Micro Behaviors: A New Perspective in E-commerce Recommender Systems

Minreng Wu
529005234

Contents

1. Introduction

2. Problem Definition & Analysis

3. Model Design

4. Evaluation

5. My Thoughts

1.1. Motivation

With the help of the explosive growth of e-commerce and the development of big data technologies, e-commerce websites accumulate more and more user micro-behavior data, such as reading the comments, carting, ordering and the dwell time on products.

Such micro behaviors offer fine-grained and deep understandings about users and provide tremendous opportunities to advance recommender systems in e-commerce.

In this paper, the authors propose an interpretable Recommendation framework, which models inherently the sequence of micro Behaviors and helps to improve the performance of the recommendation system.

1.2. Macro Interactions & Micro Behaviors

In Figure 1, it shows a real example of a user buying a mobile phone case on an e-commerce website.

- From a macro perspective, the user interacted with the following 4 products: iPhone 7, iPhone 6, iPhone 7 cases and Samsung Galaxy.
- From a micro perspective, each macro interaction includes a sequence of behaviors that can indicate how the user located the product page (e.g., the search engine or the sale promotion), whether the user clicks detailed information about a product (e.g., comments, or specifications), whether a user carts or orders a product, and how long the user dwells on a product.

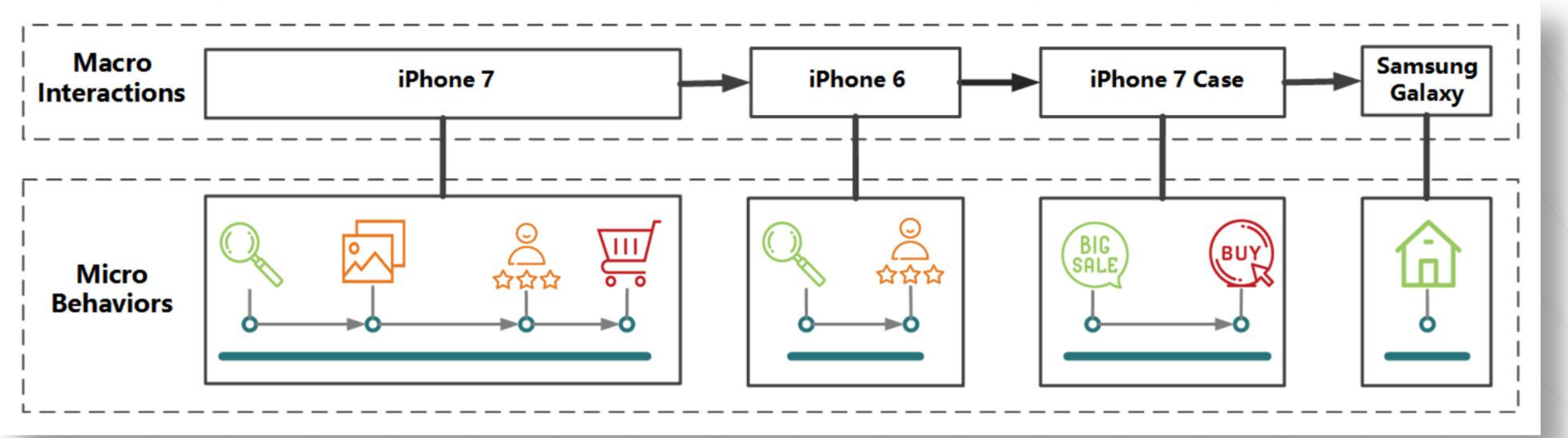


Figure 1: An illustrative example of observed data on a user from a real e-commerce site

1.3. Main Challenges & Corresponding Solutions

1. How to handle the sparse and high-dimensional input data (micro behavior sequences)?
   - Design a embedding layer to transform the input sequence into a low-dimensional dense vector by using one hot encoding and Word2vec.
2. How to model sequential information?
   - Use a RNN layer to extract the sequential features.
3. Since different micro behaviors have distinct importance, how to capture varied effects of micro behaviors?
   - Develop an attention layer to capture the varied impacts of different micro behaviors.

