics - MovieLens Dataset. The VAE model trained on MovieLens dataset is highly sensitive to changes in the dropout rate. NDCG increases linearly with the dropout rate, leading to a decrease in individual and item-level diversity metrics

7.3.3 Effect of the Number of vectors sampled from the distribution to create the recommendation list

Sampling multiple vectors from the distribution can help us find multiple ways of representing the user, which could potentially help improve the diversity of the model. The recommendations from each sampled vector z are combined to optimize intra-list diversity; hence, it would increase diversity at the expense of NDCG. We can choose as many vectors to create the final recommendation list as we want. The recommendations from all the different vectors need to be combined, and as one of the critical requirements of this method is scalability, it is useful to understand the benefit of increasing the number of vectors and evaluate the feasibility in terms of relevance, diversity, and scalability. We can see in Figure 7.6 and 7.5 that NDCG reduces with an increase in the number of vectors and diversity increases. As we use more profile samples, we explore more items in the vicinity of the user preferences. These items could be relevant to the user, but might not be captured by NDCG as this metric does not take similar products into account when measuring relevance.

We can also observe the rise in diversity plateaus with the increased number of vectors. This could be the result of repetitive recommendations provided by different vectors. It might be more feasible to choose fewer vectors to generate recommendations to ensure scalability and efficiency. Both the figures show that the proposed method of generating recommendations is sensitive to the number of sample vectors z used. For the MovieLens dataset, NDCG was reduced by 20.2%, while ILD, TILD, and aggregate diversity increased by 38.2%, 27.5%, and 12.1% between one to three vectors. For the Bol.com dataset, NDCG and aggregate diversity were reduced by 20.4% and 9.5%, respectively, while ILD and TILD increased by 13% and 12.2%, respectively, between one to three

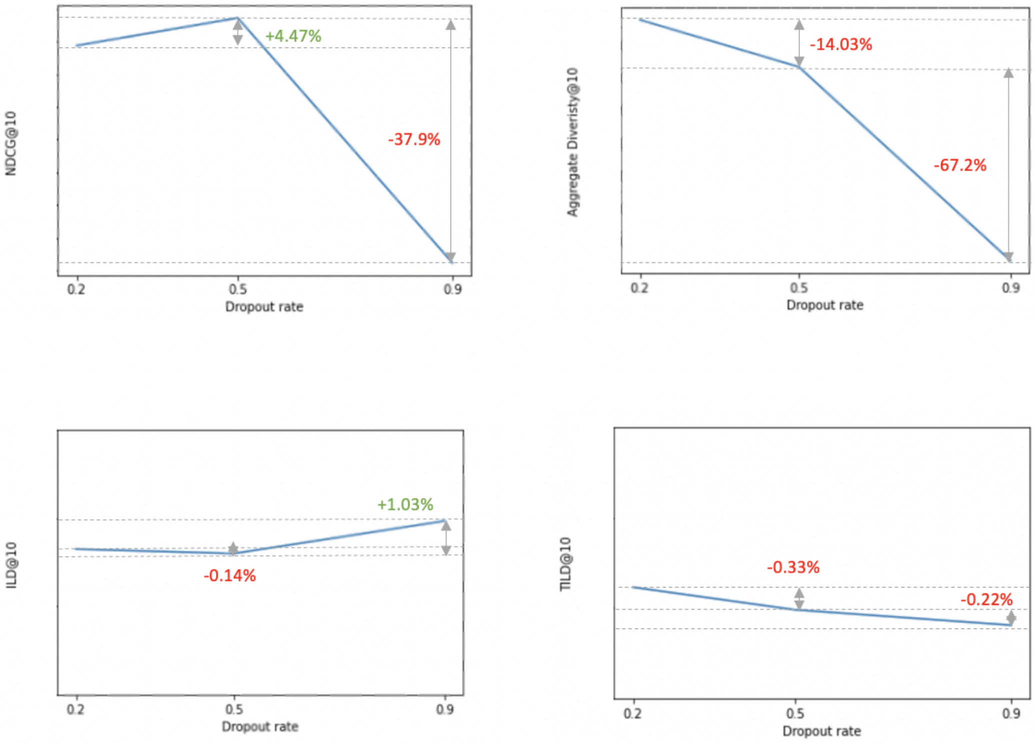


Figure 7.4: Effect of the dropout rate on relevance and diversity metrics - Bol.com Dataset. The VAE model trained on Bol.com dataset is sensitive to changes in the dropout rate. NDCG peaks at an optimal dropout rate of 0.5 with dropout rate. Aggregate diversity decreases with an increase in the dropout rate, while individual-level diversity measures are more robust to the changes in dropout. Y-axis values have been hidden to protect company confidentiality

vectors. The reduction in aggregate diversity means that the increase in the number of vectors sampled results in common popular items being recommended rather than niche personalized items. These results show that increasing the number of vectors would result in a decrease in NDCG and an increase in individual-level diversity.

Based on the experiments, we can conclude that the system is the most sensitive to dropout rate changes, followed by the number of vectors sampled to form the distribution to create the final recommendation list, followed by changes in the latent vector size. Parameter effects depend on the dataset; hence, it is important to consider each dataset's unique characteristics, such as sparsity, to decide the value of different parameters.

