Simulating Personal Food Consumption Patterns Using a Modified Markov Chain

Xinyue Pan
Purdue University
West Lafayette, Indiana, United States
pan161@purdue.edu

Jiangpeng He
Purdue University
West Lafayette, United States
he416@purdue.edu

Andrew Peng
Purdue University
West Lafayette, United States
andrew.w.peng@gmail.com

Fengqing Zhu
Purdue University
West Lafayette, United States
zhu0@purdue.edu

ABSTRACT

Food image classification serves as the foundation of image-based dietary assessment to predict food categories. Since there are many different food classes in real life, conventional models cannot achieve sufficiently high accuracy. Personalized classifiers aim to largely improve the accuracy of food image classification for each individual. However, a lack of public personal food consumption data proves to be a challenge for training such models. To address this issue, we propose a novel framework to simulate personal food consumption data patterns, leveraging the use of a modified Markov chain model and self-supervised learning. Our method is capable of creating an accurate future data pattern from a limited amount of initial data, and our simulated data patterns can be closely correlated with the initial data pattern. Furthermore, we use Dynamic Time Warping distance and Kullback-Leibler divergence as metrics to evaluate the effectiveness of our method on the public Food-101 dataset. Our experimental results demonstrate promising performance compared with random simulation and the original Markov chain method.

CCS CONCEPTS

• **Computing methodologies** → *Discrete-event simulation*; **Markov decision processes**; • **Applied computing** → Health informatics.

KEYWORDS

Food Image Classification, Personalized Classifier, Image Clustering

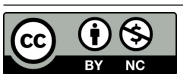
ACM Reference Format:
Xinyue Pan, Jiangpeng He, Andrew Peng, and Fengqing Zhu. 2022. Simulating Personal Food Consumption Patterns Using a Modified Markov Chain. In *Proceedings of the 7th International Workshop on Multimedia Assisted Dietary Management (MADiMa '22), Oct. 10, 2022, Lisboa, Portugal.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3552484.3555747

1 INTRODUCTION

Image-based methods have been developed to provide timely feedback on an individual's dietary intake [22] with reduced user effort compared to traditional self-reported methods [3]. Classification of food images is typically the first and most fundamental step in automated image-based food analysis [8, 10]. Most existing works focus on designing methods to improve the accuracy of food classification using static food image datasets [16–20, 23]. However, static datasets such as Food-101[2] or VireoFood-172[4] are limited to training fixed classifiers, which may not be suitable for real-life scenarios because each person has their unique food consumption patterns. In addition, accurate classification of food images is challenging due to the intra-class diversity and inter-class similarity. Food images may have diverse appearances for the same food class due to different cooking styles, and different food classes may have a similar visual appearance. To improve the accuracy of food image classification and tailor it to an individual food consumption pattern, certain researchers have designed personalized classifiers [14]. Instead of classifying images on static datasets, a personalized classifier uses food images based on a personal food consumption pattern.

Currently, there are few works that focus on designing a personalized classifier for food images. One of the main challenges is the lack of publicly available personal eating datasets. Q. Yu *et al.* designed a personalized classifier by building a personal database for each person incrementally and comparing the similarity of images at each time step with images in each food class, combined with a time-dependent model to predict the food image at each time step [27]. However, it took around 2 years to collect sufficient eating data from each person to train their model, which is time-consuming and difficult to generalize. Therefore, how to efficiently simulate personal food eating data patterns remains an open problem. A simulated data pattern that can closely mimic real life scenario would be essential for developing personalized food classification because the simulated data pattern can help train and test the personalized classifier without collecting large amounts of eating data from different people. In this paper, we focus on simulating food consumption patterns that correlates well with the initial data pattern and can be used for training and testing a personalized classifier. To our best knowledge, we are the first to simulate food consumption patterns that can be used for training personalized classifiers.

In this work, we propose a novel method to simulate personal food consumption patterns. Our method builds a food consumption data pattern of any length from an initial data pattern, which



MADiMa '22, October 10, 2022, Lisboa, Portugal

ACM ISBN 978-1-4503-9502-1/22/10.
https://doi.org/10.1145/3552484.3555747

generally consists of one to two weeks of eating data. Compared to collecting real-life personal data, our method is much more efficient while allowing the simulated data pattern to correlate well with the initial data pattern. Specifically, we leverage a Markov chain model to predict the occurrence of food in the future. The Markov chain model has the advantage of not requiring a large amount of data to train. Moreover, the logic behind the Markov chain model is applicable in our scenario because it uses conditional probability given what foods were eaten in the past, we simulate what will be eaten in the future. We modified the original model to allow more flexibility in predicting what a person tends to eat to mimic real-life scenarios.

The contribution of our work is summarized as follows.

- We modify the Markov chain model to simulate food consumption patterns that can be used for personalized classifiers, considering the case of eating foods not appeared in the initially provided pattern.
- We propose the use of Dynamic Time Warping distance and Kullback-Leibler divergence to show the success of simulated food consumption patterns.
- We sample images for the food consumption pattern by building a normal distribution based on the visual similarity clustering for each class and personal preference.

