Neurocomputing 412 (2020) 372–380

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Popularity prediction on vacation rental websites

Yang Li [a], Suhang Wang [b], Yukun Ma [d], Quan Pan [c], Erik Cambria [a,*]

[a] Nanyang Technological University, Singapore
[b] Pennsylvania State University, United States
[c] Northwestern Polytechnical University, China
[d] Continental A.G., Singapore

ARTICLE INFO

Article history:
Received 1 February 2020
Revised 10 April 2020
Accepted 28 May 2020
Available online 12 June 2020
Communicated by Oneto Luca

Keywords:
Vacation rental websites
Popularity prediction
Dual-gated recurrent unit
Inter-event time and rating score

ABSTRACT

In the personal house renting scenario, customers usually make quick assessments based on previous customers' reviews, which makes such reviews essential for the business. If the house is assessed as popular, a Matthew effect will be observed as more people will be willing to book it. Due to the lack of definition and quantity assessment measures, however, it is difficult to make a popularity evaluation and prediction. To solve this problem, the concept of house popularity is well defined in this paper. Specifically, the house popularity is decided by inter-event timeand rating score at the same time. To make a more effective prediction over these two correlated variables, a dual-gated recurrent unit (DGRU) is employed. Furthermore, an encoder-decoder framework with DGRU is proposed to perform popularity prediction. Empirical results show the effectiveness of the proposed DGRU and the encoder-decoder framework in two-correlated sequences prediction and popularity prediction, respectively.

1. Introduction

As an increasingly popular application of online booking services, vacation rental websites such as Airbnb[1] and FlipKey[2] allow people to list, discover, and book accommodations around the world at anytime and from anywhere. These services not only benefit travelers but also house owners. For example, Airbnb has helped more than 60 million people live unique travel experiences in more than 34,000 cities and 191 countries, and provided more than 20 million house owners the opportunity to earn extra money from their accommodations. In order to better understand user behavior and provide better online services, research about online rental websites has attracted increasing attention [1–3]. However, the majority of existing works in the literature mainly focus on price prediction and house recommendation to both owner and tenant. For example, Li et al. [1,2] proposed a multi-clustering model to perform price recommendation for house owners. Grbovic et al. [3] obtained consistent search ranking results by learning house-specific embeddings. To the best of our knowledge, however, there are no works on house popularity evaluation due to the lack of definition and measurement of the problem.

On online vacation websites, house popularity evaluation is a significant issue that affectsboth house owners and tenants, because popular houses will have a larger potential to provide house owners with higher earnings, provide tenants with a better experience, and also increase the popularity of the vacation rental website. However, there is still no clear definition of house popularity and few works are about house popularity evaluation in the last decade. Moreover, house popularity may be time-dependent: a house may be popular in summer for its proximity to the sea, but it may never be booked in winter.

In order to evaluate house popularity, we need to make all itsimplicit variables explicit. Traditionally, house review information which is one of the key variables is considered. As shown in Fig. 1, we know that the review information includes review content, timestamp, rating score, etc. To capture the dynamic character of house popularity, the inter-event time (IET) [4], i.e., the time gap between two successive reviews is considered. Rating score is another key variable in house popularity evaluation. These two variables are usually negatively correlated because a high rating score and a small IET usually lead to a high chance of renting.

In order to evaluate house popularity over time, we propose to predict both IET and rating score concurrently using a dual-gated recurrent unit (DGRU) network. Also, an encoder-decoder framework with DGRU is proposed to predict house popularity in an

* Corresponding author.
E-mail addresses: yang.li@ntu.edu.sg (Y. Li), szw494@psu.edu (S. Wang), yukun.ma@continental-corporation.com (Y. Ma), quanpan@nwpu.edu.cn (Q. Pan), cambria@ntu.edu.sg (E. Cambria).

[1] https://airbnb.com/.
[2] www.flipkey.com/.

https://doi.org/10.1016/j.neucom.2020.05.092
0925-2312/

Fig. 1. An example of the reviews for a house.

end-to-end way. The main contributions of this paper are listed below:

- We propose a DGRU to predict IET and rating score concurrently.
- We propose a new definition of house popularity.
- We propose a popularity prediction framework using DGRU as encoder-decoder.

The rest of the paper is organized as follows: Section 2 introduces related works; Section 3 describes the preliminaries this paper involve; Section 4 illustrates the DGRU model and the encoder-decoder framework, respectively; Section 5 discusses experimental results; finally, Section 6 concludes the paper and introduces future works.

2. Related works

In this paper, a DGRU is proposed to perform time-dependent value prediction. Therefore, related works include RNNs, time-dependent data, and popularity prediction.

2.1. Recurrent neural networks

Because of the effectiveness over series data processing, many variants of RNN have been proposed. To learn features from both directions in RNN, the forward and backward computation are stacked together [5]. To solve the gradient vanishing problem, long short-term memory (LSTM) networks [6] and gated recurrent unit (GRU) [7] are proposed. Then, plenty of works about the sequential data processing are applied with those two models [8,9]. Because there is no memory unit and controlling mechanism for the status exposing in GRU, some papers concluded that GRU is better than LSTM both in time efficiency and computation complexity [10]. Therefore, in this paper, to make the model simple, GRU is taken into consideration in our model designing.

2.1.1. GRU

GRU has become an efficient model in sequential data learning since it has been proposed [7]. The model details are described in Eq. (1). There are two gates in GRU, g_i is the reset gate, which allows the hidden states to ignore any information irrelevant to future output. z_i denotes the update gate, which controls whether the information can be carried to the current hidden state from previous hidden one. The variable $\tilde{h}_i$ is a new hidden state determined by the reset gate, and if the value is small, the previous information will be ignored. When there is much information from sequential data, the hidden state $\tilde{h}$ needs to be reconsidered.

$$\begin{aligned} g_i &= \sigma(W_g[h_{i-1}, x_i]) z_i = \sigma(W_z[h_{i-1}, x_i]) \tilde{h}_i \\ &= tanh(W_{\tilde{h}}[(g_i * h_{i-1}), x_i]) h_i = (1 - z_i) h_{i-1} + z_i \tilde{h}_i \end{aligned} \tag{1}$$

GRU has only one input, which is x in the above equation. In some scenarios, however, it is inevitable to consider two correlated variables at the same time. As mentioned earlier, IET and rating score are two negatively correlated sequential data which cannot be predicted by GRU, concurrently. Fortunately in this paper, the proposed model DGRU has such ability.

