AITA
ARTIFICIAL INTELLIGENCE THEORY AND APPLICATIONS

RESEARCH ARTICLE

A New Similarity Method for Tourism Recommendation Systems

Eren Türkel [a], **Adil Alpkoçak** [a†]

[a] Computer Engineering, Dokuz Eylül University, İzmir, Turkey,
[†] eturkel1312@gmail.com , corresponding author

RECEIVED MAY 30, 2023
ACCEPTED AUGUST 10, 2023

CITATION Türkel, E. & Alpkoçak, A. (2023). A New Similarity Method A new similarity method for tourism recommendation systems. *Artificial Intelligence Theory and Applications, 3(*2), 77-91.

Abstract

In this paper, we proposed a new similarity method to use in tourism recommendation systems. Recommendation systems highly depend on the existence of a similarity measure used to identify similar items. In tourism products such as hotels, trips, packages are all hard to judge for their similarity. The proposed method is simply based on user defined weights to calculate similarity. First, we represented each product as a vector and then weighted by user defined scores. Then it uses cosine similarity to measure similarity between items. We evaluated our method using a dataset created by the travel expert. Our experimental results indicate that the proposed method achieves a significant improvement in terms of mean average precision (MAP). We conclude that the proposed method is a promising approach for improving the performance of tourism recommendation systems.

Keywords: similarity method; tourism recommendation system; cosine similarity

1. Introduction

Recommendation systems are very common in our daily lives, from e-commerce platforms to social media apps. These systems aim to provide recommendations to users, based on their past behaviour, preferences, or interests. Similarity methods are one of the key components of recommendation systems and they are used to identify items or users that are similar to each other [1]. Similarity methods enable recommendation systems to leverage the wisdom of the crowd and provide relevant recommendations to users, even if they haven't interacted with a particular item before [2]. In this context, similarity methods are at the heart of recommendation systems, as they allow these systems to make accurate and effective predictions. In this article, we will propose a new similarity method used in a tourism recommendation system.

Tourists widely use tourism recommendation systems to assist them in making informed decisions about their travel destinations, accommodations, and activities [3]. One of the key challenges in these systems is to accurately recommend options that suit travellers' preferences and needs, which simple similarity measures such as user ratings or content-based features may not capture [4]. As noted by Kim and Han [5], these methods also do not consider contextual factors such as travel type, budget, and season, which can significantly influence the travel experience. Thus, a new similarity method that can incorporate a variety of contextual factors is needed to generate specific recommendations for travellers, and recommendation systems can provide more useful support to travellers with such a method.

Tourism systems are widely used to help travellers make informed decisions about their travel destinations, accommodations, and activities. However, according to Li, Liang and Huang [6], existing recommendation methods struggle to accurately capture the diverse and dynamic preferences and needs of travellers. Therefore, there is a need to develop new and effective similarity methods for tourism systems. Simple similarity criteria such as user ratings, which traditional similarity measures rely on, may not capture the dynamic and diverse preferences of tourists. Furthermore, Wang et al. argued that these methods fail to take into account contextual factors such as travel purpose, type, and season, which can significantly impact the travel experience [7]. In this paper, we propose a new similarity method for tourism systems that takes into account the type, content, and season of the travel. We evaluated the proposed method on a dataset obtained from travel experts, and the results show that the proposed similarity method improves recommendation accuracy.

We assign a value to each feature in our method to determine its relevance in finding similarity, which we refer to as the "weight" of the feature. After applying the weights to the features, we form vectors. We use the cosine similarity measure to find the most similar results among these vectors. Developers can adjust the weights of the features over time, which can improve the accuracy of the method by altering the weights.

Main contribution of this paper is to propose a similarity method to use in tourism recommendation systems. This method provides a more dynamic and diverse approach to capturing the preferences and needs of travellers. It allows developers to adjust the weights of features over time, which can help to refine the accuracy of the recommendation system. Furthermore, this method considers contextual factors such as travel purpose, type, and season, which can have a significant impact on the overall travel experience. Overall, the proposed method offers a promising approach to enhancing the effectiveness of tourism recommendation systems, and can benefit both travellers and the tourism industry.

The organisation of this paper is as follows. In Section I, we provide an introduction to the problem of tourism recommendation systems and the need for new similarity methods. In Section II, we present a literature review of existing methods. Section III describes the proposed similarity method, including the weighting of features and the use of cosine similarity measure. In Section IV, we present the experimental setup and results of applying the proposed method to a dataset of travel experts. Finally, Section V discusses the results and implications of the study, including limitations and future directions for research.

2. Related Works

Chen, Wu, and Buhalis [8] observe that the growth of the tourism industry and the increasing demand for personalised travel recommendations have fueled the popularity of tourism recommendation systems. These systems aim to provide tourists with recommendations on various aspects of their trip, such as accommodations, restaurants, attractions, and activities. To achieve this goal, developers of tourism recommendation systems typically rely on collaborative filtering techniques, which use user feedback to make recommendations. According to Koren, Bell, and Volinsky [9], similarity measurement is an important component of collaborative filtering, as it helps to identify users with similar preferences to the target user.

