Human-centric Computing and Information Sciences

RESEARCH Open Access

Mobile marketing recommendation method based on user location feedback

Chunyong Yin[1], Shilei Ding[1] and Jin Wang[2*]

*Correspondence: jinwang@csust.edu.cn
[2] School of Computer & Communication Engineering, Changsha University of Science & Technology, Changsha 410004, China
Full list of author information is available at the end of the article

Abstract

Location-based mobile marketing recommendation has become one of the hot spots in e-commerce. The current mobile marketing recommendation system only treats location information as a recommended attribute, which weakens the role of users and shopping location information in the recommendation. This paper focuses on location feedback data of user and proposes a location-based mobile marketing recommendation model by convolutional neural network (LBCNN). First, the users' location-based behaviors are divided into different time windows. For each window, the extractor achieves users' timing preference characteristics from different dimensions. Next, we use the convolutional model in the convolutional neural network model to train a classifier. The experimental results show that the model proposed in this paper is better than the traditional recommendation models in the terms of accuracy rate and recall rate, both of which increase nearly 10%.

Keywords: Location feedback, Mobile marketing, Convolutional neural network, Sequential behavior

Introduction

In recent years, the e-commerce industry has developed rapidly with the popularization of the Internet. At this time, famous e-commerce platforms such as Alibaba and Amazon were born. E-commerce moved physical store products to a virtual network platform. On the one hand, it is convenient for users to buy various products without leaving the home. On the other hand, it is also convenient for sellers to sell their own goods and reduce costs. However, the various products have made it more difficult for users to select products. E-commerce platform can generate a large amount of user location feedback data which contains a wealth of user preference information [1]. It is significant to predict the location of the next consumer's consumption from these behavioral data. At present, most of the recommended methods focus on the user-product binary matrix and directly model their binary relationships [2]. The users' location information and shopping location information are considered as the third factor. In this case, you can only use the limited check-in data. The users' location feedback behavior and the timeliness of behavior are often overlooked.

The mobile recommendation system takes advantage of the mobile network environment in terms of information recommendation and overcomes the disadvantages. Filtering irrelevant information by predicting potential mobile user preferences and providing

SpringerOpen

mobile users with results that meet users' individual needs gradually become an effective means to alleviate "mobile information overload" [3]. Mobile users have different preferences in different geographical locations. For this problem, how to use location information to obtain mobile users' preferences and provide accurate personalized recommendations has become a hot topic in mobile recommendation research [4]. Although there are many researches based on location recommendation, they mainly focus on service resources without positional relevance. To solve the shortcomings of research on location relevance of service resources is few [5], Zhu et al. [6] proposed the method which is based on the user's context information to analyze the user's preferences and retrograde. Their approach is to derive user preferences by proposing two different assumptions and then recommending user models based on preference analysis. Yin et al. [7] proposed LA-LDA. The method is a location-aware based generation probability model, which uses scoring based on location to model user information and recommend to users. However, these methods only treat location information as an attribute without considering the spatial information of users or items and weaken location information's role in the recommendation. There are some studies determine user preferences by the distance between the mobile user and the merchant [8], but only set the area based on the proximity of the distance and ignore the spatial activities of the mobile user [9]. However, these methods were limited to the analysis of user information and product information, and did not carefully consider the importance of user and business location information. Therefore, the user preference model based on location recommendation they created has some gap.

Considering the core of mobile marketing recommendation is location movement, Lian et al. [10] proposed an implied feature-based cognitive feature collaborative filtering (ICCF) framework, which avoids the impact of negative samples by combining conventional methods and semantic content. In terms of algorithms, the author proposed an improved algorithm that can expand according to data size and feature size. To determine the relevance of the project to user needs, Lee et al. [11] developed context information analysis and collaborative filtering methods for multimedia recommendations in mobile environments. Nevertheless, these methods only used small-scale training data and could not achieve accurate prediction of long-term interest for users. In this paper, deep learning and time stamps are used to compensate for these shortcomings.

With great achievements in visual and speech tasks, the Deep Learning (DL) model has become a novel field of study [12]. Because of the interventional optimization of deep learning algorithms, artificial intelligence has made great breakthroughs in many aspects. It is well known that models obtained through deep learning and machine learning models have very similar effects, which learns advanced abstract features from the original input features by simulating the network structure of the human nervous system. Experiments show that the deep model can express the characteristics of the data better than the shallow model [13]. Weight sharing by convolution makes CNN similar to biological neural networks, which reduces the difficulty of network structure and the number of weights. The structure of CNN is roughly divided into two layers. It is well known that the first layer is a convolutional layer. Each neuron's input is connected to the previous layer through a convolution kernel and the local features are extracted. Next layer is a pooling layer. In this layer, the neurons in the network are connected through a convolution kernel to extract the overall features. Convolutional neural networks have great advantages in processing two-dimensional features [14], such as images.

Based on our detailed comparative analysis, this paper proposes a location-based mobile marketing recommendation model by convolutional neural network (LBCNN). Firstly, we use user-product information as a training sample, and treat this problem as a two-class problem. The category of the problem is divided into the purchase behavior and the purchase behavior of the product at the next moment. In order to capture the user's timing preference characteristics, we divide the behavior of the merchandise according to a certain length of time window and dig deeper into the behavior characteristics of each time window. Secondly, we consider the users' timing preferences and overall preferences for the product. Then, the features of time window are used to train convolutional neural network models. Finally, we input the sample features of the test set into the model and generate the Top-K sample as the location-based purchase forecast results [15].