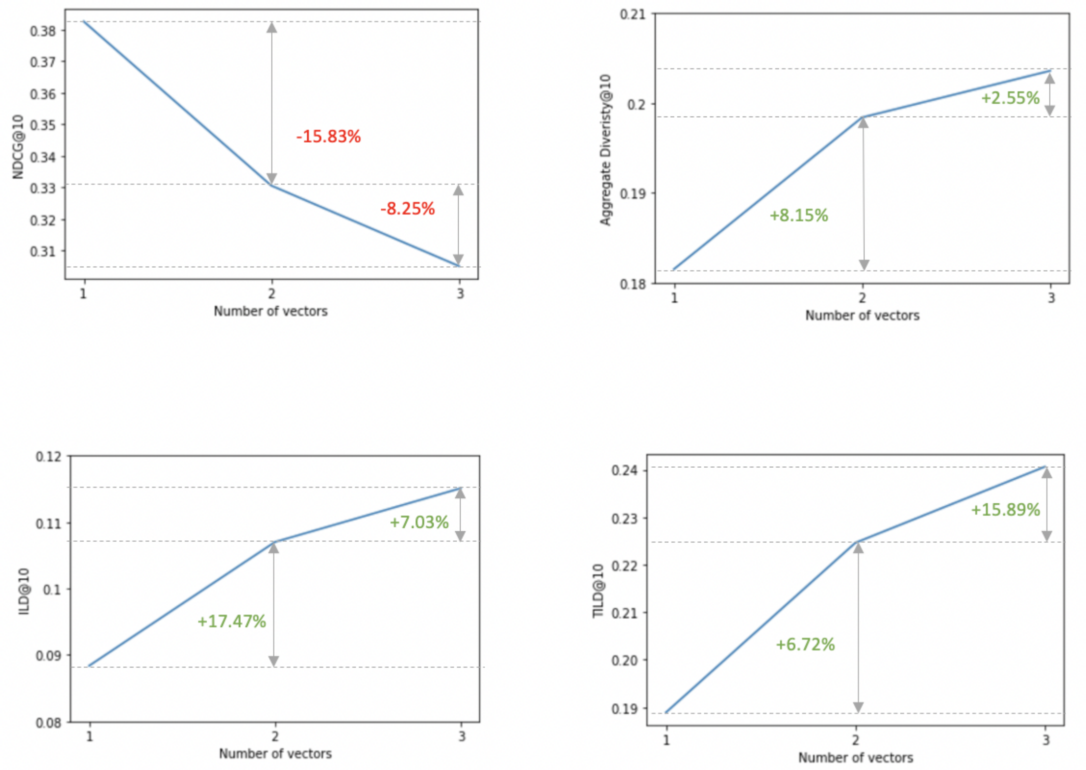


Figure 7.5: Effect of the number of vectors sampled from the distribution on relevance and diversity metrics - MovieLens Dataset. The method trained on the MovieLens dataset is sensitive to changes in the dropout rate. NDCG is negatively correlated to the number of vectors used to produce the recommendations and positively correlated with different diversity measures.

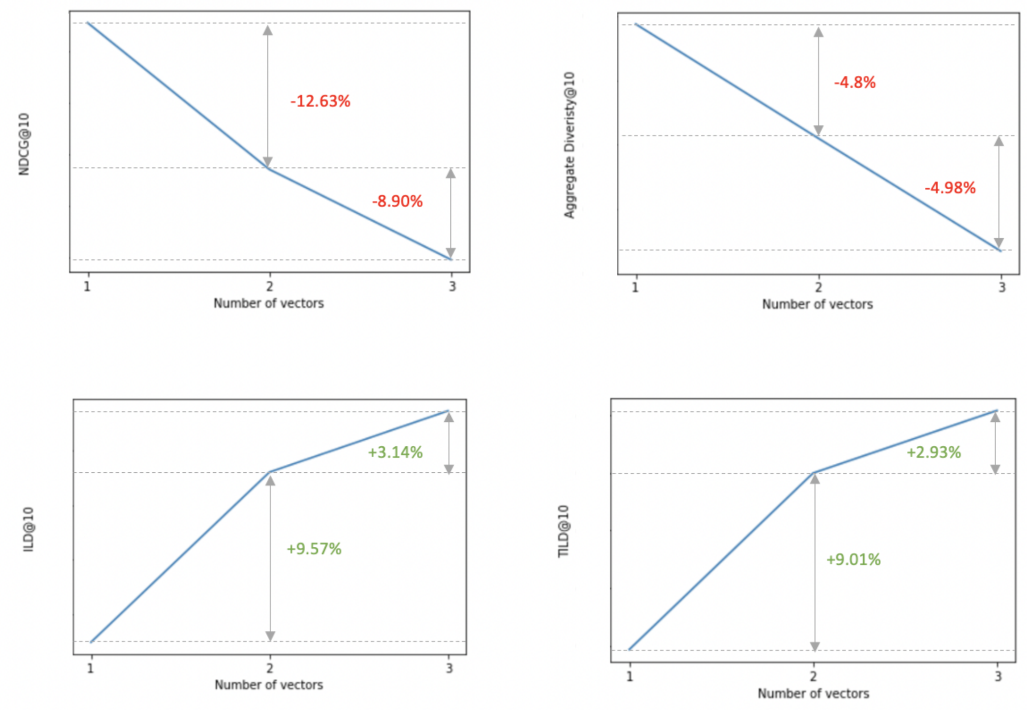


Figure 7.6: Effect of the number of vectors sampled from the distribution on relevance and diversity metrics - Bol.com Dataset. The method trained on Bol.com Dataset is sensitive to changes in the number of vectors used to represent the users. NDCG is negatively correlated to the number of vectors used to produce the recommendations and positively correlated with different diversity measures. Y-axis values have been hidden to protect company confidentiality

7.4 Limitations

One limitation of this experiment is that we have not performed a complete global sensitivity analysis due to the time constraints of this project. However, we believe that deltas and insight should be more valuable to people attempting to use this method than a complete analysis. Another limitation is that we have not covered a few parameters in this analysis due to time constraints. Other parameters that could be evaluated are the weight of KL divergence and different ways of combining the recommendation lists generated by multiple vectors z sampled from the distribution. Future work could explore these directions.

7.5 Summary

In this chapter, we analyzed the effect of changing the different parameters on the diversity and relevance of the recommendations generated by our method. 7.1 explained this experiment's motivation and connected it to the research question being answered in this study. Section 7.2 described the experiment methodology, and the chapter is concluded with the results of the experiment 7.3.

Chapter 8

Conclusions

8.1 Concluding Summary

In this research, we have focused on the diversity aspect of RSs. We seek to improve diversity by altering the conventional user representation using the proposed VAE-based user profile generation method. The main RQ we have attempted to answer using our method is -

> *"Can representing users as a distribution rather than a point estimate improve the user and item level diversity in RS while maintaining an acceptable level of relevance?"*

This question has been divided into five RQs which are described in the following sections.

8.1.1 Diversity in RSs

This chapter described the motivation for diversity, and why it is lacking in existing RSs and provided an analysis of diversity measures. We have explained how users perceive diversity as high relevance as they feel more represented when they are shown a list of items that represent their varied interests. Moreover, the section shares how users expect RSs to evolve with them and help them identify more content that is diverse and novel. We have described existing definitions of diversity and evaluated their need, advantages, and disadvantages. Using this section we have answered the first RQ of this study -

RQ1 - *"How should diversity be measured holistically in real-world RSs?"*

We have chosen ILD and aggregate diversity metrics to measure how diverse a single recommendation list is for each user and the proportion of total items that get user exposure, respectively. We have also motivated why there is a need to introduce a new diversity metric called temporal inter-list diversity to measure diversity for each user temporally. These metrics have been selected because they provide a holistic view of RS diversity from the user and vendor perspective while having low time complexity.

8.1.2 VAE for Diversity in RSs

This chapter described the existing techniques for improving diversity in RSs, which were categorized into post-processing and algorithmic techniques. We also discussed the advantages and disadvantages of both. We presented some case studies where VAEs have been used to improve diversity in different NLP tasks. Inspired by these studies and studies proving that multiple user profiles better represent users' varied interests, we proposed a method to generate multiple user profiles using VAEs. We provided a step-by-step description of the method along with an evaluation of the method against the requirements mentioned in Chapter 3. The chapter concluded with the limitations of the proposed method.