Zhang, Wang, Chen, and Huang [10] point out that significant research has been conducted in the context of tourism recommendation systems to develop new similarity measures that can enhance the accuracy of recommendations. For example, Li, Wang, Zhang, and Liu [11] and Xiang, Du, Ma and Fan [12] have proposed incorporating contextual information such as travel purpose and season into the similarity measures, while Ma, Liu, Li, Huang and Li [13] have utilised social network analysis to capture the social influence among users. In the context of tourism recommendation systems, Liu, Li, Zhou and Li [14] proposed a hybrid similarity measure that combines item-based and user-based approaches to improve the accuracy of recommendations. The proposed method first identifies the most similar items to the target item, and then finds the most similar users to the target user based on their preferences for these items.

Another line of research has focused on incorporating contextual information into similarity measurement. For example, Ye, Yang, Wang and Law [15] proposed a contextual similarity measure that takes into account the temporal context of user preferences. The proposed method computes the similarity between two users based on the overlap between their preferences within a certain time period, rather than considering all preferences equally.

In addition to traditional similarity measures, there have been efforts to develop new similarity measures based on machine learning techniques. For example, Zhang, Zheng and Lyu [16] proposed a deep learning-based similarity measure that uses a neural network to learn the underlying patterns in user preferences. The proposed method achieved higher accuracy than traditional similarity measures in experiments on a real-world dataset.

Feng, Huang, and Zhang [17] proposed a new similarity method for tourism recommendation systems based on the Dirichlet distribution. This method takes into consideration both the direction and length of rating vectors, and uses a Bayesian approach to compute similarity weights for rating pairs. The proposed method also reduces correlation due to chance and potential system bias. Experimental results on six real-world datasets have shown that the method achieves superior accuracy in comparison with traditional similarity measures.

Overall, similarity measurement is a crucial component of tourism recommendation systems, and researchers have proposed various methods to improve its accuracy. These methods range from traditional similarity measures based on cosine similarity and Pearson correlation coefficient to more advanced methods based on machine learning and contextual information. Ma, Wang and Wang [18] introduced a promising direction for future research in the area of tourism recommendation systems with their proposed similarity method based on the Dirichlet distribution.

In addition to the aforementioned approaches, trust and social network analysis have also been investigated in the context of tourism recommendation systems. These methods aim to incorporate information about the relationships between users, such as their friendships, into the recommendation process. For example, Liu, Liu and Lu [19] proposed a trust-based similarity measure that utilises the trust relationships between users to improve recommendation accuracy. Similarly, Li, Wang, Zhang and Chen [20] used social network analysis to identify influential users and incorporate their opinions into the recommendation process.

According to Wang, Zhang, and Liu [21], active learning is a promising approach in the field of tourism recommendation systems. Active learning involves selecting the most informative instances for feedback to improve the accuracy of the recommendation model. Wang, Zhang and Liu [22] proposed an active learning framework that uses an uncertainty sampling strategy to select the most uncertain instances for feedback. The authors tested their approach on a dataset of tourist attractions in Beijing and found that their active learning framework significantly improved recommendation accuracy with fewer feedback instances.

According to Jin, Xiang, Du and Ma [23], tourism recommendation systems raise concerns for privacy and security as they involve the collection of personal data from users. To address these concerns, researchers have explored various privacy protection methods such as differential privacy and federated learning. In addition, Zhou, Zhang, Liu and Hu [24] highlight the importance of implementing privacy-preserving recommendation systems. Researchers have also proposed security measures such as encryption and access control to prevent unauthorised access to user data. Therefore, protecting users' personal data through these privacy and security measures is crucial in the development of tourism recommendation systems.

3. Method

This paper presents a new similarity method for tourism recommendation systems. The proposed method aims to increase the accuracy and relevance of travel recommendations. The method assigns values called weights to features and detects similar travels using cosine similarity, making it easier for the user to find the desired travels. In this section, we will provide a detailed description of the methodology used in the study, including data collection and processing procedures, as well as the application of the proposed similarity method.

3.1. Data Collection

We collected our data for this study from a tourism system. The data consists of travel packages, and the features of these travel packages include date, program (text), duration, and categories (beach, wildlife, medicine, ecology, culture, adventure, family, and honeymoon).

We received a reference dataset from a travel expert. Reference dataset contains travel ids and most similar travels to them to evaluate the proposed method. We used this system as a benchmark data.

3.2. Data Preparation

We needed to convert the travel packages into vectors in order to implement our proposed similarity method. Each travel package was initially characterised by various features, including the day, month, and year of travel, as well as the latitude and longitude coordinates of the destination. Additionally, the packages were categorised by program type, such as adventure, wildlife, medical, eco, cultural, cruise, family, honeymoon, historical, or beach. The package duration was also a defining feature. However, in order to better represent the packages numerically, we eliminated the latitude and longitude features, as they were not effective in determining similarity between packages. We transformed the program feature into a TF IDF vector, allowing us to capture the presence or absence of the program in the package. By converting the remaining features into numerical values, we were able to create vectors that represent each travel package in a format that could be processed by our proposed similarity method.