Remain of the paper is divided into four sections. Related work is shown in "Related work" section. Necessary definitions and specific implementation of the location-based mobile marketing recommendation model by convolutional neural network (LBCNN) are shown in "Location-based mobile marketing recommendation model by CNN" section. In "Experimental analysis" section, experimental analysis is introduced. "Conclusion" section summarizes the strengths and weaknesses of the paper and proposes plans for future progress.

Related work

In the current chapter, we will review existing methods for recommending systems that can be broadly divided into three parts: content filtering, collaborative filtering and hybrid methods. We also discuss the establishment of feature models based on time series to clearly represent the differences between our research and other existing methods.

Traditional recommendation method

In the general products recommendation system, the similarity between users is calculated by the user's interest feature vector. Then, the system recommends some products with similarity greater than a certain threshold or the similar Top-N products to the target user. This is a traditional recommendation algorithm based on content and the recommendation is based on comparing users.

a. Content-based recommendation method

Content-based information filtering has proven to be an effective application for locating text documents related to topics. In particular, we need to focus on the application of content-based information filtering in the recommendation system. Content-based methods allow for accurate comparisons between different texts or projects, so the recommended results are similar to the historical content of the user's consumption. The content-based recommendation algorithm involves the following aspects. User description file describes the user's preferences, which can be filled by the user and dynamically updated based on the user's feedback information (purchasing, reading, clicking, etc.) during the operation of the system. The project profile describes the content characteristics of each project, which constitutes the feature vector of the project. In addition, the similarity calculation is the similarity between the user's description file and the item feature vector.

The similarity calculation of the content-based recommendation algorithm usually adopts the cosine similarity algorithm. The algorithm needs to calculate the similarity between the feature vector of user u and the feature vector of item i. The calculation formula is as shown in Formula (1).

$$sim(u, i) = \frac{\vec{u} \cdot \vec{i}}{|\vec{u}||\vec{i}|} \tag{1}$$

where $\vec{u}$ denotes the user feature vector, $\vec{i}$ denotes the project feature vector, $|\vec{u}|$ is the modulus of the user feature vector and $|\vec{i}|$ is the model of the project feature vector.

Representative content-based recommendation systems mainly include Lops, Gemmis, and Semeraro [16]. Compared to other methods, content-based recommendations have no cold-start issues and recommendations are easy to understand. However, the content filtering based recommendation method has various drawbacks, such as strongly relying on the availability of content and ignoring the context information of the recommended party. The content-based recommendation method also has certain requirements for the format of the project. Besides, it is difficult to distinguish the merits of the project. The same type of project may have the same type of features, which are difficult to reflect the quality of the project.

b. Collaborative filtering method

The recommendation based on collaborative filtering solves the recommendation problem by using the information of similar users in the same partition to analyze and recommend new content that has not been scored or seen by the target user.

Regarding the traditional collaborative filtering method based on memory, we understand that this method is based on the different relationships between users and projects. According to expert research, the traditional collaborative filtering method based on memory should be divided into the following three steps.

Step 1: collection of user behavior data, this step represents the user's past behavior with a m * n matrix R. The matrix U_{mn} represents the feedback that the user m has on the recommended object n. Rating is a range of values and different values represent how much the user likes the recommended object.

$$U = \begin{bmatrix} U_{11} & U_{12} & \dots & U_{1n} \\ U_{21} & U_{22} & \dots & U_{2n} \\ \dots & \dots & \dots & \dots \\ U_{m1} & U_{m2} & \dots & U_{mn} \end{bmatrix}.$$

Step 2: establishment of a user neighbor: establish mutual user relationships by analyzing all user historical behavior data.

> Step 3: generate recommendation results: find the most likely N objects from the recommended items selected by similar user sets.

Therefore, recommendations are made by mining common features in similar users' preference information [17]. The normal methods in this classification include k-nearest neighbor (k-NN), matrix decomposition, and semi-supervised learning. According to the survey, Amazon uses an item-by-item collaborative filtering method to recommend personalized online stores for each customer.

Compared to other method, collaborative filtering has the ability to filter out information that can be automatically recognized by the machine and effectively use feedback from other similar users. However, collaborative filtering requires more ratings for the project, so it is affected by the issue of rating sparsity. In addition, this method does not provide a standard recommendation for new users and new projects, which is called a cold start issue.

c. Hybrid recommendation method

The hybrid recommendation method combines the above techniques in different ways to improve the recommended performance and optimize the shortcomings of the conventional method. Projects that cannot be recommended for collaborative filtering are generally addressed by combining them with content-based filtering [18].

The core of this method is to independently calculate the recommendation results of the two types of recommendation algorithms, and then mix the results. There are two specific hybrid methods. One method is to mix the predicted scores of the two algorithms linearly. Another hybrid method is to set up an evaluation standard, compare the recommended results of the two algorithms, and take the recommendation results of the higher evaluation algorithms. In general, the hybrid recommendation achieves a certain degree of compensation between different recommendation algorithms. However, the hybrid recommendation algorithm still needs improvement in complexity.

d. Recommendation based on association rules

The association rule algorithm is a traditional data mining method that has been widely used in business for many years. The core idea is to analyze the rules of user historical behavior data to recommend more similar behavioral items [19]. Rules can be either user-defined or dynamically generated by using rule algorithms. The effect of the algorithm depends mainly on the quantity and quality of the rules so the focus of the algorithm is on how to develop high quality rules.