2.2. Time-dependent data

It can be categorized into two parts for the sequential data [11]. One is the time-independent data. It contains no time information, but it has a logical order, such as sentences. Another one is the time-dependent data, and it contains a time range i.e. IET between two events, for example, the time for next clinical visitation. One favourite topic for the time-dependent data is to make a time prediction (e.g., when the next event will happen) based on previous events. Choi et al. [12] predicted the upcoming visit by building the model over the clinical visiting using the RNN. Li et al. [11] predicted the next event by applying the IET as the regularization during the training. Also, IET can be the input for the event prediction. Using IET as the weight of LSTM cells, Baytas et al. [13] proposed the time-aware LSTM networks to enhance the time effect of patient subtyping based on clinical visitation data. Different from the previous works, our model takes IET and rating score into consideration simultaneously. Additionally, the time-elapse nature of IET can also be treated as the variable weight to adjust the model.

2.3. Popularity prediction

There are plenty of works about popularity detection, most of which are based on clustering method [14]. Generally, the concept of popularity is well defined in terms of different tasks, but none of them are similar to our case. E.g., in social networks, the popularity of an event depends on the spread of the topic within a short time [15], and it can also be determined by the trend words in the events [16]. In the tourism paths recommendation, the popularity of the point of interest (POI) is determined by the adjective words that describe over the social media [17]. In the network security, the popularity of the hidden service over the Tor network is defined by the request rate that sends from the clients [18]. In this work, we use IET and rating score these two variables to define house popularity. As we have discussed, our definition has the potential in following the instinct that high frequency reviews and high rating scores can lead to a popular house.

3. Preliminary data analysis

In this section, we will use preliminary data analysis to illustrate why IET and rating scores are important indicators of house popularity. There are three steps which are analysing the relationship between IET and rating score, defining house popularity based on this relationship, and designing the popularity prediction framework.

3.1. Relation between IET and rating score

Popular houses usually have high reservation rates and high reviews, which means popular houses usually have small IETs and high rating scores. Thus, IET and rating scores can be two variables to indicate house popularity. In this subsection, the relation between IET and rating scores is analyzed.

The house review data is time-dependent. From the observation, we can see that one bad rating score leads to the review is less likely to appear in a short time, and vice versa. These phenomena are validated in Fig. 2. In the figure, the "Before rating" denotes rating behavior before an IET, which means $((t_k - t_{k-1}), r_{k-1})$ is treated as an evidence pair. In the "Before rating" evidence pair, rating score is the independent variable with a range of $[0, 5]$, and IET is the dependent variable. Accordingly, there are only 6 points in the left subfigure. The "After rating" means rating score happens after an IET, and $((t_k - t_{k-1}), r_k)$ is treated as the evidence pair. In this situation, IET is the independent variable and rating score is the dependent variable. It can be seen from the observation that the evidence pairs of "After rating" and "Before rating" are negatively correlated.

Intuitively, IET is an essential parameter for the house renting. Besides, from Fig. 2, we can see that ("Before rating", IET) has a stronger correlation than (IET, "After rating") evidence pair. If the rating score is "Before rating", more features from rating score will be adopted in the IET prediction. Therefore, we choose ("Before rating", IET) as the data and the rating score in this paper refers to the "Before rating".

3.2. House popularity

Lemma 3.2.1. *In the rental websites, the house popularity is a trend which is decided byIET t and rating score r at the same time.*

The popularity is a trend. To quantitatively analyze the house popularity, the average IET $\tau = \frac{\sum_i^{n_h}(t_{i+1}-t_i)}{n_h}$ and the rating score summation $\gamma = \sum_i r_i$ are calculated. τ represents the interval between different bookings: the smaller the better. γ represents the tenant's satisfaction with the house: the bigger the better. The statistic results about these two values are shown in Fig. 3. In the figure, the diameter of the bubble is proportional to the renting number. Bigger bubbles mean more rents. In particular, the house with small τ and big γ usually has a large diameter bubble, which means the house is booked frequently with high satisfying accommodation, i.e., the house is popular.

At first, a house may not be popular. If most tenants are satisfied with the accommodation and give high review rates, it will stimulate more people to book. This phenomenon is shown in the small figure in Fig. 3. Each bubble denotes current τ_t and γ_t at time t. Then, the evolution of the house popularity can be studied in terms of τ and γ. In the beginning, the house is not popular when it is a small bubble, where τ is large and γ is small. Over time, however, the house becomes popular with a smaller τ and a larger γ.

Therefore, the rating summation threshold γ_r and the average IET threshold τ_t are utilized to define the house popularity. Based on those two thresholds, the panel in Fig. 3 is divided into four sections. When γ is larger than γ_r and τ is smaller than τ_t, the house will be popular , and it is the section D in Fig. 3. When γ is smaller than γ_r and τ is smaller than τ_t, the house has large potential to be popular, which is the section C in Fig. 3. When γ is smaller than γ_r and τ is bigger than τ_t, the house is not so popular and needs to improve a lot, and this is shown in section A in Fig. 3. The section B is the house with a higher rating score, but the small number of renting leads to a large τ. Thus, the house in this section needs more attention during advertising. According to the division, the house is marked by these four labels to represent the degree of popularity in different periods.

Based on these settings, w.r.t., the house popularity depends on the trend of the τ and γ, we design a house popularity prediction framework which will be introduced in Section 4.2.

