The influence of screen size of smart phone on user's searching performance

Chao Lu[1]

Research School of Computer Science, Australian National University

{u6686416}@anu.edu.au

Abstract. With the increasingly fierce competition in the mobile phone market, many new mobile phone brands and models with different screen sizes have appeared on the market. The problem of how different screen sizes of smart phone have influences on the user's search performance becomes a hot topic. In this paper, I use both classic machine learning method and deep learning method, to discover the relationship between screen size and search performance, and compare the their performance. I also applied DecryptGISData technique on my dataset, which helps improve the performance of the neural network. The result is that there is no significant difference but the medium size screen is more preferable. This shows that the screen size is not the bigger the better, of course, not too small, medium size improves users' search experience, which makes them feel more satisfied.

Keywords: Screen size, Search performance, Machine learning

1 Introduction

Now the mobile phone market is developing rapidly. There is a trend that the current mobile phone manufacturers are committed to making the mobile screen larger and larger. For example, Apple's mobile phone products, the iPhone 1 screen is only 3.5 inches, and now the iPhone XS Max with the largest screen has reached 6.5 inches, so the problem about the screen size of mobile phones arises, that is, how large screen size can make users use mobile phones more efficiently for a series of activities such as searching information. In this paper, I investigate the relationship between screen size and user's search performance. I present my experiment on eyegaze-search2 dataset. By designing a classic classification neural network model and a convolutional neural network, given the observed data about user during searching, the task is to predict which screen size the user performs the search on, the aim is to compare the performance between the two methods. In this process, I will apply DecryptGISData method to preprocess the dataset and discuss its effect in the following section.

2 Method

I use classic neural network and convolutional neural network as my deep learning method.

In terms of dataset, it is collected by the authors of Understanding Eye Movements on Mobile Devices for Better Presentation of Search Results (Kim, Thomas, Sankaranarayana, Gedeon, & Yoon, 2016). It has 162 cases with each case has 27 features measured.

I randomly split my dataset into two sets, training and testing set with 70% and 30% respectively and use the same dataset partition for both neural network to ensure consistency. To deal with NaN values or missing values, I fill them with 0. Then I am going to explain how the technique is applied.

In terms of the DecryptGISData technique, it is a technique about how to preprocess dataset to obtain more accurate result. I tried to use different normalization methods to preprocess the features, including max-min normalization, sigmoid normalization and Z-score normalization. Since the given dataset have too many features, more than 30, I removed some less relevant features which are related to other topics and pay more attention on the remaining more important features which are tightly associated with the results. Then I try to apply the technique on the features. Here are the details of how to deal with different features in different ways based on their distribution. I apply the same technique on the features which have similar distribution.

Time_to_firstclick: The time between when the user starts browsing all the search results links and when a link is clicked. Its distribution is shown as Figure 1.

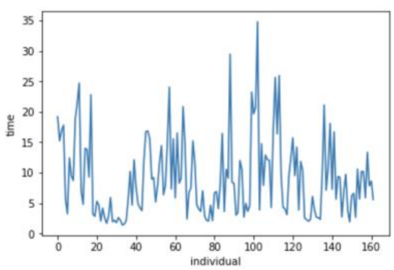

Figure 1

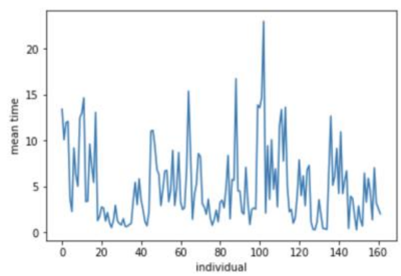

Figure 2

Mean_fixation_duration: During the search process, users will always watch the searching result, and the average time spent on each watching is what this feature means. Usually we consider that the more time we keep eyes on the result, the more difficult the problem is. Its distribution is shown as Figure 2.

Task_completion_duration: This feature just represents the time it takes from the user starts to search to give the answer. Its distribution is shown as Figure 3.

