Université de Montréal

Incorporating Health Factors into Food Recommendation – Experiments on Real-World Data from a Weight-Loss App

par

Yabo Ling

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté en vue de l'obtention du grade de
Maître ès sciences (M.Sc.)
en Discipline

March 2, 2022

Université de Montréal

Faculté des arts et des sciences

Ce mémoire intitulé

Incorporating Health Factors into Food Recommendation – Experiments on Real-World Data from a Weight-Loss App

présenté par

Yabo Ling

a été évalué par un jury composé des personnes suivantes :

Esma Aïmeur
(président-rapporteur)

Jian-Yun Nie
(directeur de recherche)

Bang Liu
(membre du jury)

Résumé

Les systèmes de recommandation typiques tentent d'imiter les comportements passés des utilisateurs pour faire des recommandations futures. Par exemple, dans le domaine des recommandations alimentaires, ces algorithmes de recommandation apprennent généralement d'abord l'historique de consommation de l'utilisateur, puis recommandent les aliments que l'utilisateur préfère. Bien qu'il existe de nombreux systèmes de recommandation d'aliments proposés dans la littérature, la plupart d'entre eux sont généralement des applications directes des algorithmes de recommandation génériques sur des ensembles de données alimentaires. Nous pensons que pour le problème de la recommandation alimentaire, les connaissances spécifiques au domaine joueraient un rôle vital dans la réussite d'un recommandeur alimentaire. Cependant, la plupart des modèles existants n'intègrent pas ces connaissances. Pour résoudre ce problème, dans cet article, nous intégrons des facteurs liés à la santé (tels que l'IMC des utilisateurs, les changements de poids sous-jacents, les calories des aliments candidats et les variétés d'aliments) dans des modèles de recommandations alimentaires séquentielles pour les utilisateurs qui souhaitent mieux gérer leur alimentation et poids. Les changements de poids sous-jacents des utilisateurs sont également traités comme leurs objectifs ou leurs intentions (perdre, maintenir ou prendre du poids). Le modèle proposé devrait adapter en douceur le flux d'articles recommandé vers l'objectif des utilisateurs en tenant compte des préférences de consommation et des facteurs de santé antérieurs de l'utilisateur.

Pour étudier les meilleures stratégies pour incorporer des facteurs de santé spécifiques à un domaine dans les recommandations alimentaires, dans cette étude, nous proposons deux approches de modélisation: la recommandation du prochain article et la recommandation du prochain panier. Ces deux méthodes prennent la séquence passée d'aliments (noms d'aliments et calories) consommés par un utilisateur comme entrée et produisent une liste classée d'aliments pour le prochain aliment (Next-item) ou le lendemain (Next-basket). En outre, les recommandations de base sont améliorées sur la base des approches de pointe de chaque approche de modélisation, qui sont respectivement GRU4Rec [**65**] et LSTM hiérarchique.

Pour étudier l'impact des facteurs de santé et ajuster le modèle vers un objectif, nous construisons des sous-modèles spécifiques pour chaque groupe d'utilisateurs en fonction de

l'IMC et de l'intention. À savoir, les utilisateurs sont regroupés en obèses, en surpoids, normaux, sous-pondérés selon l'IMC. Leurs données (par semaines) sont segmentées en semaines de perte/gain/maintien de poids en fonction du changement de poids au cours de la semaine. Cette dernière segmentation vise à saisir les habitudes de consommation alimentaire liées au poids, qui est traité comme l'intention sous-jacente de l'utilisateur.

Un modèle général formé sur l'ensemble des données historiques mixtes devrait capturer les habitudes générales de consommation alimentaire de tous les utilisateurs, tandis qu'un sous-modèle formé sur l'ensemble spécifique de données pour l'IMC et l'intention capture celles des groupes ou semaines correspondants. Pour un utilisateur au sein d'un groupe d'IMC et avec l'intention de changer de poids, nous appliquons le sous-modèle spécifique, combiné avec le modèle général, pour la recommandation alimentaire.

Nos modèles sont formés sur une grande quantité de données de comportement alimentaire d'utilisateurs réels à partir d'une application de gestion du poids, où nous pouvons observer la consommation alimentaire quotidienne et le poids corporel de plusieurs utilisateurs.

Lorsque nous combinons le modèle complet général avec les modèles spécifiques à l'IMC et spécifiques à l'intention avec un coefficient approprié, nous observons des améliorations significatives par rapport aux performances du modèle général basé à la fois sur la recommandation de l'article suivant et sur la recommandation du panier suivant. De plus, les sous-modèles spécifiques à l'IMC et spécifiques à l'intention se sont avérés utiles, ce qui donne de meilleurs résultats que le modèle complet général, tandis que les sous-modèles spécifiques à l'IMC ont plus d'impact que le modèle spécifique à l'intention.

En pratique, pour un utilisateur qui a l'intention de perdre du poids, le système peut appliquer le modèle de résultat Perte de poids (avec l'IMC correspondant) à l'utilisateur. Cela tend à ajuster en douceur le modèle général de recommandation vers cet objectif. En outre, le niveau d'ajustement pourrait être contrôlé par le coefficient de combinaison de modèles. En d'autres termes, avec un coefficient plus élevé, le sous-modèle spécifique aura un impact plus important sur la prédiction du classement final des aliments, ce qui implique que le système donnera la priorité à la réalisation de l'objectif de l'utilisateur plutôt qu'à l'imitation de ses habitudes alimentaires précédentes. Cette stratégie est plus efficace que de toujours recommander certains types d'aliments hypocaloriques, qui ne sont pas appréciés par l'utilisateur. L'intention est alignée sur le résultat de poids réel au lieu de l'intention indiquée par l'utilisateur. Ce dernier s'avère beaucoup moins performant dans nos expérimentations.

Mots-clés: Recommandation alimentaire, Facteurs de santé, Perte de poids, L'apprentissage en profondeur

Abstract

Typical recommender systems try to mimic the past behaviors of users to make future recommendations. For example, in the food recommendation domain, those recommenders typically first learn the user's previous consumption history and then recommend the foods the user prefers. Although there are lots of food recommender systems proposed in the literature, most of them are usually some direct applications of generic recommendation algorithms on food datasets. We argue that for the food recommendation problem, domain-specific knowledge would play a vital role in a successful food recommender. However, most existing models fail to incorporate such knowledge. To address this issue, in this paper, we incorporate health-related factors (such as users' BMI, underlying weight changes, calories of the candidate food items, and food varieties) in sequential food recommendation models for users who want to better manage their body weight. The users' underlying weight changes are also as treated as their goals or intents (either losing, maintaining, or gaining weight). The proposed model is expected to smoothly adapt the recommended item stream toward the users' goal by considering the user's previous consumption preferences and health factors.

To investigate the best strategies to incorporate domain-specific health factors into food recommenders, in this study, we propose two modeling approaches: Next-item Recommendation and Next-basket Recommendation. These two methods take the past sequence of foods (food names and calories) consumed by a user as the input and produce a ranked list of foods for the next one (Next-item) or the next day (Next-basket). Besides, the basic recommendations are improved based on the state-of-the-art approaches of each modeling approach, which are GRU4Rec [**65**] and hierarchical LSTM, respectively. To investigate the impact of health factors and tune the model toward a goal, we build specific sub-models for each group of users according to BMI and intent. Namely, users are grouped into Obese, Overweighted, Normal, Underweighted according to BMI. Their data (by weeks) are segmented into weight losing/gaining/maintaining weeks according to the weight change during the week. This latter segmentation aims to capture food consumption patterns related to weight outcome, which is treated as the user's underlying intent. A general model trained on the whole mixed historical data is expected to capture the general food consumption patterns of all the users, while a sub-model trained on the specific set of data for BMI and intent captures those of

the corresponding groups or weeks. For a user within a BMI group and with the intent of weight change, we apply the specific sub-model, combined with the general model, for food recommendation. Our models are trained on a large amount of eating behavior data of real users from a weight management app, where we can observe the daily food consumption and the body weight of many users.

When we combine the general full-model with the BMI-specific and intent-specific models with appropriate coefficient, we observe significant improvements compared with the performance of the general model based on both Next-item Recommendation and Next-basket Recommendation. Furthermore, both BMI-specific and intent-specific sub-models have been proved useful, which achieves better results than the general full-model, while BMI-specific sub-models are more impactful than the intent-specific model.

In practice, for a user who intends to lose weight, the system can apply the Losing-weight outcome model (with the corresponding BMI) to the user. This tends to smoothly adjust the general recommendation model toward this goal. Besides, the adjustment level could be controlled by the coefficient of model combination. In other words, with a larger coefficient, the specific sub-model will have a greater impact on predicting the final food ranking list, implying that the system will prioritize achieving the user's goal over mimicking their previous eating habits. This strategy is more effective than always recommending some types of low-calorie foods, which are not liked by the user. The intent is aligned with the actual weight outcome instead of the indicated intention by the user. This latter turns out to be much less successful in our experiments.

Keywords: Food recommendation, Health factors, Weight loss, Deep learning

Contents

List of tables

List of figures

Liste des sigles et des abréviations

NLP	*Natural Language Processing*
TF-IDF	*Term Frequency-Inverse Document Frequency*
GloVe	*Global Vectors for Word Representation*
BERT	*Bidirectional Encoder Representations from Transformers*
ELMo	*Embeddings from Language Model*
RNN	*Recurrent Neueal Network*
GRU	*Gated Recurrent Unit*
LSTM	*Long Short-Term Memory*
Uni-LSTM	*Unidirectional Long Short-Term Memory*
Bi-LSTM	*Bidirectional Long Short-Term Memory*
BPTT	*Backpropagation Through Time*

CNN	*Convolutional Neural Network*
MC	*Markov Chains*
MF	*Matrix Factorization*
FPMC	*Factorizing Personalized Markov Chains*
TansRec	*Translation-based Recommendation*
STAMP	*Short-Term Attention/Memory Priority model*
SASRec	*Self-Attention based Sequential model*
BMI	*Body Mass Index*
HHI	*Herfindahl-Hirschman Index*
mHealth	*Mobile Health*
Hit	*Hit-Ratio*
NDCG	*Normalized Discounted Cumulative Gain*
MRR	*Mean Reciprocal Rank*

ARHR *Average Reciprocal Hit Ratio*

MAT *Multi-dimensional Adaptation Transformation*

HHFR *Health-factor-aware Hierarchical Food Recommender*

Remerciements

It's my great honor to take this opportunity to express my special thanks to many people who have helped me a lot during the past two years. Without their support and encouragement, I cannot finish this thesis.

First and foremost, I would like to express my earnest gratitude to my supervisor, Professor Jian-Yun Nie. He gave me a cherishable opportunity to pursue my Master's degree under his supervision. Besides, during my research and writing processes, he guides me patiently not only on technical and methodology aspects but also helping me improve self-cultivation in my life, which benefits me a lot. Without his patient instruction, insightful criticism, expertise and constructive guidance, the completion of this thesis would not have been possible.

Second, I'm also indebted to the professors and classmates in our lab for their cordial help, steadfast favors, and academic inspiration. They spare no efforts to give me emotional support and actual deeds during the past two years. I am deeply moved by the mutual support among each other in thesis writing. Besides, I will give my sincere thanks to my friends for their help and care for me during my graduate study. It will be a beautiful and cherishing memory in my life.

Finally, I would like to express my deep appreciation to my family, especially my dear parents, brother, and sister, who are always giving their persistent support and love in my study and my life. Their constant encouragement and unconditional tolerance and considerate companion are the most powerful emotional anchor for me to complete my postgraduate study.

Chapter 1

Introduction

Overweight and obesity are among the main causes to chronic diseases [**58**]. Body weight is a health aspect that many people care about today and try to manage. Many apps and web-based programs have been developed to help users manage their body weight [**73, 46, 38, 21**]. It is known that the main factors important to body weight are related to genetics, behavior, diet, and environment. Among them, diet is one of the most important controllable factors that affect body weight and body health [**5, 31**]. However, according to the World Health Report 2020, the incidence rate of numerous diet-related diseases, such as diabetes, obesity, and malnutrition, is quickly growing over the world. [1] In particular, obesity has reached epidemic proportions, with at least 2.8 million people dying each year as a result of being overweighted or obese. Therefore, a well-balanced diet is crucial to maintain one's physical health.

Food recommendation has been an extensively studied topic because of its high potential impact on the human physical health. Food recommendation can follow dietary guidelines developed by experts. [2] For example, it is recommended to limit the amount of sodium to less than 2,300 mg per day. While these guidelines are important, they can hardly be translated by the general population into their daily life [**72**]. Recommendation systems can offer more effective solutions by recommending concrete food items or ingredients to users on the fly. The popularity of such tools is growing with the widespread utilization of mobile devices or the web.

A typical type of food recommendation system attempts to capture users' previous food preferences and provide future recommendations accordingly, regardless of health factors [**17, 56, 65, 20, 70**]. For example, a recommender system may keep suggesting 'pizza' to a user who often consumed 'pizza' in the past. The recommendation may be easily adopted by the user, but this cannot lead the user to a healthy diet or eating pattern.

[1] https://www.who.int/data/gho/data/themes/topics/topic-details/GHO/world-health-statistics
[2] https://food-guide.canada.ca/en/guidelines/; https://www.who.int/news-room/fact-sheets/detail/healthy-diet

A second type of recommendation is to impose some diet constraints to users. For example, the caloric consumption is controlled within the recommended range. In a system developed to deal with malnutrition for elderly, Aberg [**1**] proposed a recommendation approach based on constraint satisfaction to meet a set of constraints on several aspects. While such an approach can be applied in specific context which requires tight control, it is difficult to impose strict dietary constraints to general users.

A third approach tries to balance the user preferences and the dietary needs. This approach may lead users to a healthier diet while taking into account the food preferences. The recommended foods may be more easily adopted by users. Our study falls into this category. The existing approaches in this category have been simplistic. [Harvey and Elsweiler 2015] [**18**] make recommendations based on user preferences at first; then the recommended foods are filtered using the required dietary constraints (e.g. food with calories higher than a required value are filtered out). The recommendations made with such a simplistic solution may be hard to be adopted by users.

Different from the previous approaches, in our work, we propose an approach that makes food recommendations based on both user's preferences and the success stories of similar users. The approach we develop relies on a large amount of eating behavior data of real users from a weight management app, where we can observe the daily food consumption and the body weight of many users. The eating behaviors that lead to weight loss are used as good examples to guide food consumption of a similar user who wants to lose weight. This is done by employing a goal-oriented recommendation model trained with eating behavior data that truly led to the goal. This strategy differs from the one that relies on dietary guidelines in the sense that the guidelines are now hidden in the concrete successful behaviors. For a user who has a goal on body weight (e.g. losing weight), we make food recommendation by combining a general preference-based recommendation model with a goal-oriented model based on the successful behaviors of users who lost weight.

It is important to rely on the goal-oriented model of similar users. To this end, we train several goal-oriented models for different groups of users. For our study, we group users into different BMI (Body Mass Index) groups by assuming that users in the same group would have (more) similar dietary needs. These groups can be refined in the future to take into account other factors such as age, sex, and so on. To train goal-oriented models, we segment the data collected from an app into subsets of periods (weeks), where we observe that users truly lose/maintain/gain body weights. Using the subset of data corresponding to each outcome, we can train recommendation models respectively for the goals of losing/maintaining/gaining weight. The intuition is that these users would have adopted food patterns during these periods that successfully lead to the observed weight loss/maintenance/gain. These patterns can be generalized to similar users with the same goal. Therefore, the method we propose in this thesis is to make food recommendation for a user by combining a general recommendation model based on the user's past behaviors, and the goal-oriented model trained on the success periods of similar users. We believe that such a combined model can make smoother recommendations than traditional food recommendation system relying on strict food guidelines.

In this thesis, we explore two modelling approaches for food recommendation: Next-item Recommendation and Next-basket Recommendation. These two approaches use a user's previous sequence of foods (food names and calories) as input and generate a ranked list of foods for the following one (Next-item) or the next day (Next-basket).

To evaluate the effectiveness, the ideal setting would be to let the users use the recommender system and observe their final weight outcome. Unfortunately, this evaluation takes a long time and is not yet feasible. Instead, we use the recorded data as proxy in the following way: For a user with a weight goal (e.g. losing weight), we compare our recommendation approach to a general one for the test periods where the user reaches the goal. If the recommendation accuracy of our approach is higher for these periods, it is considered to be better suited to the the user and the goal. Experiments show that our combined recommendation system can produce higher recommendation accuracy for different weight goals. This suggests that the approach could effectively adapt the recommendations toward the user's goal, while also considering his/her preferences.

The main contributions of this thesis are summarized as follows: (1) We propose a new food recommendation approach that is based on the success behavior of similar users. This can be termed "following good examples". To lead user's food choices toward a goal, we combine a general model with a goal-oriented model. This model combination approach is different from the traditional one based on dietary guidelines. (2) We utilize two modeling approaches, Next-item and Next-basket goal-oriented recommendation models for different groups of users based on successful behavior data that meet the goals. (3) Our experiments on a large set of real data demonstrate the effectiveness of this data-based food recommendation approach.

The work presented in this thesis will also be presented in a paper accepted at The Web Conference 2022 [**39**].

The remaining of the thesis is organized as follows: Background and related works are summarized in Chapter 2. Problem formulation is described in Chapter 3. We introduce dataset and evaluation in Chapter 4, our proposed model in Chapter 5. Then the experimental Results and Conclusion are given in Chapter 6 and Chapter 7.

Chapter 2

Background and Related Work

In this chapter, we describe some basic deep learning techniques we use in our work, as well as the related work on food recommendation. There is a large body of work on deep learning. It is impossible to cover all of them. We limit the coverage of this chapter to only those that are strongly related to our work, i.e. word embedding, sequential neural models, recommender systems and food recommendation.

2.1. Pre-trained Word Embedding

Generally, word embedding models learn a real-valued vector to represent word semantics by taking into consideration of its neighboring words. The basic assumption is that words occurring in similar contexts have similar meanings. After the training, words that are close in this embedded vector space should have similar meanings [**29**].

Pre-training of word embedding is applied on a large set of texts. It is assumed that the word embedding obtained from this process could be transferred to other application contexts, i.e. to this is a special type of transfer learning [**83**]. Once pre-trained, those word embeddings are used in downstream tasks, usually as the initial word representations. Due to the capability to capture some basic language properties (both syntactic and semantic), pre-trained word embedding becomes a mainstream word encoding method and has proven to be highly useful in many current NLP tasks, such as sequence tagging [**34, 41**], text classification [**32**], neural machine translation [**53, 45, 3, 35**], and recommendation systems [**67, 48, 81**].

There are many ways to generate pre-trained word embeddings. Particularly, they could be categorized into two approaches: context-independent methods (Bag of Words [**23**], TF-IDF [**37**], Word2Vec [**43**], GloVe [**50**]) and context-aware methods (ELMo [**52**], Transformer [**74**], BERT [**14**], Transformer-XL [**12**]).

In our work, we adopt GloVe and BERT pre-trained word embeddings to conduct the experiments. We describe the details of these embedding methods in Sections 2.1.1 and 2.1.2.

2.1.1. GloVe Word Embedding

Glove (Global Vectors for Word Representation) [**50**] is an unsupervised context-independent generation approach for word embedding, which relies on the assumption that words occurring in similar contexts have similar meanings, thus similar representations.

The GloVe model is trained on the non-zero entries of a global word-word co-occurrence matrix, which tabulates how frequently words co-occur with one another in a given corpus. Pennington et al [**50**] observes that ratios of word-word co-occurrence probabilities have the potential for encoding some form of meaning. The most general form of the GloVe model is given by:

$$F(w_i, w_j, \widetilde{w}_k) = \frac{P_{ik}}{P_{jk}}, \tag{2.1.1}$$

where i, j are two target words, k refers to the different probe word (is also called context word); $w, \widetilde{w} \in \mathbb{R}^d$ are two sets of word vectors with only random initialization differences; P_{ik} denotes the probability of seeing word i and k together, which is computed by dividing the number of times word i and k appeared together by the total number of times word i appeared in the corpus; Similarly, P_{ik} denotes the probability of word i and k appear together. The optimization process will adjust the word embeddings w_i and w_j so that they tend to become similar if they occur frequently in the same contexts. We will not describe the details about the process, which can be found in [**50**].

The Stanford website[1] provides GloVe pre-trained word embedding, which we can download. In our task, 100-dimension and 300-dimension GloVe word vectors are used to encode food items in Section 5.2.1.1.

An alternative to GloVe is Word2Vec [**43**], which usually leads to similar results when used as the initial word embeddings.

2.1.2. BERT Word Embedding

BERT (Bidirectional Encoder Representations from Transformers) [**14**] is an unsupervised, deeply bidirectional, transformer-based [**74**] method for language representation that is pre-trained from unlabeled text data. The pre-trained BERT model that we will use is the one pre-trained on the BooksCorpus [**82**] with 800M words and English Wikipedia with 2,500M words [**2**].

In contrast to GloVe and Word2Vec, BERT pre-trained word embedding is considered to be context-aware and have the capability to disambiguate the semantic of the same word in different contexts. This is achieved by learning dynamic representations for the same word, while considering its contexts. To better understand the way BERT embeddings are learned,

[1]https://nlp.stanford.edu/projects/glove/

we present two example sentences as follows:

*He is **running** a company.*

*He is **running** a marathon.*

For the representation vectors of word “running”, the context-independent method like GloVe (described in Section 2.1.1) has the same embeddings in these two sentences, whereas the context-aware methods BERT will generate the contextualized embeddings that are more adaptive according to the individual context of these sentences. Apparently, the same word “running” has different semantics in the above two contexts: In the first sentence, running means to physically run, while in the second sentence, running means to operate a company. The different representations for the token “running” in the two sentences are created by aggregating some of the representations of the neighboring words through an attention mechanism. The basic idea is that if the word embedding of a neighboring word is similar to that of the target word (i.e. “running”), then it will be more aggregated into the word embedding of the target word at the next layer. This is the essence of transformer through self-attention [**74**]. Again, we do not present the details of BERT model, as we only use it as our initial word embedding for food names.

There are different models presented by Devlin et al. [**14**] and they are differentiated by the model size, denoted as $\text{BERT}_{\text{BASE}}$ and $\text{BERT}_{\text{LARGE}}$. In our work, we adopt $\text{BERT}_{\text{BASE}}$ model (with 12 stacked Transformer blocks, hidden size is 768, 12 self-attention heads, total parameters is 110M) in our experiments. Therefore, the dimension of our BERT word embedding is 768.

2.2. Sequential Modeling Modules

Sequential models are special machine learning models that deal with (input or output) sequential data. Real applications cvan generate different types of sequential data: Audio and video clips are the natural sequences; text streams, which can be split into words or characters sequences; and time-series data, one of it is users’ consecutive activities, such as users’ shopping records, diet records with timesteps, and so on. Sequence modeling is used in a wide range of applications in real-word scenarios based on different types of sequential data, such as speech recognition, sentiment classification, machine translation, dialog system, sequential recommendation and etc.

Recurrent Neural Network (RNN) is a popular algorithm used in sequence models. Several typical forms of sequence models are like Long Short-Term Memory (LSTM) [**27**] and Gated Recurrent Unit (GRU) [**10**], which achieve great performances on sequential tasks. As we will also deal with sequence data (the sequence behavior of food consumptions) in

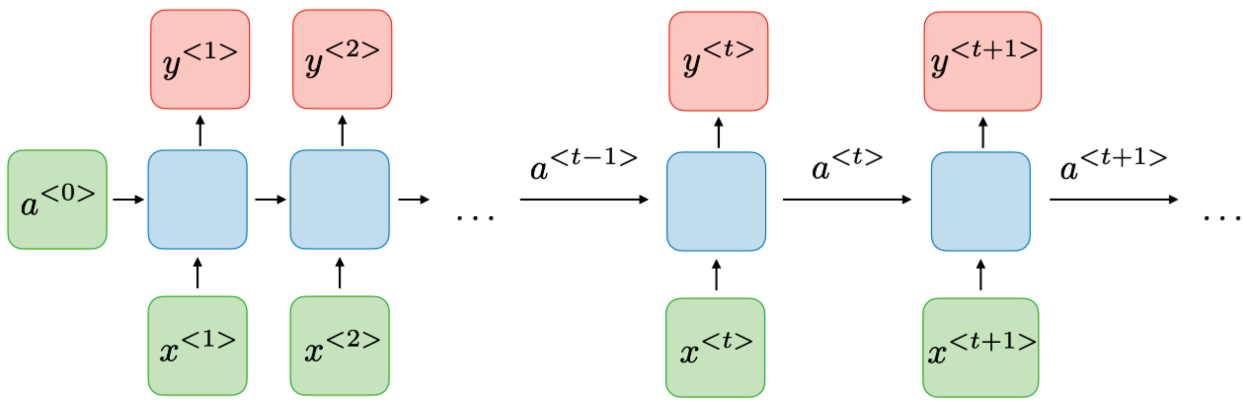

Fig. 2.1. RNNs Architecture

our work, we provide a detailed description about RNN, LSTM and GRU in the following sections.

2.2.1. RNN

One of the most basic sequential models is Recurrent Neural Networks, as also known as RNNs, which is a deep learning algorithm and a type of neural networks that allow previous outputs to be used as inputs while having hidden states. The typically RNNs architecture[2], which is also called vanilla RNN, is shown in Figure 2.1.