8.1.3 Comparative study between representing user as a point estimate vs a distribution

In the next chapter of the thesis, we described the experiment we conducted to answer the second, third, and fourth RQs of this study described in Chapter 3 -

These questions arise from the problem context and user complaints mentioned in Chapter 3. We compared recommendations from a user profile point estimate learned by a vanilla AE to the recommendations generated by the proposed method. These recommendations were compared on relevance and various diversity metrics to address all the research questions mentioned above. The following metrics were used for the experiment - NDCG, ILD, TILD, and Aggregate diversity.

RQ2 - *“Can representing users as a distribution rather than a point estimate improve user-level diversity of recommendations generated in a single session while maintaining an acceptable level of relevance?”*

NDCG and ILD can be used to answer RQ2. Our method of generating multiple user profiles drawn a distribution leads to a 31% increase in ILD for a list of size 5 on the benchmark dataset. For this dataset, there was also a significant decrease of 25% in NDCG for an increase in individual diversity. However, for the real-world dataset, the decrease in relevance was only 5% for a list of size 5, while there was a 9% increase in diversity. Hence, we can conclude that for the real-world application setting, using a distribution rather than a point estimate can improve the user-level diversity of recommendations generated in a single session while maintaining an acceptable level of relevance.

RQ3 - *“Can representing users as a distribution rather than a point estimate improve user-level diversity of recommendations generated over multiple sessions while maintaining an acceptable level of relevance?”*

NDCG and TILD can be used to answer RQ3. Our method of generating multiple user profiles using a distribution leads to a 47% increase in TILD for a list of size 5 on the benchmark dataset. For this dataset, there was also a significant decrease of 25% in NDCG for an increase in individual temporal diversity. However, for the real-world dataset, the decrease in relevance was only 5% for a list of size 5, while there was an 18% increase in diversity. We can notice that we get a significant increase in TILD, for a small decrease in relevance with our method. Hence, we can conclude that for the real-world dataset, using a distribution rather than a point estimate can improve the user-level diversity of recommendations generated over multiple sessions while maintaining an acceptable level of relevance.

RQ4 - *“Can representing users as a distribution rather than a point estimate increase the total number of products recommended to all users while maintaining an acceptable level of relevance?”*

NDCG and Aggregate diversity can be used to answer RQ4. Our method of generating multiple user profiles using a distribution leads to a 48% increase in aggregate diversity for a list of size 5 on the benchmark dataset. For this dataset, there was also a significant decrease of 25% in NDCG for an increase in individual temporal diversity. However, for the real-world dataset, the decrease in relevance was only 5% for a list of size 5, while there was a 33% increase in diversity. We can notice that we get a significant increase in aggregate diversity, for a small decrease in relevance with our method. Hence, we can conclude that for the real-world dataset, using a distribution rather than a point estimate can increase the total number of products recommended to all users while maintaining an acceptable level of relevance.

Our study shows that the proposed method can help improve user and item-level diversity in real-world RSs. This came at a slight cost towards recommendation relevance in the benchmark dataset, but the cost was minimal in the real-world dataset. As this method will be used at an exploratory stage of the user journey where diversity is more important than relevance, 5% is an acceptable decrease in relevance for the real-world dataset. These results provide empirical evidence that our method can be useful in various industrial settings to tackle several issues - 1) improve user satisfaction as it is linked with user-level diversity, 2) improve the feedback loop and popularity bias issue in RSs by increasing item-level diversity.

8.1.4 Effect of different parameters on the diversity relevance tradeoff

In this experiment, we answered the final sub-RQ of the thesis -

RQ5 - *"How does changing different parameters in the system impact the potential diversity-relevance tradeoff?"*

We evaluated the effect of the following parameters on the metrics that were used in the previous experiment: 1) changing the size of the latent vector that represents the user profile, 2) changing the number of vectors that were sampled from the distribution learned by the VAE to produce the final recommendation list, and 3) changing the dropout rate of the input fed to the encoder in the VAE. We have also described why these parameters are selected and the expected outcome of changing these parameters. Through the experiments, we found that the diversity relevance tradeoff is more prominent when evaluating relevance with user-level diversity. On some occasions, item-level diversity decreases with relevance to the change of specific parameters. Moreover, we found the relevance and diversity of the recommendations were the most sensitive to dropout rates, followed by the change in the number of user profiles sampled from the distribution. The diversity of the model was relatively robust to changes in the size of the latent vector of the user profile, specifically user-level diversity measures. We also observed that the model sensitivity to specific parameters depends on dataset properties like sparsity.

8.2 Limitations and future work

8.2.1 Diversity in RSs

In our literature review, we have tried to provide diversity definitions based on how different were the dimensions they covered. This research is not exhaustive and we cannot be sure that none of the existing research in recommendation system diversification has not been omitted. This is also the limitation of the other sections of this chapter.

8.2.2 VAE for Diversity in RSs

We have attempted to describe different techniques for improving diversity in RSs, but these techniques are not exhaustive and have been chosen to explain the breadth of research on the topic.

Moreover, the method proposed in this chapter has the following limitations. The additional step of combining recommendations from different user profiles adds latency to the method. This method is still feasible for small values of k. Another limitation is that we make recommendations under the assumption that the information we know about the users already informs us sufficiently about the users and hence the recommendations are only based on users' historical data. This assumption might not be correct in all scenarios or for all datasets, but we have limited the scope of this study by this assumption. The final limitation is that the user profiles sampled from the distribution could overlap and it might be interesting to generate discrete distributions instead of continuous ones using methods like VQ-VAE.

8.2.3 Comparative Study between a user representation as a point mass and a distribution

One limitation of this experiment is that in the real-world data the users have been sampled within a particular time window. This could lead to more active users having a higher chance of getting selected. As multiple datasets were used and the results were consistent across both datasets, we can still conclude the study to be valid.

A limitation of this experiment is that we have only compared recommendations generated by the samples of a Gaussian distribution learned by a VAE to a point estimate learned by a vanilla autoencoder. It would be interesting to evaluate whether representing the user as a different distribution (other than Gaussian) can provide an improvement in terms of diversity. The vectors sampled from the distribution (learned by a VAE) can be homogeneous. It might be interesting to use VQ-VAE, which generates discrete distribution to ensure that the vectors used to represent the user differ sufficiently.

Some definitions of diversity, like ILD and TILD depend on item representations generated by feeding metadata to pre-trained models. These pre-trained models have been shown to have some biases, so it would be interesting to explore if using different pre-trained models provides a different result. It would also be interesting to see how using additional item metadata or a different way to represent the items would impact the output.

8.2.4 Effect of different parameters on the diversity relevancy trade-off

One limitation is that we have not shown how the weight parameter of the KL Divergence term affects the diversity-relevance tradeoff. It might be an interesting angle to study in future work as most of the existing work focuses on tuning that parameter to maximize relevance.