2 RELATED WORK

2.1 Food Image Classification

Many different models of food image classification have already been published. P. Ma *et al.* compiled a food image dataset named ChinaMartFood-109 with nutrition information [16]. They tried different network architectures on this dataset, such as VGG [25], ResNet [13], Wide ResNet [28] , and InceptionV3 [26], to train the classifier and compare the classification accuracy. They found that InceptionV3 obtained the best food recognition accuracy. In another work [19], P. McAllister *et al.* found that Resnet-152 features provide better generalization for food image classification on popular datasets such as Food 5K, Food-11, RawFooT-DB [6], and Food-101 [2]. S. Phiphiphatphaisit and O. Surinta [20] also applied modified MobileNet architecture to improve the food image classification accuracy and reduce computational time. Besides, recently the food image classification has been studied under a more realistic continual learning scenario where new foods come sequentially overtime [9, 11, 12]. However, none of existing work can tailor to individual food consumption patterns, which can help provide more accurate dietary assessment results.

2.2 Personalized Classifier

S. Horiguch *et al.* proposed a dataset named FLD that compiled 1.5 million images of eating data from 20,000 people over two years [14] . It trained a fixed class classifier on static food datasets first and then applied the classifier to the collected eating data to extract features for each image at each time step. Using the nearest class mean method, a personalized classifier can be constructed. Q. Yu *et al.* improved the classification method that was originally proposed in [14] by applying a time-dependent food distribution model [27]. However, the data collection is time-consuming and the dataset is not open to the public.

3 SYSTEM OVERVIEW

The objective of this work is to simulate personalized food consumption patterns by using (1) initial data pattern, and (2) static food image datasets as the input. The initial data pattern is provided by the user, which contains what a person eats in the past few days. Allowing the user to provide a short initial data pattern ensures that the simulation is not done randomly and there are some correlations within the simulated pattern. In addition, it is easier to collect short-term food records from each person than a long-term ones. The overview of our proposed food consumption pattern simulation system is shown in Figure 1, which includes simulation based on modified Markov chain method that will be described in Section4.1 and the image sampling in Section 4.3. The output of our system is the personalized food consumption pattern that simulates what a person eats every day.

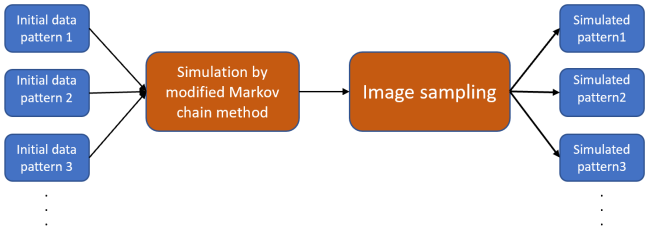


Figure 1: System overview of the simulation

The simulation using the modified Markov chain method is introduced in section 4.1.2. The purpose is to simulate a food consumption pattern that has some correlations with the initial data pattern so that the simulated data pattern is learnable by the personalized classifier. Since a person may eat new types of food from time to time, we can choose to incorporate new food classes that have not appeared in the initial data pattern, which is described in Section 4.2.

We introduce the image sampling process in Section 4.3. After the simulated data pattern is generated, which contains food types that a person may eat, we need to sample food images for each food type to complete the food image data simulation. We leverage visual similarity clustering and cluster sampling, which are described in Section 4.3.1 and section 4.3.2, respectively. The purpose is to mimic the scenario that a person typically prefers certain cooking styles for each type of food so that the simulated food consumption pattern is more realistic.

4 METHODS FOR SIMULATING FOOD CONSUMPTION PATTERN

We propose a new framework to simulate food consumption patterns of any length that closely mimics a provided initial data pattern while allowing flexibility for generalization. We assume an initial food consumption pattern is available for each individual. We then extend this data pattern by building a modified Markov chain model, which does not require a large amount of training data. A Markov chain model is suitable in our scenario since the simulation of the data pattern is based on what was eaten in the past. However, the original Markov chain method can result large

difference in the probability of food appearing between the simulated data pattern and the initial data pattern. This is because it tends to forget the probability distribution of the original data pattern in the later part of the simulation. Therefore, we propose a modified Markov chain model. We also sample images for food classes that appeared in food consumption patterns via a clustering method. Figure 2 describes the overall pipeline of our method. The user first provides an initial food consumption pattern, then we simulate the food consumption pattern by extending this initial data pattern using a modified Markov chain method. Finally, we sample images of each food class that appeared in the simulated data pattern.

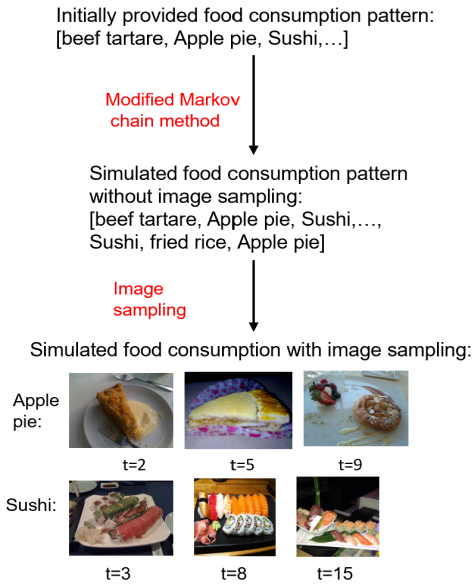


Figure 2: Pipeline for simulating an eating data pattern

4.1 Modified Markov Chain Method to Simulate food consumption Pattern

4.1.1 Original Markov Chain Method. T. Almutiri and F. Nadeem presented a survey of example applications that use the Markov chain method [1]. The Markov chain method [7] uses conditional probability to simulate the next state from only the previous state. It can be represented in the following equation 1