4. Methodology

The IET $\mathbf{t} \in \{t_1, t_2, \ldots, t_n\}$ and the rating score $\mathbf{r} \in \{r_1, r_2, \ldots, r_m\}$ are the time-dependent and correlated sequential data. To make

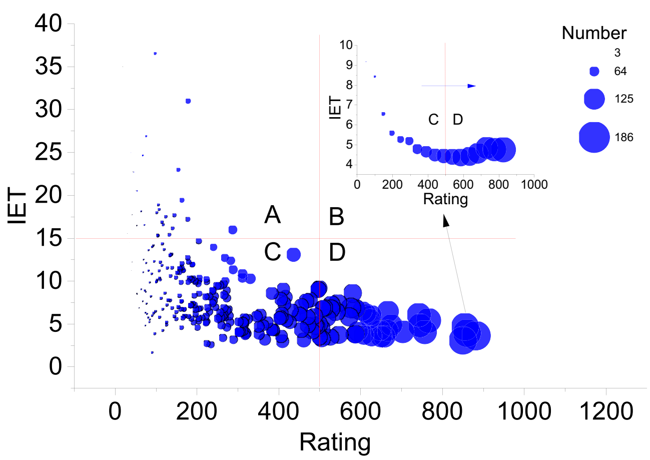


Fig. 3. The x axis is the total rating of a house that it has, the y axis denotes the average IET of the house, the diameter of the bubble denotes the renting number of a house. The small figure over the arrowed line is a single case at that point, each bubble in that small figure represents the average IET and total rating before that time.

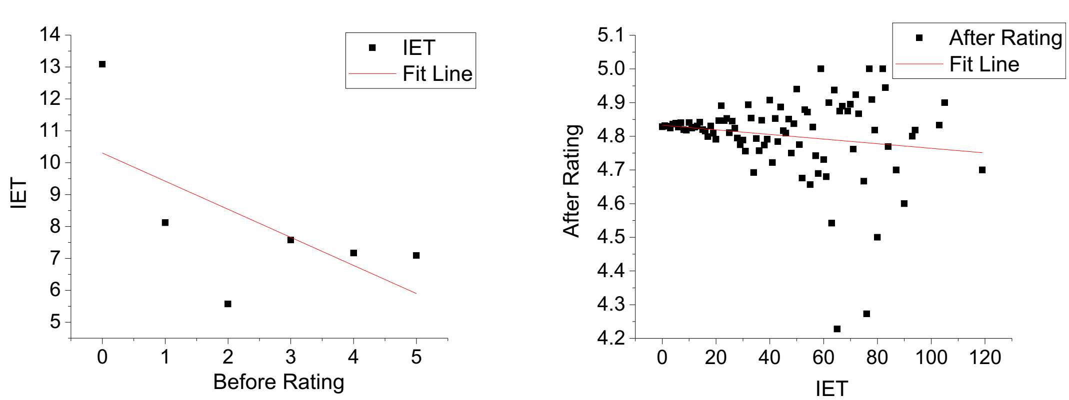


Fig. 2. The left figure shows the relation between the Before Rating and the IET, and the right figure shows the correlation between the IET and the After Rating.

IET predictable, it can be represented by one-hot vector $t_i \in R^{1\times n}$, where n means the maximum IET days. Similarly, rating score also is encoded by one-hot vector $r_j \in R^{1\times m}$, where m denotes the rating range. IET is measured by days, and its range is $[1, 400]$, while the range of the rating score is $[0, 5]$. As we know, there should be no difference in time representation when IETs are close, e.g., 7 days and 8 days , but it is hard to be embodied by one-hot vector. To overcome this disadvantage, a trainable embedding layer is added after the input layer, which maps the one-hot vector to a 16-dimensional vector.

To predict when the next comment will come and what the rating score will be based on the previous review information, we need to learn the function that maps the feature from $\{(t_1, r_1), (t_2, r_2), \ldots, (t_{k-1}, r_{k-1})\} \rightarrow (t_k, r_k)$. When predicting the house popularity, it is necessary to extract features from those sequential data and then make the prediction based on the learned features. This procedure can be described as $F(\{(t_1, r_1), (t_2, r_2), \ldots, (t_k, r_k)\}) \rightarrow Y$, where F denotes features extraction function, and Y is the popularity label of the house. Next, we will give designing details about DGRU and popularity prediction framework in this section. An overview of DGRU is shown in Fig. 4.

4.1. Dual-Gated Recurrent Unit

Time elapsed is inversely proportional to the effectiveness of online reviews as users only tend to read most recent reviews. In addition, renting behavior changes over time: if the time gap between two events is large, IET will have little impact on the prediction, and vice versa. Hence, the current IET t is transformed into the weight in our model. Inspired from [13], it can be $1/t$. In this paper, because of a large elapsed time among reviews in Airbnb, $f(t) = 1/\log(e + t)$ is applied. To strengthen the effectiveness of IET, the previously hidden states are updated with the current IET, and this is in Eq. (2).

$$\tilde{t}_i = \sigma(W_{it} t_i)\tilde{h}^t_{i-1} = f(t_i) * \tilde{t}_i + h^t_{i-1} \tag{2}$$

Except for the hidden value from the last layer, the GRU cell has only one input, which makes it difficult to learn two correlated variables simultaneously. The direct idea is to parallelize two GRU cells, and one for each variable. The relationship between the two variables, however, will be ignored and the model complexity will be doubled. Thus, a newly designed GRU is proposed. As we know, the hidden value $\tilde{h}$ is updated with the reset gate which is to decide the degree in the past information forgetting. To design a cell that can learn patterns from r and t at the same time, $\tilde{h}$ is treated as the connection point by combing these two variables. That is to say, $\tilde{h}$ is the composition of the two correlated variables. Accordingly, the designing of DGRU will start from the variable $\tilde{h}$. For GRU, the g_i and z_i is the reset gate and the update gate, respectively. In DGRU, each variable needs its gates correspondingly. The basic procedure about DGRU is depicted in Eq. (3).