Define N as the total number of transactions, R is the total project and U and V are two disjoint sets of items ($U \cap V \neq \emptyset$, $U \in R$, $V \in R$). The association rule is essentially an IF–Then statement, here is expressed by $U \rightarrow V$. The strength of the association rule $U \rightarrow V$ can be measured by two criteria: support and confidence. S is the ratio containing U and V data which both represent the number of transactions, which is shown in Formula (2).

$$S(U \rightarrow V) = \frac{N(U \cup V)}{N}. \tag{2}$$

C is the ratio of U, V data to the only U data which represents the number of transactions, as shown in Formula (3)

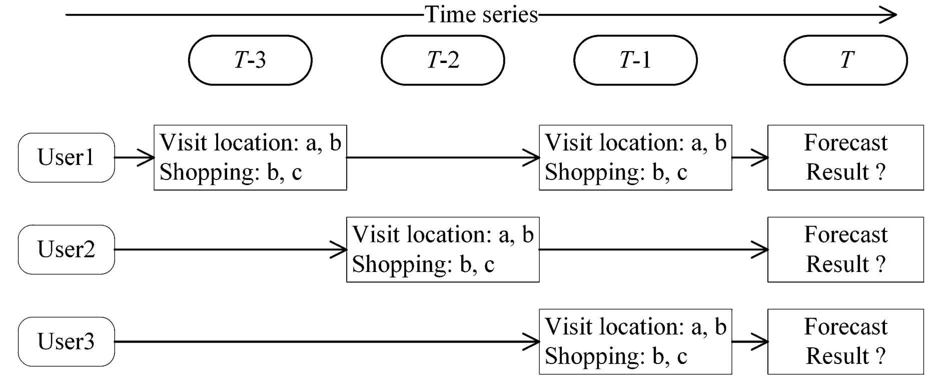

Fig. 1 The time series of user position feedback

$$C(U \rightarrow V) = \frac{N(U \cup V)}{N}. \tag{3}$$

The recommendation process of the algorithm is shown in below.

Firstly, according to the items of interest to the user, the user's interest in other unknown items is predicted by rules. Secondly, compare the support of the rules. Finally, the recommended items of TOP-N are obtained to the user.

The recommendation system based on association rules includes three parts: the keyword, the presentation and the user interface. The keyword layer is a set of keyword attributes and dependencies between keywords. The description layer connects the keyword layer and the user layer and the main function is to describe the user and the resource. The user interface layer is the layer that interacts directly with the user. However, the system becomes more and more difficult to manage as the rules increasing. In addition, there is a strong dependence on the quality of the rules and a cold start problem is existed.

Most of the recommendation systems use collaborative filtering algorithm to recommend for users. However, the traditional algorithm can only analyze ready-made data simply, and most systems simply preprocess the data. In our method, we preprocess the dataset by extending the time information of the data to a time label. The next section is an explanation of the specific implementation.

Construction of time series behavior's preference features

The timing recommendation model is based primarily on the Markov chain. This model makes full use of timing behavior data to predict the next purchase behavior based on the user's last behavior. The advantage of this model is that it can generate good recommendations by timing behavior.

As shown in Fig. 1, the prediction problem of product purchase can be expressed as predicts the user's purchase behavior at time T by a user behavior record set D before time T [20]. Different actions occur at different times. For example, user1 visit location a and b when user1 purchasing b and c at T − 3. We need to predict T-time consumer behavior based on different timing behavior characteristics.

According to relevant professional research, we divide the data sets of user behavior into three groups in a pre-processing manner. By the feature statistics method, the

Table 1 Characteristic system diagram (True/False)

Feature group	Counting feature	Mean feature	Ratio feature
User-product	True	False	True
User feature	True	True	False
Product feature	True	True	False

features are divided into two types, as shown in Table 1. "True" indicates that the feature group has corresponding features. Conversely, "False" means no such feature. Next we explain these features.

a. Counting feature

For each feature statistics window, we use the behavioral counting feature and the deduplication counting feature. The behavior count is a cumulative measure of the number of behaviors that occurred in and before the current window. For the location visit behavior, it represents the number of visits to the product location by the user, the total number of visits by the user and the total number of visits to the merchandise. The deduplication count feature is similar to the behavioral count, but only the number of non-repetitive behavioral data is counted.

b. Mean feature

In order to describe the activity of the user and the popularity of the product better, this article derives a series of mean-type features based on the counting features. Take the location visit behavior as an example, the user characteristics group includes the user's average number of visiting to the product. The average number of visiting to the product by user i is calculated as shown in Formula (4).

$$avg_{ui}(t, i, visit) = \frac{action\_count(t, U, Ui, visit)}{user\_unique\_item(t, U, Ui, visit)}. \tag{4}$$

c. Ratio feature

The ratio of user-product behavior to the total behavior of the user and the product is also an aspect affecting the user's degree of preference for the product. In the time window t, the method to calculate the ratio of the user's visit to the products' total visit is shown in Formula (5).

$$rate\_ui\_in\_u(t, i, j, visit) = \frac{action\_count(t, UI, Ui, Ij, visit)}{action\_count(t, U, Ui, visit)}. \tag{5}$$

Our work presents a mobile marketing recommendation model is trained by adding the time axis to the user position features. Contrary to current research, it is highly usable and low difficulty of achievement for real-world work applications. Considering the speed of calculation, we study the method of directly embedding time series information into the collaborative filtering calculation process to improve the recommendation quality. Specific information will be covered in the following sections.