$$P(S_t = A|S_1 ... S_{t-1}) = P(S_t = A|S_{t-1}), \tag{1}$$

where S_t represents the predicted state at time t and S_{t-1} represents the current state at time $t-1$. There is also a decision array that represents the probability of each state that can happen in the current time, as shown in equation 2

$$P_{decision} = [P_A, P_B, P_C, P_D, ...], \tag{2}$$

where P_A represents the probability of state A in the current unit of time and the same for P_B, P_C, etc. To predict the probability of the next state, we also need a transition matrix, which represents the conditional probability of transitioning from one state to another in different combinations:

$$P_{trans} = \begin{bmatrix} P_{A|A} & P_{B|A} & ... \\ P_{A|B} & P_{B|B} & ... \\ ... & ... & ... \end{bmatrix}, \tag{3}$$

where $P_{A|B}$ represents the probability of transitioning from state B to state A. Each row will be normalized such that the summation of each row equals 1. Then, the probability of predicting different states at the next unit of time is calculated using equation 4

$$P_{new\_decision} = P_{old\_decision} * P_{trans}. \tag{4}$$

The highest probability element in the decision array is then selected as the prediction.

4.1.2 Food consumption Pattern Simulation using modified Markov chain method. Figure 3 shows our modified Markov chain method for simulating food consumption patterns. The old decision array represents the probability of each food class in the last time step. The new decision array represents the probability of each food class in the current time step after multiplying the old decision array with the transition matrix. The updated decision array represents the updated probability of each food class in the current step after handling special circumstances during the simulation of food consumption patterns. Given an initial food consumption pattern, we use the Markov chain method to simulate a subsequent data pattern. However, the original Markov chain method results in repetitive occurrences of the same food class, which is unrealistic. This is expected because the goal of a Markov chain method is to predict the state a data pattern converges to. We also noticed a few other issues with the Markov chain method. For example, if multiple food classes could have the same highest probability, the first one is always chosen by default. There are also sometimes zero rows in the transition matrix if the last food class in the initial data pattern only appears once, which means that there is no data to construct conditional probability for this particular class. To address these issues, we modified the original Markov chain method to lower the probability of simulating repetitive data patterns and make decisions less biased. Our modifications include:

- If there are repetitive occurrences of one food class when simulating the food consumption pattern, we reinitialize the decision array randomly by assigning a food class with a probability of 1 in the decision array.
- If two or more food classes share the highest probability, we randomly select a food class from the classes sharing the highest probability and reinitialize the decision array.
- For any zero rows in the transition matrix, we replace the zero row with the average probabilities calculated from other non-zero rows.

4.2 Incorporating New food Classes

It is highly probable that the initial food consumption pattern does not include all the foods a person may eat, or that new foods may be consumed over time. Therefore, our method incorporates new food classes as follows.

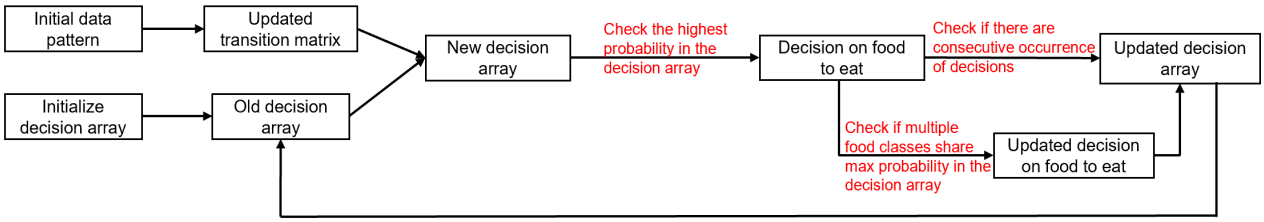


Figure 3: Modified Markov chain method for simulating data pattern

4.2.1 Adding New Food Class. We first need to decide how to define a new food class. At each time step, there is a probability that new food is consumed. Therefore, we build a probability model to estimate the likelihood of eating a new food at each time step, as represented in equation 5

$$P_{new} = \frac{1}{c_{new}+1} * \left(\frac{1}{c_{total}+1}\right)^{\frac{1}{x_t - x_{t-1}+1}} \quad (5)$$

where c_{new} represents the total number of new classes defined after the initially provided data pattern, c_{total} represents the total number of food classes in the data pattern, x_t represents the current time index, and x_{t-1} represents the last time index when a new food class is defined. The idea behind this equation is that the longer a person eats the old foods since the last time they ate new food, the more likely they are to eat another new food, and vice versa. If the probability of adding a new food class is larger than the probability of eating an existing food, a new food class will be added to the data pattern instead of inferring what food to eat from the modified Markov chain method.

4.2.2 Expansion Of Transition Matrix And Decision Array. When we define a new food class, we add one row and one column to the transition matrix. The simplest way to fill the additional row or column is to assign random values to each element. However, this could lead to a higher probability for the newly added entries during the update of the decision array. When a decision array multiplies with the transition matrix, each element in the decision matrix corresponds to the dot product between the decision array and the corresponding column in the transition matrix. If the corresponding column does not have any sparsity (defined as the proportion of zeroes), the element will have a larger value, which means a higher probability to be selected in the updated decision array. To address this issue, we randomly add sparsity in the additional column to keep the sparsity close to that of the original transition matrix. This would ensure that the decision at each time step is always alternating between new food classes and existing food classes.

