Algorithmic Approach for Learning a Comprehensive View of Online Users

Kourosh Modarresi
Adobe Inc., San Jose, U.S.
kouroshm@alumni.stanford.edu

Abstract
Online users may use many different channels, devices and venues for any online user experience. To make all services such as web design, ads, web content, shopping, personalized for every user; we need to be able to recognize them regardless of device, channels and venues they are using. This, in turn, requires building up a comprehensive view of the user which includes all of their behavioral characteristics - that are spread all over these different venues. This would not be possible without having all behavioral related data of the user which requires the capacity of connecting the user all over the devices, and channels, so to have all of their behavior under a single view. This work is a major attempt in doing this using only behavioral data of users while protecting the user’s privacy.

Keywords: Matrix Completion, Regularization, Matrix Reconstruction, Singular Value Decomposition, User Behavior Data

1 Introduction

Creating an ideal digital users’ experience has to be based on understanding the users and so creating an echo environment that is favorable to them. This way, the users would have a desirable online experience. One could expect a great digital experience may lead to a maximum return on the e-marketing different metrics such as increase in usage, purchase and so on. Often, the basis of the study of the users’ preferences is on tracking cookies, login parameters - such as email address, userid and so on. Tracking users through cookies is not very reliable method in gaining a comprehensive view of the users. One major issue about tracking by cookies is that users could delete them besides the fact that cookies cannot recognize the same users as they may use different devices or digital venues and channels. Deploying login credentials has only limited effects since users may not use any credentials in their online journeys. Also, tracking IP address has the major limitation of being restricted to only
one device as the IP address changes from one device to another. Those are some of the reasons supporting the use of behavior data for user identification for this work. Users behavior will be recorded regardless of whether they use any login credentials, or what device they are using and is also independent of whether any cookies are used to track them. One of the major challenges in user recognition based on behavioral data is that user behavior is distributed among all devices (such as phone, desktop and so on), digital venues (email, browser, …) and channels (search engine, commercial websites, .). Users may not be fully understood if we look at only a fraction of their interactional data with the digital space and thus to get a comprehensive view of a user, we need to collect all behavioral data the users have left on all these dispersed locations. To do this, i.e., to gather all these characteristics of the users - from all these channels, devices and venues – we need to find a way to recognize the user across all these devices, channels and venues. This would suggest that this process is an iterative problem and as we could recognize the same users – no matter where they are engaging the digital sphere – enabling us to bring together more behavioral data from all points, we get a better understanding of the user through all this collected data and thus we could identify the same user across all the channels more effectively.

2 The Model in Recognizing the Users

This work uses only behavioral data and assumes no access to private information of users, such as name, last name, email address, SSN, DLN, login information. Also, no information from cookies is used. The model is based on the following steps;

(A) Centering the Data Matrix
(B) Transformation of the Data Matrix X to a Higher Dimensional Space
(C) Computing SVD (Singular Value Decomposition)
(D) Completing the Matrix
(E) Computing the Best Rank-k Matrix for the Matrix X
(F) Updating the Original Matrix
(G) Projecting the New Row onto the k-dimensional Space (SVD Updating)
(H) Computing the New Distance and Detecting a Possible Match

2.1 The Format and Representation of User’s Behavioral Data

The user data entails all possible information of the interactions users have in the digital space. Matrix representation of data is used in this work and so, the data is represented in the matrix form, called matrix X. Rows of the matrix are users (i) and its columns (j) are different features that represent users’ online interactions. The columns could include interaction variables such as “browser type”, “operating system type”, “url”, “the time of interaction” and “length of interaction”. Each matrix entry - \( x_{ij} \) – displays the specific feature of user i on interaction behavior j. Thus each entry
is the relation between a specific interactive variable \( j \) and a specific user \( i \). The data matrix, \( X \) has \( m \) rows and \( n \) columns \((m \times n)\).

In general, the entries of the matrix \( X \) may be explicit such as the browser, rating and total purchases, or they may be implicit data such as like/not like of specific content derived from some explicit data. Explicit data is not always available and often not enough of that could be found [11,12,15]. Historic (logged) data or live data (streaming) data and quite often a combination of both are used to design and test the user recognition model.

2.2 Centering the Data Matrix

The data matrix contains a variety of data types that have different range and scales. The matrix also includes non-numerical data that may be transformed to numeric data using dummy variables. All types of data need to be centered to prevent domination of some variables (columns) as a results of their larger range. The process of centering [85] is done by reducing the mean of each column from all the column entries,

\[
X = X(I_n - \frac{1}{n} 1^T)
\]

Where \( I \) am the identity matrix and \( 1 \) is a column vector with all entries to be one.