Location-based mobile marketing recommendation model by CNN

Creating the model is one of the most important aspects, which is an evaluation criterion to make sure correctness of the next step. This section mainly describes the relevant definitions of LBCNN that are shown in "Relevant definitions of the LBCNN" section, and specific implementation of the model is shown in "Specific implementation of the model" section.

Relevant definitions of the LBCNN

In order to get better feature expression, we consider the user's timing sensitivity of the product preferences and the user's overall preferences comprehensively. This paper uses a convolutional neural network as the basis to build location-based mobile marketing recommendation model. In the next step, we give the relevant definition.

a. Definition 1 (Model framework): based on the above analysis and user's timing behavior preference feature. We use the convolutional neural network model shown in Fig. 2. The model is divided into four layers that are input layer, multi-window convolution layer, pooling layer and output layer. The input layer is a well-constructed input feature which transforms the input features into a two-dimensional plane by time series. Each time window is expressed as an eigenvector. The multi-window convolutional layer convolves the input feature plane through different lengths of time windows to obtain different feature maps. The pooling layer reduces the dimension of the feature map to obtain a pooled feature vector. The output layer and the pooling layer are fully connected network structures.

b. Definition 2 (Convolution layer): assume that there are N time windows of the feature and each time window has K user preference feature for the commodity. Then input sample × can be expressed as a matrix of T × K. The feature map in the convolutional layer is calculated by the input layer and the convolution kernel. The window length of the convolution kernel is h. $x_{\mathrm{i,i+j}}$ represents the eigenvector added by time window i and time window i + j. The convolution kernel w can be expressed as a vector of h × K. Feature map $f = [f_1, f_2, \ldots, f_{\mathrm{T-h+1}}]$. The i-th feature fi is calculated according to Formula (6):

$$f_i = \sigma(w \cdot x_{i,i+h-1} + b) \tag{6}$$

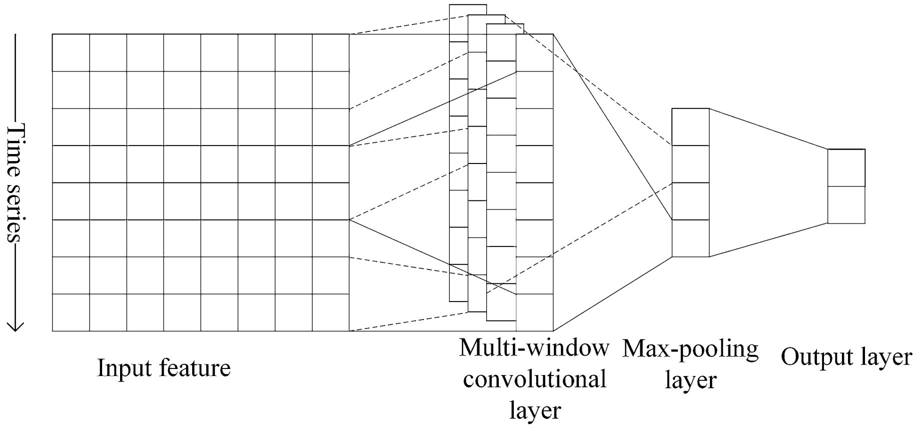

Fig. 2 The framework of the LBCNN

where b is an offset term and a real number. σ(x) is a nonlinear activation function. This paper uses ReLu and Tanh as an activation function. Relu is shown in Formula (7):

$$\begin{aligned} ReLu &= max(0, x). \\ \mathrm{Tanh}(x) &= \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}. \end{aligned} \tag{7}$$

c. Definition 3 (Max-pooling): the pooling layer is to scale the feature map while reducing the complexity of the network. The maximum features of the convolution kernel can be obtained according to the maximum pooling operation. The feature map obtained at the kth product of the convolutional kernel is $f_{\mathrm{k}} = [f_{\mathrm{k},1}, f_{\mathrm{k},2}, \ldots, f_{\mathrm{k},\mathrm{T-h}+1}]$. The pooling operation can be expressed as Formula (8):

$$Pool\_feature(j) = down(f_i). \tag{8}$$

d. Definition 4 (Probability distribution): there are M convolution kernels and the output layer has C categories [19]. The weight parameter θ of the output layer is a C × M matrix. The pooled feature $\hat{f}$ of x is an M-dimensional vector. The probability that x belongs to the i-th category can be expressed as Formula (9):

$$p(i|x, \theta) = \frac{e^{(\theta_i \cdot \widehat{f} + b_i)}}{\sum_{k-1}^{C} e^{(\theta_k \cdot \widehat{f} + b_k)}} \tag{9}$$

where $\mathrm{b_k}$ represents the k-th offset of the fully connected layer. The loss function of the model can be obtained by the likelihood probability value, as shown in Formula (10):

$$J(\theta) = -\sum_{i=1}^{k} \log(p(y_i|x, \theta)) \tag{10}$$

where T is the training data set, $\mathrm{y_i}$ is the real category of the i-th sample, xi is the characteristic of the i-th sample and θ is the model's parameters. We learn model parameters by minimizing the loss function. The training method adopts the improved gradient descent method proposed by Zeiler. In addition, we have adopted Dropout processing on the convolutional layer to prevent over-fitting of the trained model [21]. The Dropout method randomizes the neurons in the convolutional layer to 0 with a certain probability.