2.1. Problem Definition

Let:

- $P = \{p_1, p_2, ..., p_N\}$ be the set of products where N is the number of products.
- $A = \{a_1, a_2, ..., a_M\}$ be the set of activities a user can perform where M is the number of activities.
- $D = \{d_1, d_2, ..., d_K\}$ be the K different dwell time choices (discretized into K segments)

With the notations and definitions, the problem is formally stated as:

Given a set of sequences of tuples (i.e. micro behaviors) (p_i, a_j, d_k) , then aim to build a recommender system that can recommend the next product for each user.

2.2. Analysis on Effects of Micro Behaviors

Conversion Rate:

$$Conversion\ rate = \frac{\#\ behaviors\ of\ a_i\ ended\ with\ ordering}{\#\ behaviors\ of\ a_i}$$

The higher the conversion rate is, the more likely a micro behavior will lead to a user's ordering action.

2.2. Analysis on Effects of Micro Behaviors

Figure 2: Ordering vs. Other micro behaviors

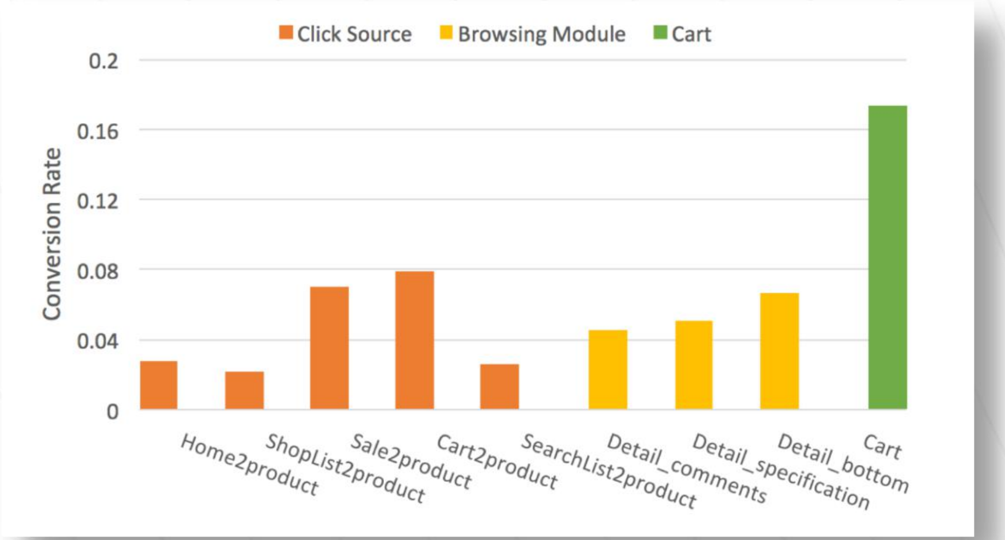
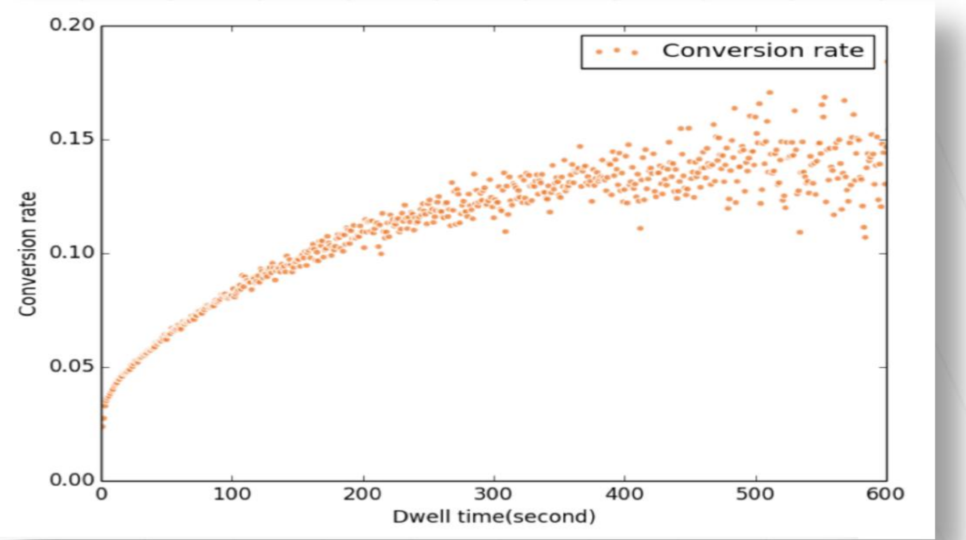


Figure 3: Ordering vs. Dwell time



In Figure 2, the micro behavior "Cart" has the highest conversion rate, which means if a user adds a product to cart, he is more likely to order it in the end.