4.3 Image Sampling For simulated Food consumption Pattern

After obtaining the food consumption pattern, we can sample images from existing food datasets such as Food-101 [2] to simulate personalized consumption data. As indicated in [14, 27], one of the characteristics of personalized data is that foods from the same class are more visually similar, as each person may prefer certain cooking styles for the same foods. This motivates us to perform visual similarity clustering within each food class first, and then sample images in the same cluster to make the data pattern more realistic.

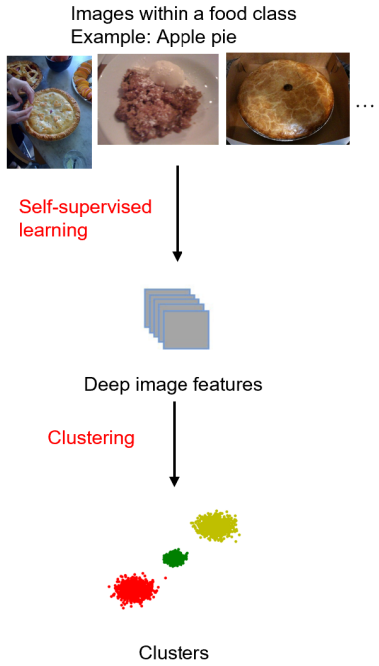


Figure 4: Pipeline for visual similarity clustering

4.3.1 Visual Similarity Clustering. To cluster visually similar images within each food class, we first need to learn the discriminative features for each food image. Figure 4 shows the pipeline for visual similarity clustering. The first step is performing image feature extraction and the second step is clustering images based on these extracted features. To extract discriminative image features, we propose to apply self-supervised learning techniques to learn the visual representation of food images with static datasets. Our pipeline can work with any existing self-supervised approaches. In this work, we apply SimSiam [5] as an example to illustrate our method.

SimSiam [5] learns the visual representation by passing two different augmented views of an input image into an encoder network and through a projection MLP head. Finally, it will pass through the

prediction MLP to maximize the agreement with the other augmentation. SimSiam is trained on minimizing negative cosine similarity between the prediction output $p(\cdot)$ on one path and a stop gradient of the projection output $stop\_grad(h(\cdot))$ on the other. The same process will be repeated for the other pair of paths. The two different negative cosine similarities sum together to form the total loss. Here, the stop gradient is a critical part to avoid a collapsing solution. The whole process can be formulated as equation 6.

$$Loss = \frac{1}{2}D(p_1(\cdot), stop\_grad(h_2(\cdot)) + \frac{1}{2}D(p_2(\cdot), stop\_grad(h_1(\cdot)) \tag{6}$$

where D refers to the negative cosine similarity between two different vectors.

Clustering: After self-supervised learning, we apply the Power Iteration Clustering (PIC) [15] as our clustering approach, which is a graph based method. One advantage of PIC is that the number of simulated clusters is not predefined, so there will be more clusters if the food class has higher intra-class variations and vice versa. Given n_c images for one food class c, we first simulate the nearest neighbor graph by connecting their 10 neighbor data points in the Euclidean space using extracted feature embeddings. Let $f(\mathbf{x}_i)$ denote the extracted feature for the i-th image. The sparse graph matrix $G = \mathbb{R}^{n_c \times n_c}$ with zeros on the diagonal and the remaining elements of G are defined by equation 7

$$e_{i,j} = exp^{-\frac{||f(\mathbf{x}_i) - f(\mathbf{x}_j)||^2}{\sigma^2}} \tag{7}$$

where σ denotes the bandwidth parameter and we empirically use $\sigma = 0.5$ in this work. Then, we initialize a starting vector $s^{n_c \times 1} = [\frac{1}{n_c}, ..., \frac{1}{n_c}]^T$ and iteratively update it using Equation 8

$$s = L_1(\alpha(G + G^t)s + (1 - \alpha)s) \tag{8}$$

where $\alpha = 0.001$ refers to a regularization parameter and $L_1(\cdot)$ denotes the L-1 normalization step. The simulated clusters are given by the connected components of a directed, unweighted subgraph of G denoted as $\tilde{G}$. We set $\tilde{G}_{i,j} = 1$ if $j = argmax_j e_{i,j}(s_j - s_i)$ where s_i refers to the i-th element of the vector. Note that no edge starts from i if $\{\forall j \neq i, s_j \leq s_i\}$, *i.e.* s_i is a local maximum.

4.3.2 Cluster Sampling. Given the simulated clusters for each food class, we can start sampling images to simulate the personalized food consumption patterns. Based on the obtained food consumption pattern as illustrated in Section 4.1, we know the appearance frequency for each food class, which allows us to learn the number of images we need to sample for each food class. In addition, we learn the user's preferred cooking style based on their provided initial food consumption pattern. Specifically, we calculate the standard deviation σ of the initial food consumption pattern, which is used to construct a Gaussian distribution with the same σ where a high σ indicates that the person prefers a diverse range of cooking styles while a low σ indicates that the person prefers a small range of cooking styles. Finally, given the appearance frequency and the constructed Gaussian distribution, we sample the appropriate number of food images from each cluster and randomly place them in the food consumption pattern to simulate the personalized food consumption pattern.