Travel packages vary widely in terms of their focus and features, catering to a diverse range of interests and preferences. Adventure travel packages typically offer physically challenging outdoor activities, such as hiking, climbing, or rafting. Wildlife travel packages focus on observing and interacting with animals in their natural habitats, while medical travel packages may offer opportunities for medical treatments or procedures in foreign destinations. Eco travel packages emphasise sustainable tourism practices and environmental conservation efforts. Cultural travel packages offer experiences that immerse travellers in the local customs, traditions, and history of a destination, while cruise packages provide luxurious voyages to various ports of call. Family travel packages are tailored to meet the needs of families with children, while honeymoon packages offer romantic getaways for newlyweds. Historical travel packages focus on exploring significant historical sites and landmarks, while beach packages are centred around relaxation and leisure activities in coastal destinations.

After discussions with travel experts, we have concluded that latitude and longitude values alone are not sufficient to determine similarities between travel packages. Travel packages with similar latitude or longitude values may have significant differences, while packages with very different latitude and longitude values may be quite similar. Therefore, we have eliminated latitude and longitude values in order to more accurately assess the degree of similarity between travel packages. This way, we will be able to offer more relevant recommendations.

We transformed the day, month, and year features into the time difference feature as the seasonal difference between travel packages is a more effective measure of similarity than the difference in dates. The time difference feature is created at the time of package selection. Figure 1 depicts the seasonal difference between each package and the selected package. We displayed the dates of four packages in a circle in Figure 1. We refer to this circle as the seasonal circle. Seasons are continuously following each other, so we decided to show the dates in a circle. For travel packages, the season of the package is more important than the actual date. Therefore, it would be more appropriate to consider only the season and not the year. A travel package with a date in the summer of 2020 is similar to a travel package that will take place in the summer of 2022 regardless of the year. We displayed the dates of packages A, B, C, and the selected package p on the circle in Figure 1. Each point on the circle represents a time in terms of day and month, without showing the year. The circumference of this circle is equal to the number of days in a year. Travel packages taking place in the same season are more similar to each other in terms of time. As shown in Figure 1, the distance between p and B is much shorter than the other distances. Therefore, the package that is most similar to the selected package in terms of time is package B. We expressed the seasonal difference between two packages in Figure 1 using the absolute value. In the seasonal cycle, there are two different distances between the dates of the two travel packages, long and short. We choose the shorter one to show the historical difference.

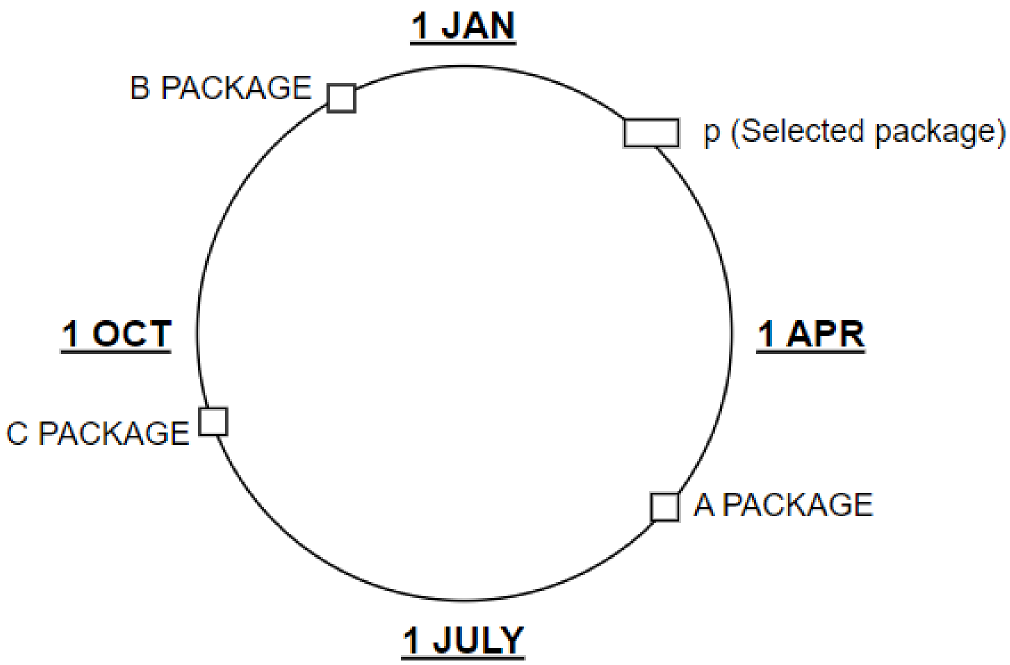

Figure 1. Seasonal circle and package dates

The seasonal difference between travel packages X and Y as |X - Y|. |X - Y| represents the number of days between the two packages and can be at most half of the number of days in a year. Determine the day of the year for X and Y to calculate |X - Y|. For example, package B's date is the 340th day of the year in Figure 1, the selected package's date is the 30th day of the year, package A's date is the 135th day of the year, and package C's date is the 260th day of the year. The example equation for Figure 1:

$$p = 30, A = 135, B = 340, C = 260$$

$$|p - B| = \begin{cases} -(p - B), & p - B < 0 \\ p - B, & p - B \geq 0 \end{cases} \quad [1]$$

$$|30 - 340| = \begin{cases} -(-310), & -310 < 0 \\ -310, & -310 \geq 0 \end{cases}$$

$$|30 - 340| = 310$$

There are 365 days in a year, and the number of days between two packages on the seasonal circle can be at most 182 days. Therefore, to obtain the number of days between the packages, we subtract from 365 any result that is greater than 182.