$$\begin{aligned} g^t_i &= \sigma(W_{gt}[\tilde{h}^t_{i-1}, \tilde{t}_i])z^t_i = \sigma(W_{zt}[\tilde{h}^t_{i-1}, \tilde{t}_i])g^r_i = \sigma(W_{gr}[h^r_{i-1}, r_i])z^r_i \\ &= \sigma(W_{zr}[h^r_{i-1}, r_i])\tilde{h}_i = tanh(W_{\tilde{h}}[g^t_i; g^r_i; \tilde{h}^t_{i-1}; h^r_{i-1}, r_i, \tilde{t}_i])h^t_i \\ &= (1 - z^t_i)\tilde{h}^t_{i-1} + z^t_i \tilde{h}_i h^r_i = (1 - z^r_i)h^r_{i-1} + z^r_i \tilde{h}_i \end{aligned} \tag{3}$$

In the equation, g^t_i, z^t_i and h^t_i are the reset gate, update gate and current hidden state for the IET variable. And g^r_i, z^r_i and h^r_i are the reset gate, update gate and current hidden state for the rating variable, $\tilde{h}_i$ is the newly designed hidden state. $\{W_{gt}, W_{gr}\}$ and $\{W_{zt}, W_{zr}\}$ are the parameters of the reset gate and update gate for the two variables. $W_{\tilde{h}}$ is the parameter for the new hidden state. The dimensions of those parameters are decided by the input, output and the hidden state. There are different ways in the composition designing for the new hidden state $\tilde{h}_i$. To fully explore the way this variable is constructed, a more comprehensive $\tilde{h}$ is designed as in Eq. (4).

$$\tilde{h} = tanh\left(W_{\tilde{h}}[\frac{g^t_i * h^t_{i-1}}{\beta_t} \circ \frac{g^r_i * h^r_{i-1}}{\beta_r}; r_i; t_i]\right) \tag{4}$$

Here $\beta_t \in \{h^t_{i-1}, 1\}$ and $\beta_r \in \{h^r_{i-1}, 1\}$ are to decide which parts to ignore. If $\beta_t = h^t_{i-1}$, it means the reset gate for IET has no effect on the IET's previous hidden state h^t_{i-1} and should be ignored, however, if $\beta_t = 1$, it means h^t_{i-1} has effect on the final output and will not be ignored. It is the same for the parameter β_r. As discussed, it is the "Before rating" which is used as the rating score in the inference. Therefore, the previous state h^r_{i-1} has a greater effect than previous state h^t_{i-1}. To make the model simple and effective, the parameter β_t and β_r are set to h^t_{i-1} and 1, respectively. And experiments also verify the effectiveness of these settings. $\circ \in \{+, *\}$ denotes the calculator for these two variables, and it decides the way that the two variables joining together. Generally, the calculator $*$ will be better than the + on account of its effect enhancement. Therefore, $\circ$ is set to $*$, and its effectiveness also is validated in the experiment part. Based on those settings, the format of $\tilde{h}$ is as follows:

$$\tilde{h}_i = tanh(W_{\tilde{h}}[g^t_i * g^r_i * h^r_{i-1}, r_i, t_i]) \tag{5}$$

Therefore, the structure of the proposed DGRU is shown in Fig. 4. The input sequential data is the review data which include IET and rating score.

4.2. House popularity detection with DGRU

We propose a new encoder-decoder framework based on DGRU, which is to extract the features from the two correlated sequential data directly. The structure of the proposed framework is shown in Fig. 5.

In the encoder, to make full exploration about the sequential data, the slide window is applied over the sequential data $\mathbf{t} = \{t_1, t_2, \ldots, t_{k-1}\}$ and $\mathbf{r} = \{r_1, r_2, \ldots, r_{k-1}\}$. The window size is w. A wider window not only means more historic information will be adopted, but it also means more computing time. The best window size w is validated in the experiment part. After the encoder, the learned latent features h^r and h_t are fed into the decoder. In the decoder, it is trained with the teacher forcing way, and the outputs are $\mathbf{t} = \{t_k, t_{k+1}, \ldots, t_{k+w}\}$ and $\mathbf{r} = \{r_k, r_{k+1}, \ldots, r_{k+w}\}$. Meanwhile, the learned latent features h^t and h^r are fed into the classifier to do the popularity prediction. The output of the classifier is the popularity probability $\hat{y}$. The classifier is a two layers fully-connected network whose hidden dimensions are 256 and 16, respectively. Because the representation of r and t are one-hot vectors, the cross-entropy loss is applied in the objective function as shown in Eq. (6).

$$L = \sum_{i=1}^{w} r_i * \log(\hat{r}_i) + \sum_{i=1}^{w} t_i * \log(\hat{t}_i) + \sum_{i=1}^{4} y_i * \log(\hat{y}_i) \tag{6}$$

The first term in the right-hand side is the loss for the rating score prediction, the second term is for the IET prediction, and the last term is for the house popularity label prediction.

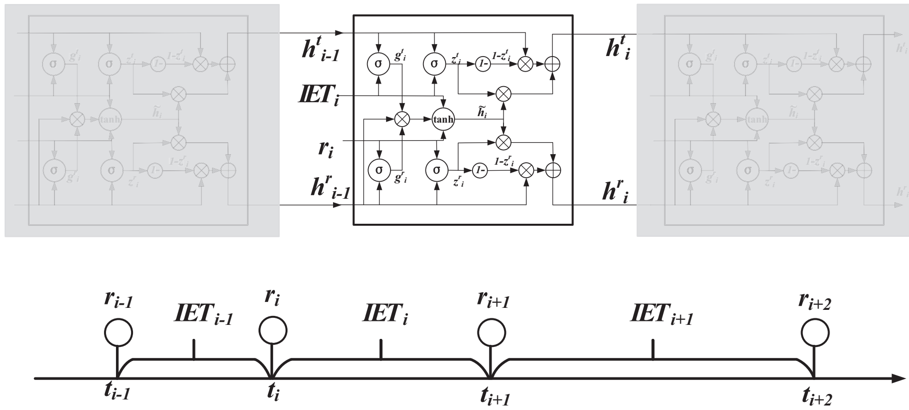


Fig. 4. The illustration of the proposed DGRU. The axis with an arrow is the event with timestamps. There are two inputs for the DGRU, which are IET and rating score, respectively.