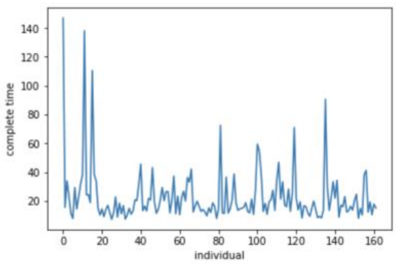

Figure 3

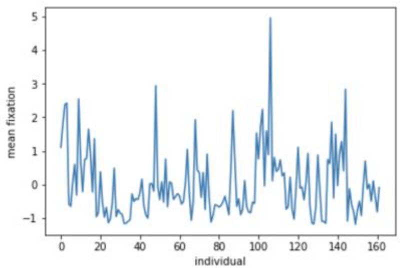

Figure 4

Mean_fixation_duration_for_onelink: Similar to Mean_fixation_duration, but only the mean time in one searching result. Its distribution is shown as Figure 4.

Therefore, based on these distributions of features, we need to normalize them to 0-1 for the neural network, however, I consider to use the sigmoid normalization and Z-score normalization, and compare their effect.

Scrolled: Whether the user turns the page to see more search results, 1 represents turning the page, 0 represents not turning the page. This feature is just 1 or 0 simply so I do not process it.

Regression: Whether the user jumps back to see the previous search results, 1 means yes and 0 means no. This feature is also just 1 or 0, thus does not need to be processed.

Compressed_scanpath_value & Minimal_scanpath_value & Compressed_M_Minimal: There is a feature used to record the order in which the user views the links, or the searching results. Then, the order is represented as a path, remove all consecutive values, we get a compressed scan path, then the node number is the value. If we remove all the revisit, we get the minimal scan path value. Compressed_M_Minimal just means the difference between these two values. Figure 5 and Figure 6 demonstrates the distribution of Compressed_scanpath_value and Minimal_scanpath_value respectively.

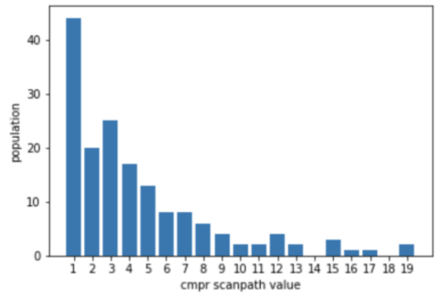

Figure 5

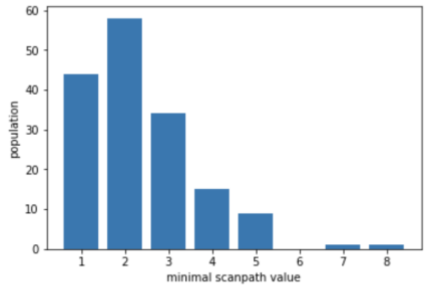

Figure 6

Regression_count: It counts how many times the user turns to the previous search results. It distribution is shown as Figure 7.

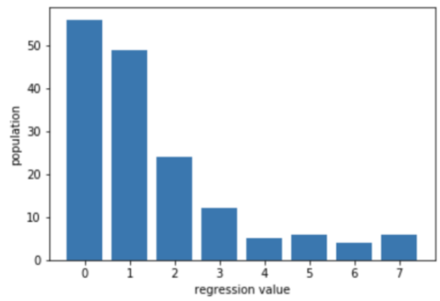

Figure 7

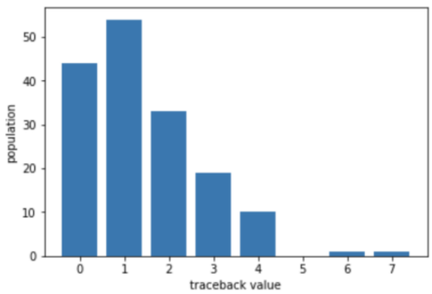

Figure 8

Traceback: This feature means how far the selected search result on which the user find the answer to the farthest search result visited. Its distribution is shown as Figure 8.

These features have similar distribution and data type. Therefore, I apply the same technique on them, using a single continuously valued unit to normalize them.

Regression_distance: This feature indicates how far the previous searching result is. Its distribution is shown as Figure 9. Since only two categories which are 0 and 1 account for the largest proportion, I modify this feature into 3 sub-features, used to distinguish between the common values and rare values, although it will lose information to some extend. It is shown as Table 1.