Another limitation of this work is that we have not evaluated how different ways of combining the recommendations from other vectors sampled from the VAE distribution affect the diversity-relevance tradeoff. It would be interesting to assess whether or not a different way of combining the results could change the diversity-relevance tradeoff.

8.3 Final remarks

With the pervasiveness of RSs increasing in our daily life, it is important to think about the algorithms used to build these systems, the biases affecting the data used to build these algorithms, and provide solutions to correct these biases. Through the experiments conducted in this study, we have shown that our proposed way of representing the users can address some common issues users have with RSs while addressing common biases found in RSs. We have combined research from different domains of machine learning and tackled diversity in RSs, along with addressing the human-computer interaction perspective of this problem.

8.3.1 Summary of Contributions

The contributions of this study can be summarized as follows -

- We have provided insight into how diversity measures should be chosen to holistically evaluate real-world RS while considering the limitations of such a system. We have also introduced a new metric called temporal inter-list diversity to measure the pairwise distance between items of two recommendation lists generated at different timestamps.

- We have proposed a scalable method to generate multiple user profiles using existing techniques and have provided empirical evidence that using multiple random samples from a distribution to represent users can help improve item and user-level diversity while maintaining an acceptable level of recommendation relevance. We have also provided a parameter sensitivity analysis to show how different parameters in the system impact the diversity-relevance tradeoff of the proposed method.

Furthermore, we have performed a literature review that discusses motivations, definitions, and techniques for enhancing diversity in RSs. This analysis would be helpful for practitioners and academics looking for a deeper understanding of the domain.

8.3.2 Business Value

This thesis has been motivated by our case study from Bol.com and interactions with real customers. Representing users as a distribution using a VAE does not require a significant change in system design over representing them as a point estimate. So, it can help practitioners improve recommendation diversity with little changes to their existing system design. Our method would expose 789 more item categories to Bol.com customers in a single run and increase individual and temporal diversity by 9% and 19% respectively for a list of size 5. As the decrease in relevance is small, exposure to more categories should improve user experience as per existing studies [22] and tackle the feedback loops created by recommending only popular items. According to research, exposure to more products is also correlated to improved sales for eCommerce companies as well [47]. More studies are needed to validate the impact of implementing this change on the company's sales or user satisfaction in this particular use case, but if we use existing studies as evidence then this low-investment change should improve the users' experience and sales while providing more opportunities to vendors.

Bibliography

[1] Text embedding models contain bias. here's why that matters. 36

[2] Gediminas Adomavicius and YoungOk Kwon. Improving aggregate recommendation diversity using ranking-based techniques. *IEEE Transactions on Knowledge and Data Engineering*, 24(5):896–911, 2012. 7, 13, 15, 16

[3] Chris Anderson. The long tail, Oct 2004. 7, 13

[4] Aqeel Anwar. Difference between autoencoder (ae) and variational autoencoder (vae), Nov 2021. 24, 25

[5] Andrea Asperti and Matteo Trentin. Balancing reconstruction error and kullback-leibler divergence in variational autoencoders. *IEEE Access*, 8:199440–199448, 2020. 27

[6] Andrea Asperti and Matteo Trentin. Balancing reconstruction error and kullback-leibler divergence in variational autoencoders. *IEEE Access*, 8:199440–199448, 2020. 27

[7] Tevfik Aytekin and Mahmut Özge Karakaya. Clustering-based diversity improvement in top-n recommendation. *Journal of Intelligent Information Systems*, 42(1):1–18, 2014. 18

[8] Daniel Billsus and Michael J. Pazzani. Learning collaborative information filters. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, page 46–54, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. 4

[9] Keith Bradley and Barry Smyth. Improving recommendation diversity, 2001. 4, 14, 18

[10] John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering, 2013. 4

[11] Erik Brynjolfsson and Michael Smith. Consumer surplus in the digital economy: Estimating the value of increased product variety. *Management Science*, 49, 11 2003. 13

[12] Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, page 335–336, New York, NY, USA, 1998. Association for Computing Machinery. 8, 18

[13] Allison J. B. Chaney, Brandon M. Stewart, and Barbara E. Engelhardt. How algorithmic confounding in recommendation systems increases homogeneity and decreases utility. In *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM, sep 2018. 11, 12, 13

[14] Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. Are we really making much progress? a worrying analysis of recent neural recommendation approaches. In *Proceedings of the 13th ACM Conference on Recommender Systems*. ACM, sep 2019. 1, 21, 33

[15] Daniel M. Fleder and Kartik Hosanagar. Recommender systems and their impact on sales diversity. In *Proceedings of the 8th ACM Conference on Electronic Commerce*, EC '07, page 192–199, New York, NY, USA, 2007. Association for Computing Machinery. 1, 13

[16] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. 22

[17] Wenshuo Guo, Karl Krauth, Michael I. Jordan, and Nikhil Garg. The stereotyping problem in collaboratively filtered recommender systems, 2021. 1, 13, 15, 18, 26, 28

[18] F. Maxwell Harper and Joseph A. Konstan. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4), dec 2015. 33

[19] Unnat Jain, Ziyu Zhang, and Alexander G. Schwing. Creativity: Generating diverse questions using variational autoencoders. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1, 19

[20] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2013. 22, 23

[21] Diederik P. Kingma and Max Welling. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019. 21, 23, 24, 25, 26

[22] Bart Knijnenburg, Martijn Willemsen, gantner, soncu, and newell. Explaining the user experience of recommender systems. *User Modeling and User-Adapted Interaction*, 22:441–504, 10 2012. 1, 12, 13, 29, 50

[23] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009. 4, 5

[24] Matevž Kunaver and Tomaž Požrl. Diversity in recommender systems–a survey. *Knowledge-based systems*, 123:154–162, 2017. 2, 8, 19

[25] Neal Lathia, Stephen Hailes, Licia Capra, and Xavier Amatriain. Temporal diversity in recommender systems. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, page 210–217, New York, NY, USA, 2010. Association for Computing Machinery. 2, 7, 12, 15

[26] Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. Variational autoencoders for collaborative filtering, 2018. 20, 26, 30, 32, 33

[27] Jian-Guo Liu, Kerui Shi, and Qiang Guo. Solving the accuracy-diversity dilemma via directed random walks. *Physical Review E*, 85(1), jan 2012. 19

[28] Masoud Mansoury, Himan Abdollahpouri, Mykola Pechenizkiy, Bamshad Mobasher, and Robin Burke. Feedback loop and bias amplification in recommender systems, 2020. 7, 13, 14

[29] C. Pichery. Sensitivity analysis. In Philip Wexler, editor, *Encyclopedia of Toxicology (Third Edition)*, pages 236–237. Academic Press, Oxford, third edition edition, 2014. 37