5 EXPERIMENTS

In this section, we conduct experiments to show that our modified Markov chain method can improve the quality of the simulated data pattern compared to the original Markov chain method or random simulation method. If the simulated data pattern correlates better with the initially provided data pattern, we know that the simulated data pattern is of higher quality. First, we manually select foods from the Food-101 dataset [2] as the initial data pattern. The image sampling process is also conducted on the Food-101 dataset [2]. We use Dynamic Time Warping distance [21] and Kullback-Leibler Divergence [24] as our evaluation metrics. These two metrics can evaluate the correlation between the simulated pattern and the initial data pattern. Note that since there is no label for the future eating pattern, we cannot use the prediction accuracy directly as our metric. In addition, the goal of the simulation is not to make correct prediction. Instead, we want to simulate a food consumption pattern that correlates well with the initial data pattern so that a personalized classifier can learn from the data distribution. We use different lengths of the initially provided food consumption pattern in the experiments to see how length may affect performance. We also show examples of image sampling results on simulated food consumption for qualitative evaluation.

5.1 Evaluation Metric

5.1.1 Dynamic Time Warping. To evaluate how well the extended data pattern infers from the initially provided data pattern, we use Dynamic Time Warping (DTW) [21] distance between the extended data pattern and the initially provided data pattern. Unlike Euclidean distance, which compares two data patterns by aligning only the corresponding points, DTW compares two data patterns by aligning one point in one data pattern to multiple points in another data pattern, which can better correlate the two data patterns. In general, DTW measures how well one sequence can follow the pattern of another sequence.

In our case, DTW only needs to be applied on the Hamming distance since we just want to see whether the inference is correct or not instead of the degree of correctness. To calculate the DTW distance between time series A and B with lengths m and n respectively, we construct an $m \times n$ matrix. Each element of the matrix represents the distance between A and B at (i, j). The distance is represented by $dis(A_i, B_j) = hamming(A_i, B_j)$. After constructing the matrix, we find the path with a minimum distance from $(0, 0)$ to (m, n) in the matrix. The cumulative distance is calculated using dynamic programming and can be represented by Equation 9:

$$\begin{aligned} D_{dtw}(i, j) = dis(A_i, B_j) + min\{&D_{dtw}(i-1, j-1), \\ &D_{dtw}(i, j-1), \\ &D_{dtw}(i-1, j)\} \end{aligned} \tag{9}$$

where D_{dtw} represents the cumulative DTW distance at (i,j). A lower DTW distance indicates a better correlation between the simulated data pattern and the initial data pattern. However, only using DTW distance as our metric is not enough to confirm that the simulated food consumption pattern has been well inferred from the initially provided food consumption pattern. For example, The DTW distance may be small when a certain food consecutively

appears multiple times in the simulated food consumption pattern since these consecutive decisions will only match one decision point in the initial data pattern and can cause a zero distance at that point, which makes total DTW distance very small. However, the simulated data pattern is likely not realistic. This is particularly true for the original Markov chain method. Therefore, we need another metric to compare the performance of the original Markov Chain method and our modified version.

5.1.2 Kullback-Leibler Divergence. Kullback-Leibler(KL) Divergence [24] is used to measure the degree of difference between two probability distributions. In our case, it measures the probability of each food appearing in the initial food consumption pattern and the simulated food consumption pattern and compares the difference between the two distributions. It is formulated as in Equation 10

$$D_{KL}(p,q) = \sum_{i=1}^{n} p_i log(\frac{p_i}{q_i}) \tag{10}$$

where p_i represents the probability of a food class appearing in the simulated food consumption pattern, q_i represents the probability of a food class appearing in the initial food consumption pattern, and n represents the number of food classes in the simulated food consumption pattern. A lower KL divergence indicates better probability distribution matching of food appearance between the simulated food consumption pattern and the initial food consumption pattern.

5.2 Experiment Setup and Results

Since our goal is to simulate a food consumption pattern that can be used for training a personalized classifier, the simulated data pattern needs to have some correlations with the initial data pattern. To check whether our simulated data pattern is successful, we randomly simulate a data pattern that has the same number of food classes as the data pattern simulated by the modified Markov Chain method. Then we compare the random simulation method with our modified Markov Chain method using DTW distance. We calculate the DTW distance and KL divergence on simulated food consumption patterns with different lengths of initial food consumption patterns and methods. We also use KL divergence to show that the modified Markov Chain method is better than the original Markov Chain method.

5.2.1 Datasets. To simulate a data pattern, we manually select food classes from the Food-101 [2] dataset to form initial food consumption patterns. Each initial data pattern can be used to form additional data patterns of different lengths to evaluate the effect of the initial data pattern length. The initial data pattern has a length of [5,10,20,30,40,50], and each length corresponds to 20 initial data patterns, giving us a total of 120 initial data patterns. We experimentally trained 20 sets for each length of the initial food consumption pattern. The image sampling is then evaluated based on the corresponding classes of images from the Food-101 dataset.

5.2.2 Comparison between using the original and our modified Markov chain method. As pointed out in Section 4.1.2, there are some issues with using the original Markov chain method for our purpose. We conduct experiments to see the difference in the simulated food consumption pattern. The sample simulated food consumption is shown in Figure 5. Different colored dots represent food class eaten at each unit of time assuming the simple case where only one type of food is consumed at a time. The first 5 units of time present the initially provided food consumption pattern. We can see the food consumption pattern simulated by the original Markov chain method always samples the same food type in later units of time, which is not realistic in real life. We also observe the food consumption pattern simulated by the modified Markov chain method better mimics the initial food consumption pattern and is more realistic. In Table 1, we use KL divergence to illustrate problems using the original Markov chain. We can see the KL divergence for the original Markov Chain method is large because there is a large difference between the probability of food appearing between the simulated data pattern and the initial data pattern. This issue can be addressed by the modified Markov chain method because if there is any consecutive occurrence of decisions, it will try to correct this bias. In addition, if a person prefers certain foods in the initial data pattern, the simulated food consumption pattern will still keep this preference. We can see from the results that the modified Markov chain method can obtain lower KL divergence, which shows a better matching to the pattern of food appearing in the initial data pattern.