We can observe that for each timestep t, RNNs essentially have two inputs, one for the current word $x^{<t>}$, and one for the accumulated input $a^{<t-1>}$, which contains the accumulated information of the previous $t-1$ words. Besides, RNNs also have two outputs, one is the primary output $y^{<t>}$ (which is used when asked to produce an output) and the other is the accumulated output $a^{<t>}$, which represents all the accumulated information of the words that have been input into the RNNs so far. Then $a^{<t>}$ is considered as one of the input at timestep $t+1$, which means the accumulated information keeps getting updated as the RNNs processes each word in a sequence. Furthermore, the formulations of the accumulation $a^{<t>}$ and the output $y^{<t>}$ are expressed in Equation :

$$\mathbf{a}^{<t>} = g_1(\mathbf{W}_{aa}\mathbf{a}^{<t-1>} + \mathbf{W}_{ax}\mathbf{x}^{<t>} + \mathbf{b}_a), \tag{2.2.1}$$

$$\mathbf{y}^{<t>} = g_2(\mathbf{W}_{ya}\mathbf{a}^{<t>} + \mathbf{b}_y), \tag{2.2.2}$$

where $\mathbf{W}_{ax}$, $\mathbf{W}_{aa}$, $\mathbf{W}_{ya}$, $\mathbf{b}_a$, $\mathbf{b}_y$ are the learnable coefficients that are shared across time and g_1, g_2 are activation functions. As a result, we can see that the fundamental benefit of adopting RNNs over other standard neural networks is that the features and weights are shared across time in RNNs, and the computation takes historical information into

[2]https://stanford.edu/ shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks

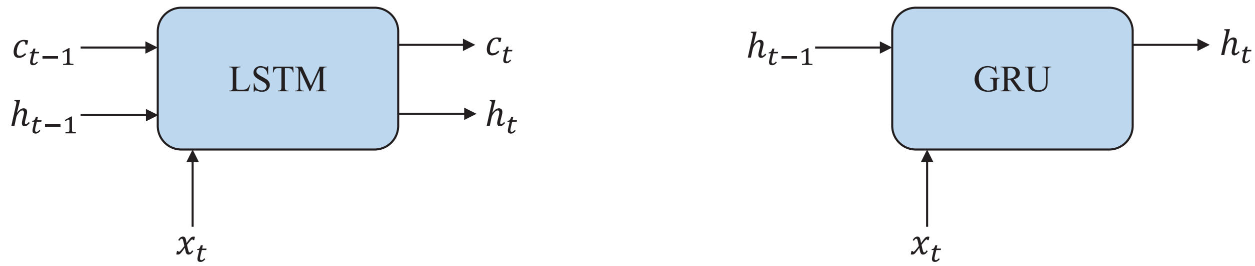

Fig. 2.2. LSTM and GRU cells at timestep t.

consideration. Furthermore, RNNs can handle a variety of input lengths, and the model size does not scale with the size of the input.

However, there are still certain shortcomings of RNNs when processing sequential data. The two most challenging issues are:

(1) **Vanishing or Exploding Gradient Problem:**
The vanishing and/or exploding gradient problems are regularly experienced with regards to RNNs. The reason for this is that any unfolded RNNs are trained in multiple timesteps, with the error gradient calculated as the sum of all gradient errors across timestamps. Then, Backpropagation Through Time, also as known as BPTT [**44**], is is used to update the weights. The dominance of the multiplicative term rises with time as a result of the chain rule while calculating error gradients, and the gradient has a propensity to explode or vanish. The gradient will vanish if the biggest eigenvalue is smaller than 1. The gradient will explodes if the biggest eigenvalue is greater than 1.

(2) **Long Term Dependency:**
Another challenging problem faced by the vanilla RNN is the long-term dependency that the network is different to capture the information which is far away of the current timestep due to vanishing gradient [**7**]. In order to remedy the long-term dependency problem, specific gates are used in some types of RNNs (such as LSTM and GRU) to more easily pass the information or forget the information from the previous timestep to the next one.

The problem of exploding gradients can be solved using gradient clipping, weight regularization [**49**], which L1 (absolute weights) and L2 (squared weights) penalty of the recurrent weights and gradients are often used. However, the problem of vanishing gradient and Long-term dependency are tricky. LSTM [**27**] and GRU [**10**] have partially solved these issues to some extent by introducing well-defined specific gates. We will provide a short description of these mechanisms in the following subsections.

2.2.2. LSTM

Traditional RNNs are not good at capturing long-time dependencies, this is mainly due to the vanishing gradient problem. To overcome these problems, Long Short-Term Memory (LSTM) was introduced by Sepp Hochreiter and Juergen Schmidhuber [**27**] in 1997. For each LSTM cell in timestep t (See the left cell of Figure 2.2), it not only has the hidden states (h_{t-1} and h_t), but also has the cell states (c_{t-1} and c_t) for previous and current timestamp respectively. Moreover, the hidden state is known as short term memory and the cell state is known as long term memory that carries the information along with all the timestamps.

In addition, memory manipulations are done by using three gates in each LSTM cell.

(1) Forget Gate: It decides whether the previous information should be kept or forgotten in the cell state..

(2) Input Gate: It is used to quantify the importance of the new information carried by the input x_t, and adds it to the cell state c_{t-1}.

(3) Output Gate: It determines the value of the next hidden state h_t, which holds information on previous inputs, and adds additional relevant information to the cell state.

The network has learned the circumstances for when to forget, ignore, or keep information in the memory cell thanks to the LSTM's gating mechanism described above.

Additionally, Bidirectional Long Short-term Memory (Bi-LSTM) [**22**] is an extension of unidirectional LSTM (Uni-LSTM). On the input sequence, Bi-LSTM trains by two ways instead of one way of LSTM. The first is from front to back and the second is from back to front. This structure allows the networks to have both backward and forward information about the sequence at every timestep. In other words, an element in a sequence will be able to take into account the context information from both sides. Bi-LSTM achieves outstanding results by better understanding the context in a variety of complex tasks [**62, 42**]. This is a structure we will use to consider the interactions between food items consumed during the same day (i.e. within a basket).

2.2.3. GRU

Gated Recurrent Unit (GRU) is a gating mechanism in recurrent neural networks, introduced by Cho et al [**10**] in 2014. From the right part of Figure 2.2, we can observe that GRU is very similar to LSTM. It also use gates to control the flow of information. However, GRU only use hidden state (h_t) to save and pass the information, unlike LSTM which has a separate cell state (c_t). Therefore, at each timestep t, it takes an input x_t and the hidden state h_{t-1} from the previous timestamp $t-1$. Then a new hidden state h_t is output and sent to the next timestep again.

Talking about the gates, in contrast to the LSTM cell, which has three gates, a GRU cell has primarily two gates:

(1) **Reset Gate:**

In essence, the reset gate is used by the network to determine how much information from the past should be erased, which is in charge of the network's short-term memory.

(2) **Update Gate:**

The update gate, which is for long-term memory (i.e the hidden state h_t), functions similarly to the LSTM cell's forget and input gates. It determines what information should be discarded and what should be included.

GRU uses less parameters and tensor operations. As a result, the training time for GRU model is a litter faster than LSTM.

2.3. General Recommender Systems

Traditional recommender systems, such as Collaborative Filtering (CF) [**33, 60**] Recommendation, model the user-item historical interactions in a static way and capture users' general preferences. However, these approaches ignore the sequential signals, such as the order of users' behaviors. So detecting the appetite of users and their evolution in time has been an active research topic to solve this problem in recent years [**56, 25, 65, 40, 30, 64**]. The problems can be further divided into Next-item and Next-basket recommendations based on the input and output (only one item vs whole basket of items). We will test both modelling methods in our work.

Three main approaches have been proposed to model the sequential behaviors of a user in recommendation systems, which are respectively based on: Markov Chains (MC), Recurrent Neural Network (RNN), and Attention Mechanism. Factorizing Personalized Markov Chains (FPMC) [**56**] and Translation based Recommendation (TransRec) [**25**] are both based on Markov Chains, which are represented in Section 2.3.1 and Section 2.3.2, respectively. We also compare the RNN-based model, GRU4Rec [**65**], in Section 2.3.3. Besides, attention-based models, namely Short-Term Attention/Memory Priority model (STAMP) [**40**], Self-Attention based Sequential model (SASRec) [**30**], and BERT4Rec [**64**], are illustrated in the Section 2.3.4, Section 2.3.5 and Section 2.3.6.

2.3.1. FPMC

In order to predict sequential user actions like the next item to consume, product to purchase, or place to visit, it is essential (and challenging) to model the third-order interactions between a user (u), the item he/she recently consumed (i), and the item to consume next (j).

Factorized Personalized Markov Chain (FPMC) [**56**] models third-order relationships between u, i, and j by a summation of two pairwise relationships: one for the compatibility between u and the next item j, and another for the sequential continuity between the previous item i and the next item j. Ultimately, it combines both a common Markov Chain (MC) and the normal Matrix Factorization (MF) model to model both sequential behaviors and general interests.

2.3.2. TransRec

Translation-based Recommendation (TransRec) [**25**] is also a type of Markov Chain model that uses a novel translation-based structure in a metric space. It has the advantages of employing a single, interpretable component as well as a metric space.

The basic concept of TransRec is as follows: items are embedded as points in a (latent) "transition space" and each user is represented as a "translation vector" in that space. The previously indicated third-order interactions are then recorded by a tailored translation operation: the coordinates of previous item i together with the translation vector of u define (approximately) the coordinates of next item j. Finally, the (u,i,j) triplet's compliance with a distance function is modeled.

The benefits of such an approach are threefold: (1) TransRec naturally models third-order interactions with only one component; (2) TransRec also reaps the generalisation benefits of the implicit metricity assumption; and (3) TransRec can easily handle large sequences (e.g., millions of instances) due to its simple form.

2.3.3. GRU4Rec

RNN is also widely used in the sequential recommendation as it has the strong capability of modeling sequential data. GRU and LSTM are the variants of RNNs to solving gradient vanishing problem, which are described in Section 2.2.

GRU4Rec [**65**] is proposed as a session-based recommendation model, which considers the first item a user clicks when entering a website as the initial input of the RNN, we then would like to query the model based on this initial input for a recommendation. Each consecutive click of the user will then produce an output (a recommendation) that depends on all the previous clicks. As such, the model architecture is relatively simple, we can apply one-hot encoding to the item sequence (input) and pass over to the GRU layer, which is passed onto the forward feed layer and ultimately predicting a (likelihood) ranked list of the next item.

2.3.4. STAMP

Short-Term Attention/Memory Priority model (STAMP) [**40**] adopts attention mechanism in which attention weights are generated from the session context and improved with the users' current interests. The output attention vector is interpreted as a compositional representation of the user's temporal interests, and it is more sensitive to the user's interests drifting across time than other neural attention-based solutions. The current interests can be thought of as a form of short-term memory for the users' preferences.

As a result, STAMP is capable of catching both the users' long-term interests in general (in response to the initial purpose) and their short-term attention at the same time (current interests).

2.3.5. SASRec

Self-attention based Sequential model (SASRec) [**30**] considers to balance two popular techniques for sequential recommendation, which are Markov Chains (MCs) and Recurrent Neural Networks (RNNs). Markov Chains presume that a user's next action can be anticipated based on only their previous (or recent) actions, whereas RNNs allow for the discovery of longer-term semantics. RNNs perform better in denser datasets when increased model complexity is affordable, whereas MC-based approaches perform best in extremely sparse datasets where model parsimony is crucial. Furthermore, SASRec allows us to capture long-term semantics (like an RNN) while making predictions based on a small number of actions (like an MC). It attempts to identify which things from a user's activity history are "relevant" at each timestep and uses them to anticipate the next item. Many empirical investigations show that SASRec outperforms several state-of-the-art sequential models (including MC/CNN/RNN-based approaches).

2.3.6. BERT4Rec

BERT4Rec [**64**] is a sequential recommendation model that uses deep bidirectional self-attention to represent user behaviour sequences. Besides, the Cloze objective is use to sequential recommendation to avoid information leakage and quickly train the bidirectional model, predicting randomly masked items in the sequence by concurrently conditioning on their left and right contexts. In this way, we learn a bidirectional representation model to make recommendations by allowing each item in user historical behaviors to fuse information from both left and right sides, rather than the left-to-right unidirectional representation models

2.4. Food Recommendation

As we mentioned earlier, approaches to food recommendation can be categorized into the following categories [**69**] that consider: (1). user's preferences [**17**]; (2). nutritional needs of users [**1**]; (3). both of them [**18**]. Our study is related to the third category, but we take a different approach than the existing work.

Current methods for food recommendation mainly consider the nutrition information via food analysis or external nutrition guide. For a weight goal, it may be recommended to consume some groups of foods instead of others, or consume different groups of foods at some proportion. Most automatic methods have tried to incorporate healthiness into the recommendation process by substituting ingredients [**68, 9**], incorporating calorie counts [**19**], and generating food plans [**18**]. A critical problem is that the recommended foods or food plans may be very different from the user's habits and food preferences. In many cases, a user will start a diet program, but abandon it after a while [**11, 61**]. The key issue is that a too drastic change in food choices can be hardly adopted by a user. An alternative approach is to make gradual changes, by proposing some food alternatives that are acceptable to the user. Over time, the accumulated changes will eventually lead the user to the goal. This gradual approach has been found more successful, especially for users who are not monitored by a professional [**28, 4**]. This is the general approach we take in this work.

Many food guidelines are created for general population, without taking into account the specific health information of the user. While personalized food guides can be built with the help of a nutrition professional, this solution is not accessible to many people. The general food recommendation systems do not incorporate user's personal health information such as BMI. Some existing recommendation methods balance user's food preference and user's health via simple fusion inside the model. For example, Ge et al. [**19**] simply calculated the weighted mean between the preference component and health component. Elsweiler and Harvey [**18**] used a simple approach to increase or reduce the amount of calories by 500 kilocalories for users who want to gain or to lose weight. However, it may also be the case that the expected outcome is not produced, or the user will not adopt the recommended foods - a critical challenge in this domain [**69**]. We believe that making food recommendation by following real success examples we observed in data can truly lead to the expected outcome. By combining goal-oriented recommendation with user preferences, the recommendations can be more easily adopted by users.

Chapter 3

Problem Description

Overweight and obesity are considered to be the fifth cause of death all over the world. As in 2008, the number of overweight adults was 1.5 billion, of which 200 million of them were obese men and nearly 300 million were obese women [**5**]. Therefore, many people are concerned about their weight and try to control it nowadays.

Many health- and nutrition-related factors are important to consider for human weight and health management. First, the main factors which may determine one's ability of weight management are genetics, behavior, diet, and environment. Among them, diet is the most important factor that affects management of body weight [**5, 31**]. To have a healthy and balanced diet, it is important to consume food which will provide enough but not too much calorie and food's calorie distribution is also an essential factor [**47, 24, 71**]. Studies have shown that calorie intake less than the required amount or changes in the food calorie distribution may have impacts on one's weight management [**75**]. To better understand the roles of calorie intake and food variety, we will describe the calorie factor and food variety factor in details in Section 3.1 and Section 3.6, respectively.

Besides, regarding the food, many research papers [**8, 16, 54, 78**] in nutrition science show that users' starting weights and their intents could have huge impacts on their choices of food, therefore resulting in quite different diets. Thus, we also consider these two health factors, which are described in details in Section 3.2 and Section 3.3, respectively.

Each health factor may have positive, negative or no impact for human body weight management. It is interesting to explore how these health factors could affect the food recommendation performances. Therefore, in our task, we will incorporate the above mentioned health factors (Calorie, BMI, Intent, and Variety) into food recommendation, and conduct experiments on the real-world data from a wildly-used weight control application.

Furthermore, these health factors are defined at different levels: Calorie is for item-level, which means there are calorie values for each corresponding items; BMI and Intent are defined at the user-level; Variety is calculated based on basket-level (a set of food items). So

Table 3.1. Average Calorie for Each Food in An Example Day

Item	Average calorie
coffee	45.61
egg	121.41
chocolate	175.9
chicken	227.62
salad	105.33
bacon	132.64
watermelon	74.39

the several preliminary experiments have been conducted to determine the best approaches to incorporate these three types of health factors.

- For Calorie, we treat it as an enhanced input of each item for the recommendation system.
- For BMI and Intent, we divide all data into sub-groups depending on the user's starting BMI and periods' intents.
- For Variety, we use greedy search to recalculate final ranking scores of every items, then consist the new recommend basket.

3.1. Calorie

Caloric restriction is the most common method to control weight [**78**]. Weight loss occurs when energy intake is less than energy expenditure [**8, 16**]. Therefore, to successfully build a health recommendation system, it would be beneficial to incorporating calories factor in to our system.

The data we collect form the weight loss applications has rich information of calories, such as item calories, whole day calories, breakfast calories, lunch calories, etc. Nevertheless, the amount of calorie input by the users for each meal is not very precise. In many cases, it is difficult to determine if a food is consumed for lunch or diner. We also observe some variations of calories for the same food item (e.g. sandwich). Therefore, we take a simplified approach: we only consider item calorie and whole day calorie, and use the average calories of a food item input by all the users. Table 3.1 lists an example day's food with its average calories among all dataset.

Table 3.2. User Classification According to Body Mass Index (BMI)

Classification	BMI Category
Underweight	< 18.5
Normal weight	18.5 - 24.9
Overweight	25.0 - 29.9
Obese	$>= 30$

Table 3.3. Top 10 items consumed based on different BMI groups

Top	Underweight	Normal Weight	Overweight	Obese
1	milk	cheese	cheese	cheese
2	cheese	milk	milk	chicken
3	yogurt	coffee	coffee	milk
4	cereal	chicken	chicken	coffee
5	coffee	egg	egg	egg
6	peanut butter	yogurt	salad	salad
7	salad	salad	yogurt	yogurt
8	egg	cereal	cereal	banana
9	chicken	banana	banana	oil
10	candy	oil	oil	cereal

3.2. BMI

Body Mass Index (BMI) is a measure of body fat based on height and weight that applies to adult men and women. BMI can be calculated by the formula 3.2.1.

$$\text{BMI} = \frac{\text{weight}(kg)}{\text{height}(m)^2}, \tag{3.2.1}$$

According to the BMI classification defined by Health Canada[1], users can be divided to 6 groups ('Underweight', 'Normal Weight', 'Overweight', 'Obese class I', 'Obese class II' and 'Obese class III') based on their BMI values. To group users in our dataset, we use BMI values from the users' initial login day. Besides, 'Obese class I', 'Obese class II' and 'Obese class III' are merged into one group called 'Obese'. Table 3.2 shows the final classification of BMI for our dataset.

In order to observe the differences of these BMI groups, we list the top 10 food items which are consumed by users of these groups in Table 3.3. Several diverse eating habits may be seen among these groups, which can be summarized as follows:

(1) The underweight group's top-1 food is 'milk', while the other groups' top-1 food is 'cheese'.

[1] https://www.canada.ca/en/health-canada/services/food-nutrition/healthy-eating/healthy-weights/canadian-guidelines-body-weight-classification-adults/body-mass-index-nomogram.html

(2) In the underweight group, ‘yogurt’ is ranked third, while it is ranked sixth, seventh, and seventh in the normal weight, overweight, and obese groups, respectively.
(3) ‘chicken’ is the second most preferred food among obese users, whereas it ranks at ninth among underweight users.
(4) There are ‘oil’ on the top-10 popular in normal weight overweight and obese groups, however, ‘oil’ is not the top-10 choose for the underweight users.
(5) ‘cereal’ ranked higher in the lower BMI groups (such as Underweight group) than it in the higher BMI groups (like Obese group).

Although the popular food among these BMI groups are similar, the different orders among different groups show the different eating patterns between Underweight, Normal Weight, Overweight and Obese weight groups. Therefore, it is appropriate to make food recommendation in taking into account the user’s BMI. The results of further experiments are discussed in Chapter 6.

3.3. User Intent

Our dataset is collected and sorted from real-world users’ food records. Users choose their long-term intents from the following five options at the beginning of their annual record:
(1) maintaining current weight;
(2) losing 0.5 pound per week;
(3) losing 1 pound per week;
(4) losing 1.5 pound per week;
(5) losing 2 pound per week.

However, users’ temporal weight change curves usually exhibit some fluctuations, no matter their long-term intents are increasing, decreasing or maintaining the weight. To demonstrate this, we randomly select three users with three different long-term intents and plot their weight trends throughout the logged days in Figure 3.1.

We can observe that the red line, which records the weight changes of a user (ID: 450), increases over time but with some fluctuations. Similarly, the blue line (user ID: 28403, whose weight drops) and green line (user ID: 10378, whose weight remains relatively consistent) also show these kind of oscillations.

Because of these fluctuations, the user’s long-term intent may not be able to accurately reflect the user’s weight changes. So the users’ underlying short-term intents become a crucial factor in our task. Dividing underlying short-term intents by week is an appropriate time granularity according to the many nutrition aspect papers [**75, 59, 51**]. Specifically, we segment users into three sub-groups as defined by whether they experience weight increase (i.e., last observed weight is higher than start weight), weight maintenance (i.e., last observed

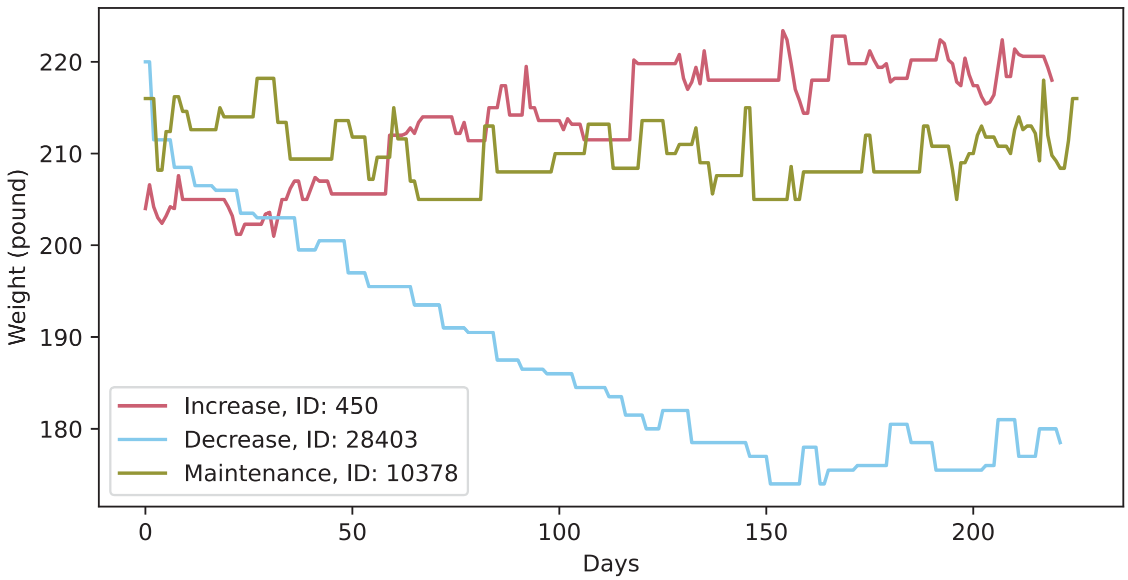

Fig. 3.1. Weight Trends for Random selected Users with Three Different Intents

Table 3.4. Average Calories Among Different Groups

Data	Average Item Calorie	Average Whole Day Calorie
Total	137.20	1758.11
Underweight	89.63	1358.09
Normal Weight	124.12	1703.74
Overweight	143.31	1806.48
Obese	147.34	1768.51
Increase	140.23	1823.15
Decrease	135.41	1722.43
Maintenance	140.27	1743.33

weight is as same as start weight), and weight decrease (i.e., last observed weight is lower than start weight) in each week.

3.4. Calorie Within BMI and Intent Groups

In order to analyze the impact of calorie factor within the BMI and intent groups, the statistic is shown in Table 3.4.

Both the average item calorie and whole day calorie in the underweight group are the lowest ones, which are 89.63 and 1358.09 separately. On the contrary, the obese group's average item calorie shows the highest value (147.34) among all the datasets. In addition, compare these three user intent groups, increase, decrease and maintenance. The decrease group's average item calorie and whole day calorie are the lowest among them, at 135.41 and

1722.43, respectively. And the average whole day calorie of the increase group is 1823.15, which is the greatest value among all the groups.

Thus, we can observe that calorie restriction indeed shows positive impacts on weight loss. Besides, lower weight groups, such as underweight group, typically consume fewer calories than higher weight groups, such as overweight group. The similar phenomenon can also be observed in intent groups, as calorie intakes are higher for increase group than decrease group.

3.5. Combine BMI and Intent

The BMI categories of users, as well as their short-term intents, are crucial for weight management. Therefore , we create sub-groups based on these two criteria. In particularly, we take into account consumers' intentions depending on distinct sub BMI groups.

As a result, we use 'Un_In', 'Un_Ma' and 'Un_De' to denote sub-groups inside the 'underweight' group, which represent 'underweight Users with Increase Intent', 'underweight Users with Maintenance Intent' and 'underweight Users with Decrease Intent', respectively. Similarly, we can also segment the 'Normal Weight', 'overweight' and 'Obese Weight' groups into these three sub-groups. All the sub-groups' statistics are shown in the last 12 rows in Table 4.1.

3.6. Variety

The role of variety in nutrition aspect is debatable. On the one hand, ecological evidence suggests that consuming greater variety in one's diet usually increases consumption in the overall diet, within food groups, and within eating bouts [**55**]. Thus, in those circumstances in which negative energy balance or maintenance of energy balance is desired (i.e., during weight loss and weight maintenance), consuming a diet with higher variety may make these periods more difficult [**47**]. However, some researchers also consider that greater variety is beneficial because occasional indulgence in energy-dense foods might help people make more disciplined dietary choices [**63, 24**].