5. Experiments

In this section, the experiments over the synthetic datasets and real-world datasets are conducted, respectively. The synthetic datasets are generated with predefined function, and the real-world datasets are crawled from Airbnb.

There are two cases for the generation of the synthetic dataset, and the sketch maps about the synthetic datasets are shown in Fig. 6. The IET t is generated with $t(k) = 5(\sin(\frac{\pi}{5}k) + 1)$ in both cases. Case one is to simulate the "Before rating" as the rating score r is generated with $r(k) = -2.5(\sin(\frac{\pi}{5}k - \frac{\pi}{2})) + 5$. It means the rating score happened $\frac{\pi}{2}$ time ago. Case two is to simulate the "After rating" as rating score t is generated with $r(k) = -2.5(\sin(\frac{\pi}{5}k + \frac{\pi}{2})) + 5$. It means the rating score will happen $\frac{\pi}{2}$ time later. Both cases follow the negative correlation between the variables of the IET t and the rating score r.

There are three cities in the real-world datasets which are Beijing, London, and Boston. We collected reviews of all the rooms in the three cities. The details about the datasets are listed in Table 1. Room Num is the room number in each city w.r.t. 302 rooms in Beijing, 271 rooms in London, and 266 rooms in Boston. Avg Review and Total Review denote the average review number for the house and the total review number in each city, respectively. For example, in Beijing, there are around 56 reviews for each house, out of a total of 16,769 reviews across the city.

All the codes are implemented in Python with Tensorflow library.[3] The parameters are listed in Table 2. The Adam [19] is applied to optimize the model.

All of the experiments are run five times to get average results. In subsection 5.1, synthetic dataset is separated into two parts, 70% data are used for training, and 30% of them are for testing. In subsections 5.2–5.4, the dataset from London is adopted. Similarly, the dataset is separated into two parts, 70% of them are used for training, and the rest are used for testing. All of the parameters are decided by the performance on the test data.

5.1. Validation with synthetic datasets

To validate the "Before rating" and the "After rating", the synthetic datasets are applied. For the description about the synthetic datasets, we generate 10,000 sequences of data with $k \in [1, 10000]$, each data has the length of 10, and the last number is treated as the target which is used for the prediction. The results are shown in Table 3. DGRU-A denotes the case one that applies the "After rating" as the rating score, DGRU-B is the case two that applies the "Before rating" as the rating score.

Acc_t is the accuracy for next IET prediction, while Acc_r denotes the accuracy for next rating score prediction. From the table, we can see that in the evaluation of Acc_t, DGRU-B performs the best, which indicates that the "Before rating" is better at the IET prediction with rating score working as the information resource. However, in the evaluation of Acc_r, DGRU-A has the best performance, which indicates that the "After rating" pair is better at rating score prediction with IET working as the information resource. On the whole, regarding the Loss and the Acc_t, the DGRU-B outperforms other models. And this shows the effectiveness of the proposed model in these two variables cooperation.

5.2. Structure validation for DGRU

To validate the effectiveness of the proposed framework, we traverse the parameters β_r, β_t, $\circ$ within their own sets. There are eight cases in total, and the results are listed in Table 4. There is a large range for the IET prediction. Thus, the tolerance for the prediction should be counted in days. Consequently, in this work, we assume that there is no difference when the error of the IET predictions is less one day. It is counted as the correct when the gap between predicted IET and real IET is smaller than 1. Therefore, the accuracy $Acc_t = \frac{\sum_i^N \mathbf{1}(|t_i - \hat{t}_i| <= 1)}{N}$. The accuracy for rating $Acc_r = \frac{\sum_i^N \mathbf{1}(|r_i = \hat{r}_i|)}{N}$, where N is the total number of the test dataset.

From Table 4, we can see that most of the cases with $\circ \rightarrow *$ come to better results compared with $\circ \rightarrow +$. Therefore, it is better to use multiply operation when doing the combination. As explained earlier, the rating score in this work denotes the "Before rating", and there is more information on the variable h^r_{i-1} than h^t_{i-1} when making the prediction. This is validated in Case 2 which has the best results compared with other cases in Acc_r and $Loss$, and it is nearly the best in Acc_r compared with case 4. These observations validate the effectiveness in ignoring the previous state h^t_{i-1} during the state forward passing. All these results prove that the proposed structure of the DGRU is the best composition.

5.3. The effect of the window size

During the IET and rating score prediction, if the window size is large, more historical information will be implied. To find out the optimal length of the window size during the feature extraction, the window size range is $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. And the

---

[3] www.tensorflow.org/.

