User Embeddings Based on Mobile App Behavior Data

Kushal Singla[1(✉)], Satyen Abrol[1], and Sungdeuk Park[2]

[1] Samsung R&D Institute, Bangalore, India
{kushal.s, satyen.abrol}@samsung.com
[2] Samsung Electronics, Suwon, Korea
sdeuk.park@samsung.com

Abstract. We consider a smart phone scenario with a number of apps used by a user. The app usage data provides information about the user behavior, which can be used to identify the user demographics and interest and in turn is used to find similar users. In this paper, we propose a method to generate a latent space user embedding using the user app usage data, which is a dense low-dimensional representation of the user. This representation is used for low latency user similarity computation and acts as the user feature representation in user demographics prediction models.

Keywords: User2vec · AutoEncoder · User embedding

1 Introduction

User Modeling and User Profiling are well-studied areas of Computer Science finding applications in broad range of areas including recommendation systems and expert systems. Understanding the user preferences, interests and characteristics allows service providers to serve more personalized content to user resulting in improved user experience and increased success (engagement, purchases, etc.) for the business. In the domain of smartphones, a potential approach for building the user profiles is based on either developing a predictive model that map the user to a pre-defined taxonomy or creating user segments based on the app installation or app usage. In this paper, we present a novel approach to map a user to a latent space embedding which implicitly captures user behavior. In our work, we make several contributions in the field of user modelling based on app usage. First, we propose a user modelling method for creating a user embedding using Auto encoders. Second, we show the effectiveness of user embedding for different use cases such as demographics prediction. Thirdly, we show that this user representation can be used for user segmentation using vector operations.

2 Related Work

User modelling and Personalization has been studied extensively, especially since the advent of smartphones. Zolna et al. [1] present a method to build LSTMs to model user behavior on a website, with applications in e-commerce domain. The approach looks at

J. Pagán et al. (Eds.): ICOST 2019, LNCS 11862, pp. 183–189, 2019.
https://doi.org/10.1007/978-3-030-32785-9_16

user's browsing pattern and maps it into a fixed-size vector. Amir et al. [2] describe a method to create user embedding from text to capture latent user aspects. App2Vec [3] and AppDNA [4] are two such papers, which present methods to create embedding which captures semantic relationship between apps. The work done by us presents a novel unified method to compute user embedding based on the apps of the user instead of deducing user embedding from the app embedding. [5] Shows that usage of user embedding for the gender prediction task where the social context of the user is available but the social context is not readily available in all cases so in the absence of social context, the proposed approach doesn't require the user relations. Mannan [6] shows the use of artificial neural network for the user similarity computation but it is computationally expensive to compute the similarity between all the set of users and it cannot perform vector-based operations using this approach but the autoencoder fills this gap.

3 User Embeddings Model

User embedding can be used for finding similar user. The app space cardinality is very high but it is very sparse per user whereas dense user embeddings are usefull in this case. User embedding can be used as a feature for predictive analytics like classification of user into various categories extracted in an unsupervised manner. User embedding can be used to perform vector operations like addition, subtraction etc. This can be used to perform user analogy tasks. Our method is based on autoencoder. Autoencoder is a neural network model, which was developed for unsupervised learning and is used for feature extraction. [7] explains the usage of autoencoder for reducing the dimensions using a multi-layer neural network, which is proven to be better than the standard PCA approach. Amiri [8] presents a method to compute similarity between text pairs using autoencoder.

3.1 Method Explanation

In this subsection, we explain the architecture of the user-embedding model based on the autoencoder architecture. The layers of the encoder model are explained below (Fig. 1).

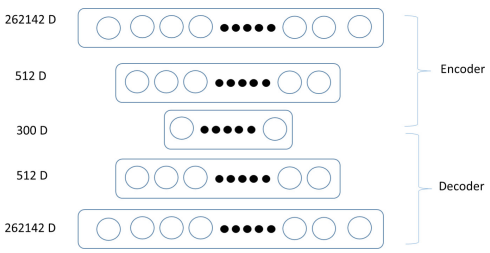

Fig. 1. Architecture for the autoencoder

(1) Input Layer – The input layer is 262,142 dimensional which is equal to the vocabulary of the app space.
(2) Hidden later – This layer is 512 dimensional which is followed by Relu, batch-norm, dropout respectively.
(3) Output later – This layer is 300 dimensional. Tanh computation is applied on the output layer

The layers of the decoder model are in the reverse order until the input layer. Sigmoid operation is applied to the output layer. Loss is optimized using the Adam optimizer and BCELoss. Number of epoch is equal to 6. Variable learning rate is employed per epoch- [0.1, 0.05, 0.025, 0.01, 0.005, 0.0001]. We ran the experiment on Amazon AWS p2.xlarge which has 12 GB of GPU memory which influences the autoencoder design (Table 1).