In Figure 3, we note that within a certain range, the longer the dwell time is, the higher the conversion rate is.

2.2. Analysis on Effects of Micro Behaviors

Figure 4: Dwell Time vs. Click Source

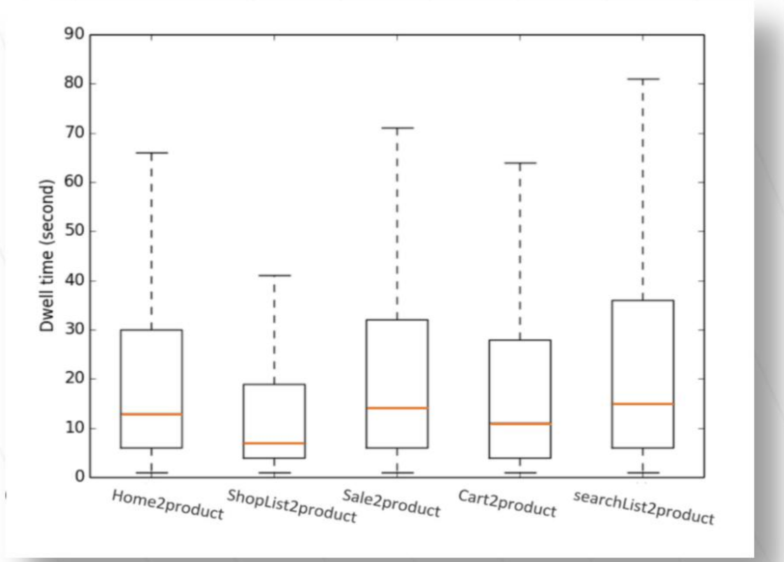
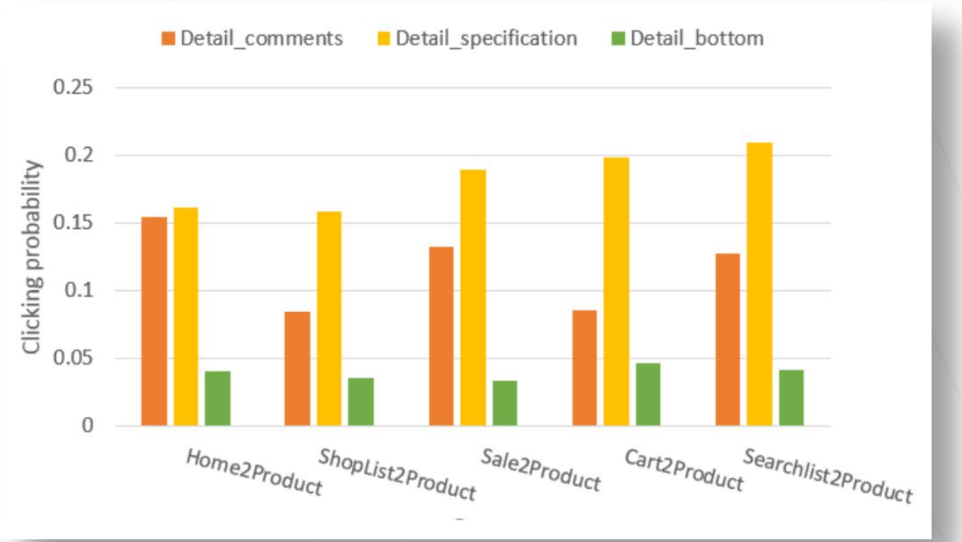


Figure 5: Click Source vs. Browsing Modules



In Figure 4, dwell time on a product is related to how a user locates the product. For example, from searching pages, one may spend longer time on this product.

In Figure 5, the likelihood of clicking detailed modules is related to the source how a user locates the project.