[30] Pleuni. Bol.com: Product range grew 42 percent, Oct 2021. 6, 9

[31] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. 32

[32] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows, 2015. 22

[33] Joseph Rocca. Understanding variational autoencoders (vaes), Mar 2021. 21

[34] David E. Rumelhart and James L. McClelland. *Learning Internal Representations by Error Propagation*, pages 318–362. 1987. 23

[35] Upendra Shardanand and Pattie Maes. Social information filtering: Algorithms for automating "word of mouth". In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 210–217, 1995. 4

[36] Barry Smyth and Paul McClave. Similarity vs. diversity. In *Proceedings of the 4th International Conference on Case-Based Reasoning: Case-Based Reasoning Research and Development*, ICCBR '01, page 347–361, Berlin, Heidelberg, 2001. Springer-Verlag. 14

[37] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015. 22

[38] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. 39

[39] Xiaoyuan Su and Taghi M Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009, 2009. 4

[40] Jake Tae. A step up with variational autoencoders, Feb 2020. 31

[41] Moussa Taifi. Mrr vs map vs ndcg: Rank-aware evaluation metrics and when to use them, Jun 2020. 33

[42] Xin Technology. *Challenges in recommender systems : scalability, privacy, and structured recommendations*. PhD thesis, 01 2015. 2, 4

[43] Clive Thompson. If you liked this, you're sure to love that, Nov 2008. 12

[44] Liwei Wang, Alexander Schwing, and Svetlana Lazebnik. Diverse and accurate image description using a variational auto-encoder with an additive gaussian encoding space. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 1, 19, 20

[45] Jacek Wasilewski and Neil Hurley. Incorporating diversity in a learning to rank recommender system. In *The twenty-ninth international flairs conference*, 2016. 19

[46] Lilian Weng. What are diffusion models?, Jul 2021. 22

[47] Hongzhi Yin, Bin Cui, Jing Li, Junjie Yao, and Chen Chen. Challenging the long tail recommendation. 2012. 1, 7, 50

[48] Mi Zhang and Neil Hurley. Avoiding monotony: Improving the diversity of recommendation lists. In *Proceedings of the 2008 ACM Conference on Recommender Systems*, RecSys '08, page 123–130, New York, NY, USA, 2008. Association for Computing Machinery. 14

[49] Yuchi Zhang, Yongliang Wang, Liping Zhang, Zhiqiang Zhang, and Kun Gai. Improve diverse text generation by self labeling conditional variational auto encoder. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2767–2771, 2019. 20, 21

[50] Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. Improving recommendation lists through topic diversification. In *Proceedings of the 14th International Conference on World Wide Web*, WWW '05, page 22–32, New York, NY, USA, 2005. Association for Computing Machinery. 14

[51] Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. Improving recommendation lists through topic diversification. In *Proceedings of the 14th International Conference on World Wide Web*, WWW '05, page 22–32, New York, NY, USA, 2005. Association for Computing Machinery. 18

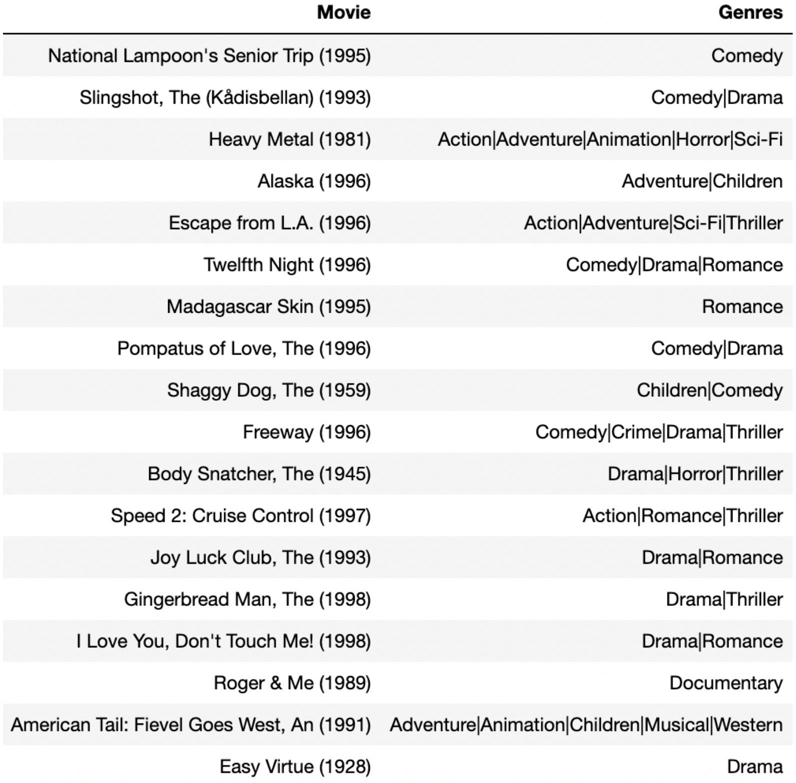
Appendix A

Examples of results

In this chapter, we have provided examples of how using two user profiles helps generate a diverse list of recommendations.

A.1 MovieLens Example

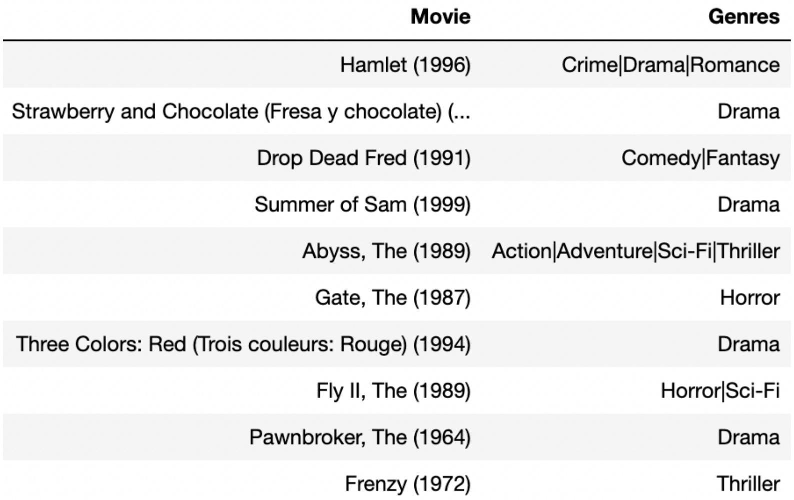
Figure A.1 shows the movies a user has rated highly. We can see that the user is interested in many different genres like 'drama', 'comedy', 'thriller', 'action', 'animation' and 'romance'.