Therefore, given the potential role of variety in users' eating habits, we would like to explore if variety could have the positive impact of our health recommendation system. Then we construct a concentration index for the share of calories consumed across each of the food items. One commonly used index to measure concentration is the Herfindahl-Hirschman Index (HHI) [**57**], which is used to measure the market concentration at the beginning. Then many nutrition researches adopt it to measure the diet variety and achieve great results [**36, 15**]. HHI can be easily computed as Equation 3.6.1 for each individual and time period:

$$\text{HHI} = \sum_{i=1}^{N} c_i^2, \tag{3.6.1}$$

$$c_i = \frac{Calorie(item\ i)}{Calorie(total)}, \tag{3.6.2}$$

where c_i represents the daily calorie share that food item i contributes to, relative to the total calories consumed during that day; N is the total number of items in a basket. A higher value of this measure indicates that an individual has placed a high concentration of their food calories towards a small set of foods (i.e., low variety). On the contrary, a lower value of HHI expresses higher variety (low concentration) of this day.

3.7. Summary

From the above discussion, we can summarize our task as follows: Our goal is to build a food recommendation system which incorporates item-level health factor (Calorie) and basket-level health factor (Variety) to recommend healthy and diverse food to users. To investigate the impact of incorporating these health factors into the recommendation model, we will conduct experiments on a real-world dataset collected from a weight losing application. Furthermore, we also experiment our models on sub-group datasets which divided according to the user-level health factors (BMI and Intent). The details of the experimental data and our methodology will be presented in Chapter 4 and Chapter 5.

Chapter 4

Dataset and Evaluation Metrics

We begin with a description of Lost-it data in Section 4.1, followed by a full description of data processing procedures in Section 4.2. The evaluation metrics, on the other hand, can be found in Section 4.3.

4.1. Lose-it Data Description

Lose-It [1] is a popular weight management app. From the users, it collects rich information about their daily food consumption ad exercises. In our work, we use a dataset that contains 12 months (January 2016 to December 2016) of food items consumption records on Lost-It mobile application as well as a detailed description of each user: the initial weight, the initial goal, age, gender, etc. The data we use have been anonymized so that the user's identity cannot be recognized (replaced by an ID number). In order to rely on the most reliable data, we identify records from users who record their meals consistently. We apply some criteria to select the users who fulfill all the following requirements:

- The user's age should between 18 and 65;
- The user's weight should change at least 5 times;
- The user should log in their meal more than 200 days (365 days in total);
- The user's total records should more than 2,000.

After the selection, we have 15,408,719 food records belonging to 7,496 users as well as 312 food items. The statistics of Lose-It dataset are summarized in Table 4.1. In addition, the most 80 popular consumed food items list can be found in the Appendix A.1. The food entries in our dataset range from individual ingredients or foods (e.g. sugar, cheese) to foods containing a mixture of ingredients/components (e.g. potato salad, burrito). Furthermore, fruits and vegetables made up a large proportion of consumption frequency, indicating that our dataset is actively attempting to lose weight through diet (and possibly also exercise).

[1] https://www.loseit.com/

Table 4.1. The statistic of Lose-It dataset based on different groups. No/Ov/Ob means the BMI groups: Normal weight/Overweight/Obese, while In/De/Ma means weight outcome/intent: increase/decrease/maintenance.

	User Number	Average BMI	Average Weight	Average Height	Total Records	Average # Records / User
Total User	7,496	27.2	175.9	67.3	15,408,719	2,055.6
Male User	2,927	28.0	198.2	70.5	6,132,061	2,095.0
Female User	4,569	26.6	160.6	65.1	9,276,658	2,030.3
Underweight	39	17.9	110.8	65.9	114,316	2,931.2
Normal Weight	2,286	22.4	141.2	66.5	5,389,511	2,357.6
Overweight	2,576	26.4	173.0	67.8	5,382,564	2,089.5
Obese	2,595	33.3	216.7	67.6	4,522,328	1,742.7
Increase	6,335	26.7	173.6	67,5	3,402,773	537.1
Decrease	7,245	27.7	179.6	67.4	5,625,669	776.5
Maintenance	6,656	27.0	174.0	67.1	6,380,277	958.6
Un_In	35	17.8	106.3	64.7	31,979	913.7
Un_De	36	17.7	107.5	65.4	35,926	997.9
Un_Ma	34	18.1	114.9	66.8	46,411	1,365.0
No_In	2,079	22.5	142.6	66.6	1,350,754	649.7
No_De	2,191	22.4	141.4	66.5	1,675,282	764.6
No_Ma	2,098	22.3	140.2	66.4	2,363,475	1,126.5
Ov_In	2,251	26.5	175.2	68.0	1,214,837	539.7
Ov_De	2,504	26.3	172.5	67.8	1,910,671	763.0
Ov_Ma	2,310	26.4	172.1	67.6	2,257,056	977.1
Ob_In	1,970	33.3	218.7	67.9	805,203	408.7
Ob_De	2,514	33.2	215.5	67.5	2,003,790	797.1
Ob_Ma	2,214	33.4	217.1	67.5	1,713,335	773.9

In order to better design and verify our frameworks, we divide several sub-datasets based on users' start weight group and their intents. More specific descriptions are given in Section 4.2.

4.2. Data Preprocessing

According to previous nutrition and health science researches [**5, 63, 31, 16, 15**], food plays the most important role in maintaining a healthy weight or losing/gaining weight. Furthermore, users in different weight groups usually require different recipes to achieve their objectives. And their underlying intents usually have a significant impact on weight loss, increase and maintenance.

The way we distinguish users' weight groups and underlying intents is according to their BMI and actual weight changes, which are explained in Section 3.2 and Section 3.3, respectively.

4.3. Evaluation Metrics

We adopt a number of common evaluation metrics to evaluate our frameworks, which are widely used in recommendation systems and other ranking tasks.

Assume a set of n objects to rank. Given a user u in the user set U, we use $\hat{R}(u)$ to represent a ranked list of items that a model outputs, and $R(u)$ to represent a ground-truth collection of items that user u has consumed. In our evaluation scenario, following the common practice in recommendation, only the top-ranked K items are taken into account, which means we truncate the recommendation list with a length K.

4.3.1. Hit@K

Hit@K (also known as Hit-Ratio at K) is a method for determining the number of 'hits' in a K-sized list of ranked items. We label a recommended item a 'hit' if it corresponds to one item in the ground-truth set of daily food items.

$$\text{Hit@}K = \frac{1}{|U|} \sum_{u \in U} \frac{|\hat{R}(u) \cap R(u)|}{|R(u)|}, \tag{4.3.1}$$

where $|R(u)|$ denotes the item count of $R(u)$.

4.3.2. Precision@K

Precision (also known as positive predictive value) is a metric for calculating the percentage of relevant items out of all the recommended items. The final result is computed by averaging the metrics for each user u.

$$\text{Precision@}K = \frac{1}{|U|} \sum_{u \in U} \frac{|\hat{R}(u) \cap R(u)|}{|\hat{R}(u)|}, \tag{4.3.2}$$

where $|\hat{R}(u)|$ denotes the item count of $\hat{R}(u)$.

4.3.3. R-Precision

R-precision is defined as the proportion of the top-R retrieved items that are relevant, where R is the number of ground-truth relevant items for the current user u, denotes as $|R(u)|$. Therefore, the calculating formula is similar with Equation 4.3.2, with the different cutoff R.

$$\text{R-Precision} = \frac{1}{|U|} \sum_{u \in U} \frac{|\hat{R}(u) \cap R(u)|}{|\hat{R}(u)|}, \tag{4.3.3}$$

where $|\hat{R}(u)|$ and $|R(u)|$ denote the item count of $\hat{R}(u)$ and $R(u)$, respectively. Therefore, $|\hat{R}(u)| = |R(u)|$.

4.3.4. Recall@K

Recall is a metric that calculates the fraction of corrected recommendation items out of all relevant items.

$$\text{Recall@}K = \frac{1}{|U|} \sum_{u \in U} \frac{|\hat{R}(u) \cap R(u)|}{|R(u)|}, \tag{4.3.4}$$

where $|R(u)|$ denotes the number of items in $R(u)$, and $|\hat{R}(u) \cap R(u)|$ represents the item count of the intersection set of $\hat{R}(u)$ and $R(u)$.

4.3.5. NDCG@K

NDCG (also known as Normalized Discounted Cumulative Gain) not only measures the proportion of correct recommended items, but also it takes the positions of correct recommended items into consideration by assigning higher scores to hits at top ranks.

$$\text{NDCG@}K = \frac{1}{|U|} \sum_{u \in U} \left(\frac{1}{\sum_{i=1}^{\min(|R(u)|,K)} \frac{1}{\log_2(i+1)}} \sum_{i=1}^{K} \delta(i \in R(u)) \frac{1}{\log_2 i + 1} \right), \tag{4.3.5}$$

where $\delta(\cdot)$ is an indicator function, which $\delta(x) = 1$ if x is true and 0 otherwise.

Basically, the NDCG will be high if the ground-truth items are ranked high in the recommended list. The lower the rank for ground-truth items, the more their gains are discounted (in the logarithm form of its rank).

4.3.6. MRR@K

MRR (also known as Mean Reciprocal Rank) calculates the reciprocal rank of the first relevant item retrieved by an algorithm. It is also known as the Average Reciprocal Hit Ratio (ARHR).

$$\text{MRR@}K = \frac{1}{|U|} \sum_{u \in U} \frac{1}{rank_u^*}, \tag{4.3.6}$$

$rank_u^*$ is the rank position of the first relevant item retrieved by an algorithm for a user u.

Chapter 5

Healthy Food Recommendation System

In this chapter, we will describe the approaches we explore. We will start by some preliminary exploration to determine the basic recommendation approach. Then a model combination approach will be proposed to combine a general recommendation model with a goal-oriented recommendation model for a user to achieve the goal. The chapter is organized as follows:

(1) We will describe the two basic modeling approaches for our task based on the entire and sub datasets described in Section 4. The first approach is the next-item recommendation, as discussed in Section 5.1, in which a sequence of previously consumed items are fed into the model, and the model recommends new items one by one. Another method is the next-basket recommendation, described in Section 5.2, takes the same sequence of previously consumed items, but instead of recommending new items one by one, the model outputs a basket of candidate items at a time, which is more realistic to many real-life scenarios.

(2) We will then describe methods to encode food items (through their names) and calories amounts. The sequence of encoded information will be used as the input to a recommendation model, which is then asked to recommend the next food item or next basket of foods.

(3) A specific base recommendation model, called HHFR, will be described.

(4) The reason and approaches of our proposed model combination approach will be described. Firstly, the "*following good examples*" philosophy of our food recommendation will be explained. Then we will show how different BMI- and Int- models will be trained, and how different models work in combination to make food recommendations.

5.1. Next-item Recommendation

The sequence of users' consumed food items usually has a significant impact on their future meal choices. There are certain predictable patterns in the sequence of users' food selection. For example, if a user chooses to consume a lot of dairy in the previous days, it is possible that he or she has a strong dairy consumption habit and will probably consume

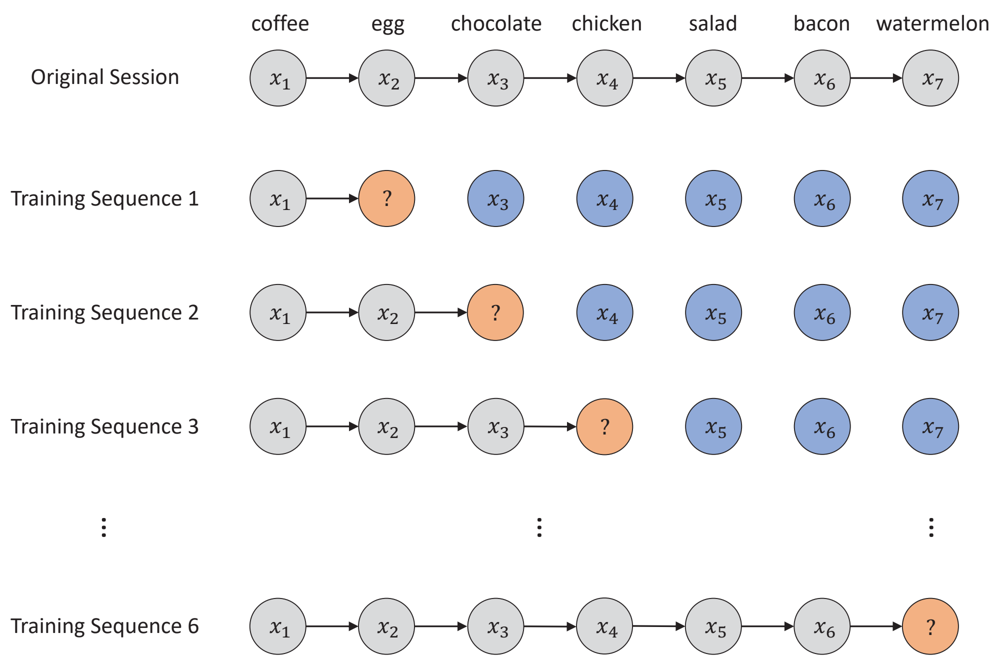

Fig. 5.1. An example of the training process of next-item recommendation modeling method

dairy again in the near future. Therefore, it is reasonable to employ a sequential prediction model to predict the user's future consumption. Among those sequential recommendation approaches, a popular one is next-item recommendation, which aims to predict a user's next item choice based on the sequential interactions in the past [**65, 66, 25, 26, 76**].

Following the proposed sequence preprocessing method, which is introduced in [**13**]. We treat each entire day's food items for every user as a new training session. Therefore, there are many new training sessions for each user, depending on how many days the user has logged in. Given an input training session $[x_1, x_2, \cdots, x_n]$, we generate the sequences and corresponding labels $([x_1], V(x_2)), ([x_1, x_2], V(x_3)), \cdots, ([x_1, x_2, \cdots, x_{n-1}], V(x_n))$ for training. Figure 5.1 shows how we generate training sequences for an original session: [***coffee, egg, chocolate, chicken, salad, bacon, watermelon***]. In this example, there are in total 7 items in the session, so we can generate 6 training sequences from it.

5.1.1. Preliminary Experiments

The first question we are faced with is what basic recommendation method is the best for the food recommendation application. To this end, we conduct several preliminary experiments of previous models on Lose-it full data, which are described in Chapter 4. The Results are shown in Table 5.1, which RNN based GRU4Rec model [**65**] shows the superior performance among all the other models in full data. Thus, our subsequent experiments will use this model as the basic recommendation model.

Table 5.1. Preliminary Test of Next-item Recommendation based on Lose-it full data

Groups	Models	Recall@10	MRR@10	NDCG@10
Markov Chain	FPMC	0.4014	0.1784	0.2332
	TransRec	0.4010	0.1609	0.2169
RNN	GRU4Rec	**0.4291**	**0.1945**	**0.2493**
Attention	STAMP	0.4258	0.1869	0.2426
	SASRec	0.4141	0.1838	0.2375
	BERT4Rec	0.4033	0.1739	0.2274

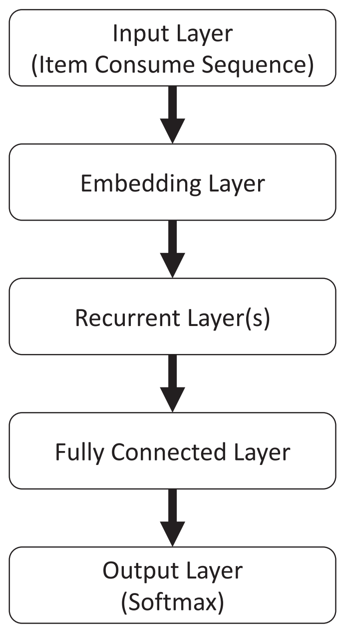

Fig. 5.2. Generic structure of the network used in our Next-item Recommendation.

5.1.2. Model Architecture

GRU4Rec [**65**] model is chosen as the base next-item recommendation model in our future experiments and it has been described in Section 2.3.3. GRU4Rec follows the generic structure of the RNN model shown in Figure 5.2. The Gated Recurrent Unit (GRU) [**10**] is used as the recurrent layer. The 300-dimensional GloVe Pre-trained [**50**] item embeddings are adopting in our model.

As for encoding the enhanced input health factor (Calorie) of each food items, we adopt the traditional Look-up embedding, which will be described in the Section 5.2.1.2.

Then it is trained using standard mini-batch gradient descent on the cross-entropy loss via Backpropagation Through Time (BPTT) [**44**] for a fixed number of time steps.

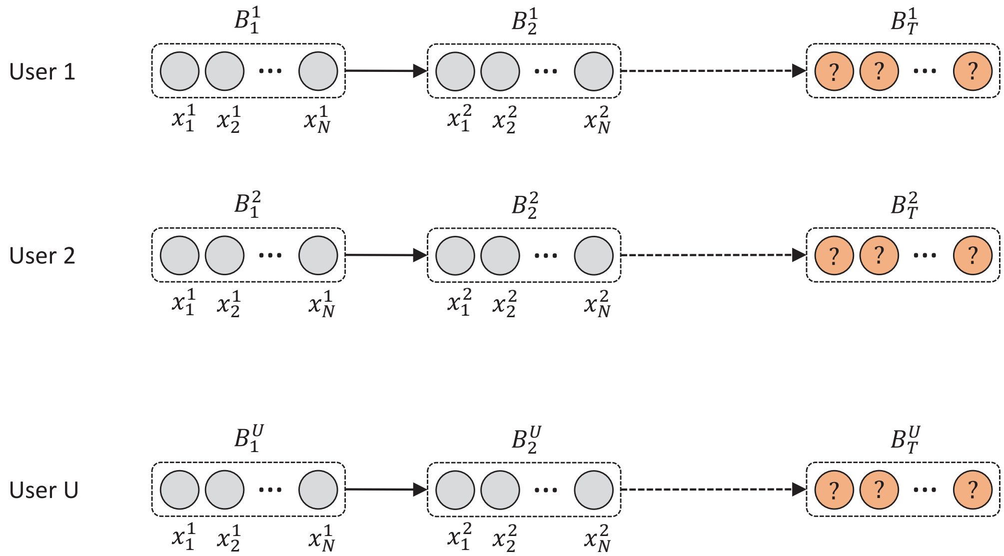

Fig. 5.3. The training process of next-basket recommendation modeling method

5.2. Next-basket Recommendation

Next-basket modeling approach is an alternative choice. Instead of recommending a single item at a time, Next-basket recommending approach recommends a set of items, or a basket, to the user based on his or her historical baskets [**6, 56, 77, 79, 80**]. In our task, we segment the items logged in a day as a basket for each user. Since we are recommending a whole day's foods to the user, the relative positions of each item in the basket is not very important, which means [***coffee, egg, chocolate***] and [***chocolate, egg, coffee***] are considered as the same basket.

The training procedure is illustrated in Figure 5.3. Given a sequential basket records $B^u = \{B_1^u, B_2^u, \cdots, B_T^u\}$ of user u, we predict the last basket B_T^u based on the previous basket sequence $\{B_1^u, B_2^u, \cdots, B_{T-1}^u\}$.

5.2.1. Preliminary Experiments

As for the next-item recommendation, we also conduct several preliminary experiments based on Lose-it full data to compare GloVe and Bert embeddings (Section 5.2.1.1), the different encoding ways (Section 5.2.1.2) and incorporate health factors (Section 5.2.1.3). Those preliminary results are described in Section 5.2.1.4.

5.2.1.1. Item Embedding.

For item embedding, we compare the commonly used GloVe and BERT word vectors, which are represented in Section 2.1.1 and Section 2.1.2, respectively. GloVe word vectors have

Table 5.2. Statistic of GloVe and Bert word embedding

	Average	**Minimum**	**Maximum**	**Abs Average**
GloVe 100d	-0.0010	-2.2466	2.4304	0.4004
GloVe 300d	-0.0069	-2.1935	1.8505	0.2959
Bert 768d	-0.0048	-1.0000	1.0000	0.4011

Table 5.3. Statistic of Healthy Factor: Calorie.

	Average	**Minimum**	**Maximum**	**Median**	**Standard**
Calorie	137.2	0.0	1000.0	100.0	134.0

100 and 300 dimensions, while Bert word vectors have 768 dimensions. There are 100-dimension and 300-dimension for GloVe word vectors, and 768-dimension for Bert word vector. Table 5.2 shows the statistics for Glove and Bert embeddings.

From Table 5.2 we can observe that the average values and extrema of the GloVe and Bert embeddings are 10^{-1} and 10^{-3}, respectively.

5.2.1.2. Health Factor Embedding.

For each item, our health factor is a real value and the order of magnitude is usually not matched with the GloVe and Bert embedding (the order of magnitude of GloVe and Bert embeddings' average values are are 10^{-3}, while it is 10^{2} for the value of calorie). Therefore, we cannot directly input it into the model. The statistics of health factors in our data are summarized in Table 5.3.

Therefore, there are two ways to generate health factor embeddings in our work, traditional Look-up Embedding and Multi-dimensional Adaptation Transformation:

(1) **Look-up Embedding:** It is the one of the most often used strategies for generating health factor's embeddings to a neural model. It means we randomly initialize a trainable embedding matrix for health factor, such as Calorie. Then we round-up each health factor's value to the closest integer as its index of embedding matrix. Therefore, the size of calorie embedding matrix is $[1001, \textit{embedding size}]$, and can be trained throughout the training process.

(2) **Multi-dimensional Adaptation Transformation (MAT):** Another option to incorporate the health factor into our model is to build a multi-dimensional encoding. Inspired by the Transformer [**74**], which employs sinusoidal vectors to encode the position of each term, we could also construct transformations which maps the health factor into an encoding vector. Here we follow the Transformers to map the scalar health factor into a multi-dimensional vector in order to encode more discriminative information to help the model to learn the impact of different health factor values.

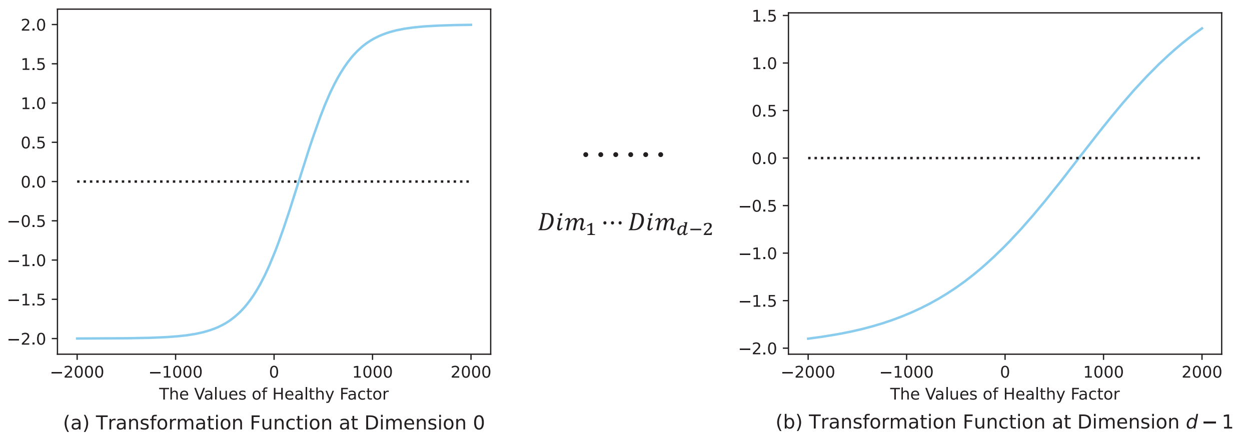

Fig. 5.4. Illustration of the Adaptation Transformation Functions at Each Dimension

We call this mapping the Multi-dimensional Adaptation Transformation. The desired transformation should have the following design criteria:

(a) **Monotonously increasing:** As stated before, the health factors are real-valued scalars such as Calories, which carries physical meanings. Therefore, given two health factor values x_1, x_2 with $x_1 < x_2$, the desired transformation should map a higher-value health factor to a vector in which each dimension is greater than that of the smaller-value health factor. i.e. $f(x_1)_{(i)} < f(x_2)_{(i)}, \forall i = 1 \cdots n$, where $f(x_1)_{(i)}$ indicates the i^{th} dimension of the mapped vector for health factor x_1.

(b) **Non-periodicity:** Moreover the transformation need to be strictly monotonous, i.e. given $x_1 \neq x_2$, each dimension of the mapped vector should have different values $f(x_1)_{(i)} \neq f(x_2)_{(i)}, \forall i = 1 \cdots n$. Since we do not want to map two health factors with the different values to vectors having same values at certain dimensions, which will cause confusions to the model.

Taking into considerations of the above desired properties, we choose hyperbolic tangent (tanh) with some translation and stretching as our transformation function. Therefore, for a float health factor value, the transformation function generates $v = [v_0, v_1, \cdots, v_{d-1}]$ as its encoding. The calculation of the value in ith dimension v_i is presented in Equation 5.2.1.

$$v_i = 2\tanh\left(\frac{x}{500 + \frac{1000}{d-1}i} - 0.5\right), 0 \leq i < d, \tag{5.2.1}$$

where d is the dimension of the mapped vector, x represents the value of health factor (i.e., calorie) that $x \in [0,1000]$.

Therefore, for every dimensions, the different degrees of translation and stretching are dynamically employed. For example, when the dimension of vector d is set as 300,

Table 5.4. Description of Next-basket Preliminary Models

Model Name	Item Emb	HF Emb	Dim_{item}	Dim_{HF}	Incorporation Type
Glove_100d_cat	GloVe	MAT	100	100	Concatenation
Glove_100d_plus	GloVe	MAT	100	100	Plus
Glove_300d_cat	GloVe	MAT	300	300	Concatenation
Glove_300d_plus	GloVe	MAT	300	300	Plus
Glove_300d_train	GloVe	Look-up	300	300	Concatenation
BERT_768d_cat	BERT	MAT	768	768	Concatenation
BERT_768d_plus	BERT	MAT	768	768	Plus

$\frac{1000}{d-1} = 3.345$, then the denominators in Equation 5.2.1 are [500, 503.345, 506.690, $\cdots$, 1500].