2.3 Transformation of the Data Matrix \( X \) to a Higher Dimensional Space

The data matrix \( X \) has \( n \) variables (columns). In general, for a unique definition and identification of each user, this number of dimensions may not be sufficient. Thus, to make the users separable and distinct from each other, the data matrix is transformed to a higher dimensional space. The number of dimensions of the new space will be computed so that every user is defined and identified uniquely. Modern data possesses many characteristics such as high correlation. To make the transformation of the data matrix \( X \), we use new coordinates of \( n+1 \) to \( n+p \), where \( p \) is the required number of added features so we could identify all users uniquely.

\[
x_w = \sum_{l=n}^{n+p} a_l x_l + \varepsilon_l
\]

Where \( \varepsilon_l \) is random white noise and \( w = n+1:n+p \).
2.4 Computing SVD

For this work the singular value decomposition (SVD) of the matrix X is defined the usual way [33], as;

\[ X = UDV^T \]

Where: U, the left singular vectors, is m×n orthogonal matrix,

\[ UU^T = U^TU = I \]

V, the right singular vectors, is n×n orthogonal matrix

\[ VV^T = V^TV = I \]

and D = diag \( (d_1, d_2, \ldots, d_n) \) with the singular vectors;

\[ d_1 \geq d_2 \geq \ldots \geq d_n \geq 0 \]

Another view of the computation of SVD is by using minimum reconstruction error,

\[ \min \|X - UD'V'\| \]

Which could be rewritten as,

\[ \arg\min_{(u,v,d)} \|X - XuVu^T\|_2^2 \]

Equivalently,

\[ \min_{u,v,d} \sum [x_{ij} - u_id_iv_j^T]^2 \]

2.5 Completing the Matrix

Modern data is often very sparse [58] and the data under study in this work is even more sparse. That is due to the fact that for every user entry (row) of the data matrix X, even when coming from the same channel or device, there are many missing (unknown) entries. This is due to the fact the number of variables is large and it is unlikely that user’s interaction involves all different variables. Since the columns of data matrix X is a combination of all variables from all possible devices and entries and although these channels and devices have many similar and overlapping features (columns), still each of these venues have many unique variables that are not shared by many other venues. As a result of these two factors, i.e., the missing values for each channel and the uniqueness of many variables of these channels, each row contains many unknown (missing) entries.

At this step, the unknown entries of the data matrix are computed using an iterative SVD algorithm [75,76]. In short, at the first step, the missing entries in each column of X are replaced with the median of the column. Then, SVD is computed for the matrix X. The third step involves the computation of the best k-rank matrix (k<< min (m, n)). The forth step requires the reconstruction of X using the low rank (k-rank) singular vectors and singular values matrix. Then, the newly computed values of the missing entries are compared with the values from the previous iteration. All of the above steps are repeated until convergence. The summary of the algorithm in this part is;
For a centered $X$:

Step (1) compute the
\[
\min_{U_q V_q D_q} \| X - U_q D_q V_q \|
\]

Where $q$ is computed using a threshold of variation of the original data in $X$.

Step (2) computes the new $X$ – of $q$-rank
\[
X_q = U_q D_q V_q
\]

Using newly computed $X_q$, we have new values for the missing entries.

Step (3) Iterate steps (1) and (2) till convergence,
\[
|X_q (i+1) - X_q (i)| / |X_q (i)| \leq \delta
\]

For small $\delta$.

2.6 Computing the Best Rank-$k$ Matrix for the Matrix $X$

At this step, using the results in the previous sections, we need to compute the best low dimensional space that we should project matrix $X$ onto. Matching of any new user will take place in this new space. This new space has $k$-dimensions (section 2.5). The computation of $k$ is based on the preservation of a minimum threshold of the original information (variation) of the data. The determination of a specific threshold depends on many considerations including the specific applications, the required accuracy and the computational cost.

2.7 Updating the Original Matrix

For any incoming user for whom the identity is unknown, update the original matrix by adding the new row and do all steps of 2.2-6.

2.8 Projecting the New Row onto the $k$-dimensional Space (SVD Updating)

Project the new row in SVD space to get the new coordinates of the new data point.

For any new entry $x_{m+1}$:

1. Compute the missing entry (section 2.5) to get $x_{new}$.
2. Compute its projection in the new (low dimensional) space;
2.9 Computing the New Distance and Detecting a Possible Match

For the new row in $U$, $u_{\text{new}}$, compute the distance between the new row (vector) with all of the previous rows in the matrix $U$ (left singular vector of $X$). In other words, compute the closest user in the space to the new one, in the sense of Euclidean (2nd) norm.

$$
\min_{i} \left( \| u_{\text{new}} - u_i \|_2 \right) \leq \varepsilon
$$

For a small $\varepsilon$. If the equation condition is not satisfied i.e., if there is no user satisfying the inequality then there is no match with the known users and the $x_{\text{new}}$ is flagged as a new row (new user) in the data matrix $X$.

2.10 SAMPLE OF RESULTS

Different data set is used to test this model. For all different examples of data matrices, k-fold cross validation is used to validate the model. The model was applied on $X=200,000$ by 234 data matrix. The obtained accuracy was 92%.

References