5.2.3 Results Of Data Pattern Simulation. The results of simulating food consumption patterns are shown in Table 1, and Table 2. In these tables, $\boldsymbol{Markov_{orig}}$ means the original Markov chain method without adding new food classes during simulation. $\boldsymbol{Markov_{orig}}$ ***with new*** means the original Markov Chain with added new food classes during simulation. $\boldsymbol{Markov_{ours}}$ means our modified Markov chain method without adding new food classes. $\boldsymbol{Markov_{ours}}$ ***with new*** means modified Markov Chain method with added new food classes during simulation. ***Random*** means random simulation method without adding new classes. ***Random with new*** means random simulation method with added new classes. The bold number in the table means the performance is better than another method.

Results without adding new classes. To show the results of simulating data patterns without adding new food classes on different methods, we ran experiments on 20 different initial data patterns. We then took the average DTW distance (for comparison between the random simulation method and modified Markov chain method) and average KL divergence(for comparison between the original Markov chain method and modified Markov chain method). We conduct experiments using initial data pattern lengths of [5,10,20,30,40,50]. In the second and third columns of Table 1, we observe higher KL divergence of the original Markov Chain method. Worse performance in the original Markov chain method is due to the unbalanced likelihood of food appearing in the simulated food consumption pattern. Therefore, the original Markov Chain method is not suitable to simulate the food consumption pattern. In the second and third columns of Table 2, we observe higher DTW distance of the random simulation method. Worse performance in the random simulation method is because the simulated pattern has almost no correlation with the initially provided pattern. The simulation does not follow the pattern in the initial data pattern. Finally,

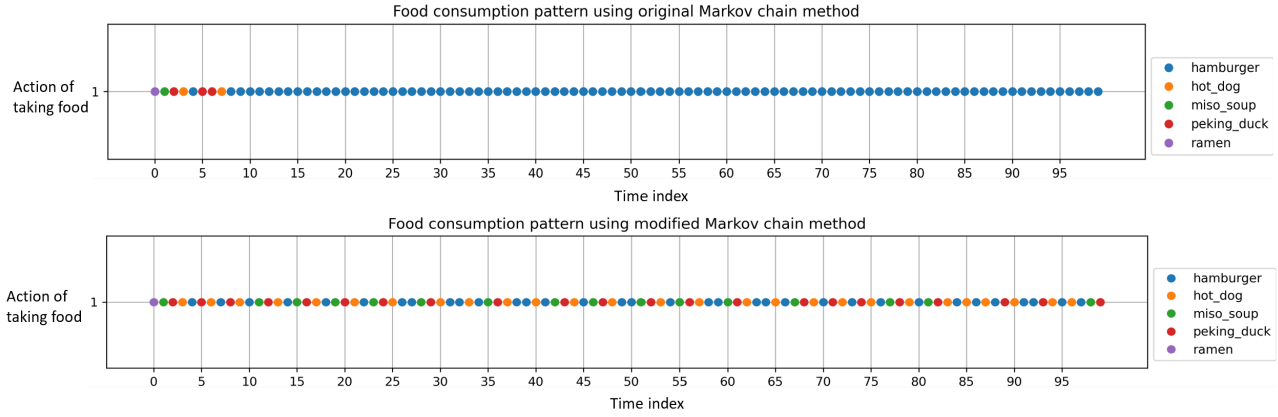


Figure 5: Comparison of food consumption pattern using original and modified Markov chain method

Length of initial pattern	$Markov_{orig}$	$Markov_{ours}$	$Markov_{orig}$ with new	$Markov_{ours}$ with new
5	1.9 ± 0.158	**0.531±0.031**	1.371 ± 0.28	**0.551±0.12**
10	1.35 ± 0.54	**0.237±0.176**	0.957 ± 0.371	**0.382±0.16**
20	1.07 ± 0.34	**0.0756±0.04**	0.78 ± 0.135	**0.301±0.09**
30	0.903 ± 0.251	**0.085±0.043**	0.649 ± 0.119	**0.269±0.076**
40	0.758 ± 0.171	**0.123±0.061**	0.566 ± 0.073	**0.282±0.093**
50	0.652 ± 0.114	**0.138±0.069**	0.503 ± 0.065	**0.238±0.06**

Table 1: Comparison between the original Markov chain method and the modified Markov chain method using *KL Divergence* ± *standard deviation* **between the initial provided food consumption pattern and the simulated food consumption pattern (smaller value indicates better performance). Second and third columns show the results without adding new classes. The last two columns show the results with added new classes.** $Markov_{orig}$ **means original Markov chain method,** $Markov_{ours}$ **is our modified Markov chain method,** *with new* **indicates with added new food classes**

Length of initial data pattern	*Random*	$Markov_{ours}$	*Random* with new	$Markov_{ours}$ with new
5	66.3 ± 2.74	**55.6±2.21**	79.6 ± 3.67	**57.15±14.79**
10	69.2 ± 2.64	**64.95±7.85**	76.4 ± 4.95	**63±7.42**
20	67.75 ± 3.73	**56.7±2.85**	73.5 ± 2.78	**59.55±5.32**
30	68.4 ± 3.7	**49.2±5.62**	73.95 ± 2.64	**58.1±6.54**
40	68.25 ± 3.04	**50.5±4.96**	72.85 ± 3.41	**59.8±6.15**
50	71.6 ± 2.5	**54.75±5.52**	75.1 ± 3.48	**62.7±4.27**