To better demonstrate the proposed transformation, we visualize the transformation functions for dimension 0 and $d-1$ and plot them in Figure 5.4, and the curves of intermediate dimensions (dim 1 to dim $d-2$) are omitted. We can observe that the tanh function could transfer different health factor values in to different scalars for each dimension of the vector while preserving the trends. Besides, the order of magnitude and value domain of transformed scalars are similar as GloVe and BERT embeddings that is shown in Table 5.2.

5.2.1.3. Incorporation of Health Factor.

There are two most frequent methods to incorporate the item embedding with health factor embedding, which are concatenation and sum by each dimension.

(1) **Concatenation:** The straightforward approach of incorporating health factor into item embedding is concatenation operation, which is also a stacking operation. Figure 5.5 shows the concatenation of item and health factor representation as input of our model. Thus, the dimension of vector after concatenation would become item embedding size + health factor embedding size.

(2) **Sum:** According to the 'positional encoding' in transformer model [**74**]. Each health factor embedding is generated with the same dimension as the item embedding dimension, so that these two can be summed by each dimension. As a result, the dimension of the vector after plus is the same as the size of the item embedding and the size of the health factor embedding.

5.2.1.4. Results of Preliminary Experiments.

The preliminary experiments are conducted on different Item Embedding (Section 5.2.1.1), Health Factor Embedding (Section 5.2.1.2) and Incorporation method (Section 5.2.1.3). A summary of these preliminary models is presented in Table 5.4. The performances of the above models are compared in Table 5.5.

Table 5.5. Preliminary of next-basket recommendation based on Lose-it full data. P@K and R@K represent precision@K and recall@K, respectively. The results of best model is boldfaced.

Model Name	P@1	P@5	P@10	R-Prec	R@5	R@10	NDCG@5	NDCG@10
Glove_100d_cat	0.4371	0.3257	0.2633	0.2775	0.1923	0.2990	0.3196	0.2592
Glove_100d_plus	0.4367	0.3251	0.2628	0.2764	0.1920	0.2989	0.3199	0.2590
Glove_300d_cat	**0.4420**	**0.3261**	**0.2654**	**0.2784**	**0.1926**	**0.3023**	**0.3219**	**0.2616**
Glove_300d_plus	0.4408	0.3256	0.2641	0.2773	0.1926	0.3002	0.3204	0.2602
Glove_300d_train	0.4387	0.3252	0.2627	0.2761	0.1919	0.2991	0.3210	0.2594
BERT_768d_cat	0.4357	0.3242	0.2638	0.2772	0.1916	0.3003	0.3181	0.2595
BERT_768d_plus	0.4351	0.3250	0.2611	0.2733	0.1916	0.2974	0.3186	0.2567

First of all, we can observe that the BERT embedding employed for our dataset does not show superior performance to the 300-dimension GloVe embedding. One possible reason could be that BERT is better at capturing the semantics of long text since it is a pre-trained language model, whereas the item names in our dataset are usually constituted of one to three words, which may not exploit the capacity of BERT to aggregate context information. Therefore, the simpler word embeddings such as GloVe can work equallt well.

Second, we compare the results of two health factor embedding approaches, Look-up Embedding and MAT, which are given Section 5.2.1.2. We can see the '**Glove_300d_cat**' model outperforms the '**Glove_300d_train**' model, implying that Multi-dimensional Adaptation Transformation outperforms Look-up Embedding for health factors in our dataset.

Then in terms of incorporation type, we discovered that concatenation of item and health factor embeddings is superior to dimension-wise addition regardless of the item and health factore embedding approaches.

Furthermore, among all the preliminary models, the '**Glove_300d_cat**' model achieves the best experimental results. As a result, the following further tests are based on the '**Glove_300d_cat**' model.

5.2.2. Model Architecture of Basic Next-Basket Recommendation

In this chapter, we propose a **H**ealth-factor-aware **H**ierarchical **F**ood **R**ecommender (HHFR) architecture to incorporate health-related information, which is illustrated in Figure 5.5. This architecture is an adaptation of those proposed for next-basket recommendation to our problem. HHFR utilizes a hierarchical RNN to model the user's sequential behavior over time, and incorporates health information (calorie) of each item into a basket.

In the following, we first model the information of a basket: encoding information of items and health factors (calorie) in each basket; and learning the intra-basket representation. Then we model user's sequential inter-basket behavior by the basket-level RNN.

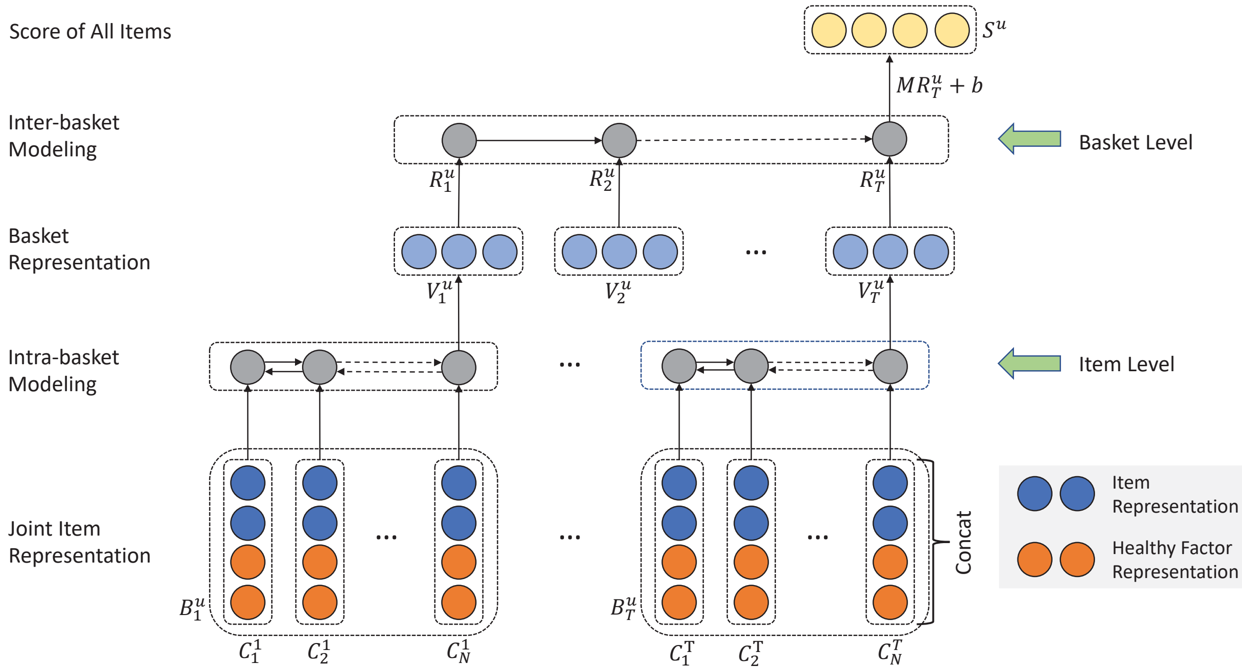

Fig. 5.5. Next-basket Recommendation Model Architecture

5.2.2.1. Item and Health Factor Encoding.
The input instances of HHFR model are the representations of a sequence of baskets $\{B_1, B_2, \cdots B_T\}$ where T denotes the total number of basket. Every basket representation is of the form $B_t = \{C_j^t | j = 1, 2, \cdots, N\}$, where N is the number of items in basket B_t. Each C_j^t is a concatenated representation of a food item and its amount of calories.

For item representation, our model adpot the commonly used pre-trained GloVe word vectors [**50**] to map each food item I to a d_a-dimension embedding $E^I = \{e_1^I, \cdots, e_{d_a}^I\}$.

We design **M**ulti-dimensional **A**daptation **T**ransformation (MAT) method for mapping each float health factor value to a d_b-dimension embedding $E^H = \{e_1^H, \cdots, e_{d_b}^H\}$ as health factor embedding. The detail of MAT method is represented in Section 5.2.1.2.

Then the joint item representation is created by concatenating the item and health factor representations, i.e. for j-th item:

$$C_j = \text{CONCAT}(E_j^I, E_j^H), \tag{5.2.2}$$

where $C_j \in \mathbb{R}^{|d_a + d_b| \times 1}$. The representation of the t-th basket B_t with N item is $B_t \in \mathbb{R}^{|d_a + d_b| \times N}$.

5.2.2.2. Intra-basket Representation Modeling.
Given a sequence of items in a basket $B_t^u = \{C_j^t | j = 1, 2, \cdots, N\}$ of user u, we first apply an item-level bi-directional Long Short-Term Memories (BiLSTM) [**27**] to capture the interactions between the items within the basket. Then the resulting representation of the last item in the basket is used as the representation V_t^u of the whole basket B_t^u.

5.2.2.3. Inter-basket Sequential Modeling.
In order to model a user's sequential inter-basket behavior, another basket-level LSTM is applied to the basket representations. Given a sequence of baskets $\{V_1^u, V_2^u, \cdots, V_T^u\}$ for user u, this LSTM will generate another representation for each basket. We take the representation of the last basket as the representation R_T^u for the whole sequence of baskets of user u.

The output scores S^u can be calculated through multiplication of item matrix M and the whole basket sequence representation R_T^u, which is formulated as follows:

$$S^u = MR_T^u + b. \tag{5.2.3}$$

We have $S_i^u \in \mathbb{R}^{|I|\times 1}$, i.e., an element of S_i^u represents the interaction score between an item i and a user u. A higher score indicates that the user is more likely to consume the corresponding item.

5.2.2.4. The Loss Function for Optimization.
For a user u and his/her previous baskets $B_{1,t}^u$, we define the probability of an item i being taken in the next basket B_{t+1}^u by sigmoid function:

$$p(i \in B_{t+1}^u | u, B_{1,t}^u) = \frac{1}{1 + e^{-S_i^u}}, \tag{5.2.4}$$

where S_i^u represents the interaction score between user u and item i in Section 5.2.2.3.

To effectively learn from the training data, we adopt a weighted cross-entropy as the optimization objective at each step of LSTM, which is defined as:

$$L = \sum_{u \in U} \sum_{B_t^u \in B^u} \sum_{I \in I_t} (-m \cdot y_i \log p_i - n \cdot (1 - y_i) \log(1 - p_i)), \tag{5.2.5}$$

where p_i is the probability of an item i being consumed in the next basket in our model. If item i is consumed in the the next basket, $y_i = 1$, otherwise, $y_i = 0$. m and n are the weights of positive and negative instances (consumed or not in the next basket). The reason of using different weights is to cope with the fact that there are usually much more negative instances than positive instances in our dataset. After training, given a user's historical records, we can obtain the probability of each item i being taken in the next basket according to Equation 5.2.4. We then rank the items according to their probability, and select top K results as the final recommended items to the user.

5.3. Model Combination

The data we collected suggest that the food choices made by users are not always consistent with their global intent. Imposing food selections according to the global intent would make the food choices very different from users' eating habits, thereby jeopardizing their adoption by the users. On the other hand, the real weight changes during a period of time may better reflect the true impact of food choices during the period, i.e. the food choices truly led to the weight outcome. Therefore, we can consider such a period as a good example that accomplishes an implicit goal corresponding to the weight outcome. Yet the foods chosen by the user during the period can be considered as an acceptable food pattern by the user. An intent model trained on such examples may be a smoother adaptation of the user's normal eating pattern toward the implicit goal, than the traditional food recommendation approaches based on dietary guidelines [**1**]. Based on the above observation, we propose an approach to model intents based on behavior data of the periods (i.e. good examples) that produced the intended weight change. The philosophy of our approach to food recommendation can be stated as "*following good examples*".

Therefore, we train different models to capture the eating patterns of different groups of users, including:

- a general/full model for all the users (**General/Full-model**): the model is trained with all the training data mixed together, which can capture general eat patterns of all the users in all period.
- models for each BMI group (**BMI/Middle-models**): A BMI model is trained with data from each BMI group ('Underweight', 'Normal Weight', 'Overweight' or 'Obese').
- models for each intent (**Int**): These models are trained on data from periods corresponding to the intent ('Increase', 'Decrease', or, 'Maintenance'). We expect these models capture some common eating patterns of users with the same intent.
- models for different intent/outcome groups within each BMI group (**BMI+Int/Sub-models**): Each model is trained with the specific intent sub-group (e.g. No_In) within a BMI group.

The middle and sub models are trained on the "*good example*" periods, the combination of these models with full-model could enhance guide model smoothly toward the goal. Furthermore, combine fuse full-model and sub-models by recalculating the ranking score of each item according to the scores of the corresponding models, and testing it on the sub-data.

There are two ensemble methods are adopted in our experiments, which are two models ensemble (Section 5.3.1) and three models ensemble (Section 5.3.2). In the next chapter, we will test different combination of models to test its effectiveness on our data.

5.3.1. Two Models Ensemble

We introduce a parameter α to control the proportion of full-model and sub-model. The recalculation equation is given in Equation 5.3.1:

$$S_{\text{new}} = (1-\alpha)\cdot S_{\text{full}} + \alpha \cdot S_{\text{sub}}, \quad \alpha \in [0,1], \tag{5.3.1}$$

where α is the sub-model ratio, when α equals 0, it means we apply the completely full model score as the new score. Similarly, it means we only utilize the sub-model to predict when α equals 1. Besides, we also combine the full-model and sub-models by taking α values in the range of [0,1] with a 0.1 increment.

5.3.2. Three Models Ensemble

In this scenario, the proportions of full-model, middle-model, and sub-model are controlled by two parameters: α and β. The re-calculation details are given by Equation 5.3.2:

$$\begin{aligned} S_{\text{new}} =& (1-\alpha-\beta)\cdot S_{\text{full}} + \alpha \cdot S_{\text{middle}} + \beta \cdot S_{\text{sub}}, \\ &\alpha, \beta \in [0,1], \quad \alpha+\beta \leq 1, \end{aligned} \tag{5.3.2}$$

where α and β are two ratios that represent the proportions of the middle-model and sub-model, respectively. Besides, the value ranges of α and β are identical to the parameter α in Equation 5.3.1, which is in the range of [0,1] with increment of 0.1. When $\alpha = \beta = 0$, which means we only use full-model to predict; When $\alpha = 1$ and $\beta = 0$, it means new score only based on the middle-model. Similarly, the sub-model score is considered as the final new score when $\alpha = 0$ and $\beta = 1$.

Chapter 6

Experimental Results

Extensive fine-grained experiments have been conducted to validate the models' performances and the impacts of the health-factors, which are represented in Chapter 3. In this Chapter, we first introduce two model ensemble methods in Section 5.3; then the experimental results of two modeling approaches, Next-item Recommendation and Next-basket Recommendation are described in Section 6.2 and Section 6.3, respectively. For Next-item Recommendation, the present sequences are Intent and sub BMI groups. Whereas, we discuss the results for Next-basket Recommendation by the effects of BMI, Intent and Variety.

6.1. Experimental Settings

In the evaluation process, the week that achieves the user's goal (e.g. losing weight) is treated as the ground truth. The proposed recommendation system is asked to recommend the foods for each day of the week. If the performance of our model is more successful than the general model, we consider it more effective than the latter. For example, there is an obese user consumes many "pizza, hamburger, French fries" during his/her weight increase period. Whereas, he/her consumes more "vegetable, fruit" in his/her losing weight period. The general model is treated as the baseline model, which is trained based on the whole mixed weight change periods. The expected model is to capture the eating patterns of the period with the given goal rather than the other or whole periods.

Following the previous work [**6**], we take the last basket of each user as the testing data, the penultimate basket as the validation set to optimize parameters, and the remaining baskets as the training data. Therefore, the statistic of these three sets among all the subdatasets is shown in Table 6.1.

Table 6.1. Statistics of the dataset. No/Ov/Ob means the BMI groups: Normal weight/Overweight/Obese, while In/De/Ma means weight outcome/intent: increase/decrease/maintenance.

			# Records		
Datasets	**# User**	**# Baskets**	**Train**	**Validation**	**Test**
General	7,496	1,300,177	15,257,122	77,462	74,135
Underweight	39	7,926	113,329	512	475
Normal Weight	2,286	424,814	5,340,033	25,441	24,037
Overweight	2,576	459,566	5,330,912	26,542	25,110
Obese	2,595	407,871	4,472,848	24,967	24,513
Increase	6,335	281,751	3,261,940	70,376	70,457
Decrease	7,245	465,270	5,464,203	81,800	79,666
Maintenance	6,656	553,156	6,242,724	69,235	68,318
Un_In	35	1,897	30,934	519	526
Un_De	36	2,147	34,936	523	467
Un_Ma	34	3,882	45,565	418	428
No_In	2,079	105,271	1,301,246	24,823	24,685
No_De	2,191	129,999	1,622,665	26,719	25,898
No_Ma	2,098	189,544	2,316,590	23,745	23,140
Ov_In	2,251	102,778	1,165,460	24,603	24,774
Ov_De	2,504	158,977	1,855,092	28,178	27,401
Ov_Ma	2,310	197,811	2,209,588	23,996	23,472
Ob_In	1,970	71,805	764,300	20,431	20,472
Ob_De	2,514	174,147	1,951,510	26,380	25,900
Ob_Ma	2,214	161,919	1,670,981	21,076	21,278

6.2. Next-item Recommendation

The first modeling approach for our task is the next-item recommendation, which is described in Section 5.1. In this section, we will introduce and analyze the results based on different sub-groups relating to intents and BMI.

6.2.1. Intent Groups

We first test the combination of a model trained for each intent group and the general model. There are three intent groups in our experiments, which are segmented by user's underlying short-term intents (Section 3.3). Therefore, we conduct experiments on ensembling two models based on 'Increase', 'Maintenance', and 'Decrease' intents groups. The results is shown in Figure 6.1.

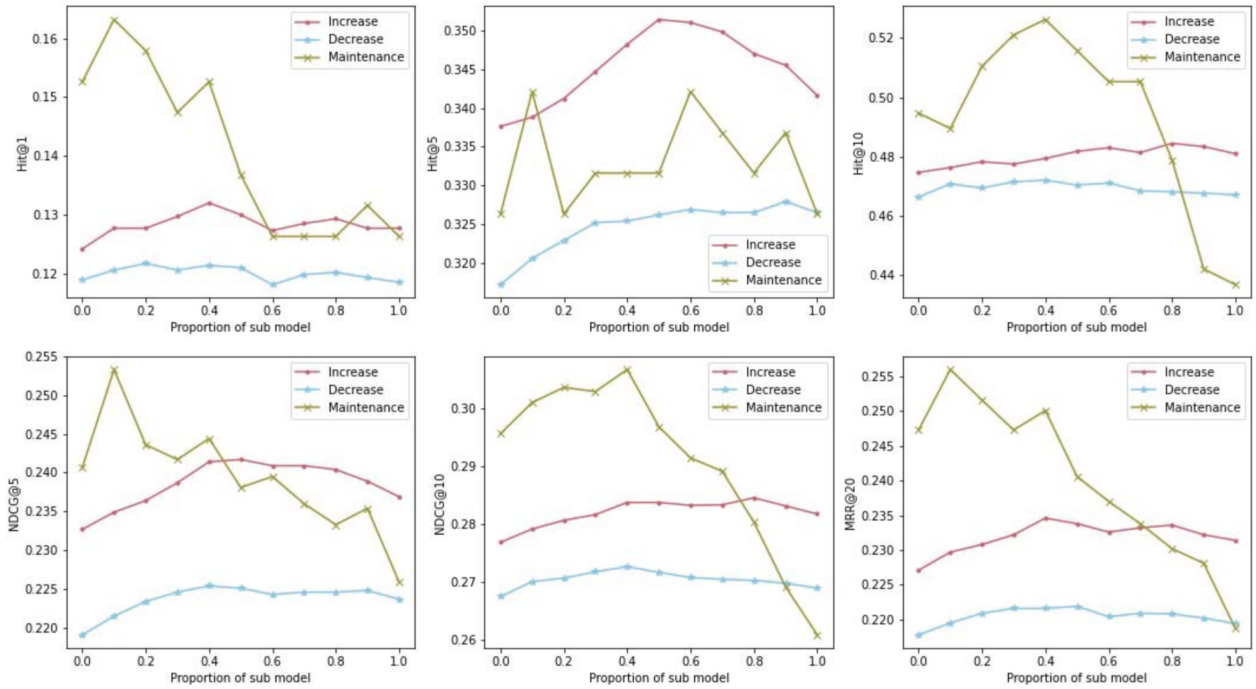

Fig. 6.1. Ensemble Two Models for Next-item Recommendation Based on Intent Groups by Varying α.

(1) **Increase Group:** It contains the weeks which the end weight is higher than the start weight of every user. We visualize the result in Figure 6.1.

We can observe that the performance curve of our ensemble model is convex, which means total full-model ($\alpha = 0$) or total sub-model ($\alpha = 1.0$) does not show the better results than the fusion model. Besides, the full-model's performance is not superior than sub-model's performance, even the training data amount of full-model is far more than it of sub-model. It indicates that intent health factor is an important factor we could consider, and it shows a positive impact to our food recommendation system.

In addition, We can see that the best performance occurs when $\alpha = 0.4$, which means the proportions of full-model and sub-model are 0.6 and 0.4, respectively.

(2) **Decrease Groups:** Decrease group is opposite of the increase group, and is constituted by the weeks which the end weight is lower than the start weight. The ensemble experiment results can be found in the Figure 6.1.

The results are shown the same pattern with the results of the increase group. For example, the best ensemble performance is also shown when $\alpha = 0.4$, which is as same as the best α in the increase group. Furthermore, total sub-model also has a superior result than the total full-model.

(3) **Maintenance Group:** As for the maintenance group, it formed by the weeks which the end weight is as same as the start weight of each user.

We can discover the same tendency with the increase group in, while the best performance occurs when $\alpha = 0.1$ (sub-model proportion = 0.1 and full-model proportion = 0.9). Moreover, the model performances show the fluctuation when ensemble specific sub-model based on the maintenance group. Specifically, the specific sub-model, which based on the maintenance group, cannot better capture users' preferences and trend features than the general full-model. It means the result of total full-model ($\alpha = 0$) is higher than it of the total sub-model ($\alpha = 1$) in the Figure 6.1. However, it also shows that the ensemble model is better than the individual models.

According to the experiments which are conducted on different intent groups, the ensemble models' performances are greater than the individual full- and sub-models. Besides, the total full-models are not better than total sub-models based on increase and decrease groups, which indicate the user's underlying short-term intent is a positive health factor for weight management. Furthermore, the way we consider the intent factor (segment data based on the week underlying intents) is successful. Therefore, ensemble models based on full-model and different intent groups sub-models can better capture users' eating patterns and improve the performance.

6.2.2. Underweight Group

User's start weight is also an important health factor. In this section, model ensemble experiments are conducted based on the underweight group, which user's start BMI is less than 18.5. We test the combination of the Underweight model (the model trained on all the data of underweight users) and the general model. The result is visualized in Figure 6.2.

The data amount of underweight group is way more less than other groups (around 39 vs 2500 users). Thus, the performances are not stable on the underweight group. However, the similar trends can also be observed from underweight groups. The best result appears when $\alpha = 0.8$ (full-model proportion = 0.2 and sub-model proportion = 0.8).

In order to further analyze the effect of users' intent within this underweight group, we further conduct fine-grained experiments on weeks with intents of increasing, decreasing and maintaining weights in this underweight group, and the results are presented as follows:

(1) **Results on underweight Group with Weight Increase Intent:**
We can observe that when $\alpha = 0.2$, the ensemble model has the best result based on 'Un_In' data. And the total full-model shows the superiors performance than total sub-model, which means it is hard to capture the specific users eating habits based on a small data set for sub-model. While when we incorporate a small portion of sub-model (0.2), it achieves the best performance, which indicate that considering the intent factor inside the underweight group is helpful.

(2) **Results on underweight Group with Weight Decrease Intent:**

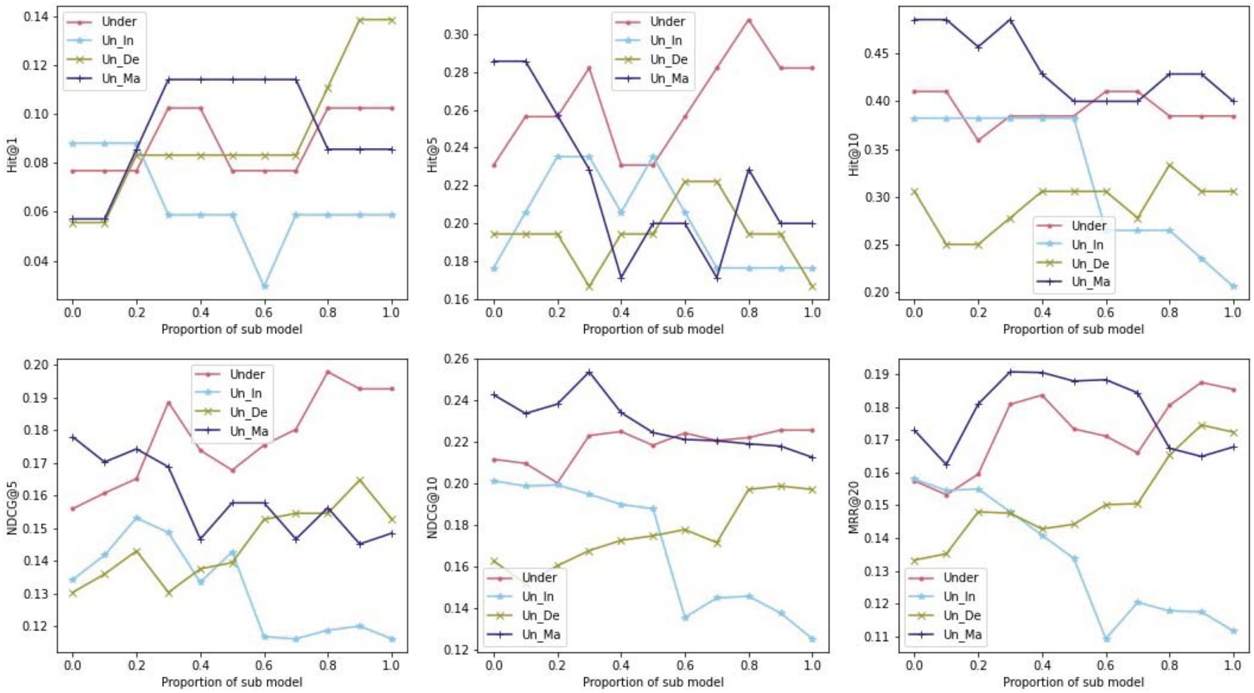

Fig. 6.2. Ensemble Two Models for Next-item Recommendation Based on underweight Groups

As for the weight decrease intent group, which is presented by the green lines in Figure 6.2, total sub-model shows a better result than total full-model. Besides, when sub-model proportion is 0.8 and full-model proportion is 0.2 ($\alpha = 0.8$), it achieves the best performance.