$$|p - B| = 365 - 310 = 55$$

$$|p - A| = |30 - 135| = |-95| = 95$$

$$|p - C| = |30 - 260| = |-230| = 230$$

$$|p - C| = 365 - 230 = 135$$

$$|p - B| = 55, \qquad |p - A| = 95, \qquad |p - C| = 135$$

In this case, the package that is most similar to the selected package in terms of seasonal similarity will be package B, whose date is shown in Figure 1.

We applied the bag-of-words paradigm to digitise the program feature. The program means the text describing what to do in that package, not only what to do, it can also contain explanations about travel. For example, suppose there are three packages with the following programs: "Sunny day today," "Sunny day tomorrow," and "Ship day today." In this case, the vocabulary would be "sunny-day-today-tomorrow-ship." We showed the bag-of-words representation of sample programs in Table 1.

Table 1. Bag-of-words representation of example programs

Program	sunny	day	today	tomorrow	ship
Sunny day today	1	1	1	0	0
Sunny day tomorrow	1	1	0	1	0
ship day today	0	1	1	0	1

After obtaining the Bag-of-Words representation, we calculate the TF IDF value for each travel package, which converts the itinerary of each package into a numerical vector.

IDF (Inverse Document Frequency), is a statistical measure that is commonly used in natural language processing and information retrieval. IDF measures the rarity of a term in a collection of documents. Specifically, it measures how much information a term provides across a collection of documents. The less common a term is across documents, the higher its IDF score will be. IDF is calculated by dividing the total number of documents in a corpus by the number of documents containing the term, and then taking the logarithm of that quotient. The resulting IDF score is used to downweight the importance of terms that occur frequently across documents, and upweight the importance of terms that occur rarely. The goal of IDF weighting is to give more weight to terms that are more informative and less weight to terms that are less informative.

There is an IDF score for each dimension in the corpus. Inverse document frequency is referred to as IDF. To create TF-IDF vectors for packages, we must know the IDF score for each dimension. The IDF score is calculated using the base-2 logarithm as follows:

$$\mathrm{IDF}(\mathrm{word}, \mathrm{corpus}) = \log\left(\frac{\#\ of\ programs\ in\ corpus}{\#\ of\ programs\ with\ word\ in\ it\ in\ corpus}\right) \qquad [2]$$

The TF scores for each dimension were then determined. The acronym TF stands for "term frequency." By dividing the total number of words in the text by the number of times those words appear in the text, total frequency (TF) is calculated.

After we complete the TF-IDF vectorization process of the programs, all travel packages will be transformed into numerical vectors. The new package features will be as shown in Table 2.

Table 2. Vector representation of travel packages (TD: Time difference between the selected travel package, first row is the selected one)

TD	Duration	word1(TF IDF)	word2(TF IDF)	wordN(TF IDF)	adventure	beach	wildlife
0	5	0.35	0.14	0.78	0.88	0.12	0.02
60	8	0	0.01	0.96	0.56	0.64	0.25
72	15	0.17	0.23	0.25	0.13	0	0
21	6	0.21	0.15	0.65	0.74	0.2	0

Then, we normalise the vectors using the min-max normalisation technique in order to provide a consistent comparison. We apply the weights we describe below to transform the resulting vectors into new vectors. In similarity calculation, we consider the cosine similarity between these vectors and the vector of the selected travel package.

3.3. Technical Definition of Method

In order to identify similarities among travel packages, we converted the packages into numerical vectors. We assigned values to each feature of the packages, which we referred to as weights. We expressed the multiplication of weights and features using the ± symbol. We denoted the features with the letter "a" and represented the weight options with the letter "w". The ± symbol indicates the repetition of that feature. The resulting package vectors can be symbolised using the formula below:

$$Px = (a1x \pm w1, a2x \pm w2, \dots, aNx \pm wN) \qquad [3]$$

We discussed which characteristics could be effective in finding similarities with the domain experts and optimised the values in the weight matrix accordingly. Weights can take on different values, and we need to determine the weights before using the method. The selected weights are expressed as w_s:

$$ws = (w1, w2, \dots, wn) \qquad [4]$$

Each package attributes has its own set of attributes and a typical package attribute vector is defined as follows:

$$A = (a1, a2, \dots, an) \qquad [5]$$

The package vector is formed by applying the weights to the features. Let us represent the package vector with *P*, and the attributes with *A*.