Table 2: Comparison between the original Markov chain method and the modified Markov chain method using *DTW Distance* ± *standard deviation* **between the initially provided data pattern and the simulated data pattern with adding new food classes (smaller value indicates better performance). Second and third columns show the results without adding new classes. The last two columns show the results with added new classes.** *Random* **means Random simulation method,** $Markov_{ours}$ **is our modified Markov chain method,** *with new* **indicates with added new food classes**

the modified Markov chain method can obtain a good performance when compared with both the original Markov chain method and random simulation method. The modified Markov chain method can simulate a pattern that has some correlation with the initial food consumption pattern. In addition, if the length of the initial food consumption pattern is large, the modified Markov chain can output competitive results in terms of KL divergence.

Results with adding new classes. We apply the same experiment setup when adding new classes. From the last two columns of Table 1, when new food classes are included in simulating future data patterns, we observe the KL divergence still decreases compared to the original Markov Chain method. From the last two columns of Table 2, we can observe the DTW distance also decreases compared to the random simulation method. Although the

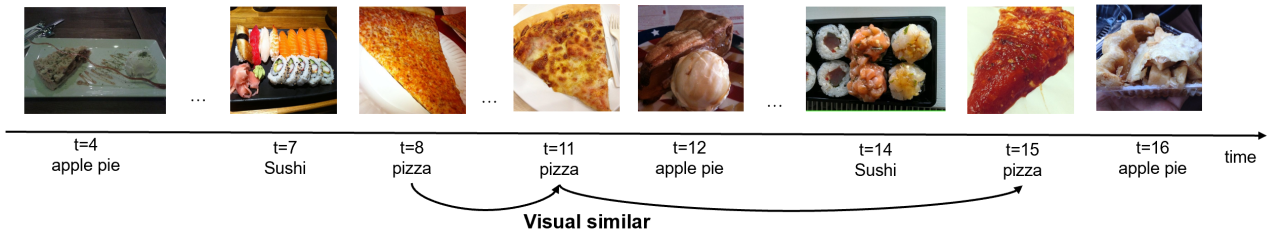


Figure 6: Image sampling over simulated data pattern

resulting values of using both metrics for comparing the methods are increased compared with the case without adding new classes, this is within expectation because in the simulated data pattern there are food classes that are not in the initial data pattern. With longer initial data patterns, the KL divergence is lower for the modified Markov chain method as shown in Table 1. However, the DTW distance does not decrease in this case as shown in Table 2 because the number of new food classes included in the simulated pattern is increasing.

5.2.4 Image sampling over simulated food consumption pattern. Figure 6 shows the sample results of image sampling over a simulated data pattern. We select sample images of apple pie, pizza, and sushi from the Food-101 dataset. We notice that image samples for the same food category appear visually similar in the data pattern at different time steps, which aligns with our assumption that a person typically prefers the same cooking style resulting in a visually similar appearance of the foods over time.

6 CONCLUSION

In this paper, we proposed to simulate food consumption pattern using a modified Markov chain method. Our method can accommodate new foods not included in the initial data pattern and closely mimic user preference of food choices and their occurrence as provided in the initial data pattern. Experimental results show that our proposed method produces realistic data pattern compared to using the original Markov chain method and a random simulation method.

Our future work will focus on improving the simulation of food consumption patterns incorporating other aspects. For example, we can consider the timing of food consumption such as breakfast, lunch, and dinner. We can use a hierarchical classification of food types and use such hierarchy to simulate what someone may eat during a meal. Methods to simulate food consumption patterns efficiently and accurately could benefit the development of personalized classifiers to improve food image classification.

ACKNOWLEDGMENTS

This work was supported by the National Institutes of Health under Grant 1U24CA268228-01.

REFERENCES

[1] Talal Almutiri and Farrukh Nadeem. 2022. Markov Models Applications in Natural Language Processing: A Survey. *International Journal of Information Technology and Computer Science(IJITCS)* 14, 2 (Apr 2022), 1–16. https://doi.org/10.5815/ijitcs.2022.02.01

[2] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. 2014. Food-101 – mining discriminative com- ponents with random forests. In *Proc. Eur. Conf. Comput. Vis.* 446–461.

[3] C. J. Boushey, M. Spoden, F. M. Zhu, E. J. Delp, and D. A. Kerr. 2017. New mobile methods for dietary assessment: review of image-assisted and image-based dietary assessment methods. *Proceedings of the Nutrition Society* 76, 3 (aug 2017), 283–294.

[4] Jingjing Chen and Chong wah Ngo. 2016. Deep-based Ingredient Recognition for Cooking Recipe Retrieval. In *Proceedings of the 24th ACM international conference on Multimedia (MM'16)*. Association for Computing Machinery, New York, NY, 31–41.

[5] Xinlei Chen and Kaiming He. 2021. Exploring Simple Siamese Representation Learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 15745–15753. https://doi.org/10.1109/CVPR46437.2021.01549

[6] Claudio Cusano, Paolo Napoletano, and Raimondo Schettini. 2015. Evaluating color texture descriptors under large variations of controlled lighting conditions. *Journal of the Optical Society of America A* 33, 1 (dec 2015), 17. https://doi.org/10.1364/josaa.33.000017

[7] Richard M. Feldman and Ciriaco Valdez-Flores. 2010. *Applied Probability and Stochastic Processes*. Springer Berlin and Heidelberg.