(3) **Results on underweight Group with Weight Maintenance Intent:**
In Figure 6.2, the best result based on maintenance groups occurs when $\alpha = 0.3$, which the sub-model portion is lower than the best α in decrease intent group. It could be caused by the less amount data (59 average records per user VS 3,424 records per user). Furthermore, the performance of $\alpha = 0$ (total full-model) is worse than it of $\alpha = 1$ (total sub-model).

Although the data amount of underweight group is not as rich as other groups, the results are affected. We can also figure out that consider underweight groups separately could help us to better capture these specific users' eating habits, with the total sub-model has a superior result than the total full-model. Furthermore, model ensemble is necessary on the all underweight groups, as the best performance is always shown on the fusion models with different ensemble proportion among these groups.

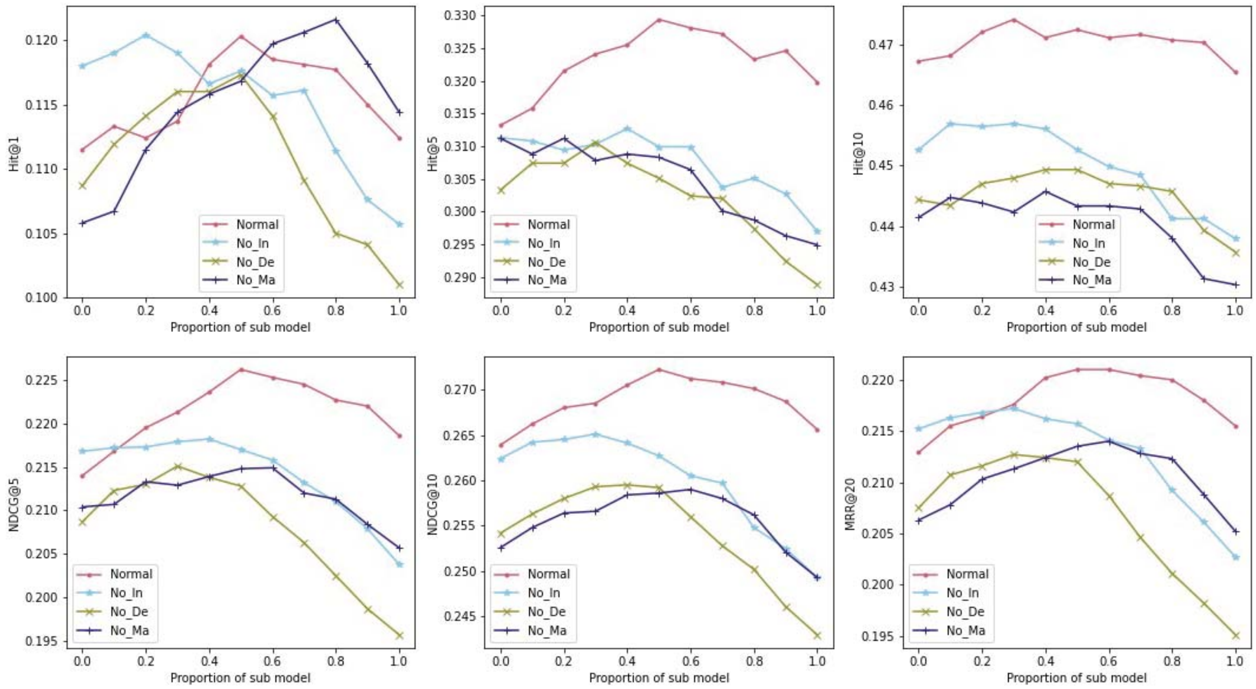

Fig. 6.3. Ensemble Two Models for Next-item Recommendation Based on Normal Weight Groups

6.2.3. Normal Weight Groups

The normal weight group is formed by the users whose start BMI is from 18.5 through 24.9, so there are total 2,286 users in this group (See Table 4.1). The similar ensemble experiments are also conducted based on 'No', 'No_in', 'No_de', and 'No_ma' groups. The result figure can be found in Figure 6.3.

The result of normal weight groups seems more stable than the underweight groups in Section 6.2.2. Besides, the red lines in Figure 6.3 show that the best α of underweight group is 0.5, which means half full-model and half sub-model combination.

The more detailed experiment results of users' intents within this normal weight group are described in the following:

(1) **Results on Normal Weight Group with Weight Increase Intent:**
The light blue lines in Figure 6.3 denote the increase intent group of normal weight group. When $\alpha = 0.3$, the ensemble model has the best results.

(2) **Results on Normal Weight Group with Weight Decrease Intent:**
As for the decrease intent group, we can observe the same trend with increase group, and the best performance also appears when $\alpha = 0.3$. It means a small portion of sub-model can already improve the model's ability and achieve a reasonable performance.

(3) **Results on Normal Weight Group with Weight Maintenance Intent:**

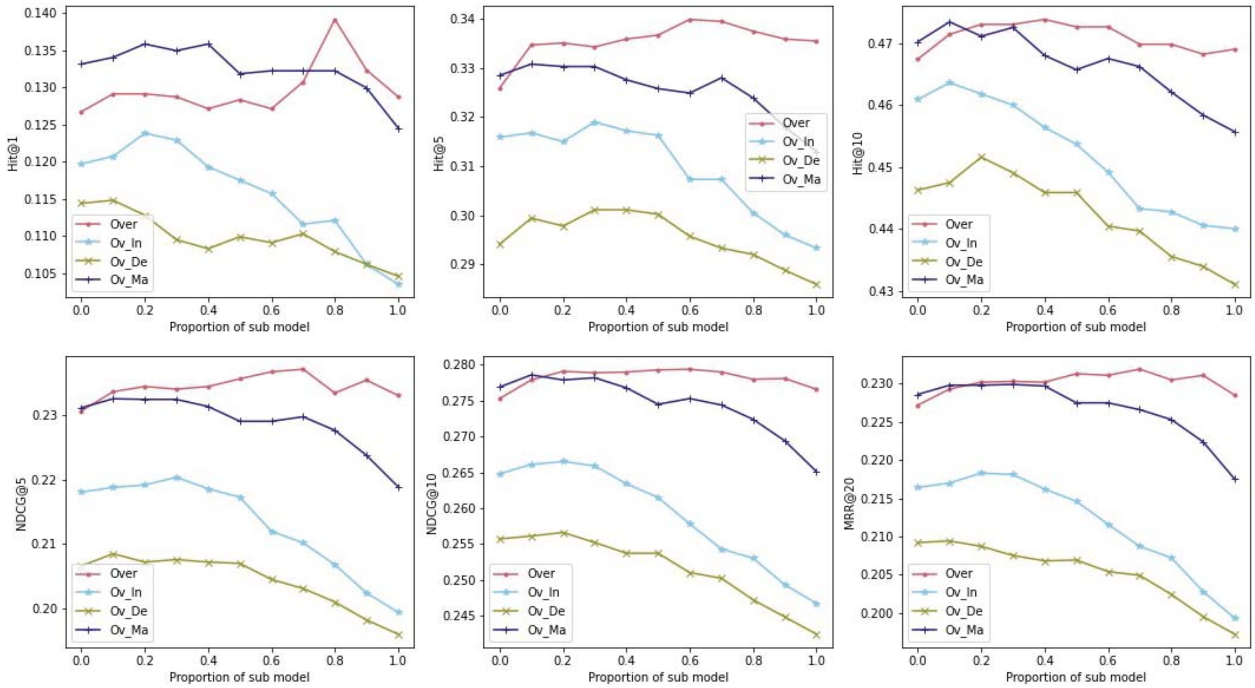

Fig. 6.4. Ensemble Two Models for Next-item Recommendation Based on Overweight Groups

In maintenance group, more incorporation of sub-model is better for the fusion model compare to the increase and decrease groups. So the best performance occurs when sub-model portion is 0.6.

For normal weight groups, the ensemble models also show the superiors results than the results of total full-model and total sub-model. Therefore, it is important to consider users' start BMI into our food recommendation. It would help us to better capture specific weight group users' features, and recommend more precise food to them.

6.2.4. Overweight Groups

Overweight group refers to the users whose BMI is in the range of [25.0,29.9] (See Table 3.2). So as Table 4.1 shows, there are 2,503 users in this group, and the average records number for each user is around 2,085. Then Figure 6.4 gives us the experiments results based on these overweight groups ('Ov', 'Ov_In', 'Ov_De', and 'Ov_Ma').

In Overweight Group, it has the same trends with the previous BMI groups, which the model performances curve shows a convex trend with the increase of α. And when $\alpha = 0.6$, which means the portion of full-model is 0.4 and it for sub-model is 0.6, the ensemble model achieves the best performances among these evaluation metrics.

(1) **Results on Overweight Group with Weight Increase Intent:**

The blue lines in Figure 6.4 show that when $\alpha = 0.3$, the ensemble model can best capture the habits of the users with increase intent. However, the total full-model's performance is better than total sub-model, which NDCG@10 are 0.2648 and 0.2467, respectively.

(2) **Results on Overweight Group with Weight Decrease Intent:**
As for the decrease intent group, the total full-model also shows the better performance than the total sub-model. And when sub-model portion is 0.1 and full-model is 0.9, the evaluation metrics show the highest values.

(3) **Results on Overweight Group with Weight Maintenance Intent:**
The experiment results of the users with maintenance intent, which are drawn in dark blues in Figure 6.4, illustrate that the total full-model is better than the total sub-model. Moreover, we could achieve the best fusion model when $\alpha = 0.1$.

Therefore, we can observe that except 'Ov' group, the rest three sub-groups' (Ov_In, Ov_De, and Ov_Ma) total full-models are more advanced than total sub-models, and the best α are smaller than other sub-groups. However, the ensemble models are still superior to the individual models, even the sub-model proportion is not large. Therefore, we can also say that consider users different underlying intents in the overweight group has the positive impact for our model performances. Besides, the results on 'Ov' group indicates it is necessary to consider overweight users' eating habits separately and incorporate to the general model. Thus, we can better fit the recommendation situation.

6.2.5. Obese Groups

The last BMI group is obese group, which contains the users whose start BMI is great and equal than 30. In this group, users are more need to lose weight due to the health consideration. Besides, the weight changes also have the same embodiment with the consideration, as the user numbers of increase, decrease, and maintenance intents sub-groups are 538, 1,977 and 40, respectively. The experiment results for each sub-groups is visualized in Figure 6.5.

The best ensemble model based on the obese group occurs when $\alpha = 0.3$.

(1) **Results on Obese Weight Group with Weight Increase Intent:**
When sub-model portion is equal to 0.4 and full-model portion is 0.6, the fusion model based on 'Ob_In' reaches the best performance among the evaluation metrics, which are 0.3269 for Hit@5, 0.2673 for NDCG@10, etc.

(2) **Results on Obese Weight Group with Weight Decrease Intent:**
0.4 is also the best sub-model fusion ratio for the users who have the decrease intent in the obese group.

(3) **Results on Obese Weight Group with Weight Maintenance Intent:**

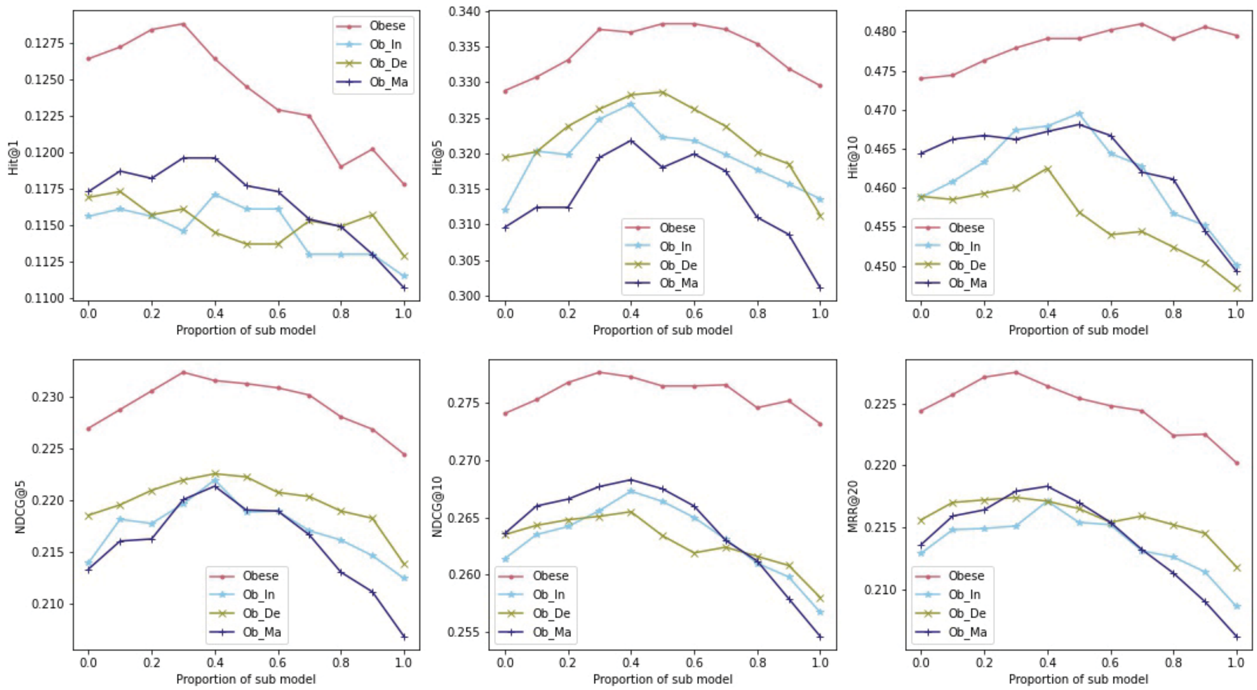

Fig. 6.5. Ensemble Two Models for Next-item Recommendation Based on Obese Groups

As for the users with maintenance intents in the obese weight group, the best fusion model is achieved when $\alpha = 0.4$, which is as same as the previous sub-groups ('Ob_In' and 'Ob_De').

From the above experiments based on obese groups, it is critical to integrate the sub-models which trained based on each obese sub-groups into full-model with the specific fusion ratios. It not only works on the entire obese group, 'Ov', but also works on the sub-groups in the obese group ('Ov_In', 'Ov_De', and 'Ov_Ma').

Therefore, BMI and intent are the useful health factors, which could help recommendation model better capture users' eating patterns, recommend appropriate food based on their intents, and help the users achieve their health goals more smoothly.

6.3. Next-basket Recommendation

Next-basket recommendation is the second modeling method for our task, which recommend a entire set of one-day's meal at each timestep and is evaluated based on the whole set (as shown in Section 5.2). Therefore, we adopt several different evaluation metrics for the next-basket modeling method, which can better consider the format of the recommended list (a whole day food set). Specifically, we use Precision@1, Precision@5, Precision@10, R-Precision, Recall@5, Recall@10, NDCG@5, and NDCG@10.

In order to analyze the impacts of BMI and Intent health factors on our recommendation system. We compare the performances of our system on various Intent groups ('Increase',

Table 6.2. Performance comparison of different models based on NDCG@10. The best performance based on each sub-dataset is boldfaced. Improvement of baselines (i.e., General) are statistically significant with $p < 0.01$.

Datasets	Un_In	Un_De	Un_Ma	No_In	No_De	No_Ma	Ov_In	Ov_De	Ov_Ma	Ob_In	Ob_De	Ob_Ma	Full
General	0.2144	0.2701	0.2134	0.2412	0.2470	0.2399	0.2251	0.2243	0.2294	0.2133	0.2257	0.2201	0.2308
Intent	0.2554	0.2843	0.2489	0.2625	0.2494	0.2486	0.2402	0.2347	0.2305	0.2199	0.2326	0.2222	0.2386
BMI	0.2907	0.3009	0.2614	0.255	0.2758	0.266	0.2403	0.2494	0.2495	0.2206	0.2503	0.2455	0.2532
Intent+BMI	0.2755	0.3014	0.2708	0.2694	0.2736	0.2609	0.2449	0.2578	0.2464	0.2435	0.2433	0.2354	0.2535
BR(I+IB)	0.2952	0.3071	0.2732	0.2764	0.2778	0.2664	0.2526	0.2603	0.2506	0.2515	0.2488	0.2382	0.2584
Improvement	37.7%	13.7%	28.0%	14.6%	12.5%	11.0%	12.2%	16.0%	9.2%	17.9%	10.2%	8.2%	12.0%
BR(B+IB)	**0.3028**	0.3067	0.2843	0.2779	0.2843	0.2730	0.2517	0.2601	0.2518	0.2508	0.2574	0.2476	0.2626
Improvement	41.2%	13.6%	33.2%	15.2%	15.1%	13.8%	11.8%	16.0%	9.8%	17.6%	14.0%	12.5%	13.8%
BR(G+B+IB)	**0.3028**	**0.3236**	**0.2871**	**0.2793**	**0.2851**	**0.2731**	**0.2541**	**0.2609**	**0.2546**	**0.2529**	**0.2574**	**0.2476**	**0.2635**
Improvement	41.2%	19.8%	34.5%	15.8%	15.4%	13.8%	12.9%	16.3%	11.0%	18.6%	14.0%	17.0%	12.5%

Table 6.3. Performance comparison of different models based on R-Precision. The best performance based on each sub-dataset is boldfaced. Improvement of baselines (i.e., General) are statistically significant with $p < 0.01$.

Datasets	Un_In	Un_De	Un_Ma	No_In	No_De	No_Ma	Ov_In	Ov_De	Ov_Ma	Ob_In	Ob_De	Ob_Ma	Full
General	0.2354	0.2316	0.2185	0.2399	0.2412	0.2419	0.2298	0.2310	0.2327	0.2141	0.2314	0.2208	0.2328
Intent	0.2562	0.2380	0.2718	0.2624	0.2450	0.2423	0.2434	0.2444	0.2351	0.2274	0.2395	0.2296	0.2412
BMI	0.2898	0.2966	0.2970	0.2586	0.2738	0.2691	0.2472	0.2593	0.2620	0.2278	0.2570	0.2591	0.2602
Intent+BMI	0.2910	0.2748	0.2771	0.2636	0.2691	0.2625	0.2505	0.2680	0.2541	0.2484	0.2566	0.2508	0.2592
BR(I+IB)	0.2931	0.3008	0.2907	0.2733	0.2752	0.2648	**0.2592**	0.2694	0.2597	0.2530	0.2585	0.2522	0.2635
Improvement	24.5%	29.9%	33.0%	13.9%	14.1%	9.5%	12.8%	16.6%	11.6%	18.2%	11.7%	14.2%	13.2%
BR(B+IB)	**0.2939**	0.3073	0.3040	0.2775	**0.2809**	0.2740	0.2578	0.2695	0.2643	**0.2542**	0.2643	**0.2639**	0.2687
Improvement	24.9%	32.7%	39.1%	15.7%	16.5%	13.3%	12.2%	16.7%	13.6%	18.7%	14.2%	19.5%	15.4%
BR(G+B+IB)	**0.2939**	**0.3100**	**0.3098**	**0.2787**	**0.2809**	**0.2742**	0.2588	**0.2698**	**0.2650**	**0.2542**	**0.2646**	**0.2639**	**0.2691**
Improvement	24.9%	33.9%	41.8%	16.2%	16.5%	13.4%	12.6%	16.8%	13.9%	18.7%	14.3%	19.5%	15.6%

'Decrease' and 'Maintenance') and BMI groups ('Underweight', 'Normal Weight', 'Overweight' and 'Obese'), see Section 3.3 and Section 3.2, respectively. First, we incorporate BMI factor into three different intent groups (Section 6.3.1) to discover the impact of it. Then, intent factor also be merged into four BMI groups (Section 6.3.2) to evaluate its effect.

The performances of NDCG@10 and R-Precision of all the data groups for ensemble models are presented in Table 6.2 and Table 6.3, respectively. 'General' row reflect the performance of the general full-model, 'Intent' presents the 'In', 'De', 'Ma' models' results. 'BMI' shows the 'Un', 'No', 'Ov' 'Ob' models' results. 'Intent+BMI' is the specific sub-models' performances, such as 'Un_In', 'No_In', etc. 'BR(I+IB)', 'BR(B+IB)' and 'BR(G+B+IB)' are the three ensemble models with best fusion ratios, which G, B, I, and IB denotes 'General' model, 'BMI' model, 'Intent' model, and 'Intent+BMI' model, respectively.

6.3.1. Effect of BMI

The rows 'BMI' in Table 6.2 and Table 6.3 reflect the BMI-specific models performances, which are average 0.2532 for NDCG@10 and average 0.2602 for R-Precision. The average performances of general full-model are 0.2308 and 0.2328. BMI-specific model outperforms

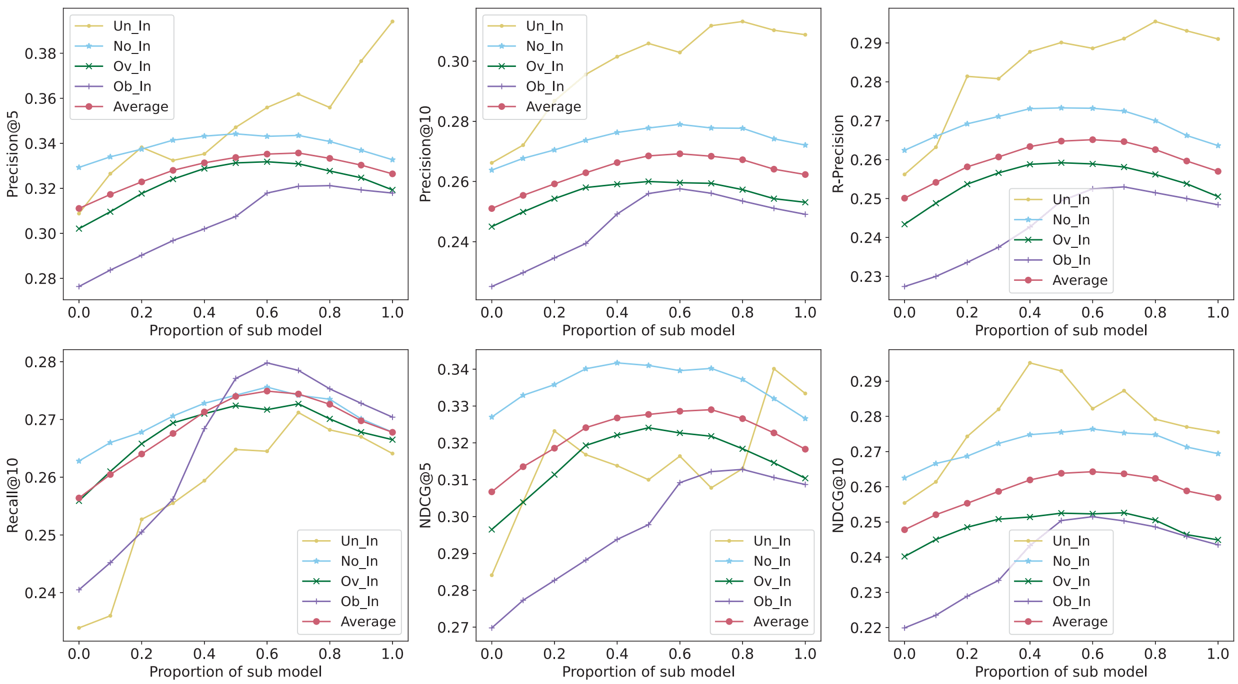

Fig. 6.6. Ensemble different BMI groups' mdoels ('Un', 'No', 'Ov', 'Ob') to 'Increase' model for next-baskets recommendation

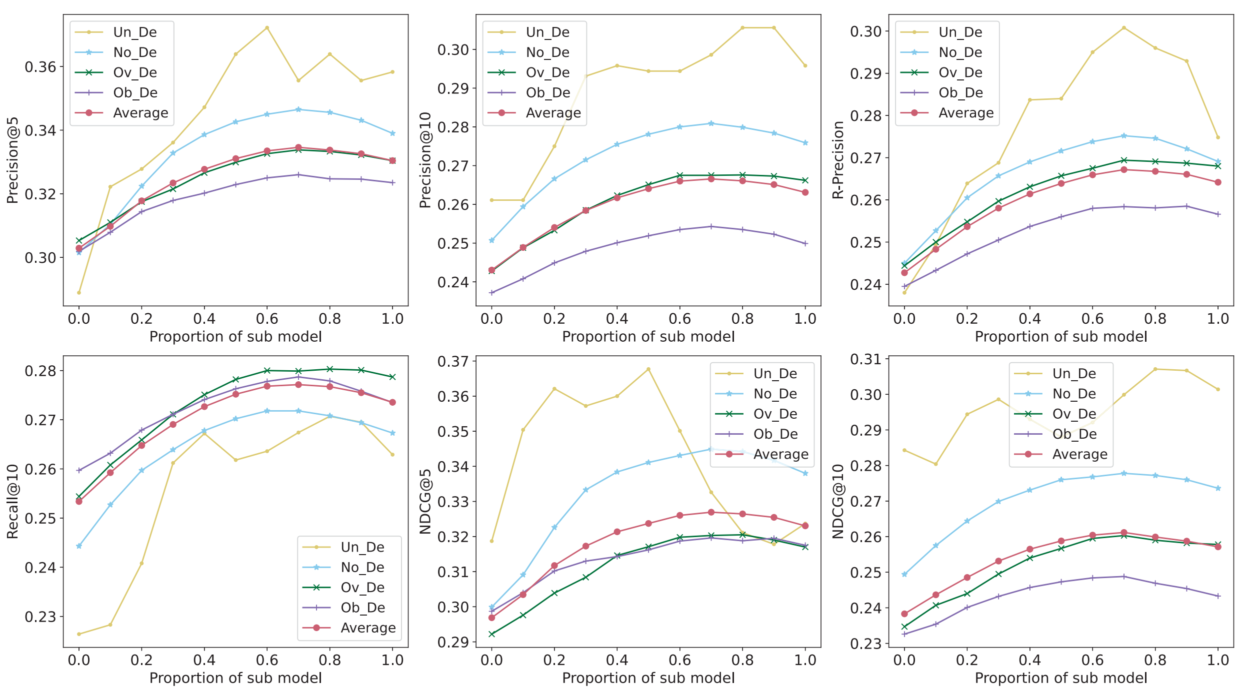

Fig. 6.7. Ensemble different BMI groups' mdoels ('Un', 'No', 'Ov', 'Ob') to 'Decrease' model for next-baskets recommendation

than the geberal full-model and the same phenomenon can be observed among all the sub-datasets. Therefore, BMI shows the positive impact for health food recommendation systems, and can help models to better capture the eating patterns of specific sub-groups.