e. Definition 5 (Latent factor): the value of the latent factor vector is true [22]. Whether an item belongs to a class is determined entirely by the user's behavior. We assume that two items are liked by many users at the same time, then these two items have a high probability of belonging to the same class. The weight of an item in a class can also be calculated by itself. The implicit semantic model calculates the user's (u) interest in the item (i) are shown in Formula (11):

$$R(u, i) = r_{ui} = p_u^T q_i = \sum_{f=1}^{F} p_{u,k} q_{i,k} \tag{11}$$

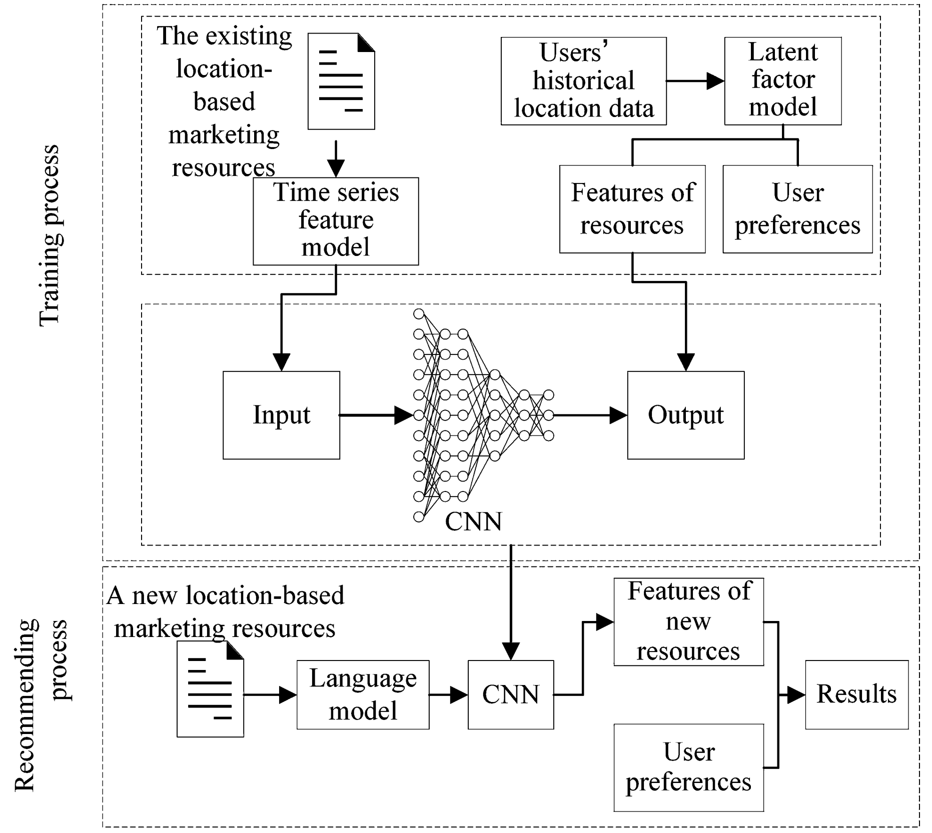
Training process
The existing location-based marketing resources
Time series feature model
Users' historical location data
Latent factor model
Features of resources
User preferences
Input
CNN
Output
Recommending process
A new location-based marketing resources
Language model
CNN
Features of new resources
User preferences
Results

Fig. 3 Location-based mobile marketing recommendation model by convolutional neural network

where p is the relationship between the user interest and the kth implicit class. q is the relationship between the kth implicit class and the item i. F is the number of hidden classes, and r is the user's interest in the item.

Specific implementation of the model

We can draw from Fig. 3 that the proposed model is divided into two processes. The first process is the training process and includes two parts. The top module shows how to generate CNN inputs and outputs from historical data. The other module in the training process shows that the traditional CNN parameters are trained by provided data. The second process finished a new location-based marketing resources recommendation. The recommendation process can work through the CNN parameters provided by the training process.

To achieve the features of users and location-based mobile marketing resources, the latent factor model (LFM) is used. In traditional LFM, L_2-norm regularization is often used to optimize training results. However, using L_2-norm regularization often leads to excessive smoothing problems. In our model, LFM results are used to represent the characteristics of the training data. In this kind of thinking, we can learn from the training method of regression coefficient in regression analysis, and construct a loss function. Therefore, it is more reasonable to use sparseness before the specification results. Based on these analyses, we propose an improved matrix decomposition method and try to normalize the solution by using the premise of verifying the sparseness of the matrix. The model is presented as Formula (12):

$$J(U, V) = \sum_{u,i \in K} \left(r_{u,i} - \sum_{k=1}^{k} p_{u,k} q_{i,k} \right)^2 + \lambda \|p_{uk}\|^2 + \lambda \|q_{ik}\|^2. \quad (12)$$

The next question is how to calculate these two parameters p and q. For the calculation of this linear model, this paper uses the gradient descent method. In the Formula (12), p_{uk} is a user bias item that represents the average of a user's rating. q_{ik} is an item offset item that represents the average of an item being scored. The offset term is an intrinsic property that indicates whether the item is popular with the public or a user is harsh on the item. For positive samples, we specify $r_{u,i} = 1$ based on experience and negative sample $r_{u,i} = 0$, which is shown in Formula (11). The latter λ is a regularization term to prevent overfitting.

a. Description of the training section

In Fig. 3, If you want to train CNN, the first thing you need to solve is its input and output problems. For input, a language model is usually used.