[8] Jiangpeng He, Runyu Mao, Zeman Shao, Janine L. Wright, Deborah A. Kerr, Carol J. Boushey, and Fengqing Zhu. 2021. An End-to-End Food Image Analysis System. *Electronic Imaging* 2021, 8 (2021), 285–1–285–7. https://doi.org/doi:10.2352/ISSN.2470-1173.2021.8.IMAWM-285"

[9] Jiangpeng He, Runyu Mao, Zeman Shao, and Fengqing Zhu. 2020. Incremental Learning In Online Scenario. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2020), 13926–13935.

[10] Jiangpeng He, Zeman Shao, Janine Wright, Deborah Kerr, Carol Boushey, and Fengqing Zhu. 2020. Multi-Task Image-Based Dietary Assessment for Food Recognition and Portion Size Estimation. *arXiv preprint arXiv:2004.13188* (2020). arXiv:2004.13188 [cs.CV]

[11] Jiangpeng He and Fengqing Zhu. 2021. Online Continual Learning for Visual Food Classification. *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops* (October 2021), 2337–2346.

[12] Jiangpeng He and Fengqing Zhu. 2022. Exemplar-free Online Continual Learning. *arXiv* (2022). https://doi.org/10.48550/ARXIV.2202.05491

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. https://doi.org/10.48550/ARXIV.1512.03385

[14] Shota Horiguchi, Sosuke Amano, Makoto Ogawa, and Kiyoharu Aizawa. 2018. Personalized Classifier for Food Image Recognition. *IEEE Transactions on Multimedia* 20, 10 (2018), 2836–2848. https://doi.org/10.1109/TMM.2018.2814339

[15] Frank Lin and William W. Cohen. 2010. Power Iteration Clustering. In *Proceedings of the 27th International Conference on International Conference on Machine Learning* (Haifa, Israel) *(ICML'10)*. Omnipress, Madison, WI, USA, 655–662.

[16] Peihua Ma, Chun Pong Lau, Ning Yu, An Li, and Jiping Sheng. 2022. Application of deep learning for image-based Chinese market food nutrients estimation. *Food Chemistry* 373 (2022), 130994. https://doi.org/10.1016/j.foodchem.2021.130994

[17] Runyu Mao, Jiangpeng He, Luotao Lin, Zeman Shao, Heather A. Eicher-Miller, and Fengqing Zhu. 2021. Improving Dietary Assessment Via Integrated Hierarchy Food Classification. *2021 IEEE 23rd International Workshop on Multimedia Signal Processing (MMSP)* (2021), 1–6. https://doi.org/10.1109/MMSP53017.2021.9733586

[18] Runyu Mao, Jiangpeng He, Zeman Shao, Sri Kalyan Yarlagadda, and Fengqing Zhu. 2020. Visual aware hierarchy based food recognition. *arXiv preprint arXiv:2012.03368* (2020).

[19] Patrick McAllister, Huiru Zheng, Raymond Bond, and Anne Moorhead. 2018. Combining deep residual neural network features with supervised machine learning algorithms to classify diverse food image datasets. *Computers in Biology and Medicine* 95 (2018), 217–233. https://doi.org/10.1016/j.compbiomed.2018.02.008

[20] Sirawan Phiphiphatphaisit and Olarik Surinta. 2020. Food Image Classification with Improved MobileNet Architecture and Data Augmentation. In *Proceedings of the 2020 The 3rd International Conference on Information Science and System*

(Cambridge, United Kingdom) *(ICISS 2020)*. Association for Computing Machinery, New York, NY, USA, 51–56. https://doi.org/10.1145/3388176.3388179

[21] Hiroaki Sakoe and Seibi Chiba. 1978. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 26 (1978), 159–165.

[22] Zeman Shao, Yue Han, Jiangpeng He, Runyu Mao, Janine Wright, Deborah Kerr, Carol Jo Boushey, and Fengqing Zhu. 2021. An Integrated System for Mobile Image-Based Dietary Assessment. *Proceedings of the 3rd Workshop on AIxFood* (2021), 19–23. https://doi.org/10.1145/3475725.3483625

[23] Z. Shao, F. Shao, R. Mao, J. He, J. Wright, D. Kerr, C. Boushey, and F. Zhu. 2021. Towards Learning Food Portion From Monocular Images With Cross-Domain Feature Adaptation. In *Proceedings of the IEEE International Workshop on Multimedia Signal Processing*. Tampere, Finland.

[24] Jonathon Shlens. 2014. Notes on Kullback-Leibler Divergence and Likelihood. https://doi.org/10.48550/ARXIV.1404.2000

[25] Karen Simonyan and Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. https://doi.org/10.48550/ARXIV.1409.1556

[26] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1–9. https://doi.org/10.1109/CVPR.2015.7298594

[27] Qing Yu, Masashi Anzawa, Sosuke Amano, Makoto Ogawa, and Kiyoharu Aizawa. 2018. Food Image Recognition by Personalized Classifier. In *2018 25th IEEE International Conference on Image Processing (ICIP)*. 171–175. https://doi.org/10.1109/ICIP.2018.8451422

[28] Sergey Zagoruyko and Nikos Komodakis. 2016. Wide Residual Networks. https://doi.org/10.48550/ARXIV.1605.07146