Table 1. Algorithm to compute the user embedding using the mobile app usage data

ALGORITHM: Algorithm to compute the user-embedding model using app usage

Input: User app usage vector for the app $A_1\ A_2\ A_{3\ldots\ldots}\ A_n$
Output: User embedding vector for the user $U_1\ U_2\ U_{3\ldots\ldots}\ U_n$

1. Pre-process the app usage vector and one-hot encode the vector
2. Train the auto encoder model using the user one-hot encoded vector.
3. Generate the user embedding from the output layer of the encoder section of the auto encoder.
4. *exit*: end procedure

4 Experiments

We evaluate the user-embedding model using a private dataset collected from a survey of the user app usage. We collected the app usage data of the users over a period of 2 months. The User set is composed of both the male and female users within the age group from 16 to 80. App usage data consists of the app ids (For example - com. whatsapp) of the apps used by the users within this period in the csv format. This dataset is used to evaluate the user embedding using various tasks. The app usage data is preprocessed using Spark to create the one hot encodings of the user app usage vector.

4.1 User Embedding Use Cases

4.1.1 Gender Clustering

In this sub section, we try to find the relation between the embedding and the user gender. In this case, we sampled 1000 user randomly from the two classes and generate the user embedding using the model training earlier and generate t-sne plots.

4.1.2 Age Clustering

In this sub section, we try to find the relation between the embedding and the user age. In the case of the age attribute based clustering of the user, we divider the user into two groups based on different point of age as the separation point and observe that the distinct clusters are formed at various age separation points. Figure 3 shows the user clusters based on different age threshold point. Equal numbers of users are sample from the two sets and t-sne plot are computed for the various threshold ages. This shows that distinct clusters are observed under various age thresholds (Fig. 2).

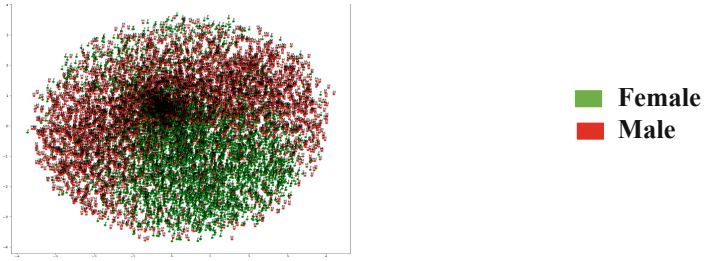

Fig. 2. T-SNE based gender plot of the users based on the autoencoder embedding

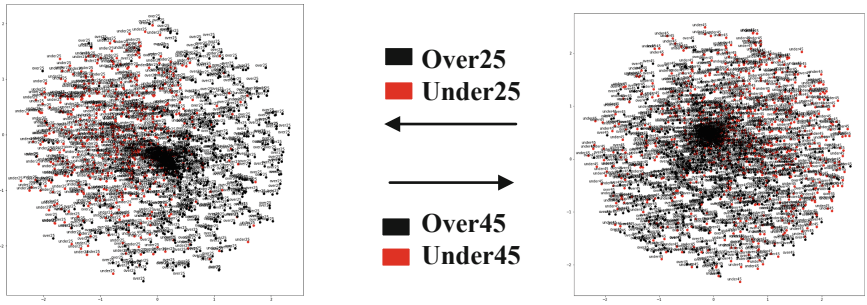

Fig. 3. T-SNE based age plots of the users based on the autoencoder embedding

4.1.3 Embedding for User Similarity Task

In this sub section, we evaluate the quality of finding users with similar attributes.

Figure 4a shows the quality of the user similarity task. In this experiment, we sampled a subset of users from the two age groups. This figure shows the probability of finding a user within the same age group by using the cosine distance between the user embedding lies within the range of 0.6 to 0.8. The app usage behavior of the people in the age group under 35 and over 35 is the least non-distinguishable in the experiment as it can be shown in lowest cosine distance of 0.6 whereas it's most distinguishable in the

age-group under 75 and over 75 as shown by the cosine distance of 0.8. Figure 4b shows the quality of the user similarity task on dividing the user sample into two sets based on the age threshold.

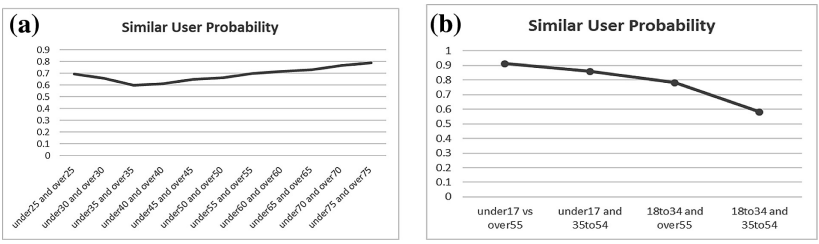

Fig. 4. (a) Probability of similar user in various age groups (b) Probability of similar user in various age groups

4.1.4 Validation Using Analogy Tasks

In this sub-section, we try to derive relationships based on multiple attributes. For e.g. the deduced semantic relation between an old woman user group and old man user group is similar to the semantic relation between young girl user group and young boy user group. These multi-attribute semantic relations can be evaluated by performing vector operation on the embedding. For example - To compute the semantic relation old_woman - old_man = young_girl - X, we sample 1000 users randomly from the each user group, compute the average of the vectors, perform this process for the old_woman, old_man, young_girl user groups and then search the vector space for the user_group vector closest to X using the cosine distance method and found that the closest vector is equal to the young_boy user group. We performed eight more experiments like this and the results are shown in the Table 2.