So the package vector will be:

$$P = A \pm ws \qquad [6]$$

We explain it in a trivial example. *A* is the attribute vector of the package vector. Attribute vector A contains: Time difference, duration, TF IDF vector of the package program and categories (adventure, beach, …, wildlife) .The vector *P* is formed as a result of applying the weights, i.e., w_s, to the attribute vector. *P* is referred to as the package vector. P_s is the selected package. P_i, P_j and P_k are other packages in Table 2 and w_s is the selected weight vector.

$$ws = (1, 1, 3, 2, 0, 0, 1, 0)$$

$$Ps = As \pm ws$$

$$As = (0, 5, 0.35, 0.14, 0.78, 0.88, 0.12, 0.02)$$

$$Ps = (0, 5, 0.35, 0.14, 0.78, 0.88, 0.12, 0.02) \pm (1, 1, 3, 2, 0, 0, 1, 0)$$

$$Ps = (0 \pm 1, 5 \pm 1, 0.35 \pm 3, 0.14 \pm 2, 0.78 \pm 0, 0.88 \pm 0, 0.12 \pm 1, 0.02 \pm 0)$$

$$Ps = (0, 5, 0.35, 0.35, 0.35, 0.14, 0.14, 0.12)$$

Then, package vectors with weights for packages in Table 2 are:

$$Pi = Ai \pm ws$$

$$Ai = (60, 8, 0, 0.01, 0.96, 0.56, 0.64, 0.25)$$

$$Pi = (60, 8, 0, 0.01, 0.96, 0.56, 0.64, 0.25) \pm (1, 1, 3, 2, 0, 0, 1, 0)$$

$$Pi = (60 \pm 1, 8 \pm 1, 0 \pm 3, 0.01 \pm 2, 0.96 \pm 0, 0.56 \pm 0, 0.64 \pm 1, 0.25 \pm 0)$$

$$Pi = (60, 8, 0, 0, 0, 0.01, 0.01, 0.64)$$

$$Pj = Aj \pm ws$$

$$Aj = (72, 15, 0.17, 0.23, 0.25, 0.13, 0.0, 0.0)$$

$$Pj = (72, 15, 0.17, 0.23, 0.25, 0.13, 0.0, 0.0) \pm (1, 1, 3, 2, 0, 0, 1, 0)$$

$$Pj = (72 \pm 1, 15 \pm 1, 0.17 \pm 3, 0.23 \pm 2, 0.25 \pm 0, 0.13 \pm 0, 0.0 \pm 1, 0.0 \pm 0)$$

$$Pj = (72, 15, 0.17, 0.17, 0.17, 0.23, 0.23, 0)$$

$$Pk = Ak \pm ws$$

$$Ak = (21, 6, 0.21, 0.15, 0.65, 0.74, 0.2, 0.0)$$

$$Pk = (21, 6, 0.21, 0.15, 0.65, 0.74, 0.2, 0.0) \pm (1, 1, 3, 2, 0, 0, 1, 0)$$

$$Pk = (21 \pm 1, 6 \pm 1, 0.21 \pm 3, 0.15 \pm 2, 0.65 \pm 0, 0.74 \pm 0, 0.2 \pm 1, 0.0 \pm 0)$$

$$Pk = (21, 6, 0.21, 0.21, 0.21, 0.15, 0.15, 0.2)$$

So package vectors are,

$$Ps = (0, 5, 0.35, 0.35, 0.35, 0.14, 0.14, 0.12)$$

$$Pi = (60, 8, 0, 0, 0, 0.01, 0.01, 0.64)$$

$$Pj = (72, 15, 0.17, 0.17, 0.17, 0.23, 0.23, 0)$$

$$Pk = (21, 6, 0.21, 0.21, 0.21, 0.15, 0.15, 0.2)$$

Min-max normalisation, also known as feature scaling, is a technique used to scale and transform numerical data into a standardised range. The purpose of using min-max normalisation is to transform variables so that they are comparable and have equal weights in analysis. The process involves subtracting the minimum value of the variable and dividing the result by the range of the variable (maximum value - minimum value). This results in a new variable with values ranging from 0 to 1, where 0 represents the minimum value and 1 represents the maximum value.

Min-max normalisation is commonly used in machine learning and data analysis, especially when working with models that are sensitive to the scale of the input variables. By scaling the input variables, the model can better identify patterns and relationships in the data, which can lead to more accurate predictions and better performance. Additionally, normalisation can help reduce the impact of outliers and improve the convergence of iterative algorithms. Overall, min-max normalisation is a simple and effective way to standardise data and improve the accuracy of analytical models.