In terms of output, we propose an improvement in model training by LFM, which is constrained by the regularization of the L1-norm [23]. LFM training data is a historical score between the user and the location-based marketing resources. The rating score can be explicit because it is based on a user tag or an implied tag and it is predicted from the user's behavior. In this model, in order to ensure that the trained model is representative, the training data we input is to select the existing authoritative standard training set.

b. Description of the recommended part

Once the LBCNN model structure is established and the model parameters are trained using the training data set, the recommended real-time performance can be achieved. The real-time performance is based on the update of network model parameters in the background, and it uses some past behavior data and information of the recommended people and products.

User information and product information can be obtained in advance and digitized. In the offline training model phase, digitized user information, product information, and behavior information are utilized [24]. The same model is trained for the same type of users, and the parameters of the model are periodically updated within a certain period of time. In the real-time recommendation stage, real-time recommendation can be realized only by integrating the collected behavior data with the previous data and inputting it into the model.

Experimental analysis

In order to verify the advantages of convolutional neural network in capturing user's timing preferences for product and mining users' temporal behavior characteristics, we compare several commonly used classification models under the same conditions of training features. They are Linear Logistic Regression Classification Model (LR), Support

Vector Machine (SVM), Random Forest Model (RF) and Gradient Boosting Regression Tree Model (GBDT) [25]. We also compare the products that have been visited for the last 8 h. Experimental tool is sklearn kit. The hyper parameter settings for each model during the experiment are:

a. LR: select L2 regular and the regularization coefficient is 0.1.
b. SVM: choose radial basis kernel function (RBF) and gamma of kernel function is 0.005.
c. RF: the number of trees is 200, the entropy is selected as the feature segmentation standard and the random feature ratio is 0.5.
d. GBDT: the number of trees is 100, the learning rate is 0.1 and the maximum depth of the tree is 3.

Description of the data set

The experiment in our paper uses the dataset disclosed according to the Alibaba Group's mobile recommendation algorithm contest held in 2015. This data set contains 1 month of user behavior data and product information. The user's behavior data includes 10 million users' various behaviors on 2,876,947 items. Behavior types include clicks, shopping carts and purchases. In addition, each behavior record identifies behavior time that is accurate to the hour. The product information includes product category information, and identifies whether the product is an online to offline type. In a real business scenario, we often need to build a personalized recommendation model for a subset of all products. In the process of completing this task, we not only need to take advantage of the user's behavior data on this subset of goods, but also need to use more abundant user behavior data. We need to define the following symbols: U (User collection), I (Product collection), P (Product subset, $P \subseteq I$), D (User behavior data collection for the complete set of products). Our goal is to use D to construct a recommendation model for users in U to products in P.

The data mainly consists of two parts. The first part is the mobile behavior data (D) of 10 million users on the product collection, including the following fields, as shown in Table 2.

For example, "141278390, 282725298, 1, 95jnuqm, 5027, 2014-11-18 08" is one of the data. The Behavior_type and the Time in these fields contain the largest amount

Table 2 The mobile behavior data of the Ali mobile recommendation data set

Field	Field description	Extraction instruction
User_id	User differentiation	Sampling and data masking
Item_id	Product differentiation	Data masking
Behavior_type	The type of behavior of the user on the product	Including browsing, collecting, adding shopping carts, and purchasing, the values are 1, 2, 3, 4 respectively
User_geoinfo	The spatial reference identifier of the user's location	Formed by latitude and longitude data through a secret algorithm
Item category	Product classification identifier	Field masking
Time	Action time	Accurate to hour level

Table 3 The product subset of the Ali mobile recommendation data set

Field	Field description	Extraction instruction
Item_id	Product differentiation	Sampling and data masking
Item_geohash	Spatial information of the product location, which can be empty	Formed by latitude and longitude data through a secret algorithm
Item_category	Product classification identifier	Data masking

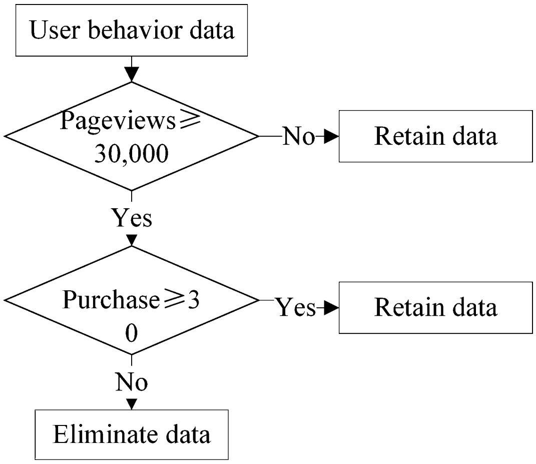

Fig. 4 Data filtering rules

of information. The User_geohash field is basically unusable due to too many missing values.

The second part is the product subset (P), which contains the following fields, as shown in Table 3.

Similar to the above, "117151719, 96ulbnj, 7350" is one of the product information. The training data contains the mobile behavior data (D) of a sample of a certain user within 1 month (11.18–12.18). The scoring data is the purchase data of the product subset (P) by these users 1 day (12.19) after this 1 month. We should be training the data model to output the predicted results of the user's purchase behavior on the next day.

Data preprocessing

We found that there are some users have a lot of page views (maximum of 2 million), which is beyond reasonable levels. We analyze that these users may be crawler users, so the behavior of these users on the goods is not the basis for predicting the user's purchase. At the same time, we predict the user product pairs that have appeared in all historical records. The existence of these users will undoubtedly increase our forecasting amount and interfere with our normal model training. Therefore, we choose to filter out these users, the filtering rules are shown as Fig. 4.