Table 2. Analysis of the classification quality for gender prediction using user embeddings and shallow network

Analogy	Actual test case
old_woman-old_man = young_girl-young_boy	60 to 65_F-60 to 65_M = 20 to 25_F-25 to 30_M
old_woman-young_girl = old_man-young_boy	70 to 75_F-20 to 25_F = 70 to 75_M-25 to 30_M
young_girl-young_boy = old_woman-old_man	20 to 25_F-20 to 25_M = 70 to 75_F-65 to 70_M
young_girl-old_woman = young_boy-old_man	20 to 25_F-70 to 75_F = 20 to 25_M-65 to 70_M
old_man-old_woman = young_man-young_girl	60 to 65_M-60 to 65_F = 20 to 25_M-25 to 30_F
old_man-young_boy = old_woman-young_girl	60 to 65_M-20 to 25_M = 60 to 65_F-25 to 30_F
young_boy-old_man = young_girl-old_woman	20 to 25_M-70 to 75_M = 20 to 25_F-65 to 70_F
young_boy-old_woman = young_girl-old_man	20 to 25_M-70 to 75_F = 20 to 25_F-60 to 65_M
young_boy-young_girl = old_man-old_woman	20 to 25_M-20 to 25_F = 70 to 75_M-65 to 70_F

4.1.5 User Embedding for Classification Task

In this sub-section, we use the embedding as feature for user attribute prediction and compare its performance with a shallow prediction model with manual feature engineering. The shallow prediction model is based on SVM [9] model with the following parameters: gamma = 1 and C = 0.1. Input data to the embedding based model is the app-based user embedding whereas the shallow network is build using the tf-idf features of the metadata of the app ids used by the user. Table 3. shows that the embedding based binary prediction model achieves accuracy similar to the shallow model without hand engineered features and this method helps to build the solutions faster. Table 4. shows the quality of the age prediction model. The recall of the classes 0 to 17 and over 55 is higher using the embedding approach because of the generalization aspect of embedding.

Table 3. Analysis of the classification quality for gender prediction using user embeddings and shallow network respectively

		precision	recall	f1-score	support
	F	0.8889	0.889	0.889	5083
	M	0.8853	0.8851	0.8852	4917
weighted	avg	0.8871	0.8871	0.8871	10000

	precision	recall	f1-score	support
F	0.8872	0.869	0.878	89299
M	0.8961	0.9109	0.9034	110701
avg/total	0.8921	0.8922	0.8921	200000

Table 4. Analysis of the classification quality for age prediction using embedding vector feature and shallow network

	precision	recall	f1-score	support
0to17	0.684211	0.537931	0.602317	145
18to34	0.767632	0.740511	0.753827	4189
35to54	0.693874	0.766928	0.728574	4711
over55	0.65047	0.434555	0.52103	955
avg	0.720486	0.7208	0.717502	10000

	precision	recall	f1-score	support
0to17	0.734463	0.463954	0.568679	1401
18to34	0.761585	0.759016	0.760298	41206
35to54	0.694594	0.774033	0.732165	46945
over55	0.695152	0.381508	0.492646	10448
avg / total	0.722815	0.722490	0.716442	100000

5 Conclusion and Future Work

In this paper, we have proposed a method to create user embedding for a smartphone user based on their app usage. Our experiments highlight the usefulness of such an embedding in capturing user behavior and for binary prediction tasks and vector operations. In future, we plan to evaluate the use of the user embedding for the recommendation task.

References

1. Zołna, K.: User modeling using LSTM networks (2017)
2. Amir, S., Coppersmith, G., Carvalho, P., Silva, M.J., Wallace, B.C.: Quantifying mental health from social media with neural user embeddings. arXiv preprint arXiv:1705.00335 (2017)

3. Ma, Q., Muthukrishnan, S., Simpson, W.: App2vec: vector modeling of mobile apps and applications. In: Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 599–606. IEEE Press (2016)
4. Xue, S., Zhang, L., Li, A., Li, X.Y., Ruan, C., Huang, W.: Appdna: app behavior profiling via graph-based deep learning. In: IEEE INFOCOM 2018-IEEE Conference on Computer Communications, pp. 1475–1483. IEEE (2018)
5. Chen, L., Qian, T., Zhu, P., You, Z.: Learning user embedding representation for gender prediction. In: 2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI), pp. 263–269. IEEE (2016)
6. Mannan, N.B., Sarwar, S.M., Elahi, N.: A new user similarity computation method for collaborative filtering using artificial neural network. In: Mladenov, V., Jayne, C., Iliadis, L. (eds.) EANN 2014. CCIS, vol. 459, pp. 145–154. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11071-4_14
7. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006)
8. Amiri, H., Resnik, P., Boyd-Graber, J., Daumé III, H.: Learning text pair similarity with context-sensitive autoencoders. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Long Papers, vol. 1, pp. 1882–1892 (2016)
9. Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: Proceedings of the Fifth Annual Workshop on Computational Learning Theory, pp. 144–152. ACM (1992)