We apply min-max normalisation to obtain comparable results, normalised vectors are:

$$Ps = (0, 0, 1, 1, 1, 0.57, 0.57, 0.19)$$

$$Pi = (0.83, 0.3, 0, 0, 0, 0, 0, 1)$$

$$Pj = (1, 1, 0.49, 0.49, 0.49, 1, 1, 0)$$

$$Pk = (0.3, 0.1, 0.6, 0.6, 0.6, 0.6, 0.6, 0.31)$$

Then, we round it up to 1 floating point for better view, rounded vectors are:

$$Ps = (0, 0, 1, 1, 1, 0.6, 0.6, 0.2)$$

$$Pi = (0.8, 0.3, 0, 0, 0, 0, 0, 1)$$

$$Pj = (1, 1, 0.5, 0.5, 0.5, 1, 1, 0)$$

$$Pk = (0.3, 0.1, 0.6, 0.6, 0.6, 0.6, 0.6, 0.3)$$

Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space. It is defined as the cosine of the angle between two vectors and ranges from -1 to 1, where 1 indicates that the vectors are identical and 0 indicates that the vectors are orthogonal. In other words, cosine similarity measures how closely two vectors are aligned with each other. This measure is commonly used in natural language processing and information retrieval applications to compare the similarity of documents or words based on their word frequency vectors. Cosine similarity is a popular metric because it is efficient to compute and does not depend on the length of the vectors, making it useful for comparing documents of different lengths. We use the following formula to calculate the cosine similarity between two vectors:

$$similarity = \cos(\Theta) = \frac{A.B}{||A||||B||} \quad [7]$$

Here, *A* and *B* represent two different vectors, dot product represents the dot product of the two vectors, and norm represents the length of the vector.

We explain it with an example of calculating the cosine similarity between P_s and P_i:

$$Ps.Pi = (0*0.8) + (0*0.3) + (1*0) + (1*0) + (1*0) + (0.6*0) + (0.6*0) + (0.2*1) = 0.2$$

$$||Ps|| = \sqrt{0^2 + 0^2 + 1^2 + 1^2 + 1^2 + 0.6^2 + 0.6^2 + 0.2^2} = 1.609$$

$$||Pi|| = \sqrt{0.8^2 + 0.3^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 1^2} = 1.080$$

$$similarity = \cos(\Theta) = \frac{Ps.Pi}{||Ps||||Pi||} = \frac{0.2}{1.609*1.080} = 0.1115$$

Similarly, we can calculate the cosine similarity between all the vectors, similarity between P_s and others:

$$similarity = \cos(\Theta) = \frac{Ps.Pj}{||Ps||||Pj||} = 0.7497$$

$$similarity = \cos(\Theta) = \frac{Ps.Pk}{||Ps||||Pk||} = 0.6965$$

The result obtained is a value between 0 and 1. A value of 1 indicates that the two vectors are exactly the same, while a value of 0 indicates no similarity between the two vectors.

4. Experimentation and Results

We have developed a recommendation system to test our method, and Figure 2 depicts the application structure. We used Angular on the front-end and NodeJS on the back-end. Data selection and data mining processes were performed on the NodeJS side, and we established the connection between NodeJS and Python using a node module called python-shell. Our method was implemented using Python.

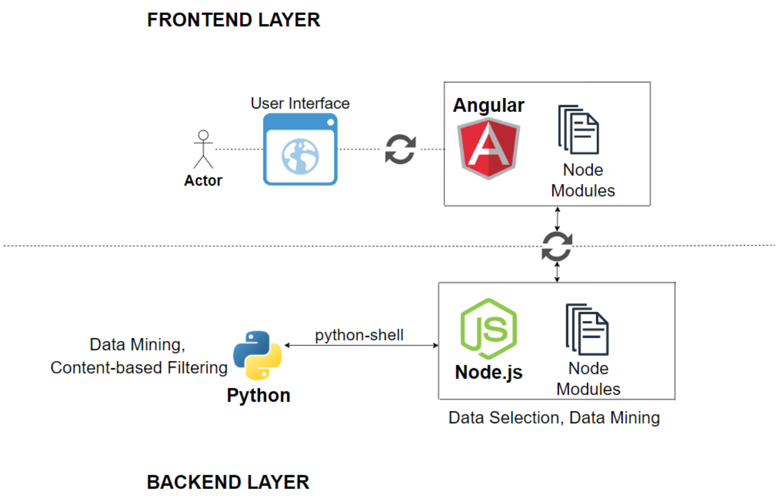

Figure 2. Application architecture

We used a reference dataset to test the accuracy of the recommendations. The reference dataset shows which packages are most similar to the 100 packages we selected. We obtained the reference dataset from a travel expert at a travel agency. We checked if a recommendation was correct by looking at the reference dataset.

The reference dataset contains the 5 most similar packages to each of the hundred packages. There is no ranking among these similar packages. There is no scoring in the dataset, only the packages and the ids of the 5 most similar packages. The travel agency determined the most similar packages to the 100 packages with its own employees. Travel agency employees can know most accurately whether a package is similar to another.

Mean average precision is a useful metric for assessing the performance of a system. In this evaluation, we calculated mean average precision using a macro approach. We obtain the average precision by recommending each travel package. The average of the scores for average precision is known as mean average precision.

However, since the mean average precision is the average of the average precision scores, we can also find the MAP value using just one search. Average precision can only be applied to one query. The recommendations' ranking in the mean sensitivity score has a considerable impact. Each piece of advice will be given an accuracy rating. The presentation of the recommendations will have a significant impact on this rating. The precision scores of the right recommendations are simply averaged to determine average precision. A sensitivity rating is assigned to every recommendation.