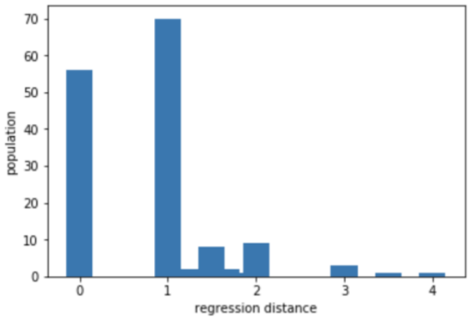

Figure 9

Value	G1	G2	G3
0	0.9	0.1	0.1
1	0.1	0.9	0.1
else	0.1	0.1	0.9

Table 1

In terms of neural network, I tried different number of hidden layers and different number of hidden neurons, the same random seed was used for weight initialisation to ensure consistency of comparisons of results.

In terms of convolutional neural network, I tried to use one and two convolution layers followed by max pooling to extract features, and then followed by linear layers and softmax function which generates the probabilities of different screen sizes. To deal with overfitting, I use dropout to add some variate on weights, in addition, batch normalization is also applied to the output of every layer to ensure the generalization. For both neural network, the learning rate is 0.001, the loss function is crossentropy and the optimization function is SGD.

3 Results & Discussion

ML method	Structure	Epochs	Accuracy (%)
NN	1 hidden layer 50 hidden neurons	100	23.68
		500	28.95
		1000	31.58
	1 hidden layer 100hidden neurons	100	26.32
		500	23.68
		1000	31.58
	2 hidden layers 50 hidden neurons 50 hidden neurons	100	26.32
		500	29.17
		1000	31.58
	2 hidden layers 100hidden neurons	100	28.57
		500	28.57

	100hidden neurons	1000	31.25
CNN	1 conv layer 50 channels	100	34.21
		500	47.37
		1000	52.63
	1 conv layer 100 channels	100	23.68
		500	44.74
		1000	42.17
	2 conv layers 10 channels 50 channels	100	34.21
		500	34.21
		1000	50.00
	2 conv layers 50 channels 100 channels	100	36.84
		500	39.47
		1000	52.24

Table 2

From the result above as shown by Table 2, the general performance of CNN is better than NN as the highest accuracy reached by CNN is above 50% while the highest accuracy of NN is approximately 30%.

As the structure of NN changes, the accuracy did not change too much. For CNN, the increase of kernel number has more influence on the increase of accuracy than that of convolutional layer number. It should be resulted from the feature number, because after preprocessing, the input feature number is 15, the more kernel to extract more feature is more important. A same thing is that with the increase of epochs, the accuracy of both neural networks have some increase while the increase of CNN is obviously larger than that of NN.

However, the accuracy is still at a low level. I think an important reason is the dataset, first of all, the dataset is too small, only 162 cases, if I split it into training set and testing set according to 70% and 30%, then the training set will only contain 113 cases which is obviously not enough for the neural network to find any patterns (D'souza, Huang & Yeh, 2020). Secondly, the problem is also hard to predict because actually the performance of users with different screen sizes itself does not vary much.

4 Conclusion

In conclusion, the search performance of the users is not highly affected by the screen size, there is only a slight difference, which is the small screen is the worst for users to search information because usually the users need to scroll again and again to find what they need, then comes the large screen because of too much information, generally speaking, medium screen is the best according to the dataset. However, in terms of neural network, the performance of CNN is much better than neural network but it is still difficult to make any prediction because the given dataset is too insufficient, the future work can be based on more data and train neural network again to check if we could have any interesting findings.

Reference

1. Kim J, Thomas P, Sankaranarayana R, Gedeon T, Yoon H. Understanding eye movements on mobile devices for better presentation of search results. *Journal of the Association for Information Science and Technology.* 2016;67:2607-2619 (2016)

2. Bustos R, Gedeon T. Decrypting neural network data: a GIS case study (1995)

3. D'souza, R.N., Huang, P. & Yeh, F. Structural Analysis and Optimization of Convolutional Neural Networks with a Small Sample Size. *Sci Rep* **10**, 834 (2020). https://doi.org/10.1038/s41598-020-57866-2

4. Lagun, D., Hsieh, C.-H., Webster, D., & Navalpakkam, V. (2014). Towards better measurement of attention and satisfaction in mobile search. In Proceedings of the 37th international ACM SIGIR conference on research & development in information retrieval (pp. 113–122).