

Technical Disclosure Commons

Defensive Publications Series

August 18, 2017

A Method for Population Segmentation by Event Based Machine Learning

David Parkinson

Follow this and additional works at: http://www.tdcommons.org/dpubs_series

Recommended Citation

Parkinson, David, "A Method for Population Segmentation by Event Based Machine Learning", Technical Disclosure Commons, (August 18, 2017)
http://www.tdcommons.org/dpubs_series/641



This work is licensed under a [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/).

This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

A METHOD FOR POPULATION SEGMENTATION BY EVENT BASED MACHINE LEARNING

Introduction

The present disclosure provides systems and methods for automating the process of segmentation analytics for users of an application. Segmentation analysis involves the bucketing of individuals (users) into different groups each describing the type of user, usage, and/or interactions the group exhibits with the application. Such knowledge allows businesses to better understand and support specific user group's needs. Existing methods of segmentation analysis involve multiple manual steps: (1) define groups; (2) build a list of business rules to calculate the correct categorization for each user; (3) execute the algorithm over all users; and (4) repeat the process to update users as new users join and previous users change groups over time. As the number of users and variance between groups increase within an application, each step becomes more expensive and less sustainable. With nearly unlimited uses it becomes very difficult to know how users are interacting with the application and which are using it the most.

Summary

The present disclosure proposes to solve the challenges described above by automating the process of segmentation analysis using a machine learning algorithm. Such automation can decrease time and resources expended and remove the manual maintenance required of a typical segmentation pipeline. By loading event data into a statistical model (e.g., a neural network or other model implemented through a machine learning process) that will look for trends in application usage, session time, application interaction, and/or user behavior, the statistical model can track such trends over time and use an automated process of segmentation to determine how many and where user groups exist and tag them as such.

The embodiments according to example aspects of the present disclosure can automatically create segmentations and groupings of an application's users. In general, this can be accomplished by providing activity logs, populated with a plurality of events, as an input to a statistical model (e.g., a neural network or other model implemented through a machine learning process). A notion of session, or of what events are tied together in a single use of the application, can also be provided as input. Additionally, an indication of the user associated with an event can be provided as input. The machine learning algorithm can then be applied to one or more of the inputs, through which a plurality of segments and associated metadata can be generated and provided as an output of the statistical model. Users of an application can then be grouped together with a plurality of other users within the same segment.

Types of user segmentations can vary in different embodiments of the disclosed technology. In some examples, the applications can correspond to education software such that segmentations are based in part on educational patterns. In some examples, the applications relate to word processing software such that segmentations are based in part on writing patterns. In other examples, the applications can relate to multimedia software such that segmentations are based in part on viewing patterns. Any type of application can benefit from the disclosed technology and segmentations can be determined depending on the nature of the application or segmentations can be predetermined despite the application.

The generated plurality of user segmentations provided as output from a neural network or other statistical model can be utilized in a variety of specific applications. In some examples, the users or other event information provided as input to the neural network are subsequently tagged with one or more of the plurality of segments generated as output. In some examples, an association between the users or other event information and the plurality of generated segments

can be stored in a database. In some examples, the event information corresponding to users of the application and the database of stored associations includes user information as well as the associations between the users associated with each event and the plurality of generated segments. In some examples, events can be matched to a user in the database using the plurality of generated segments at least in part to perform the matching.

Detailed Description

Example aspects of the present disclosure are directed to systems and methods of segmenting application users based on event information. Manually segmenting application users can be time consuming and expensive. The task is more challenging with large scale applications boasting a vast number of users all with varying needs in the product. For instance, teachers and students in the classroom can use an application for educational lessons; academics could use the same application as a rendering engine for complicated GIS analytics; or the average user can simply plan a vacation, look at their house, or scout an area for a hike or camping trip. The uses for certain applications are nearly unlimited and as such, difficult to predict.

In addition to the complexities of classifying application users, classification maintenance can be a continuous task as new users join while others alter their usage of the application. A fully automated segmentation process can automatically update classifications as needed; essentially removing the time, expense, and continuing manual maintenance of a typical segmentation pipeline.

In some embodiments, in order to obtain the benefit of the techniques described herein, a user can be required to allow the collection and analysis of user activity, and other relevant information collected by an application's activity logs. For example, in some embodiments,

users can be provided with an opportunity to control whether programs or features collect such data or information. If the user does not allow collection and use of such signals, then the user may not receive the benefits of the techniques described herein. The user can also be provided with tools to revoke or modify consent. In addition, certain information or data can be treated in one or more ways before it is stored or used, so that personally identifiable data or other information is removed.