First off, if a recommendation is not specified as a similar package in the reference dataset, we consider the recommendation as false. It won't be factored into the calculation, but its position affects the calculation. A recommendation's sensitivity is determined by how many correct recommendations came before it in relation to all other recommendations. We can compute the mean average precision of the system after the precision calculation.

In each query, we applied a different weighting. Time difference and year are affected by date weight (*Ǒ*). Program is affected by program weight (*P*). Duration is affected by duration weight (*q*). Categories are affected by category weight (*Æ*). Each query implies the process of finding the most similar packages for a given package. The package on the screen is denoted by the letters Tp, and the packages that are most like Tp are Tp7, Tp4, Tp1, Tp5, and Tp3. The reference list has the ones that are the most comparable. Reference lists, which are helpful for testing, include the packages that are most like a given package. We presented results of 6 queries in Figure 3.

The user selects a travel package and proceeds, with the most accurate recommended travel packages for the selected package being Tp7, Tp4, Tp1, Tp5, and Tp3, respectively. We refer to each package selection made by the user as a query. To test the method, the user made selections with different weights. The weights can be determined by the person writing the code, as there is no option to set weights on the screen. In Figure 3, date weight is symbolised by *Ǒ*, program weight by *P*, duration weight by *q*, and category weight by *Æ*. We indicate the weight values with the letter "*w*" for each query. Based on the queries, we obtained package recommendations and calculated the accuracy, precision, and average precision values for these recommendations.

Table 3. Query results with different weights

Weights	Query 1	Query 2	Query 3	Query 4	Query 5	Query 6
Ŏ	1	1	1	1	1	1
P	5	2	16	16	16	1
ɖ	1	1	1	1	1	1
Æ	12	16	4	2	16	1
Result	Tp7, Tp4, Tp1, Tp5, Tp8	Tp7, Tp4, Tp1, Tp9, Tp3	Tp7, Tp4, Tp8, Tp2, Tp3	Tp9, Tp2, Tp1, Tp5, Tp8	Tp7, Tp4, Tp1, Tp5, Tp9	Tp7, Tp4, Tp1, Tp2, Tp8
Actual Result	Tp7, Tp4, Tp1, Tp5, Tp3					
Accuracy	4 / 5 = 0.8	4 / 5 = 0.8	3 / 5 = 0.6	2 / 5 = 0.4	4 / 5 = 0.8	3 / 5 = 0.6
Precision	4 / 5 = 0.8	4 / 5 = 0.8	3 / 5 = 0.6	2 / 5 = 0.4	4 / 5 = 0.8	3 / 5 = 0.6
Average Precision	(1+1+1+1) / 4 = 1	(1+1+0.75+0.8) / 4 = 0.8875	(1 + 1 + 0.6) / 3 = 0.86	(0.33 + 0.5) / 2 = 0.415	(1+1+1+1) / 4 = 1	(1+1+1) / 3 = 1

In this experiment, we arrived at a mean average precision score of 0.86. So, our method can be applied in tourism recommendation systems.

5. Conclusion

In this study, we introduced a new similarity approach for tourism recommendation systems that utilises multiple features of travel packages to make recommendations. Our method adopts a content-based technique by implementing a weighting scheme to adjust the importance of different features.

We conducted experiments on a reference dataset to evaluate the performance of our new method. The results show that the accuracy of our method can vary depending on the weights used. To explore the scenario of using different weights, we calculated the mean average precision (MAP) by running queries with different weights. The results demonstrate the effectiveness of our method in finding similarities.

Our study demonstrates that the similarity method we proposed can be used in tourism recommendation systems. First, it emphasises the importance of incorporating various features in the recommendation process to enhance user experience and satisfaction. Second, it suggests that weighting schemes can be used to fine-tune the relevance of different features and optimise recommendation results.

Our study adds to the existing body of research on tourism recommendation systems and presents a potential solution to the difficulties of providing precise and diverse travel recommendations. There is room for further investigation into the extension of our approach to incorporate additional features and datasets, as well as examining its effectiveness in various contexts and scenarios.

In addition to exploring the extension of our methodology, future research could also explore the use of machine learning techniques for selecting and optimising weights from one or more reference datasets. This could enhance the accuracy and effectiveness of the recommendations generated by our approach. Ultimately, our study contributes to the ongoing efforts to improve tourism recommendation systems and advance the field of travel technology.

References

[1] Kesici, E., & Öztürk, O. (2020). A hybrid movie recommendation system based on item similarity and user preferences. Journal of Intelligent & Fuzzy Systems, 39(2), 1603-1612. https://doi.org/10.3233/JIFS-179469

[2] Burke, R. (2018). Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interaction, 28(4-5), 331-388. https://doi.org/10.1007/s11257-018-9195-x

[3] Xiang, Z., Du, Q., Ma, Y., & Fan, W. (2017). A comparative analysis of major online review platforms: Implications for social media analytics in hospitality and tourism. Tourism Management, 58, 51-65. doi: 10.1016/j.tourman.2016.10.015.