Evaluation index

The purpose of the proposed method is to predict the user's purchased business in the next position based on the user's historical behavior record. Therefore, we evaluate the model with the data of the last day. The sample construction of time series method is shown in Fig. 1. F1-score can be viewed as a harmonic mean of accuracy and recall. At present, F1-score has been widely used in the evaluation of the recommendation system.

$$precision = \frac{\left|prediction\_set \cap answer\_set\right|}{\left|prediction\_set\right|} \tag{13}$$

$$Recall = \frac{\left|prediction\_set \cap answer\_set\right|}{|answer\_set|} \tag{14}$$

$$F1 - score = \frac{2 \times precision \times recall}{precision + recall} \tag{15}$$

where Formula (13) is the calculation method of the accuracy rate, Formula (14) is the calculation method of the recall rate, and Formula (15) is the calculation method of F1-score. Prediction_set is the predicted purchase of the user-item. Answer_set is a real-purchased user-item collection.

The distribution of positive and negative samples used in this experiment is extremely unbalanced, and negative samples contain more noise. In order to make the model more suitable for learning under unbalanced data, we perform under sampling on negative samples. The model training process adopts AdaDelta Update Rule to adjust the parameters by using the stochastic gradient descent method. Hyper Parameters of the model are described in Table 4. The value in the table is the final hyper parameter when the error of the validation set is minimal. Convolution time window in convolution kernel is 2 and 3. The number of convolution kernels for two different length windows is 200. In this experiment, the training process needs to iterate ten times. To achieve the convergence of the model, we observe the accuracy of the training set every iteration in the model training process.

In Fig. 5, the abscissa indicates the number of iterations, and the ordinate indicates the accuracy of the sample. As we can see from the figure, the accuracy of the training set has been increasing and the verification set accuracy has declined after the fifth iteration of the model [26]. This situation shows that the model training has been overfitting after

Table 4 Parameter settings of convolutional neural networks

Parameter name	Parameter value
Activation function of convolution kernel	Tanh
Size of convolution kernel window	[2, 3]
Number of convolution kernel	400
Dropout ratio	0.5
Batch size	64
Epoch	5

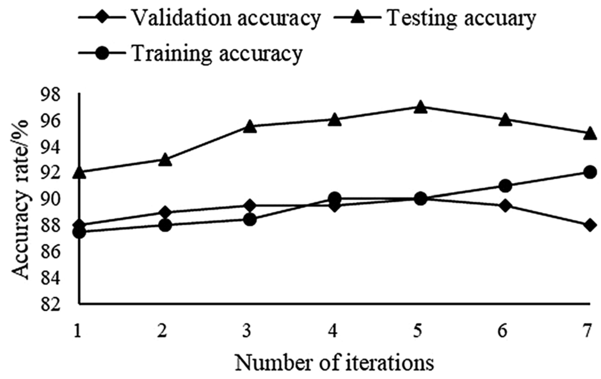

Fig. 5 Training process of the LBCNN

Table 5 Comparing the experimental results of each model (%)

Model	Accuracy	Recall rate	F1-score
Traditional method	31.4	5.60	4.02
LR	75.0	7.63	7.57
SVM	70.0	7.12	7.06
RF	57.5	5.85	5.80
GBDT	62.5	6.36	6.13
LBCNN	80.0	8.14	8.07

the 5th iteration. In addition, we found that the test set accuracy is higher than the training set and verification set.

Experimental results and comparison

The experimental results obtained using the above parameters are shown in Table 5. As can be seen from Table 5, the machine learning model using the features designed in this paper is superior to the traditional method. Our model achieves an 80% accuracy in predicting the accuracy of user behavior, which is significantly better than traditional models at least 10%. In terms of recall rate, LBCNN reached 8.14%, which is at least 2% higher than the traditional method. Similarly, our model is up to 8.07% in F1-score.

This result shows that the user's time-series behavior preference model is reasonable. This solution works well for improving the accuracy and quality of recommendations. In a single model, the LBCNN model works best. Since the linear model assumes that each feature is independent, it is impossible to excavate the intrinsic relationship between features. The proposed method can mine the intrinsic link between user timing preference features better. The experimental results show that the user preferences we build are more accurate and convolutional neural networks have strong capabilities of feature extraction and model generalization.

Conclusion

The current mobile marketing recommendation system only treats location information as a recommended attribute, which weakens the role of the location information in the recommendation. For the implicit feedback behavior of users, this paper proposes a location-based mobile marketing method by convolutional neural network. First, we divide the user location-based behaviors into several time windows according to the timestamp of these behaviors, and model the user preference in different dimensions for each window. Then we utilize the convolutional neural network to train a classifier. Finally, the experimental process of this paper is introduced, and a good prediction effect is obtained on effective data sets. The final experimental results express that the proposed method has different feature extraction perspectives from other models. Because of using convolutional neural networks, the proposed method has stronger capability of feature extraction and generalization. This method helps to change the accuracy and quality of the recommendation system and user satisfaction.

The work introduced here is to show the prospects for further research. The method proposed in this paper has a certain dependence on the user's geographical location information during the training process of the user preference model. In addition, the recommendation system will encounter a cold-start problem with sparse user information. For dealing with these discovered issues, we plan to use the hot start case to improve the recommended cold start problem. Meanwhile, we are investigating new method which uses a better big data framework (such as Hadoop MapReduce) to ensure the efficiency of training large data sets. In the future, we will show recommended methods to improve performance in other applications.