2.2. Analysis on Effects of Micro Behaviors

Key Obervations Based on the above Analysis:

1. Micro behaviors are correlated.
2. The effects of one micro-behavior on others are varied.

3.1. Model Architecture Overview

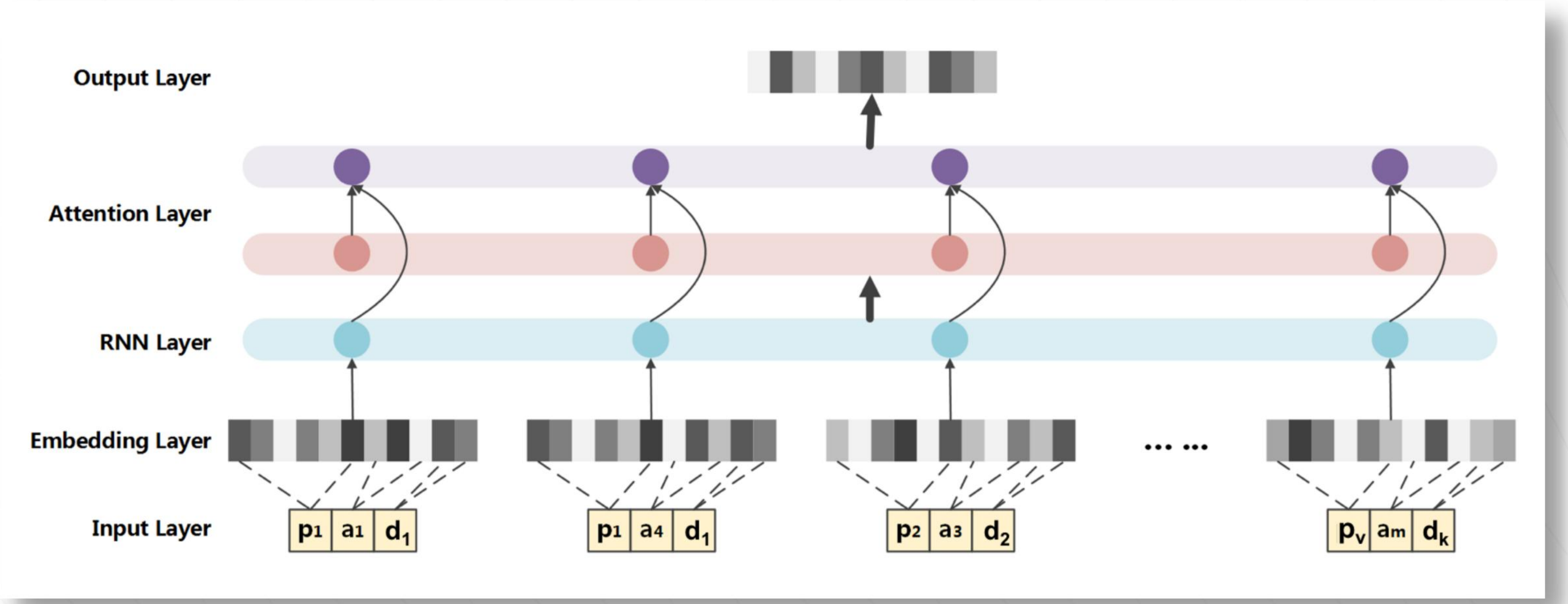


Figure 6: The architecture of the proposed framework

3.2. The Input and Embedding Layers

The input of the model is the data of a user u with a sequence of n micro behaviors.
We formally define it as a sequence $S_u = \{x_1, x_2, ..., x_n\}$,where $x_t = (p_v, a_m, d_k)$ and p_v, a_m, d_k are all one-hot encoding vectors.

The vocabulary sizes of P,A,D are V,M,K respectively, and there are V × M × K tuples in total. Therefore, the input data is extremely sparse and high-dimensional.

The authors design an embedding layer to transform the input x_t into a low-dimensional dense vector e_t using **Word2Vec**, which is formally defined as: $e_t = concatenate(W_P p_v, W_A a_m, W_D d_k)$
where $W_P \in R^{d_P \times V}$, $W_A \in R^{d_A \times M}$, $W_D \in R^{d_D \times K}$ where $d_P \ll N, d_A \ll M \ and \ d_D \ll K$ are the numbers of latent dimensions for products, activities and dwell time, separately.