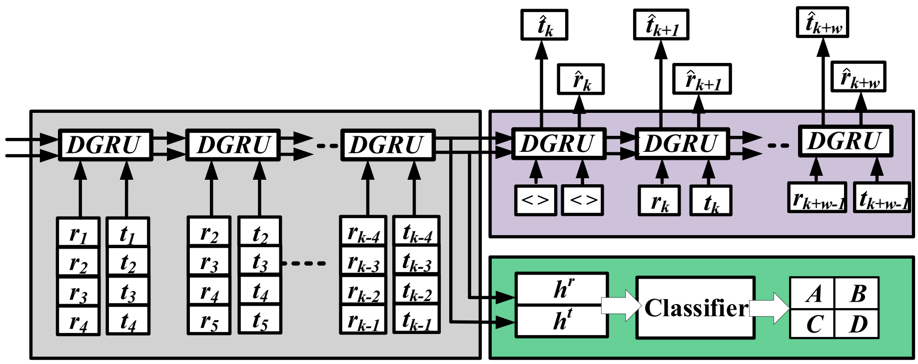
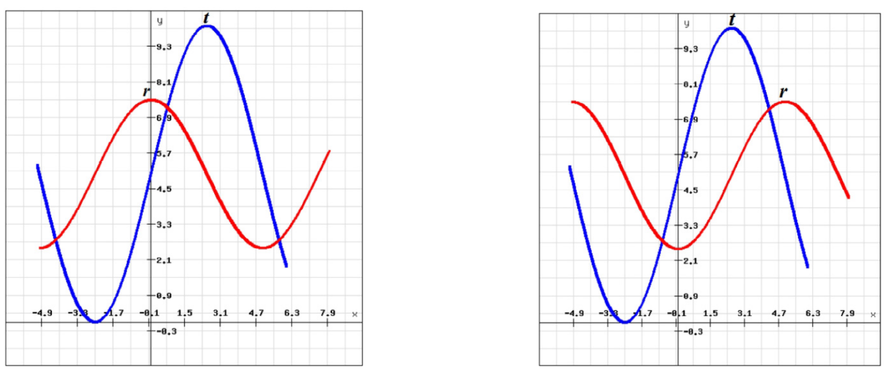


Fig. 5. The structure of the encoder-decoder framework based on the DGRU.



(a) Case of the “Before rating” (b) Case of the “After rating”

Fig. 6. The sketch maps about the synthetic data generation. The red line is the rating score r, and the blue line is the IET t. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
Details of the datasets.

City	Beijing	London	Boston
Room Num	302	271	266
Avg Review	56	95	112
Total Review	16,769	25,774	29,885

Table 2
Parameter setting.

Parameter	Value
Learning Rate	0.001
Batch Size	128
Hidden Dimension (DGRU)	50
Hidden Dimension (Classifier)	{256, 16}

Table 3
Results over synthetic data.

Model	Acc_t	Acc_r	Loss
LSTM	86.02%	83.77%	0.751
GRU	81.13%	87.77%	0.686
DGRU-A	72.56%	96.67%	1.059
DGRU-B	88.22%	92.48%	0.632

evaluation metrics are Acc_t, Acc_r and Time, respectively. Results are depicted in Fig. 7.

From the figure, we can see there is a dramatical improvement in Acc_t when the window size is smaller than 4. This shows that more information could be involved in large window size, but this leads to a longer computing time which is depicted in the third figure. However, in the rating score prediction, the model performs better when the length is 4. When the length is larger than 4, the performance becomes unstable both in IET prediction and rating score prediction. Therefore, the length of window size is set to 4 in overall consideration.

5.4. Grid search

In our proposed model, there are five parameters which are “Before rating/After Rating”, “$\circ$”,“β_r”, “β_t” and “Window Size” in total. We have a basic analysis over them in a sequential manner in Sections 5.1, 5.2, 5.3, respectively. To find the optimal parameter combination, the grid search on the real-world dataset is conducted. In the grid search, there are eight combinations within {“$\circ$”,“β_r”, “β_t”}which are listed in Table 4, and we refer to the case number in that table as the searching index. To make a clear illustration, we use two heat maps to show the grid search results which are shown in Fig. 8, one for the “Before rating” case and one for the “After rating” case. From this figure, we can see that the “After rating” panel (right one) is darker than the “Before rating” panel (left one). That means when using “After rating” pair as the input data, it is hard to train the model. Therefore, “Before rating” pair will help the model gain more features in the real world dataset. Similarly, we can draw the same conclusions as before, Case 2 (where $\circ$ is $*, \beta_r = 1$ and $\beta_t = h^t_{t-1}$) has the smallest loss in the test dataset when the length is set to 4, especially in the “Before rating” panel.

Table 4
Composition selection within $\tilde{h}$ (bold values mean the best composition for the proposed DGRU).

Cases	$\circ$	β_r	β_t	Acc_r	Acc_t	Loss
1	*	1	1	68.80% (±0.381)	40.21% (±0.003)	4.937 (±0.488)
2	*	1	h^t_{i-1}	**85.86% (±0.000)**	40.39% (±0.002)	**4.108 (±0.626)**
3	*	h^r_{i-1}	1	68.80% (±0.381)	40.18% (±0.004)	4.680 (±0.427)
4	*	h^r_{i-1}	h^t_{i-1}	51.74% (±0.467)	**40.41% (±0.003)**	4.926 (±0.713)
5	+	1	1	68.80% (±0.381)	40.21% (±0.006)	4.432 (±0.707)
6	+	1	h^t_{i-1}	**85.86% (±0.000)**	40.15% (±0.003)	4.571 (±0.415)
7	+	h^r_{i-1}	1	51.74% (±0.467)	39.52% (±0.023)	4.942 (±0.752)
8	+	h^r_{i-1}	h^t_{i-1}	68.80% (±0.381)	39.95% (±0.012)	4.712 (±0.319)

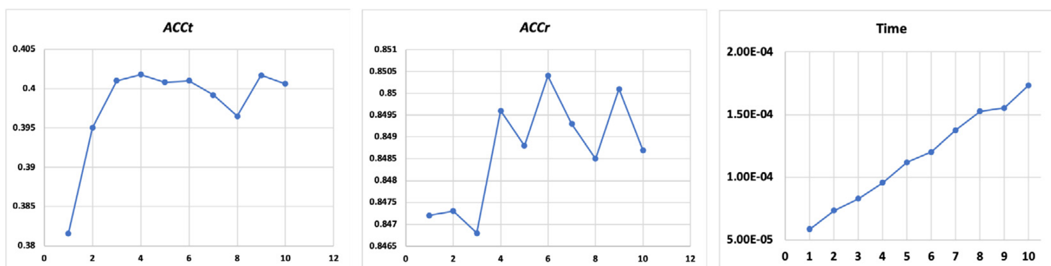


Fig. 7. The length effects about the model.