Movie	Genres
National Lampoon's Senior Trip (1995)	Comedy
Slingshot, The (Kådisbellan) (1993)	Comedy|Drama
Heavy Metal (1981)	Action|Adventure|Animation|Horror|Sci-Fi
Alaska (1996)	Adventure|Children
Escape from L.A. (1996)	Action|Adventure|Sci-Fi|Thriller
Twelfth Night (1996)	Comedy|Drama|Romance
Madagascar Skin (1995)	Romance
Pompatus of Love, The (1996)	Comedy|Drama
Shaggy Dog, The (1959)	Children|Comedy
Freeway (1996)	Comedy|Crime|Drama|Thriller
Body Snatcher, The (1945)	Drama|Horror|Thriller
Speed 2: Cruise Control (1997)	Action|Romance|Thriller
Joy Luck Club, The (1993)	Drama|Romance
Gingerbread Man, The (1998)	Drama|Thriller
I Love You, Don't Touch Me! (1998)	Drama|Romance
Roger & Me (1989)	Documentary
American Tail: Fievel Goes West, An (1991)	Adventure|Animation|Children|Musical|Western
Easy Virtue (1928)	Drama

Figure A.1: Example from MovieLens Dataset: Movies rated by a user

Figure A.2 shows the movies that were recommended to this user by a single user profile through a vanilla AE. We can see that these recommendations are mostly in the 'Drama' and 'Action' genres, with one recommendation from 'romance' and one from 'comedy' genre.

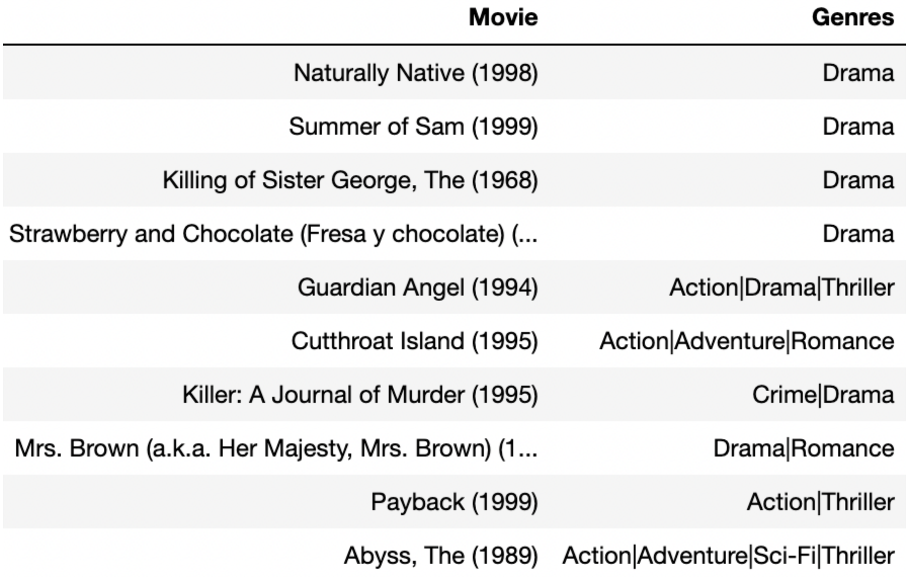


Movie	Genres
Hamlet (1996)	Crime|Drama|Romance
Strawberry and Chocolate (Fresa y chocolate) (...	Drama
Drop Dead Fred (1991)	Comedy|Fantasy
Summer of Sam (1999)	Drama
Abyss, The (1989)	Action|Adventure|Sci-Fi|Thriller
Gate, The (1987)	Horror
Three Colors: Red (Trois couleurs: Rouge) (1994)	Drama
Fly II, The (1989)	Horror|Sci-Fi
Pawnbroker, The (1964)	Drama
Frenzy (1972)	Thriller

Figure A.2: Movies recommended by a vanilla AE where a user is represented by a point estimate. Movies are mostly from 'Drama', 'Horror, 'Thriller' genres

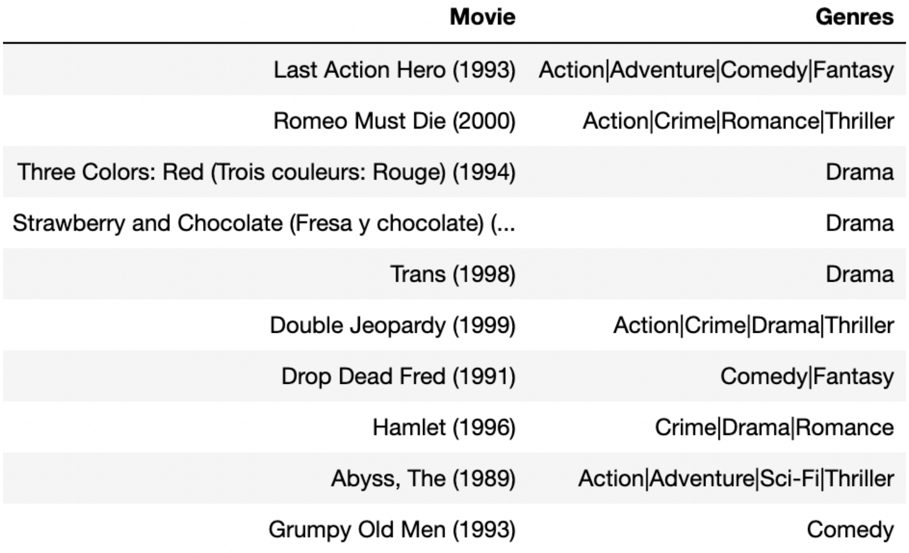
Figure A.5 shows the movies that were recommended to this user by two user profiles sampled from the distribution generated by a VAE. The recommendations generated by the first user profile are in the 'drama', 'action', and 'thriller' genres, while the second list has movies from the 'crime', 'drama', 'romance', and 'comedy' genres.

Figure A.6 shows the list generated by combining the two lists from Figure A.5. We can see that in comparison to the list from Figure A.2, the resulting list one has two 'comedy' movies and three 'romance' movies apart from 'drama' and 'action' movies.



Movie	Genres
Naturally Native (1998)	Drama
Summer of Sam (1999)	Drama
Killing of Sister George, The (1968)	Drama
Strawberry and Chocolate (Fresa y chocolate) (...	Drama
Guardian Angel (1994)	Action|Drama|Thriller
Cutthroat Island (1995)	Action|Adventure|Romance
Killer: A Journal of Murder (1995)	Crime|Drama
Mrs. Brown (a.k.a. Her Majesty, Mrs. Brown) (1...	Drama|Romance
Payback (1999)	Action|Thriller
Abyss, The (1989)	Action|Adventure|Sci-Fi|Thriller