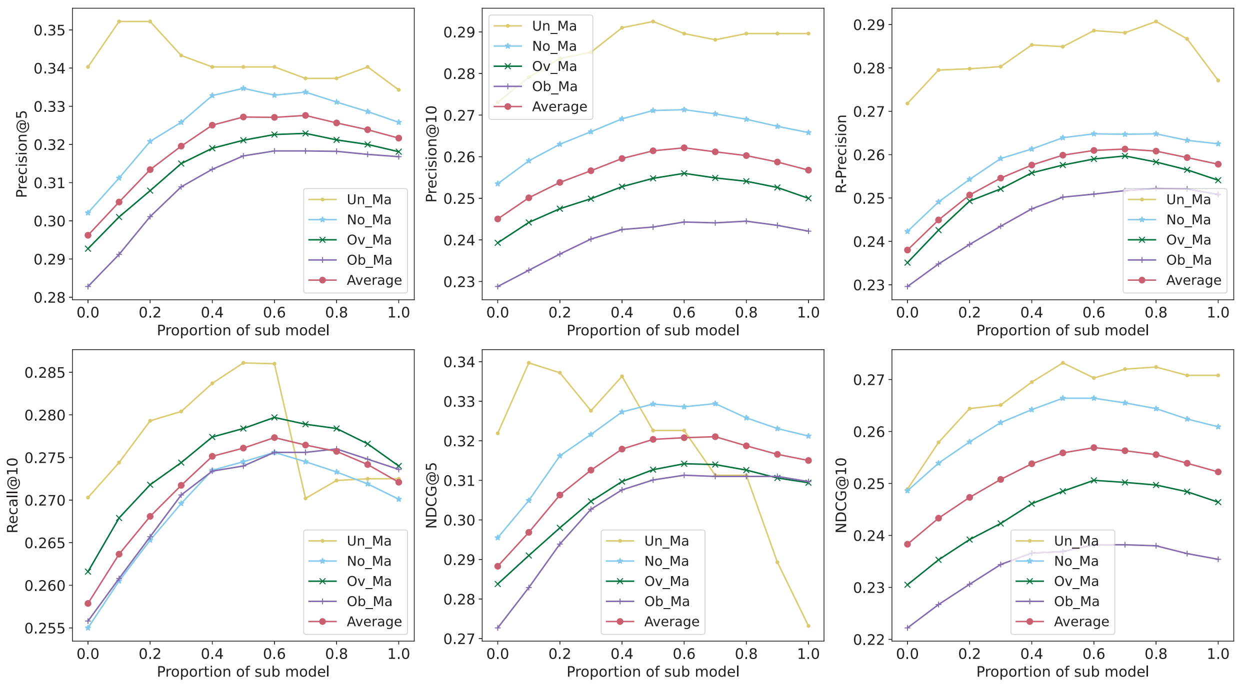

Fig. 6.8. Ensemble different BMI groups' mdoels ('Un', 'No', 'Ov', 'Ob') to 'Maintenance' model for next-baskets recommendation

Table 6.4. Best proportion by incorporating BMI into Intent groups

	Underweight	**Normal Weight**	**Overweight**	**Obese**
Increase	0.9	0.6	0.6	0.7
Decrease	0.9	0.7	0.7	0.7
Maintenance	0.4	0.6	0.6	0.8

Moreover, we would like to incorporate BMI into intent groups to evaluate if BMI have the positive impacts. The performance of 'BR(I+IB)' model are higher than it of 'Intent' (0.2584 vs 0.2386 in NDCG@10). Then, we visualize the performances curves of incorporating BMI factor on three different intent groups: 'Increase', 'Decrease' an 'Maintenance', which are given in Figure 6.6, Figure 6.7, and Figure 6.8, respectively. The x-axis denotes the ratios α of incorporating BMI factor into final scores, it gradually increases along left to right from 0 to 1.0. When ratio equals 0, which means the final results only predict by 'Intent' groups, whereas the final ranking score of each item is only comes from the corresponding sub-model (i.e., 'Un_In', 'No_In', etc.) when the ratio equals 1.0. Besides, the red line called 'Average' is the weighted average based on the data amount of these plotted sub-datasets.

For all the 'Intent' groups, we can observe that the performance curves of our ensemble model are convex, which means total intent model ($\alpha = 0$) or total sub-model ($\alpha = 1.0$) does not show the better results than the fusion model.

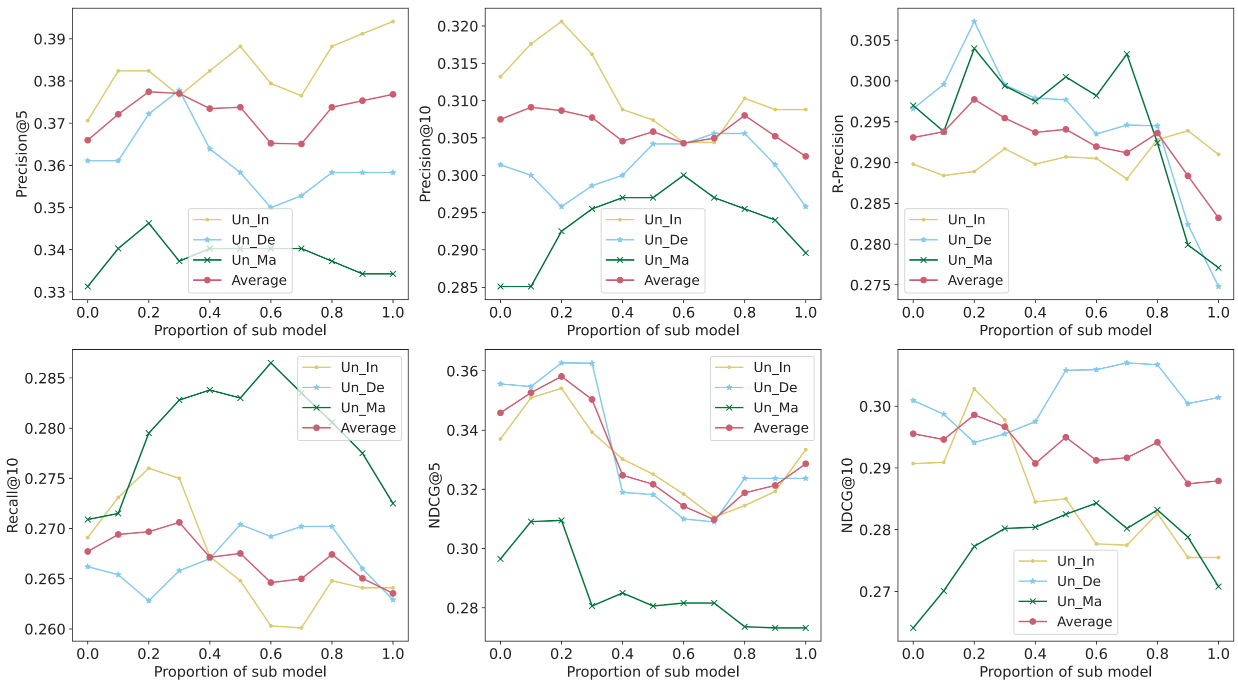

Fig. 6.9. Ensemble different Intent groups' mdoels ('In', 'De', 'Ma') to 'Underweight' model for next-baskets recommendation

Table 6.5. Best proportion by incorporating Intent into BMI groups

	Increase	**Decrease**	**Maintenance**
Underweight	0.2	0.3	0.2
Normal Weight	0.6	0.5	0.3
Overweight	0.6	0.7	0.3
Obese	0.6	0.5	0.4

Furthermore, the best incorporation portions differ among all these four BMI groups, which are given in Table 6.4. We can observe that for the 'Increase' group, the best ratios are 0.9, 0.6, 0.6 and 0.7 for 'Underweight', 'Normal Weight', 'Overweight' and 'Obese', respectively.

6.3.2. Effect of Intent

Intent (Section 3.3) is also an another important health factor, which is evaluated on this section. First of all, we can observe that the performance of Intent-specific model is higher than general full-model on both NDCG@10 (average 0.2386 vs 0.2308) and R-Precision (average 0.2412 vs 0.2328).

Then, the experiments are conducted by incorporating three Intent groups models ('Increase', 'Decrease' and 'Maintenance') into BMI groups ('Underweight', 'Normal Weight',

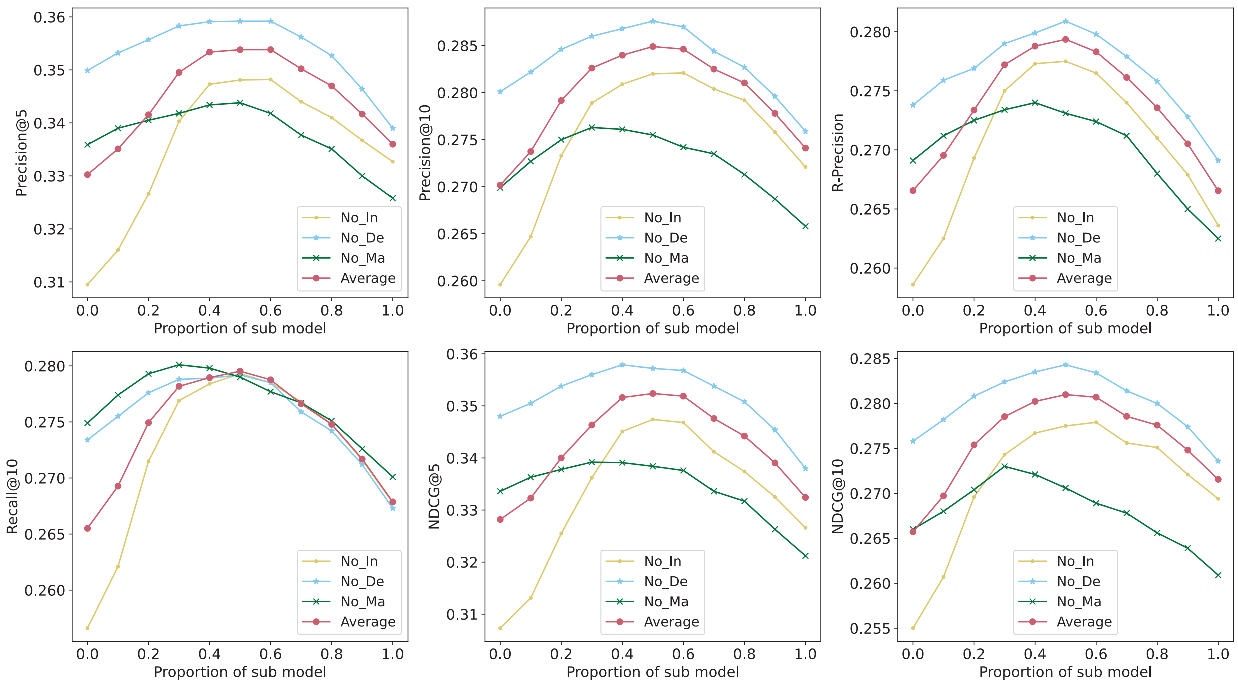

Fig. 6.10. Ensemble different Intent groups' models ('In', 'De', 'Ma') to 'Normal Weight' model for next-baskets recommendation

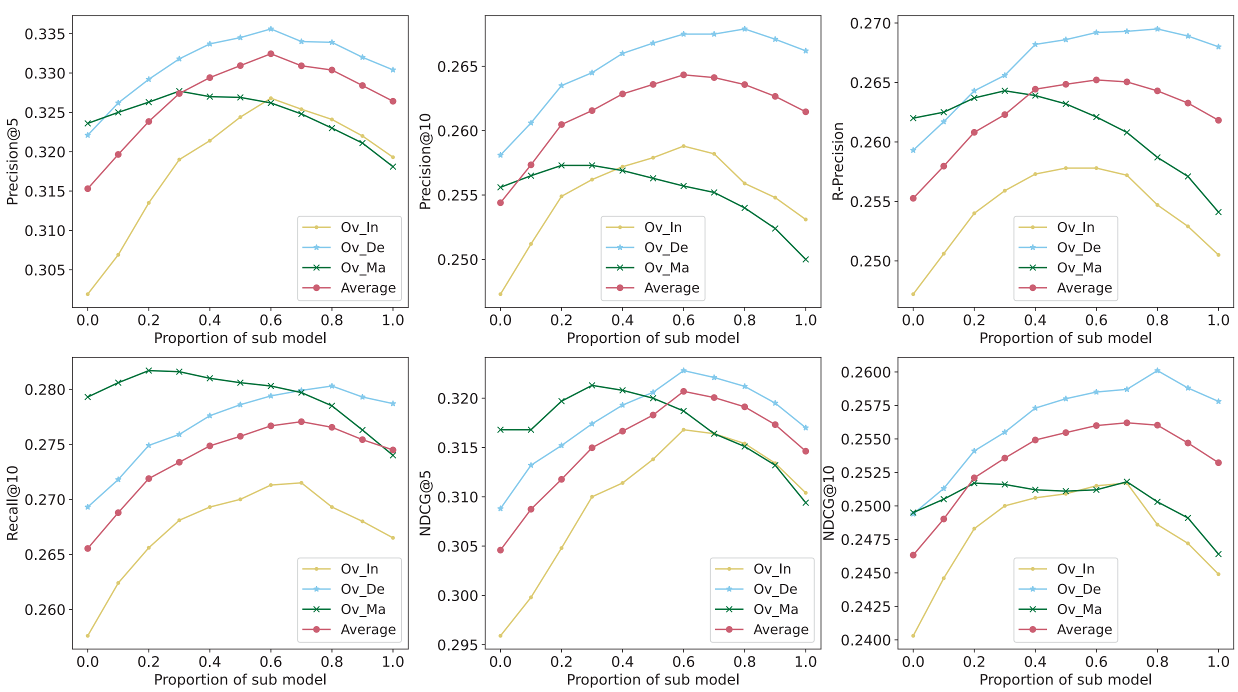

Fig. 6.11. Ensemble different Intent groups' models ('In', 'De', 'Ma') to 'Overweight' model for next-baskets recommendation

'Overweight' and 'Obese') on the country of Section 6.3.1. Table 6.2 and Table 6.3 present

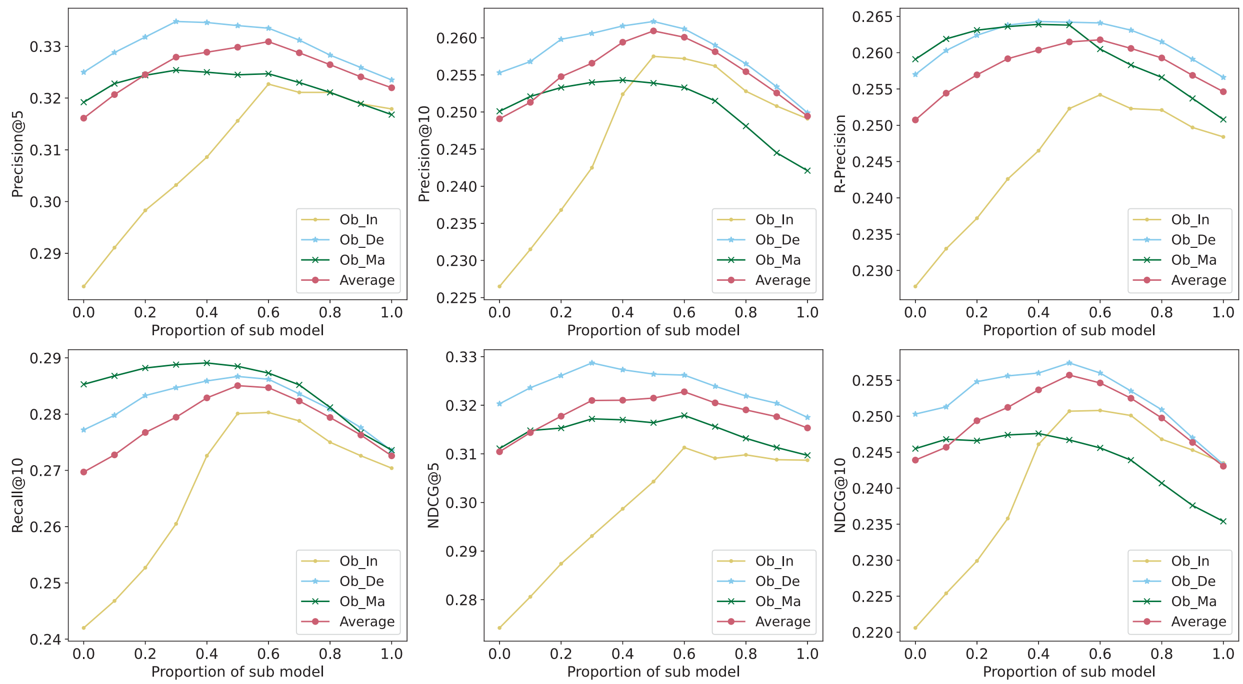

Fig. 6.12. Ensemble different Intent groups' models ('In', 'De', 'Ma') to 'Obese' model for next-baskets recommendation

the final results, which 'BMI' row lists the results of the models do not consider Intent factor; row 'Intent+BMI' denotes the results trained on sub-datasets only; then the best two and 3 models ensemble results are shown in rows 'BR(B+IB)' and 'BR(G+B+IB)', respectively.

It also shows the same phenomenon as Section 6.3.1 that the the 'BR(B+IB)' row's performances are superior than the 'BMI' and 'BMI+Intent' columns. And three ensemble models achieves the best performances among all the models. The improvements are also given in the result table. Therefore, Incorporating Intent factor into food recommendation can indeed achieve better performances than the results without Intent factor.

As in Section 6.3.1, the incorporation curves of Intent factor based on four BMI groups are visualized in Figure 6.9, Figure 6.10, Figure 6.11 and Figure 6.12, respectively. From left to right of x-axis, the incorporation ratio α of intent is more and more higher until $\alpha = 1.0$, which means the the prediction are made by the sub-model only. The performance curves are also as convex as the curves of BMI incorporation (Section 6.3.1). Besides, Table 6.5 presents the the best fusion ratios of each sub-datasets.

In addition, we can see that the results of incorporating intents based on 'Underweight' data are not as stable as other groups. The reason could be the data amount of 'Underweight' group is way more less than the other groups (39 users VS around 2,500 users). Thus, it is difficult for our model to capture the users' diet patterns and habits. However, the ensemble model with the specific fusion ratio is also better-performed than the separate

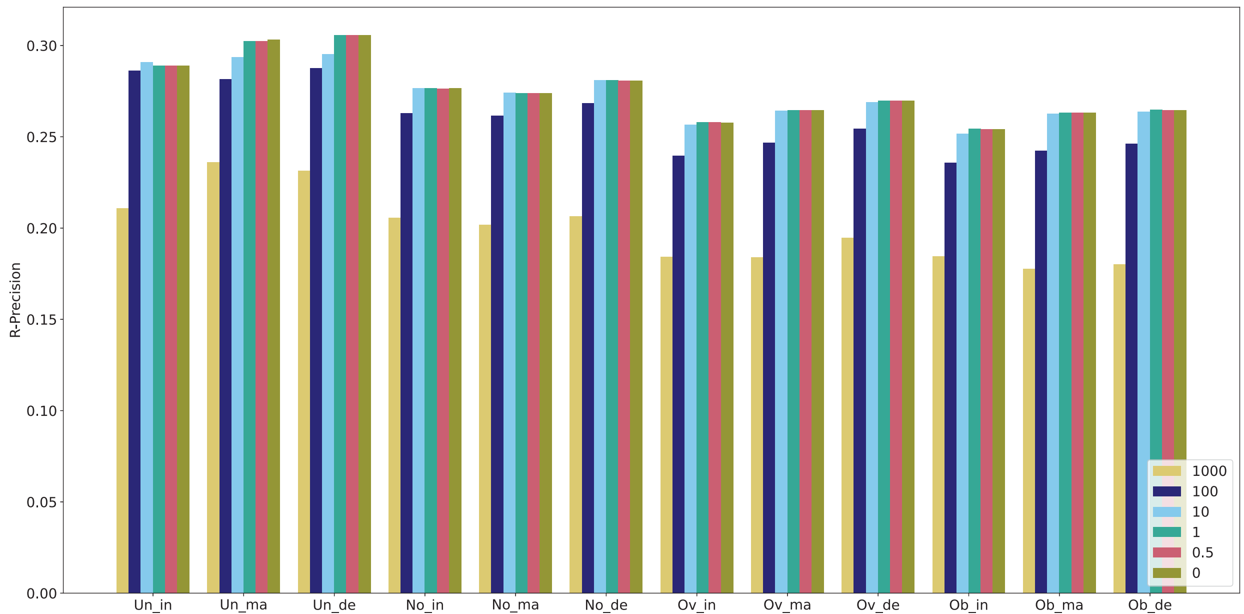

Fig. 6.13. R-Precision of Variety

Intent and sub-set models. It indicates that incorporating the Intent health factor into our food recommendation system can help the systems achieve greater performances based on the corresponding sub-groups. Furthermore, it can help the users achieve their intent gradually and smoothly.

6.3.3. Effect of Variety

We adopt greedy search algorithm to select food items into basket according to the Equation 6.3.1 at each step. A greedy search algorithm makes the optimal choice that seems to be the best at that moment at each step. This means that it makes a locally-optimal choice in the hope that this choice will lead to a globally-optimal solution.

$$Max(s_i - \alpha \cdot \mathrm{HHI}), \quad i \in [1,N], \tag{6.3.1}$$

where $\alpha \in \{0,0.5,1,10,100,1000\}$, and the calculation of HHI is given in Equation 3.6.1. s_i is the ranking scores of item i which is generated by the best 3 ensemble model, α is a parameter to adjusted the impact extent of HHI (variety), bigger α represents higher impact of variety. It means variety (HHI) does not have any impact for generating recommend basket when $\alpha = 0$.

The R-Rrecision and NDCG@10 performances of different α based on 12 sub-groups are presented in Figure 6.13 and Figure 6.14. We can observe that when $\alpha = 0, 0.5, 1$ and 10, the performances are similar, smaller impact of variety does not improve or reduce the performances. However, higher impact of variety ($\alpha = 100, 1000$), the performances dropped

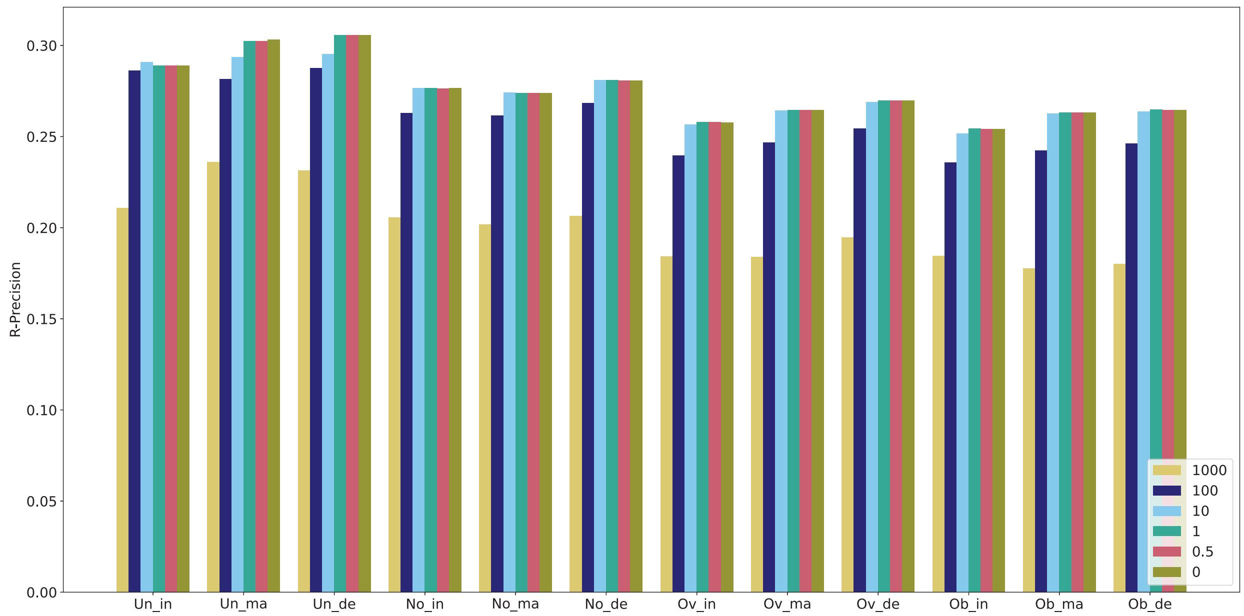

Fig. 6.14. NDCG@10 of Variety

significantly. Therefore, the variety does not have the positive impact for our health food recommendation system.