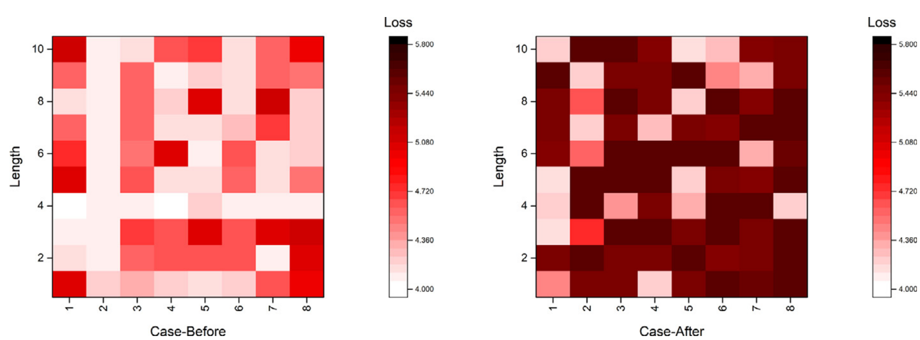


Fig. 8. The grid search over those four hyper-parameters.

5.5. The popularity prediction

To predict the popularity of the house, we slide the data from each house review with the fixed length 10, and set the window size to 4. The threshold γ_r and τ_t are two super-parameters which decide the house popularity directly. Based on the statistics and past experience, γ_r is set to 500, and τ_t is set to 15. After getting the data, we split them into two parts, one is for training the model, and the other one is used as the test data. The details of the prepared data are shown in Table 5.

The compared methods include LSTM, TLSTM, GRU, and TGRU. All of these models are fitted into the encoder-decoder framework we proposed in Fig. 5. TLSTM is cited from [13], and TGRU is implemented with the same idea from the TLSTM. The results are shown in Table 6.

In the table, the t and r behind the model name mean the data type that the model is used during the popularity detection. For example, LSTM (t) indicates the model's input is an IET sequence. When there are two sequences as input, the encoder and decoder are paralleled to both models. Then, the learned features are concatenated together as the input of the classifier.

From the results, we can see that the popularity accuracy with IET as input is far better than that with rating sequence as input, and this implies that there are more features in the IET sequence when doing the popularity detection. When using the IET and rating score sequences at the same time, the frameworks with LSTM, GRU, TLSTM as the encoder and decoder, e.g., LSTM (t+r), GRU (t+r),

Table 5
Training and test data.

City	Beijing	London	Boston
Train Num (P)	10,858	17,647	20,918
Test Num (P)	1,119	3,669	4,774

Table 6
The prediction results over different datasets.

City	Beijing	London	Boston
LSTM (t)	32.46%	52.89%	62.52%
LSTM (r)	26.19%	36.73%	41.80%
LSTM (t+r)	30.11%	43.37%	41.19%
GRU (t)	31.37%	53.62%	59.56%
GRU (r)	19.67%	31.63%	36.34%
GRU (t+r)	33.40%	40.44%	45.15%
TLSTM (t)	34.96%	50.59%	64.23%
TLSTM (t+r)	37.50%	42.79%	52.86%
TGRU (t)	32.56%	39.80%	50.23%
TGRU (t+r)	40.89%	41.50%	51.78%
DGRU	41.66%	55.73%	65.37%

TLSTM (t+r), do not perform very well, and in most cases, these results are no better than that with IET sequence as input, e.g., LSTM (t), GRU (t), TLSTM (t). That means when there are two inputs, the features from LSTM, GRU and TLSTM cannot cooperate well. However, it is not the same in TGRU, where the results in TGRU (t+r) are better than that with TGRU (t). That means the features learning from TGRU can cooperate better. All in all, DGRU achieves the best results, and this proves that the DGRU model has learned the latent feature well from the IET and rating score.

6. Conclusions & future works

In this paper, we proposed a new framework for predicting the popularity of vacation rental websites. In particular, the framework is based on a dual-gated recurrent unit that models inter-event time and rating score. The framework is validated through extensive experiments over both synthetic and real-world datasets. An important contribution of our work is also a clear definition of house popularity, which was a rather opaque entity in previous works. As future work, we plan to include more parameters in such a definition, including variables to model house facilities, location, and other properties that can be automatically discovered by applying aspect-based sentiment analysis to available reviews [20].

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

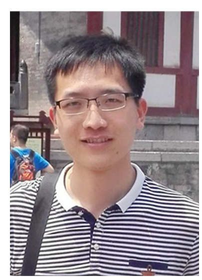
Yang Li: Conceptualization, Methodology, Software, Validation, Formal analysis, Data curation, Writing - original draft, Visualization, Resources. **Suhang Wang:** Conceptualization, Methodology, Writing - review & editing. **Yukun Ma:** Validation, Formal analysis, Investigation. **Quan Pan:** Supervision, Resources. **Erik Cambria:** Writing - review & editing, Investigation, Supervision, Resources, Funding acquisition.

Acknowledgement

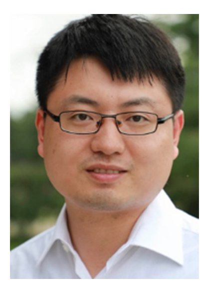
This research is supported by the Agency for Science, Technology and Research (A*STAR) under its AME Programmatic Funding Scheme (Project #A18A2b0046).