Referring now to the drawings, FIG. 1 depicts a sample embodiment **100** depicting an exemplary machine learning statistical model providing segmentations of an application's users and/or event information. Schematic **100** generally includes an event **110**, an associated session **120**, and/or an associated user **130** provided as one or more inputs to a statistical model **100**, such as but not limited to a neural network, which generates one or more outputs.

An event **110** describes any user interaction with an application. For instance, in certain applications an event **110** takes place when a user searches for a location, zooms in/out on a location, alters a map view, and/or interacts with the user interface in any way. In some embodiments, a plurality of events can be logged by an application in an activity log. Additionally, in some examples, each event can be stored with an associated session and/or user.

An associated session **120** provides an indication of which events are tied together in a single use of the application. An associated session **120** can be determined in various ways, including time period of activity, subject matter of activity and/or nature of activity. For instance, if a user is planning a trip using an application, a session **120** can include every interaction with the user interface within the time period in which the user is searching hotels. In some embodiments, an event **110** and an associated session **120** can be stored together in a database and/or an application's activity logs.

An associated user **130** provides an indication of the individual who performed the logged event. A user can be identified in a plurality of ways. For instance, a user can be assigned an identification number, a username and/or any other type of identification information. In some embodiments, although not required, user identification information is stored in a database and/or an application's activity log with an associated event.

A statistical model **100** can be implemented in a variety of manners. In some embodiments, machine learning can be used to evaluate training events and develop classifiers that correlate predetermined event features to specific categories. For example, event features can be identified as training classifiers using a learning algorithm such as a Neural Network, Support Vector Machine (SVM) or other machine learning processes. Once classifiers within the statistical model are adequately trained with a series of training events, the statistical model can be employed in real time to analyze subsequent events provided as input to the statistical model.

In examples when the statistical model **100** is implemented using a neural network, the neural network can be configured in a variety of particular ways. In some examples, the neural network can be a deep neural network and/or a convolutional neural network. In some examples, the neural network can be a distributed and scalable neural network. The neural network can be customized in a variety of manners, including providing a specific top layer such as but not limited to a logistics regression top layer. A convolutional neural network can be considered as a neural network that contains sets of nodes with tied parameters. A deep convolutional neural network can be considered as having a stacked structure with a plurality of layers.

Although statistical model **100** of FIG. 1 is illustrated as a neural network having three layers of fully-connected nodes, it should be appreciated that a neural network or other machine learning processes in accordance with the disclosed techniques can include many different sizes,

numbers of layers and levels of connectedness. Some layers can correspond to stacked convolutional layers (optionally followed by contrast normalization and max-pooling) followed by one or more fully-connected layers. For neural networks trained by large datasets, the number of layers and layer size can be increased by using dropout to address the potential problem of overfitting. In some instances, a neural network can be designed to forego the use of fully connected upper layers at the top of the network. By forcing the network to go through dimensionality reduction in middle layers, a neural network model can be designed that is quite deep, while dramatically reducing the number of learned parameters.

Referring still to FIG. 1, after the statistical model **100** is applied to inputs **110**, **120**, and/or **130**; one or more outputs **140** can be generated. In some examples, outputs **140** of the statistical model include a plurality of segmentations **140** for the users and/or event information based in part on determined patterns. Types and amounts of classification labels **140** can vary in different embodiments of the disclosed technology. In some examples, the applications correspond to education software such that the segmentations **140** are based in part on educational patterns. In some examples, the applications are types of word processing software such that segmentations **140** are based in part on writing patterns. In other examples, the applications are types of multimedia software such that segmentations **140** are based in part on viewing patterns. Segmentations can also be predetermined despite the application. For instance segmentations can be based in part on predefined grouping characteristics. Although two different segmentations **140** are shown in the example of FIG. 1, other specific numbers and segmentation parameters can be established in accordance with the disclosed technology.

An example method for segmenting event information includes training a statistical model using a set of training events and data identifying the characteristics associated with the

training events. A statistical model can be trained in a variety of particular ways. Training the statistical model can include using a relatively large set of training events coupled with ontology-based segmentation labels. The training events can be associated with different users and include data identifying a plurality of user segments, such that the statistical model outputs a plurality of segmentations for each training event. In some examples, training a statistical model can include pre-training using a predetermined subset of event information and ground truth labels with a Soft Max top layer.