Authors' contributions

CY conceptualized the study and analyzed all the data. SD performed all experiments and wrote the manuscript. JW advised on the manuscript preparation and technical knowledge. All authors read and approved the final manuscript.

Author details

[1] School of Computer and Software, Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science & Technology, Nanjing 210044, China. [2] School of Computer & Communication Engineering, Changsha University of Science & Technology, Changsha 410004, China.

Acknowledgements

It was supported by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD), Postgraduate Research & Practice Innovation Program of Jiangsu Province (KYCX18_1032).

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

We declared that materials described in the manuscript will be freely available to any scientist wishing to use them for non-commercial purposes.

Funding

This work was supported by the National Natural Science Foundation of China (61772282, 61772454, 61811530332, 61811540410).

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 25 September 2018 Accepted: 4 April 2019
Published online: 01 May 2019

References

1. Fernández-Tobías I, Braunhofer M, Elahi M, Ricci F, Cantador I (2016) Alleviating the new user problem in collaborative filtering by exploiting personality information. User Model User Adap Inter 26(2–3):221–255

2. Koohi H, Kiani K (2016) User based collaborative filtering using fuzzy C-means. Measurement 91:134–139
3. Xu X, Fu S, Qi L, Zhang X, Liu Q, He Q, Li S (2018) An IoT-oriented data placement method with privacy preservation in cloud environment. J Netw Comput Appl 124:148–157
4. Yingyuan X, Pengqiang A, Ching-Hsien H, Hongya W, Xu J (2015) Time-ordered collaborative filtering for news recommendation. China Commun 12(12):53–62
5. Kaminskas M, Ricci F (2011) Location-adapted music recommendation using tags. In: International conference on user modeling, adaptation, and personalization. pp 183–194
6. Zhu H, Chen E, Xiong H, Yu K, Cao H, Tian J (2015) Mining mobile user preferences for personalized context-aware recommendation. ACM Trans Intell Syst Technol TIST 5(4):58
7. Yin H, Cui B, Chen L, Hu Z, Zhang C (2015) Modeling location-based user rating profiles for personalized recommendation. ACM Trans Knowl Discov Data TKDD 9(3):19
8. Li X, Xu G, Chen E, Li L (2015) Learning user preferences across multiple aspects for merchant recommendation. In: 2015 IEEE international conference on data mining (ICDM). pp 865–870
9. Yin C, Xi J, Sun R, Wang J (2018) Location privacy protection based on differential privacy strategy for big data in industrial internet-of-things. IEEE Trans Industr Inf 14(8):3628–3636
10. Lian D, Ge Y, Zhang F, Yuan NJ, Xie X, Zhou T, Rui Y (2015) Content-aware collaborative filtering for location recommendation based on human mobility data. In: 2015 IEEE international conference on data mining. pp 261–270
11. Lee WP, Tseng GY (2016) Incorporating contextual information and collaborative filtering methods for multimedia recommendation in a mobile environment. Multimedia Tools Appl 75(24):16719–16739
12. Bengio Y (2009) Learning deep architectures for AI. Found Trends Mach Learn 2(1):1–127
13. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444
14. Yuan C, Li X, Wu QJ, Li J, Sun X (2017) Fingerprint liveness detection from different fingerprint materials using convolutional neural network and principal component analysis. Comput Mater Continua 53(4):357–372
15. Yin C, Wang J, Park JH (2017) An improved recommendation algorithm for big data cloud service based on the trust in sociology. Neurocomputing 256:49–55
16. Längkvist M, Karlsson L, Loutfi A (2014) A review of unsupervised feature learning and deep learning for time-series modeling. Pattern Recogn Lett 42:11–24
17. Tu Y, Lin Y, Wang J, Kim JU (2018) Semi-supervised learning with generative adversarial networks on digital signal modulation classification. Comput Mater Continua 55(2):243–254
18. Nilashi M, Bin Ibrahim O, Ithnin N (2014) Hybrid recommendation approaches for multi-criteria collaborative filtering. Expert Syst Appl 41(8):3879–3900
19. Zeng D, Dai Y, Li F, Sherratt RS, Wang J (2018) Adversarial learning for distant supervised relation extraction. Comput Mater Continua 55(1):121–136
20. Yin C, Xia L, Zhang S, Sun R, Wang J (2018) Improved clustering algorithm based on high-speed network data stream. Soft Comput 22(13):4185–4195
21. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105
22. Li X, Yao C, Fan F, Yu X (2017) A text similarity measurement method based on singular value decomposition and semantic relevance. J Inf Process Syst 13(4):863–875
23. Li CN, Shao YH, Deng NY (2015) Robust L1-norm two-dimensional linear discriminant analysis. Neural Networks 65:92–104
24. Fattah MA (2017) A novel statistical feature selection approach for text categorization. J Inf Process Syst 13(5):1397–1409
25. Wang Y, Feng D, Li D, Chen X, Zhao Y, Niu X (2016) A mobile recommendation system based on logistic regression and Gradient Boosting Decision Trees. In: International joint conference on neural networks. pp 1896–1902
26. Yin C, Zhang S, Xi J, Wang J (2017) An improved anonymity model for big data security based on clustering algorithm. Concurr Comput Pract Exp 29(7):e3902