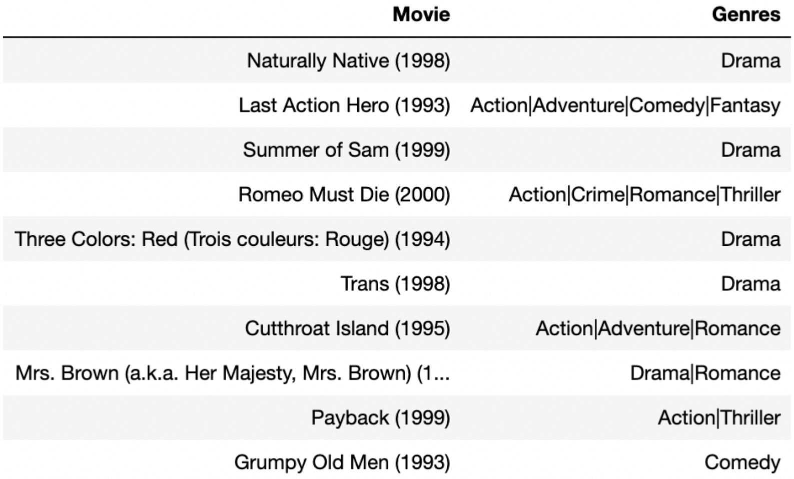
Figure A.3: Movies recommended by the first user profile



Movie	Genres
Last Action Hero (1993)	Action|Adventure|Comedy|Fantasy
Romeo Must Die (2000)	Action|Crime|Romance|Thriller
Three Colors: Red (Trois couleurs: Rouge) (1994)	Drama
Strawberry and Chocolate (Fresa y chocolate) (...	Drama
Trans (1998)	Drama
Double Jeopardy (1999)	Action|Crime|Drama|Thriller
Drop Dead Fred (1991)	Comedy|Fantasy
Hamlet (1996)	Crime|Drama|Romance
Abyss, The (1989)	Action|Adventure|Sci-Fi|Thriller
Grumpy Old Men (1993)	Comedy

Figure A.4: Movies recommended by the second user profile

Figure A.5: Recommendations from two user profiles sampled from the user profile distribution learned by the VAE. We can see the first list has a lot of 'drama', 'action' and 'thriller' movies. The second recommendation list has movies from 'drama', 'crime', 'comedy' and 'romance' genres.

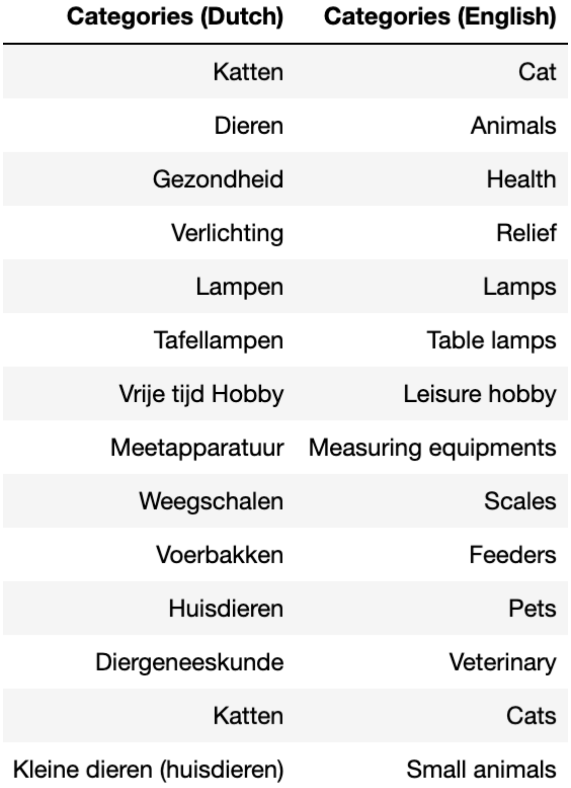


Movie	Genres
Naturally Native (1998)	Drama
Last Action Hero (1993)	Action|Adventure|Comedy|Fantasy
Summer of Sam (1999)	Drama
Romeo Must Die (2000)	Action|Crime|Romance|Thriller
Three Colors: Red (Trois couleurs: Rouge) (1994)	Drama
Trans (1998)	Drama
Cutthroat Island (1995)	Action|Adventure|Romance
Mrs. Brown (a.k.a. Her Majesty, Mrs. Brown) (1...	Drama|Romance
Payback (1999)	Action|Thriller
Grumpy Old Men (1993)	Comedy

Figure A.6: Movies recommended by combining the two lists generated using the VAE. This list has more movies from ‘comedy’ and ‘romance’ genres apart from ‘drama’ and ‘action’ movies than the list from Figure A.2

A.2 Bol.com Example

Figure A.7 shows the categories user has purchased products from. We can see that the user is interested in many different categories like ‘pets’, ‘lamps’, ‘measuring equipments’, ‘health’, and ‘leisure hobbies’.



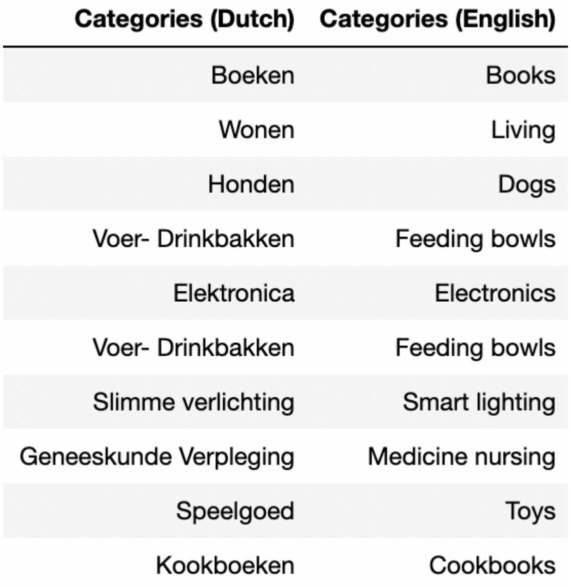
Categories (Dutch)	Categories (English)
Katten	Cat
Dieren	Animals
Gezondheid	Health
Verlichting	Relief
Lampen	Lamps
Tafellampen	Table lamps
Vrije tijd Hobby	Leisure hobby
Meetapparatuur	Measuring equipments
Weegschalen	Scales
Voerbakken	Feeders
Huisdieren	Pets
Diergeneeskunde	Veterinary
Katten	Cats
Kleine dieren (huisdieren)	Small animals

Figure A.7: Example from Bol.com Dataset: Categories user has purchased from includes ‘pets’, ‘lamps’, ‘leisure hobby’, ‘health’ and ‘measuring equipments’

Figure A.8 shows the categories that were recommended to this user by a single user profile through a vanilla AE. We can see that these recommendations are mostly in the ‘pets’ and ‘electronics’ genres, with one recommendation from ‘medicine’, ‘toys’ and ‘cookbook’ category.