6.3.4. Summary

The above sections examined both user's BMI and underlying Intent. They show positive impacts for health food recommendation. Comparing the improvements of BMI and Intent incorporation (Table 6.3 and Table 6.2), BMI shows the larger improvement than Intent among all the sub-groups, which indicates that BMI has a bigger role than Intent. Table 6.4 and Table 6.5 also shows the same findings, as the incorporation ratios of BMI for best ensemble models are much higher than Intent, with average 0.68 and 0.43, respectively.

Moreover, ensemble three models (general full-model, BMI-specific, and Intent-specific model) achieves the best performances among all the models based on all sub-groups. It indicates that considering both BMI and Intent into general models could not only cater users' general and specific diet patterns, but also gradually guide users towards to their goals. The visualizations of ensemble these three model based on 'Ov_In' and 'Ov_De' sub-groups are presented in Figure 6.16 and Figure 6.16, respectively. Besides, the fusion figures based on other sub-datasets are attached in the Appendix A.2.

In addition, variety is proved it does not have positive impact of our recommender system. The variety measurement metrics (HHI) and incorporation method (greedy search) could be further validation in the future.

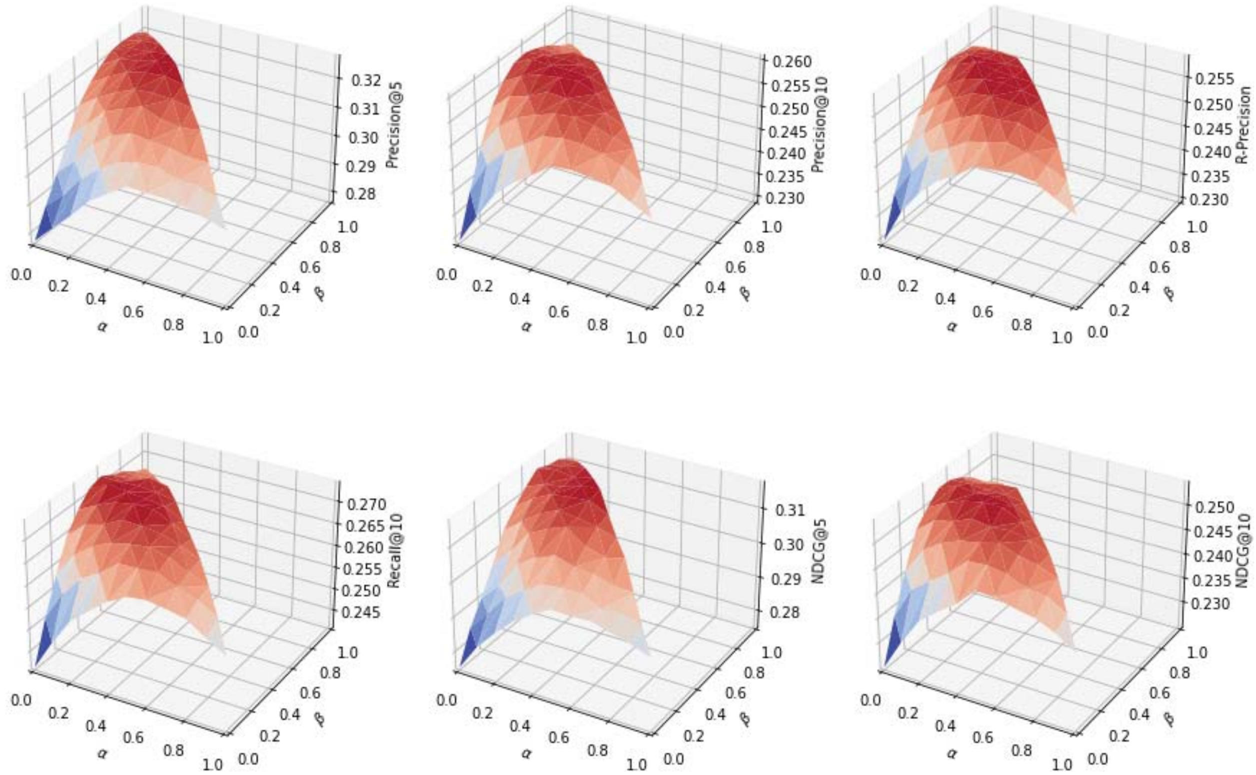

Fig. 6.15. Ensemble Three Models (BR(G+B+IB)) for Next-basket Recommendation Based on Ov_In Group

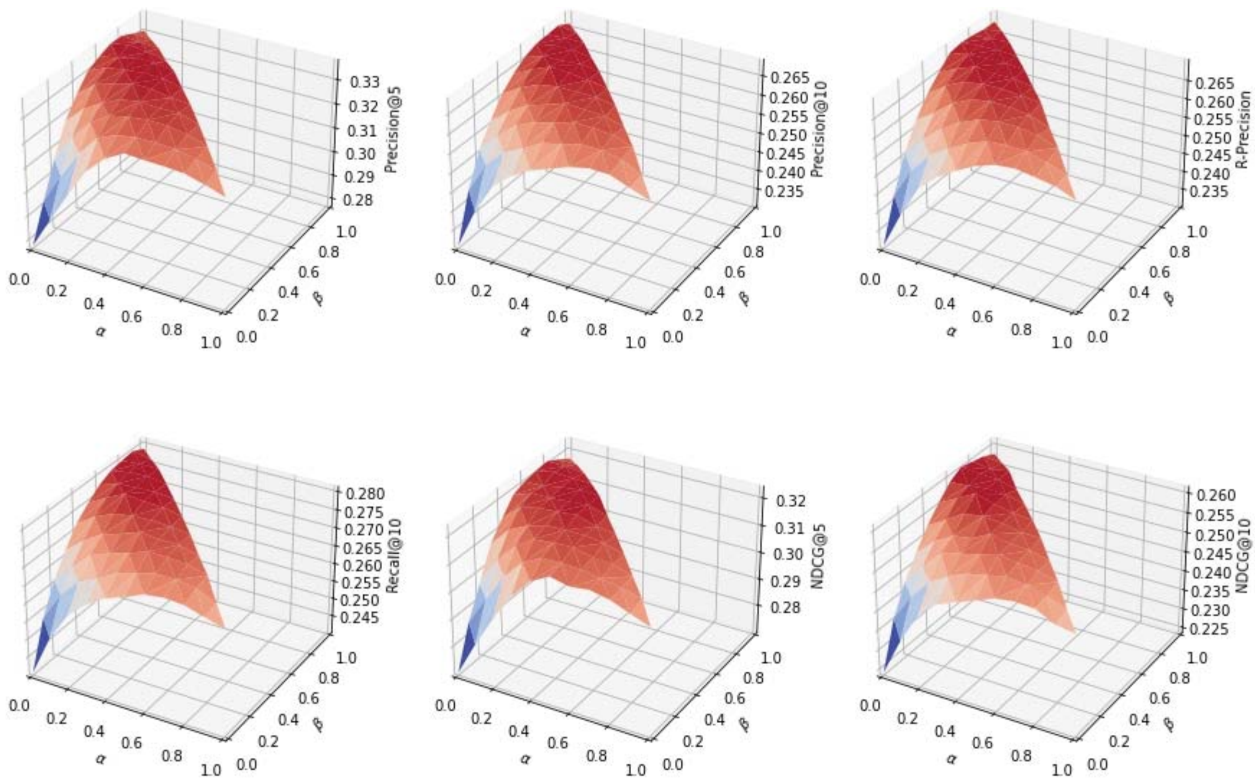

Fig. 6.16. Ensemble Three Models (BR(G+B+IB)) for Next-basket Recommendation Based on Ov_De Group

Chapter 7

Conclusion

This thesis presents the novel health and nutrient-based food recommendation with two modeling approaches: Next-item recommendation and Next-basket recommendation. These models incorporate nutrition information (calories) as the enhanced input of each food item, the intent and BMI of each user as the sub-group classification criteria, variety of each basket as part of the objective for the greedy search algorithm.

Then the combined models show the effective capabilities to capture users' eating habits with different BMI groups, and also consider users' goals. Experimental results on the real-world mobile health (mHealth) dataset demonstrated the effectiveness of our combined model for guiding users to achieve their goals as well as considering their eating habits. Besides, the combined ratio regulates the intensity of the user's goal achievement (i.e., a higher combine ratio means higher intensity), which could be adjusted by the user.

BMI and Intent have positive impacts on our health food recommendation and BMI shows the superior effect than Intent. On the other hand, Variety does not show the positive impact as we expected before. This series of experiments confirm the previous results that the nutrition health factors about food items and users provide some useful information for food recommendation. Besides, the superior performance of our ensemble model demonstrates that food recommendations should consider multi-faceted information.

As for future work, it would be interesting to investigate more nutrient health factors (e.g., food calorie density), and explore the different ensemble approaches. Besides, we would also test other variety measurement metric than HHI and their incorporation into the recommendation model. Other base recommendation models can also be used instead of GRU4Rec. When more detailed description of foods is available, it would be interesting to explore the utilization of BERT to create contextualized representations.

References

[1] J. Aberg. Dealing with malnutrition: A meal planning system for elderly. In *AAAI Spring Symposium: Argumentation for Consumers of Healthcare*, 2006.

[2] Issa Annamoradnejad. Colbert: Using BERT sentence embedding for humor detection. *CoRR*, abs/2004.12765, 2020.

[3] Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. Unsupervised neural machine translation. *CoRR*, abs/1710.11041, 2017.

[4] Damoon Ashtary-Larky, Reza Bagheri, Amir Abbasnezhad, Grant M Tinsley, Meysam Alipour, and Alexei Wong. Effects of gradual weight loss v. rapid weight loss on body composition and rmr: a systematic review and meta-analysis. *British journal of nutrition*, 124(11):1121–1132, 2020.

[5] Fatemeh Azizi-Soleiman and Leila Azadbakht. Weight loss maintenance: A review on dietary related strategies. *Journal of research in medical sciences : the official journal of Isfahan University of Medical Sciences*, 19:268–275, 03 2014.

[6] Ting Bai, Jian-Yun Nie, Wayne Xin Zhao, Yutao Zhu, Pan Du, and Ji-Rong Wen. An attribute-aware neural attentive model for next basket recommendation. In Kevyn Collins-Thompson, Qiaozhu Mei, Brian D. Davison, Yiqun Liu, and Emine Yilmaz, editors, *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, pages 1201–1204. ACM, 2018.

[7] Yoshua Bengio, Paolo Frasconi, and Patrice Y. Simard. The problem of learning long-term dependencies in recurrent networks. In *Proceedings of International Conference on Neural Networks (ICNN'88), San Francisco, CA, USA, March 28 - April 1, 1993*, pages 1183–1188. IEEE, 1993.

[8] Ruth Brown, Karissa Canning, Michael Fung, Dishay Jiandani, Michael Riddell, Alison Macpherson, and Jennifer Kuk. Calorie estimation in adults differing in body weight class and weight loss status. *Medicine and science in sports and exercise*, 48, 10 2015.

[9] Jingjing Chen, Chong-Wah Ngo, and Tat-Seng Chua. Cross-modal recipe retrieval with rich food attributes. In Qiong Liu, Rainer Lienhart, Haohong Wang, Sheng-Wei "Kuan-Ta" Chen, Susanne Boll, Yi-Ping Phoebe Chen, Gerald Friedland, Jia Li, and Shuicheng Yan, editors, *Proceedings of the 2017 ACM on Multimedia Conference, MM 2017, Mountain View, CA, USA, October 23-27, 2017*, pages 1771–1779. ACM, 2017.

[10] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734. ACL, 2014.

[11] Zafra Cooper and Christopher G Fairburn. A new cognitive behavioural approach to the treatment of obesity. *Behaviour research and therapy*, 39(5):499–511, 2001.

[12] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc Viet Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2978–2988. Association for Computational Linguistics, 2019.

[13] Alexandre de Brébisson, Étienne Simon, Alex Auvolat, Pascal Vincent, and Yoshua Bengio. Artificial neural networks applied to taxi destination prediction. In Adolfo Martínez Usó, João Mendes-Moreira, Luís Moreira-Matias, Meelis Kull, and Nicolas Lachiche, editors, *Proceedings of the ECML/PKDD 2015 Discovery Challenges co-located with European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD 2015), Porto, Portugal, September 7-11, 2015*, volume 1526 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2015.

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019.

[15] Larissa Drescher. *Healthy food diversity as a concept of dietary quality: Measurement, determinants of consumer demand, and willingness to pay.* Cuvillier Verlag, 2007.

[16] Cem Ekmekcioglu and Yvan Touitou. Chronobiological aspects of food intake and metabolism and their relevance on energy balance and weight regulation. *Obesity reviews : an official journal of the International Association for the Study of Obesity*, 12:14–25, 01 2011.

[17] M. Elahi, M. Ge, F. Ricci, I. Fernandez-Tobıas, and D. Berkovsky, S. andMassimo. Interaction design in a mobile food recommender system. In *IntRS@recsys, CEUR-WS.org, CEUR workshop proceedings*, volume 1438, pages 49–52, 2015.

[18] David Elsweiler and Morgan Harvey. Towards automatic meal plan recommendations for balanced nutrition. In Hannes Werthner, Markus Zanker, Jennifer Golbeck, and Giovanni Semeraro, editors, *Proceedings of the 9th ACM Conference on Recommender Systems, RecSys 2015, Vienna, Austria, September 16-20, 2015*, pages 313–316. ACM, 2015.

[19] Mouzhi Ge, Francesco Ricci, and David Massimo. Health-aware food recommender system. In Hannes Werthner, Markus Zanker, Jennifer Golbeck, and Giovanni Semeraro, editors, *Proceedings of the 9th ACM Conference on Recommender Systems, RecSys 2015, Vienna, Austria, September 16-20, 2015*, pages 333–334. ACM, 2015.

[20] Gijs Geleijnse, Peggy Nachtigall, Pim van Kaam, and Luciënne Wijgergangs. A personalized recipe advice system to promote healthful choices. In Pearl Pu, Michael J. Pazzani, Elisabeth André, and Doug Riecken, editors, *Proceedings of the 16th International Conference on Intelligent User Interfaces, IUI 2011, Palo Alto, CA, USA, February 13-16, 2011*, pages 437–438. ACM, 2011.

[21] Drishti P Ghelani, Lisa J Moran, Cameron Johnson, Aya Mousa, and Negar Naderpoor. Mobile apps for weight management: A review of the latest evidence to inform practice. *Frontiers in Endocrinology*, 11:412, 2020.

[22] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5-6):602–610, 2005.

[23] Zellig Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

[24] Kelly L. Haws, Peggy J. Liu, Joseph P. Redden, and Heidi J. Silver. Exploring the relationship between varieties of variety and weight loss: When more variety can help people lose weight. *Journal of Marketing Research*, 54(4):619–635, August 2017. Publisher Copyright: © 2017, American Marketing Association. Copyright: Copyright 2018 Elsevier B.V., All rights reserved.

[25] Ruining He, Wang-Cheng Kang, and Julian J. McAuley. Translation-based recommendation. In Paolo Cremonesi, Francesco Ricci, Shlomo Berkovsky, and Alexander Tuzhilin, editors, *Proceedings of the Eleventh ACM Conference on Recommender Systems, RecSys 2017, Como, Italy, August 27-31, 2017*, pages 161–169. ACM, 2017.

[26] Balázs Hidasi, Massimo Quadrana, Alexandros Karatzoglou, and Domonkos Tikk. Parallel recurrent neural network architectures for feature-rich session-based recommendations. In Shilad Sen, Werner Geyer, Jill Freyne, and Pablo Castells, editors, *Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15-19, 2016*, pages 241–248. ACM, 2016.

[27] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, 1997.

[28] Rosemary Johnson. Restructuring: An emerging theory on the process of losing weight. *Journal of advanced nursing*, 15(11):1289–1296, 1990.

[29] Dan Jurafsky and James H. Martin. *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition, 2nd Edition*. Prentice Hall series in artificial intelligence. Prentice Hall, Pearson Education International, 2009.

[30] Wang-Cheng Kang and Julian J. McAuley. Self-attentive sequential recommendation. In *IEEE International Conference on Data Mining, ICDM 2018, Singapore, November 17-20, 2018*, pages 197–206. IEEE Computer Society, 2018.

[31] Anna-Maria Keränen, Markku Savolainen, Annakaisa Reponen, Mona-Lisa Kujari, Sari Lindeman, Risto Bloigu, and Jaana Laitinen. The effect of eating behavior on weight loss and maintenance during a lifestyle intervention. *Preventive medicine*, 49:32–8, 05 2009.

[32] Yoon Kim. Convolutional neural networks for sentence classification. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751. ACL, 2014.

[33] Yehuda Koren and Robert M. Bell. Advances in collaborative filtering. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, *Recommender Systems Handbook*, pages 145–186. Springer, 2011.

[34] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In Kevin Knight, Ani Nenkova, and Owen Rambow, editors, *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 260–270. The Association for Computational Linguistics, 2016.

[35] Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[36] Jonq-Ying Lee and Mark G Brown. Consumer demand for food diversity. *Journal of Agricultural and Applied Economics*, 21(2):47–53, 1989.

[37] Jure Leskovec, Anand Rajaraman, and Jeffrey D. Ullman. *Mining of Massive Datasets, 2nd Ed.* Cambridge University Press, 2014.

[38] Jessica RL Lieffers, Jose F Arocha, Kelly Grindrod, and Rhona M Hanning. Experiences and perceptions of adults accessing publicly available nutrition behavior-change mobile apps for weight management. *Journal of the Academy of Nutrition and Dietetics*, 118(2):229–239, 2018.

[39] Yabo Ling, Jian-Yun Nie, Daiva Nielsen, Bärbel Knäuper, Nathan Yang, and Laurette Dubé. Following good examples - health goal-oriented food recommendation based on behavior data. In *WWW '22: The Web Conference 2022, Virtual Event / Lyon, France, April 25–29, 2022.* ACM / IW3C2, 2022.

[40] Qiao Liu, Refuoe Mokhosi, Yifu Zeng, and Haibin Zhang. Stamp: Short-term attention/memory priority model for session-based recommendation. pages 1831–1839. Association for Computing Machinery, 7 2018.

[41] Xuezhe Ma and Eduard H. Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016.

[42] Sanidhya Mangal, Poorva Joshi, and Rahul Modak. LSTM vs. GRU vs. bidirectional RNN for script generation. *CoRR*, abs/1908.04332, 2019.

[43] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In Yoshua Bengio and Yann LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013.

[44] Michael C. Mozer. A focused backpropagation algorithm for temporal pattern recognition. *Complex Syst.*, 3(4), 1989.

[45] Masato Neishi, Jin Sakuma, Satoshi Tohda, Shonosuke Ishiwatari, Naoki Yoshinaga, and Masashi Toyoda. A bag of useful tricks for practical neural machine translation: Embedding layer initialization and large batch size. In Toshiaki Nakazawa and Isao Goto, editors, *Proceedings of the 4th Workshop on Asian Translation, WAT@IJCNLP 2017, Taipei, Taiwan, November 27- December 1, 2017*, pages 99–109. Asian Federation of Natural Language Processing, 2017.

[46] Charoula K Nikolaou and Michael EJ Lean. Mobile applications for obesity and weight management: current market characteristics. *International Journal of Obesity*, 41(1):200–202, 2017.

[47] Marcia Otto, Cheryl Anderson, Jennifer Dearborn, Erin Ferranti, Dariush Mozaffarian, Goutham Rao, Judith Wylie-Rosett, and Alice Lichtenstein. Dietary diversity: Implications for obesity prevention in adult populations: A science advisory from the american heart association. *Circulation*, 138, 09 2018.

[48] Makbule Gulcin Ozsoy. From word embeddings to item recommendation. *CoRR*, abs/1601.01356, 2016.

[49] Razvan Pascanu, Tomás Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 1310–1318. JMLR.org, 2013.

[50] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. ACL, 2014.

[51] Michael G Perri, Arthur M Nezu, Eugene T Patti, and Karen L McCann. Effect of length of treatment on weight loss. *Journal of consulting and clinical psychology*, 57(3):450, 1989.

[52] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In Marilyn A. Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics, 2018.

[53] Ye Qi, Devendra Singh Sachan, Matthieu Felix, Sarguna Padmanabhan, and Graham Neubig. When and why are pre-trained word embeddings useful for neural machine translation? In Marilyn A. Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 529–535. Association for Computational Linguistics, 2018.

[54] Vithala R Rao, Gary J Russell, Hemant Bhargava, Alan Cooke, Tim Derdenger, Hwang Kim, Nanda Kumar, Irwin Levin, Yu Ma, Nitin Mehta, John Pracejus, and R Venkatesh. Emerging Trends in Product Bundling: Investigating Consumer Choice and Firm Behavior. *Customer Needs and Solutions*, 5(1):107–120, 2018.

[55] Hollie A Raynor. Can limiting dietary variety assist with reducing energy intake and weight loss? *Physiology & behavior*, 106(3):356–361, 2012.

[56] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In Michael Rappa, Paul Jones, Juliana Freire, and Soumen Chakrabarti, editors, *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*, pages 811–820. ACM, 2010.

[57] Stephen A Rhoades. The herfindahl-hirschman index. *Fed. Res. Bull.*, 79:188, 1993.

[58] A. Robertson. Food and health in europe: a new basis for action. *Academic Search Complete, WHO Regional Office for Europe*, 2004.

[59] Kathryn Ross, Peihua Qiu, Lu You, and Rena Wing. Week-to-week predictors of weight loss and regain. *Health Psychology*, 38, 09 2019.

[60] Badrul Munir Sarwar, George Karypis, Joseph A. Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In Vincent Y. Shen, Nobuo Saito, Michael R. Lyu, and Mary Ellen Zurko, editors, *Proceedings of the Tenth International World Wide Web Conference, WWW 10, Hong Kong, China, May 1-5, 2001*, pages 285–295. ACM, 2001.

[61] Julie T Schaefer and Amy B Magnuson. A review of interventions that promote eating by internal cues. *Journal of the Academy of Nutrition and Dietetics*, 114(5):734–760, 2014.

[62] Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin. The performance of LSTM and bilstm in forecasting time series. In Chaitanya Baru, Jun Huan, Latifur Khan, Xiaohua Hu, Ronay Ak, Yuanyuan Tian, Roger S. Barga, Carlo Zaniolo, Kisung Lee, and Yanfang Fanny Ye, editors, *2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, December 9-12, 2019*, pages 3285–3292. IEEE, 2019.

[63] Nicolette Sullivan, Gavan Fitzsimons, Michael Platt, and Scott Huettel. Indulgent foods can paradoxically promote disciplined dietary choices. *Psychological Science*, 30:095679761881750, 01 2019.

[64] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In Wenwu Zhu, Dacheng Tao, Xueqi Cheng, Peng Cui, Elke A. Rundensteiner, David Carmel, Qi He, and Jeffrey Xu Yu, editors,

Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019, pages 1441–1450. ACM, 2019.

[65] Yong Kiam Tan, Xinxing Xu, and Yong Liu. Improved recurrent neural networks for session-based recommendations. In Alexandros Karatzoglou, Balázs Hidasi, Domonkos Tikk, Oren Sar Shalom, Haggai Roitman, Bracha Shapira, and Lior Rokach, editors, *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, DLRS@RecSys 2016, Boston, MA, USA, September 15, 2016*, pages 17–22. ACM, 2016.

[66] Jiaxi Tang and Ke Wang. Personalized top-n sequential recommendation via convolutional sequence embedding. *WSDM 2018 - Proceedings of the 11th ACM International Conference on Web Search and Data Mining*, 2018-Febua:565–573, 2018.

[67] Narges Tavakolpoursaleh, Johann Schaible, and Stefan Dietze. Using word embeddings for recommending datasets based on scientific publications. In Robert Jäschke and Matthias Weidlich, editors, *Proceedings of the Conference on "Lernen, Wissen, Daten, Analysen", Berlin, Germany, September 30 - October 2, 2019*, volume 2454 of *CEUR Workshop Proceedings*, pages 365–370. CEUR-WS.org, 2019.

[68] ChunYuen Teng, Yu-Ru Lin, and Lada A. Adamic. Recipe recommendation using ingredient networks. In Noshir S. Contractor, Brian Uzzi, Michael W. Macy, and Wolfgang Nejdl, editors, *Web Science 2012, WebSci '12, Evanston, IL, USA - June 22 - 24, 2012*, pages 298–307. ACM, 2012.

[69] T.N. Trang Tran, M. Atas, and A. et al. Felfernig. An overview of recommender systems in the healthy food domain. *J Intell Inf Syst*, 50:501—-526, 2018.