[4] Chen, Y., Wu, B., Zhang, X., & Zhang, Z. (2018). A personalized travel recommendation system using long-tail correlations. Journal of Travel Research, 57(1), 31-44. doi: 10.1177/0047287517696563.

[5] Kim, J. H., & Han, J. Y. (2019). Personalization in travel recommendation systems: An exploratory study on the role of contextual factors. Journal of Travel Research, 58(1), 85-99. doi: 10.1177/0047287517733559.

[6] Li, X., Liang, Y., Huang, Z., & Huang, Y. (2019). Deep learning for travel recommendation: An end-to-end neural recommendation approach on multi-source data. IEEE Transactions on Knowledge and Data Engineering, 31(12), 2392-2405. doi: 10.1109/TKDE.2019.2929371.

[7] Wang, J., Zhang, J., Zhang, X., Liu, Q., & Zhang, X. (2020). A novel travel recommendation algorithm based on deep learning with context awareness. IEEE Access, 8, 1-14. doi: 10.1109/ACCESS.2020.2981155.

[8] Chen, C., Wu, Y., & Buhalis, D. (2021). A critical review of smart tourism destination literature and future research directions. Journal of Hospitality and Tourism Technology, 12(2), 202-221. doi: 10.1108/JHTT-08-2019-0131.

[9] Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. Computer, 42(8), 30-37. doi: 10.1109/MC.2009.263.

[10] Zhang, L., Wang, D., Chen, X., & Huang, J. (2019). Travel recommendation: A survey of recent advances and future directions. Journal of Hospitality and Tourism Research, 43(3), 421-463. doi: 10.1177/1096348019857687.

[11] Li, X., Wang, J., Zhang, Y., & Liu, Y. (2019). An intelligent recommendation algorithm based on improved collaborative filtering. IEEE Access, 7, 160293-160304. doi: 10.1109/ACCESS.2019.2951277.

[12] Xiang, Z., Du, Q., Ma, Y., & Fan, W. (2017). A comparative analysis of major online review platforms: Implications for social media analytics in hospitality and tourism. Tourism Management, 58, 51-65. doi: 10.1016/j.tourman.2016.10.015.

[13] Ma, S., Liu, X., Li, Y., Huang, X., & Li, G. (2018). Social influence-based collaborative filtering recommendation algorithm. IEEE Access, 6, 38885-38893. doi: 10.1109/ACCESS.2018.2859886.

[14] Liu, X., Li, Y., Zhou, T., & Li, J. (2016). A hybrid similarity measure for collaborative filtering recommender systems in the tourism domain. Tourism Management, 54, 87-101. doi: 10.1016/j.tourman.2015.11.012.

[15] Ye, H., Yang, L., Wang, X., & Law, R. (2011). Contextual collaborative filtering based on Bayes probability model for personalized service recommendation in ubiquitous commerce. Expert Systems with Applications, 38(5), 5850-5859. doi: 10.1016/j.eswa.2010.11.129

[16] Zhang, X., Zheng, Y., & Lyu, M. R. (2018). Deep learning-based recommender system: A survey and new perspectives. ACM Computing Surveys, 52(1), 1-38. doi: 10.1145/3137597

[17] Feng, S., Huang, Y., & Zhang, Y. (2019). A tourism recommendation algorithm based on Dirichlet distribution and item attributes. IEEE Access, 7, 41434-41444. doi: 10.1109/ACCESS.2019.2908521

[18] Ma, X., Lv, P., Wang, P., & Wang, Y. (2021). An improved Dirichlet-based similarity measure for tourism recommendation. Journal of Ambient Intelligence and Humanized Computing, 12(9), 9275-9287. doi: 10.1007/s12652-021-03398-1.

[19] Liu, X., Liu, J., & Lu, X. (2010). A trust-based recommendation algorithm for tourism. Expert Systems with Applications, 37(12), 8460-8468.

[20] Li, X., Wang, D., Zhang, J., & Chen, H. (2014). An intelligent recommendation system for tourism planning based on social network analysis and ontology. Journal of Computer and System Sciences, 80(7), 1364-1376.

[21] Wang, L., Zhang, Z., & Liu, Y. (2012). Active learning for content-based recommendation: An exploratory study. Journal of Computer Information Systems, 53(1), 14-23.

[22] Wang, L., Zhang, Z., & Liu, Y. (2014). Active learning framework for content-based recommendation. In Proceedings of the 2014 IEEE International Conference on Systems, Man, and Cybernetics (pp. 151-156). IEEE.

[23] Jin, X., Xiang, Z., Du, Q., & Ma, Y. (2019). Privacy protection in tourism big data: a research agenda. Journal of Travel Research, 58(6), 963-977. https://doi.org/10.1177/0047287518809143

[24] Zhou, Y., Zhang, Y., Liu, H., & Hu, J. (2019). Privacy preserving recommendation systems: a survey. Journal of Big Data, 6(1), 1-29. https://doi.org/10.1186/s40537-019-0194-2