References

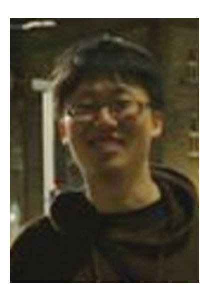
[1] Y. Li, S. Wang, T. Yang, Q. Pan, J. Tang, Price recommendation on vacation rental websites, in: Proceedings of the 2017 SIAM International Conference on Data Mining, SIAM, 2017, pp. 399–407.
[2] Y. Li, Q. Pan, T. Yang, L. Guo, Reasonable price recommendation on airbnb using multi-scale clustering, in: Control Conference (CCC), 2016 35th Chinese, IEEE, 2016, pp. 7038–7041.
[3] M. Grbovic, H. Cheng, Real-time personalization using embeddings for search ranking at airbnb, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ACM, 2018, pp. 311–320.
[4] A.-L. Barabasi, The origin of bursts and heavy tails in human dynamics, Nature 435 (7039) (2005) 207.
[5] M. Schuster, K.K. Paliwal, Bidirectional recurrent neural networks, IEEE Transactions on Signal Processing 45 (11) (1997) 2673–2681.
[6] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Computation 9 (8) (1997) 1735–1780.
[7] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using rnn encoder-decoder for statistical machine translation, arXiv preprint arXiv:1406.1078.
[8] Y. Li, Q. Pan, S. Wang, T. Yang, E. Cambria, A generative model for category text generation, Information Sciences 450 (2018) 301–315.
[9] H. Peng, Y. Ma, Y. Li, E. Cambria, Learning multi-grained aspect target sequence for chinese sentiment analysis, Knowledge-Based Systems 148 (2018) 167–176.
[10] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, arXiv preprint arXiv:1412.3555..
[11] Y. Li, N. Du, S. Bengio, Time-dependent representation for neural event sequence prediction, arXiv preprint arXiv:1708.00065.
[12] E. Choi, M.T. Bahadori, A. Schuetz, W.F. Stewart, J. Sun, Doctor ai: Predicting clinical events via recurrent neural networks, in: Machine Learning for Healthcare Conference, 2016, pp. 301–318..
[13] I.M. Baytas, C. Xiao, X. Zhang, F. Wang, A.K. Jain, J. Zhou, Patient subtyping via time-aware lstm networks, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2017, pp. 65–74.
[14] H. Borges, A. Hora, M.T. Valente, Predicting the popularity of github repositories, in: Proceedings of the 12th International Conference on Predictive Models and Data Analytics in Software Engineering, ACM, 2016, p. 9.
[15] X. Zhang, X. Chen, Y. Chen, S. Wang, Z. Li, J. Xia, Event detection and popularity prediction in microblogging, Neurocomputing 149 (2015) 1469–1480.
[16] M. Gupta, J. Gao, C. Zhai, J. Han, Predicting future popularity trend of events in microblogging platforms, Proceedings of the American Society for Information Science and Technology 49 (1) (2012) 1–10.
[17] G. Jossé, M. Franzke, G. Skoumas, A. Züfle, M.A. Nascimento, M. Renz, A framework for computation of popular paths from crowdsourced data, in: 2015 IEEE 31st International Conference on Data Engineering (ICDE), IEEE, 2015, pp. 1428–1431.
[18] A. Biryukov, I. Pustogarov, F. Thill, R.-P. Weinmann, Content and popularity analysis of tor hidden services, in: 2014 IEEE 34th International Conference on Distributed Computing Systems Workshops (ICDCSW), IEEE, 2014, pp. 188–193.
[19] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980..
[20] Y. Ma, H. Peng, T. Khan, E. Cambria, A. Hussain, Sentic LSTM: A Hybrid Network for Targeted Aspect-Based Sentiment Analysis, Cognitive Computation 10 (4) (2018) 639–650.



Yang Li Obtained his Bachelor of Information Security from Northwestern Polytechnical University in 2014, and his PhD in 2018 from the same University. Currently, he is a postdoc under the supervision of Erik Cambria in the School of Computer Science and Engineering in Nanyang Technological University. His main research interests are sentiment analysis, deep learning, and knowledge representation.



Suhang Wang is an Assistant Professor with the College of Information Sciences and Technology at Penn State University - University Park. He received his Ph.D. of Computer Science from Arizona State University in 2018, his MS degree in EE: Systems from University of Michigan – Ann Arbor in 2013, and his BS degree in ECE from Shanghai Jiao Tong University, Shanghai, China, in 2012. His research interests are in graph mining, data mining and machine learning. He has published innovative works in top conference proceedings and journals such as WWW, AAAI, IJCAI, CIKM, SDM, WSDM, ICDM, CVPR and TKDE. He serves on journal editorial boards and numerous conference program committees.



Yukun Ma is currently a senior specialist in AI - Big Data Lab of Continental Automotive Group. He obtained his PhD from NTU in the year 2018. He has a broad interest in natural language processing including sentiment analysis, representation learning, and language mode

Quan Pan was born in China, in 1961. He received the B.S. degree in automatic control from the Huazhong University of Science and Technology, in 1982, and the M.S. and Ph.D. degrees in control theory and application from Northwestern Polytechnical University. From 1991 to 1993, he was an Associate Professor with Northwestern Polytechnical University, where he has been a Professor with the Automatic Control Department, since 1997. He has authored 11 books, more than 400 articles. His research interests include information fusion, target tracking and recognition, deep network and machine learning, UAV detection navigation and security control, polarization spectral imaging and image processing, industrial control system information security, commercial password applications and modern security technologies. He is an Associate Editor of the journal Information Fusion and Modern Weapons Testing Technology.

Erik Cambria is an Associate Professor at NTU, where he also holds the appointment of Provost Chair in Computer Science and Engineering. Prior to joining NTU, he worked at Microsoft Research Asia and HP Labs India and earned his Ph.D. through a joint programme between the University of Stirling and MIT Media Lab. He is recipient of many awards, e.g., the 2018 AI's 10 to Watch and the 2019 IEEE Outstanding Early Career award. He is Associate Editor of several journals, e.g., NEUCOM, INFFUS, KBS, IEEE CIM and IEEE Intelligent Systems (where he manages the Department of Affective Computing and Sentiment Analysis), and is involved in many international conferences as PC member, program chair, and speaker.