[70] Mayumi Ueda, Syungo Asanuma, Yusuke Miyawaki, and Shinsuke Nakajima. Recipe recommendation method by considering the users preference and ingredient quantity of target recipe. In *Proceedings of the international multiconference of engineers and computer scientists*, volume 1, pages 12–14, 2014.

[71] Erica van Herpen and Rik Pieters. The variety of an assortment : An extension to the attribute-based approach. *Marketing Science*, 21:331–341, 06 2002.

[72] Y. Van Pinxteren, G. Geleijnse, and P. Kamsteeg. Deriving a recipe similarity measure for recommending healthful meals. In *Proceedings of the 16th international conference on intelligent user interfaces*, pages 105—114, 2011.

[73] Corneel Vandelanotte, Andre M Müller, Camille E Short, Melanie Hingle, Nicole Nathan, Susan L Williams, Michael L Lopez, Sanjoti Parekh, and Carol A Maher. Past, present, and future of ehealth and mhealth research to improve physical activity and dietary behaviors. *Journal of nutrition education and behavior*, 48(3):219–228, 2016.

[74] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017.

[75] Neeltje Vogels, Kristel Diepvens, and Margriet Westerterp-Plantenga. Predictors of long-term weight maintenance. *Obesity research*, 13:2162–8, 01 2006.

[76] Jianling Wang, Kaize Ding, Liangjie Hong, Huan Liu, and James Caverlee. Next-item recommendation with sequential hypergraphs. In Jimmy Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu, editors, *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 1101–1110. ACM, 2020.

[77] Pengfei Wang, Jiafeng Guo, Yanyan Lan, Jun Xu, Shengxian Wan, and Xueqi Cheng. Learning hierarchical representation model for next basket recommendation. In Ricardo Baeza-Yates, Mounia Lalmas, Alistair Moffat, and Berthier A. Ribeiro-Neto, editors, *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015*, pages 403–412. ACM, 2015.

[78] Edward Weiss, Deborah Galuska, Laura Kettel Khan, and Mary Serdula. Weight-control practices among u.s. adults. *American journal of preventive medicine*, 31:18–24, 08 2006.

[79] Haochao Ying, Fuzhen Zhuang, Fuzheng Zhang, Yanchi Liu, Guandong Xu, Xing Xie, Hui Xiong, and Jian Wu. Sequential recommender system based on hierarchical attention networks. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 3926–3932. International Joint Conferences on Artificial Intelligence Organization, 7 2018.

[80] Feng Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. A dynamic recurrent model for next basket recommendation. In Raffaele Perego, Fabrizio Sebastiani, Javed A. Aslam, Ian Ruthven, and Justin Zobel, editors, *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016*, pages 729–732. ACM, 2016.

[81] Zheni Zeng, Chaojun Xiao, Yuan Yao, Ruobing Xie, Zhiyuan Liu, Fen Lin, Leyu Lin, and Maosong Sun. Knowledge transfer via pre-training for recommendation: A review and prospect. *Frontiers Big Data*, 4:602071, 2021.

[82] Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 19–27. IEEE Computer Society, 2015.

[83] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proc. IEEE*, 109(1):43–76, 2021.

Appendix A

Food List and Additional Results

A.1. Food Item List

Table A.1. Top 80 Most Popular Consumed Food Item List in the Dataset

No.1-20	**No.21-40**	**No.41-60**	**No.61-80**
cheese	tomato	coca cola	hamburger
milk	cracker	carrot	mayonnaise
coffee	sandwich	strawberry	cake
chicken	soup	snack	alcohol
egg	fish	ice cream	tea
salad	sauce	sausage	pretzel
yogurt	candy	white bread	sour cream
cereal	cereal bar	blueberry	bagel
oil	pasta	tortilla	pickle
banana	almond	corn	roll
toast	turkey	orange	vegetable
apple	bar	avocado	grapes
cookie	milkshake	broccoli	seed
butter	pizza	tortilla chip	deli meat
water	pork	grain	peanut
chocolate	beer	french fries	jam
rice	potato chip	sugar white	honey
beef	bacon	onion	wine white
potato	juice	biscuit	oatmeal
peanut butter	spinach	wine red	mushroom

A.2. Three Models Ensemble Figures

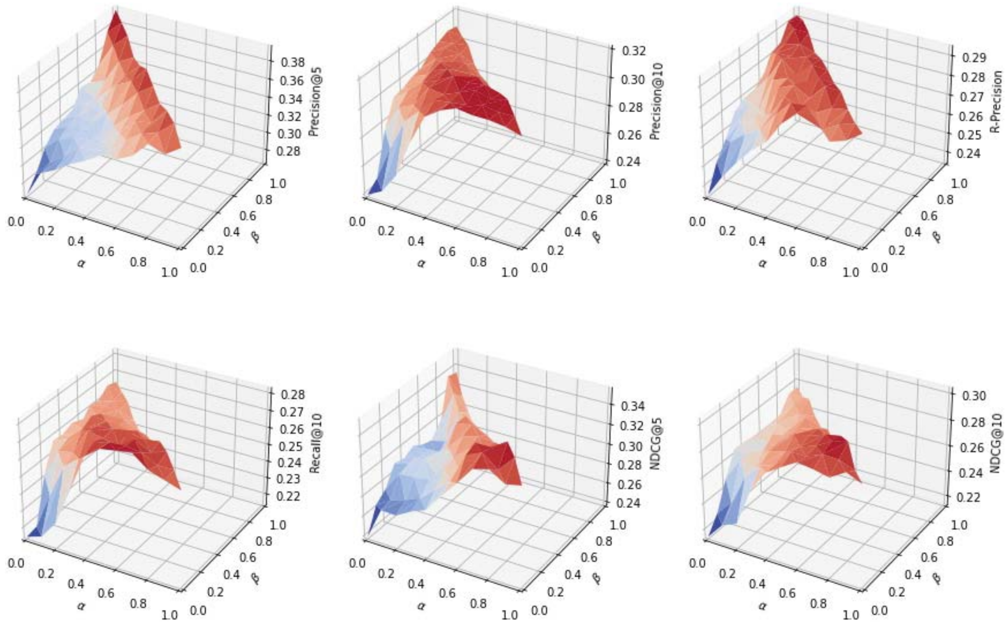

Fig. A.1. Ensemble Three Models (BR(G+B+IB)) for Next-basket Recommendation Based on Un_In Group

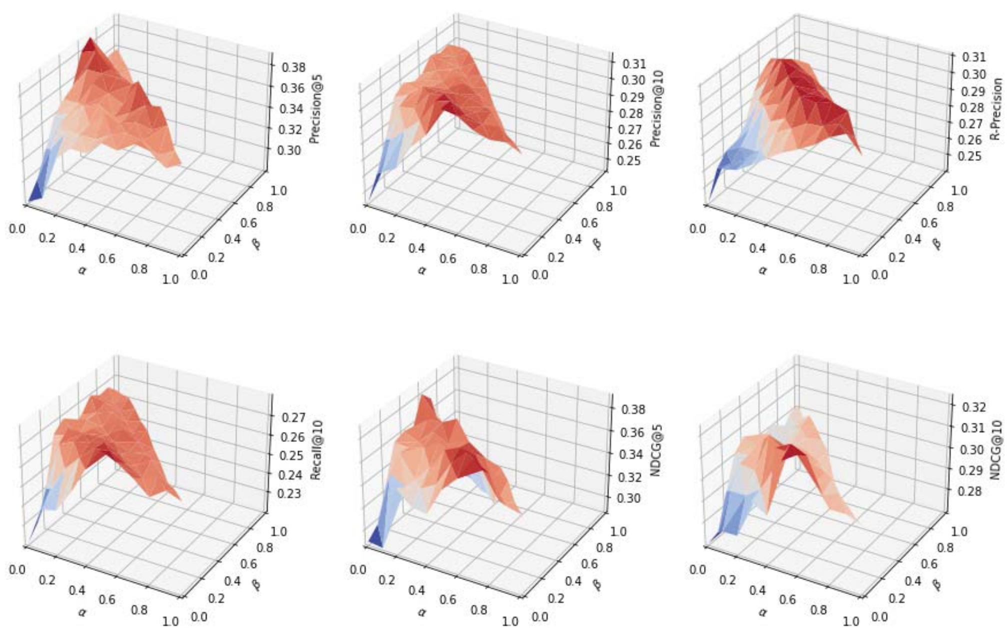

Fig. A.2. Ensemble Three Models (BR(G+B+IB)) for Next-basket Recommendation Based on Un_De Group

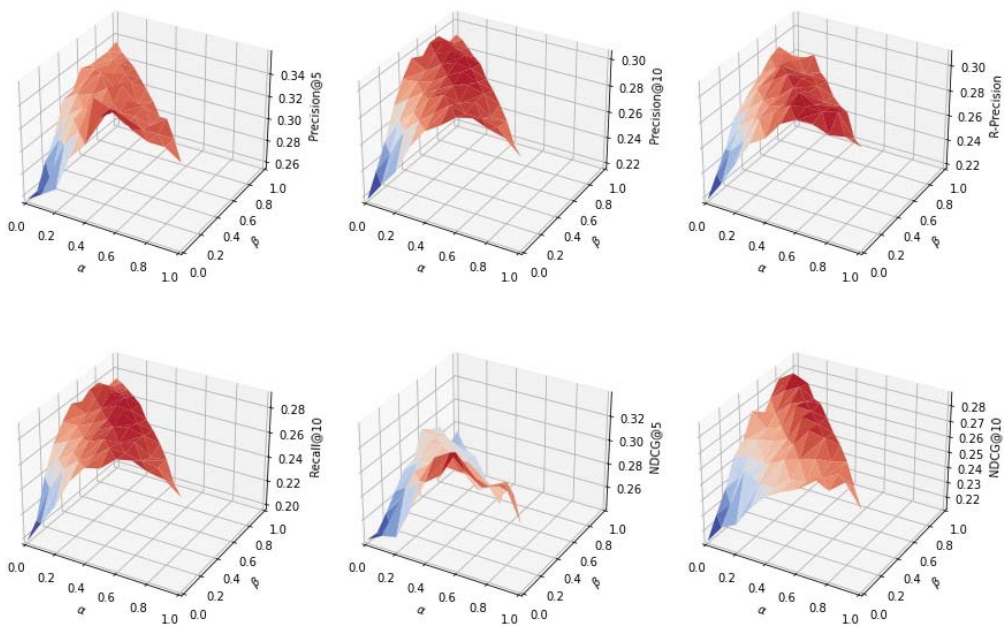

Fig. A.3. Ensemble Three Models (BR(G+B+IB)) for Next-basket Recommendation Based on Un_Ma Group

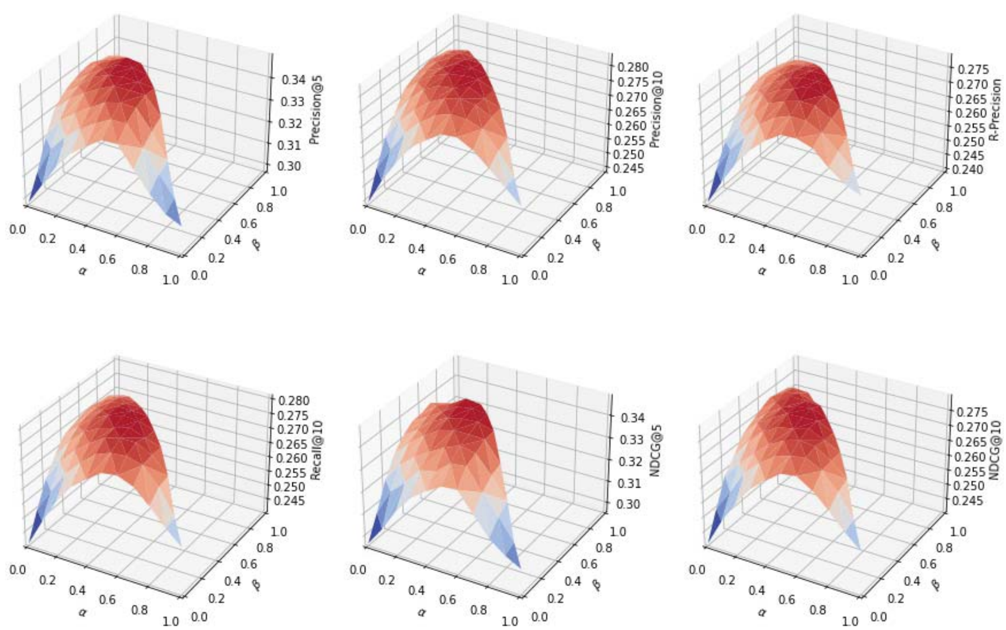

Fig. A.4. Ensemble Three Models (BR(G+B+IB)) for Next-basket Recommendation Based on No_In Group

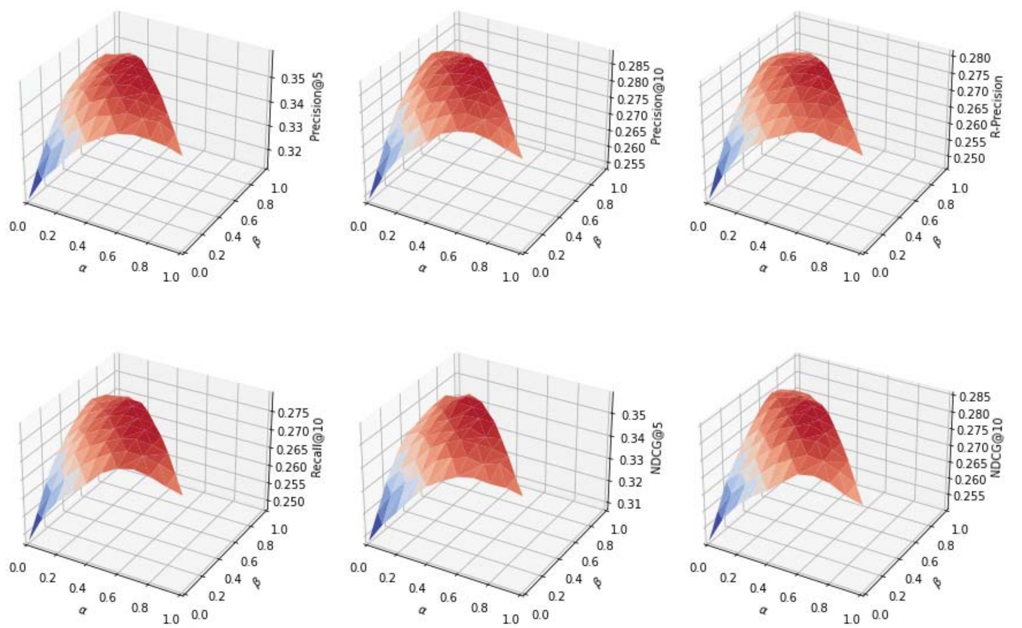

Fig. A.5. Ensemble Three Models (BR(G+B+IB)) for Next-basket Recommendation Based on No_De Group

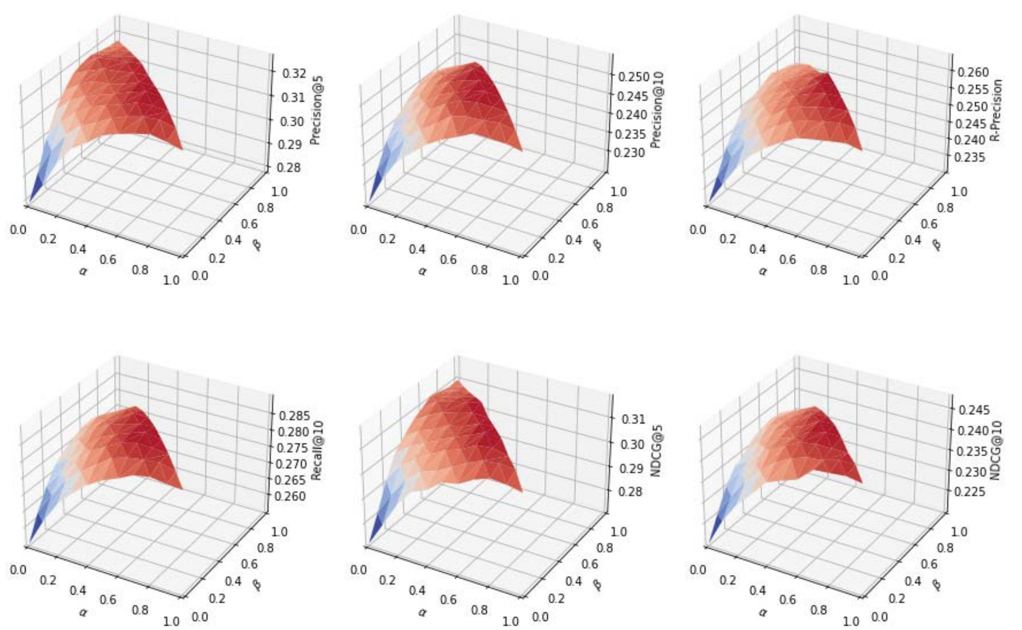

Fig. A.6. Ensemble Three Models (BR(G+B+IB)) for Next-basket Recommendation Based on No_Ma Group

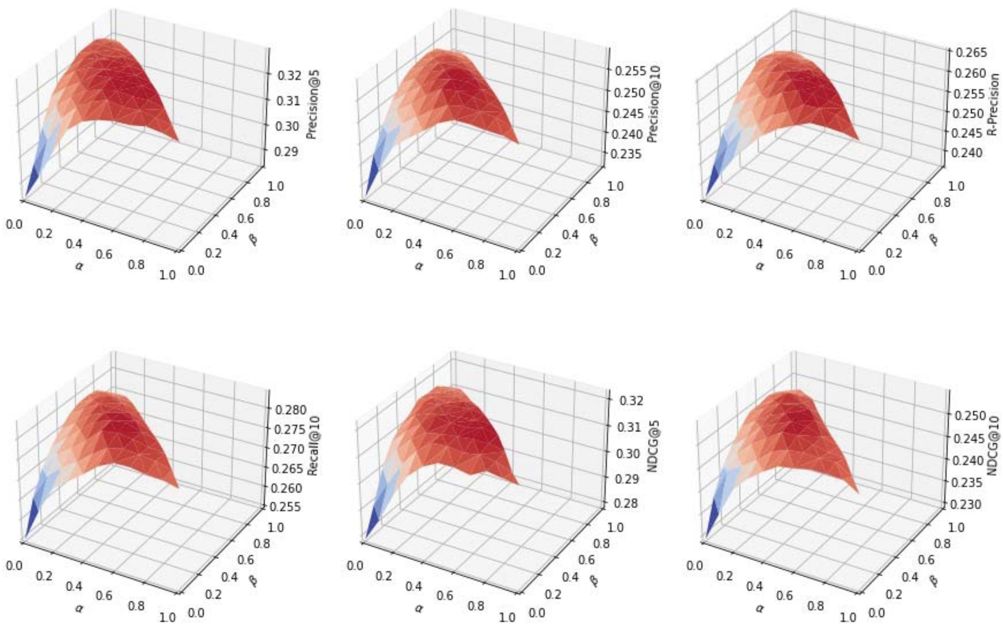

Fig. A.7. Ensemble Three Models (BR(G+B+IB)) for Next-basket Recommendation Based on Ov_Ma Group

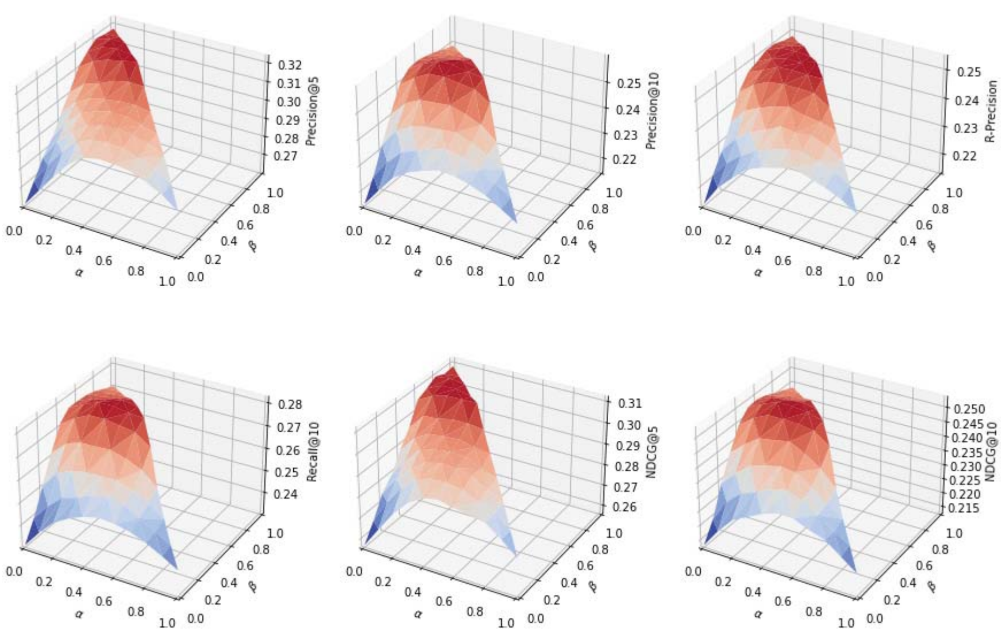

Fig. A.8. Ensemble Three Models (BR(G+B+IB)) for Next-basket Recommendation Based on Ob_In Group

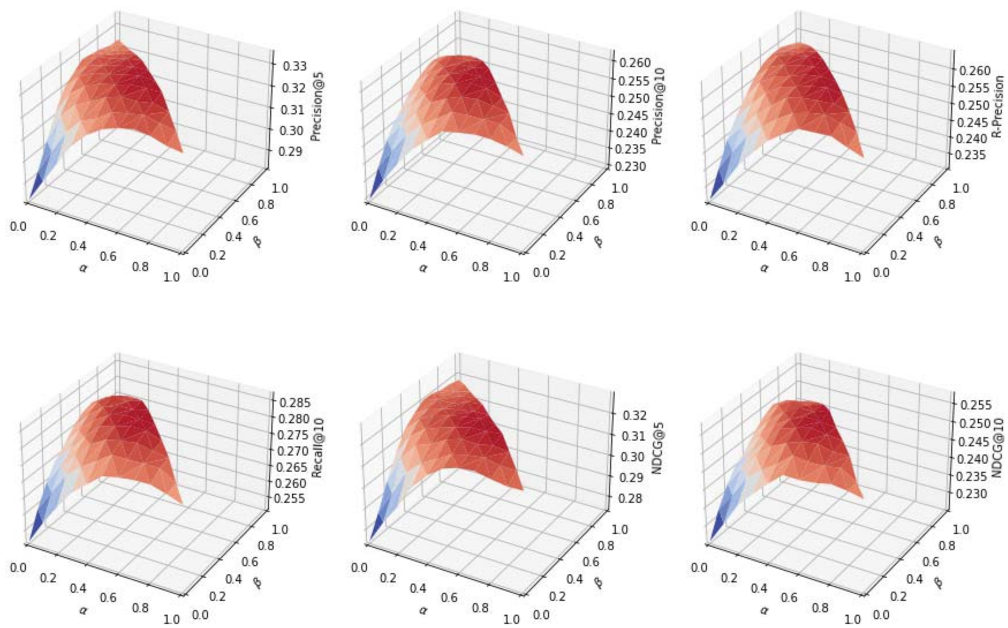

Fig. A.9. Ensemble Three Models (BR(G+B+IB)) for Next-basket Recommendation Based on Ob_De Group

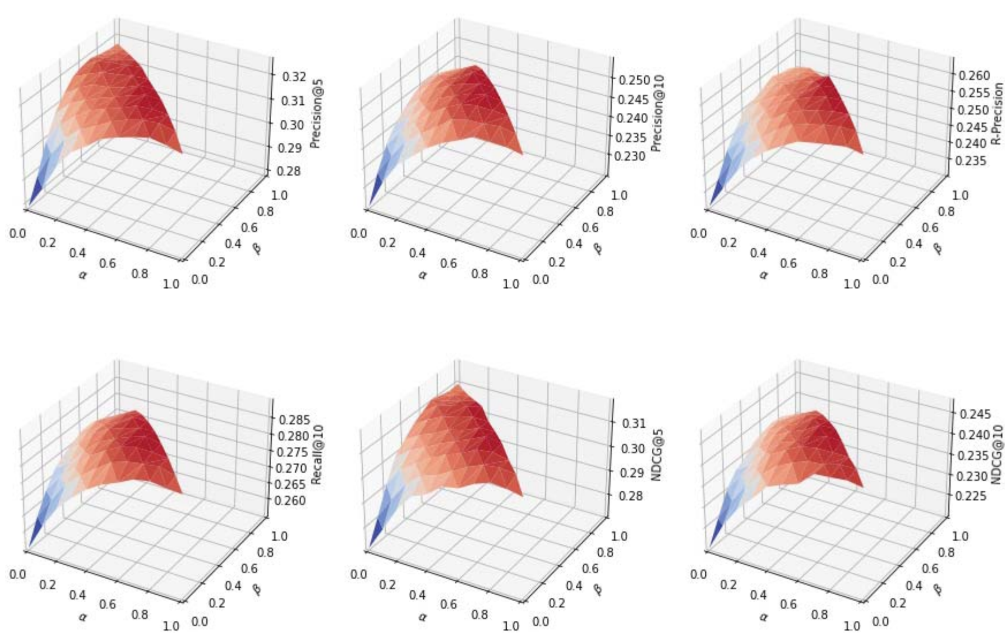

Fig. A.10. Ensemble Three Models (BR(G+B+IB)) for Next-basket Recommendation Based on Ob_Ma Group