The new representation of x_t , e_t is dense with dimension of $d_P + d_A + d_D$, which is much smaller than V × M × K.

3.3. The RNN Layer

The authors build a RNN layer to capture the sequential information of micro behaviors. The output of the embedding layer e_t is the input of the RNN layer.

The authors have tried two kinds of RNN model in this section: one is LSTM, and the other is GRU.

They find that LSTM and GRU achieve very similar performance in the evaluation. Given the simplified structure and faster training speed of GRU, they choose GRU for the proposed framework.

3.4. The Attention Layer

The attention is formed as:

$$M_t = tanh(W_m h_t + b_m), M_t \in R^{k \times L}$$

$$att_t = softmax(W_a M_t + b_a), att_t \in R^L$$

$$output = \sum_{t=1}^{T} att_t h_t, output \in R^K$$

where at is the attention weight on time t. The attention weight is mapped from the hidden layer vector into a real valued score by a function σ (·). In order to achieve enough expressive ability, the σ (·) is usually implemented by a neural network layer. One of the implementation is achieved by the above transformation, including tanh and softmax activation functions. Then the final output is an attention weighted pooling of the RNN layer.

4. Evaluation

Data type	Model	Appliances		Computers	
		Recall@20	MRR@20	Recall@20	MRR@20
SKU	POP	0.0088	0.0028	0.0149	0.0085
	BPR-MF	0.1255	0.0578	0.0747	0.0271
	Item-KNN	0.1806	0.0738	0.1101	0.0464
	Word2vec	0.3645	0.1295	0.3012	0.1044
	Word2vec Avg.	0.3668	0.1268	0.3152	0.1088
	RIB-Attention	0.4587(+25.84%)	0.1676(+29.42%)	0.3816(+21.07%)	0.1362(+25.18%)
	RIB	0.4732(+29.82%)	0.1718(+32.66%)	0.4043(+28.27%)	0.1456(+33.82%)
SKU+Activity	RIB-Attention	0.4615(+26.61%)	0.1724(+33.13%)	0.4092(+29.82%)	0.1443(+32.63%)
	RIB	0.4842(+32.82%)	0.1776(+37.14%)	0.4204(+33.38%)	0.1481(+36.12%)
SKU+Dwell	RIB-Attention	0.4673(+28.2%)	0.1745(+34.75%)	0.4108(+30.33%)	0.1482(+36.21%)
	RIB	0.4822(+32.29%)	0.1766(+36.37%)	0.4269(+35.44%)	0.149(+36.95%)
SKU+Micro-behaviors	RIB-Attention	0.474(+30.04%)	0.1784(+37.76%)	0.4227(+34.11%)	0.1516(+39.34%)
	RIB	0.4889(+34.13%)	0.1793(+38.46%)	0.4332(+37.44%)	0.1533(+40.9%)

Table 1: Performance Comparison

The authors collect two datasets with different product categories and sizes from a real e-commerce site. They are named following their product categories, “Appliances" and “Computers".

Then the authors compare their model with the following representative methods: POP, BPR-MF, Item-KNN, Word2vec, Word2vec Avg., RIB-Attention(removing attention layer).

They find that RIB obtains the best performance among all the methods.

5. My Thoughts

- This paper introduces a neural IR approach for product recommendation by utilizing the user micro-behavior data.

- It first abstracts the user micro behaviors into sequence data. Then, do one-hot encoding for the sequence data and use Word2Vect to reduce the vector dimensions. Next, use a RNN layer to capture the sequential features of the user micro behavior sequences. Finally, set a attention layer to capture the varied impacts of different micro behaviors and output the prediction results.

- The whole process is very concise and delicate, and has obtained good results. I think it's a very good example of neural IR. It helps me to understand how to apply the techniques (e.g. Word2Vec) we learned in class into practice.

- For the further improvement, there're mainly two directions:
  1. Combine the other traditional recommendation approaches (like Collaborative Filtering) and design a hybrid system
  2. Add more types of micro behaviors

Thank you