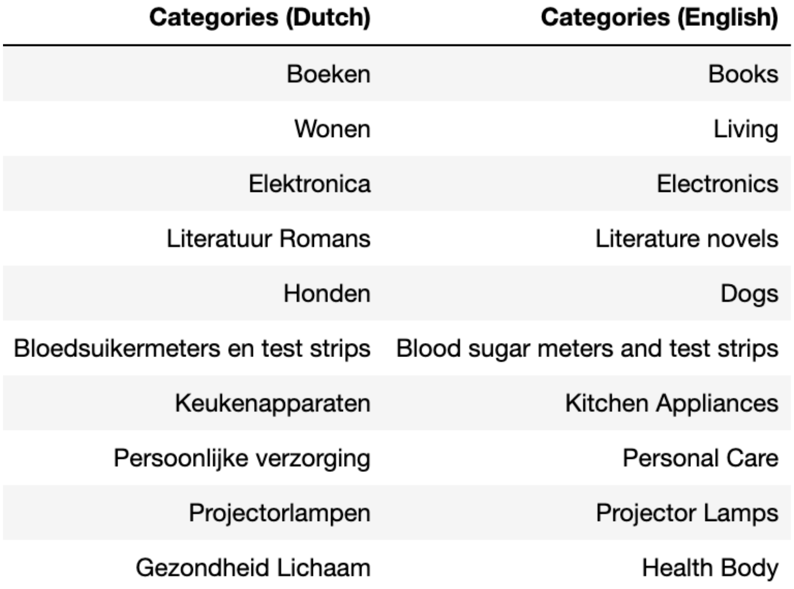
Figure A.11 shows the categories that were recommended to this user by two user profiles sampled from the distribution generated by a VAE. The recommendations generated by the first user profile have a lot of different categories like ‘books’, ‘living’ and ‘electronics’, ‘measuring appliances’, ‘kitchen appliances’, ‘personal care’, ‘lamps’, and ‘health’, but the second list is mostly focused on ‘pets’, ‘books’, ‘living’ and ‘smart lighting’.

Figure A.12 shows the list generated by combining the two lists from Figure A.11. We can see that in comparison to the list from Figure A.8, the resulting list is more diverse and includes ‘measuring equipment’ which was missed in the latter.



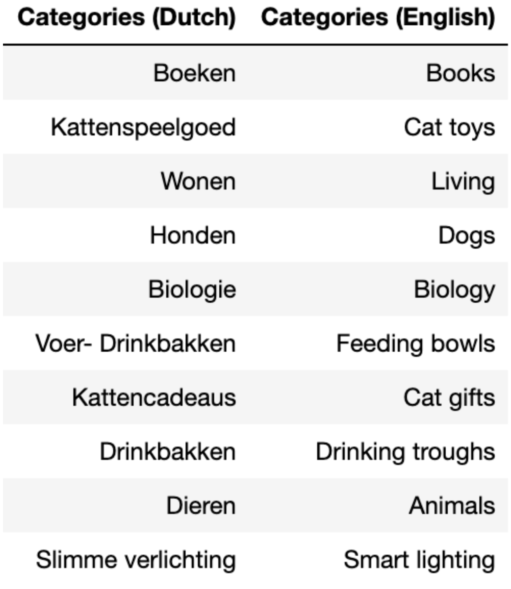
Categories (Dutch)	Categories (English)
Boeken	Books
Wonen	Living
Honden	Dogs
Voer- Drinkbakken	Feeding bowls
Elektronica	Electronics
Voer- Drinkbakken	Feeding bowls
Slimme verlichting	Smart lighting
Geneeskunde Verpleging	Medicine nursing
Speelgoed	Toys
Kookboeken	Cookbooks

Figure A.8: Categories recommended by a vanilla AE where a user is represented by a point estimate. Categories are mostly related to 'pets', 'lighting', 'toys', 'medicine' and 'cookbooks'



Categories (Dutch)	Categories (English)
Boeken	Books
Wonen	Living
Elektronica	Electronics
Literatuur Romans	Literature novels
Honden	Dogs
Bloedsuikermeters en test strips	Blood sugar meters and test strips
Keukenapparaten	Kitchen Appliances
Persoonlijke verzorging	Personal Care
Projectorlampen	Projector Lamps
Gezondheid Lichaam	Health Body

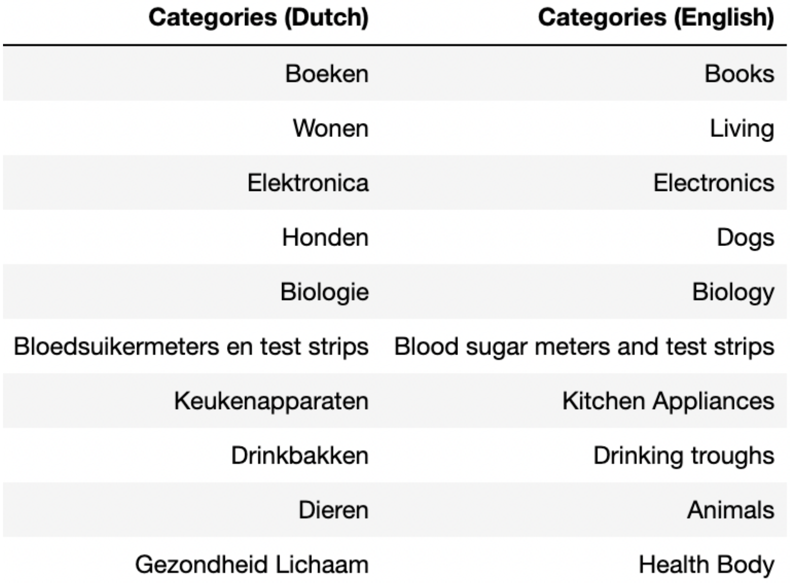
Figure A.9: Categories recommended by the first user profile



Categories (Dutch)	Categories (English)
Boeken	Books
Kattenspeelgoed	Cat toys
Wonen	Living
Honden	Dogs
Biologie	Biology
Voer- Drinkbakken	Feeding bowls
Kattencadeaus	Cat gifts
Drinkbakken	Drinking troughs
Dieren	Animals
Slimme verlichting	Smart lighting

Figure A.10: Categories recommended by the second user profile

Figure A.11: Recommendations from two user profiles sampled from the user profile distribution learned by the VAE. We can see the first list has a lot of different categories like ‘books’, ‘living’ and ‘electronics’, ‘measuring appliances’, ‘kitchen appliances’, ‘personal care’, ‘lamps’ and ‘health’. The second recommendation list has categories mostly related to ‘pets’, ‘books’, ‘living’, and ‘smart lighting’



Categories (Dutch)	Categories (English)
Boeken	Books
Wonen	Living
Elektronica	Electronics
Honden	Dogs
Biologie	Biology
Bloedsuikermeters en test strips	Blood sugar meters and test strips
Keukenapparaten	Kitchen Appliances
Drinkbakken	Drinking troughs
Dieren	Animals
Gezondheid Lichaam	Health Body

Figure A.12: Categories are recommended by combining the two lists generated using the VAE. This list has more categories like 'measuring equipments' which were missed in the vanilla AE list from Figure A.8, like 'measuring equipments'