FIG. 2 depicts an exemplary flow diagram **200** depicting an exemplary process for segmenting an application's users and/or event information. At step **210** input data is fed into a machine learning algorithm such as a neural network. In some embodiments, input data can include activity logs obtained from an application. In some embodiments, activity logs are made up of a plurality of events. In some embodiments, activity logs can include event information such as an indication of an associated user and/or an indication of session (i.e. events tied together in a single use of an application).

At step **220** the algorithm collects inputted data and inspects the group of events. In some embodiments, the algorithm can constantly inspect data as it is received. In other embodiments, the algorithm can inspect data at predetermined times. Alternately, the algorithm can be configured to inspect data after receiving a predetermined number of events.

At step **230** the algorithm determines patterns exemplified by the inputted data. Patterns can be indicated by matching different characteristics shared by the inputted data. For example, patterns can be determined by matching events based on users, sessions, nature of activity etc. At step **240** the algorithm generates a list of segments with associated metadata based on the observed patterns. The accompanying metadata can include characteristics about each segment

such as those described above. Indications such as application features that users in a given segment tend to use regularly, the frequency by which those features are used, the length of an average session, etc., can be further utilized.

After generating a plurality of segments at **240** the algorithm can group application users into the appropriate segment at step **250**. Each segment will be indicative of shared characteristics between groups of users. In some examples, the users or other event information provided as input to the machine learning algorithm are subsequently tagged with one or more of the plurality of segments generated as output. In some examples, an association between the users or other event information and the plurality of generated segments can be stored in a database. In some examples, the event information corresponding to users of the application and the database of stored associations includes user information as well as the associations between the users associated with each event and the plurality of generated segments. In some examples, events can be matched to a user in the database using the plurality of generated segments at least in part to perform the matching.

The process can then return to step **210**, repeating as additional data is gathered and inputted into the algorithm. In this way, applications can stay up to date with current users and better track and understand the needs of particular groups of current users.

Figures

Figure 1

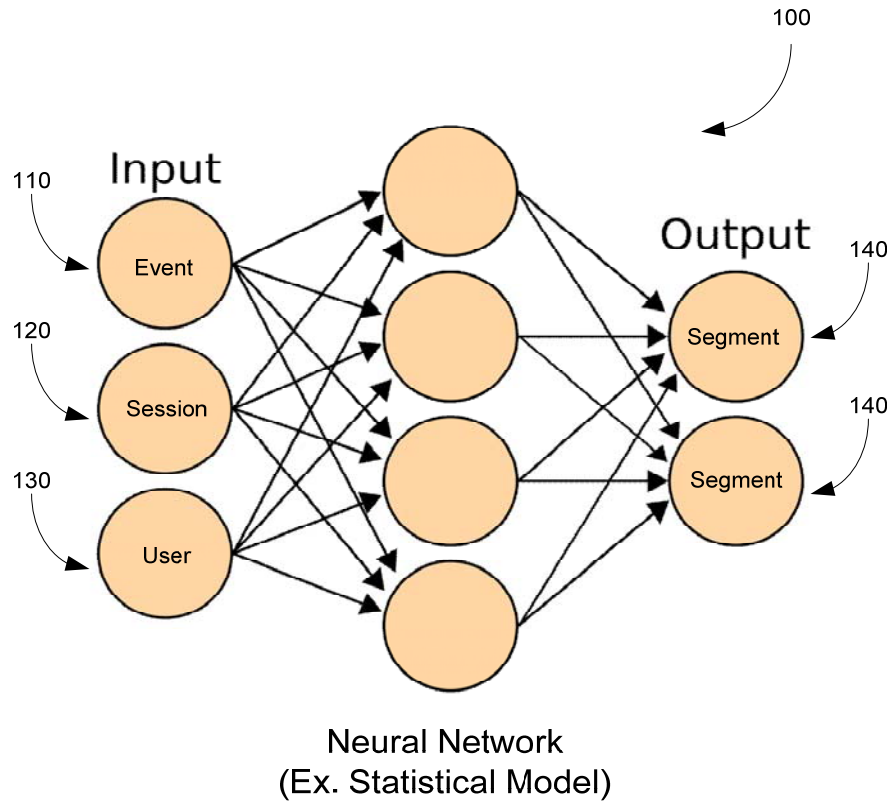
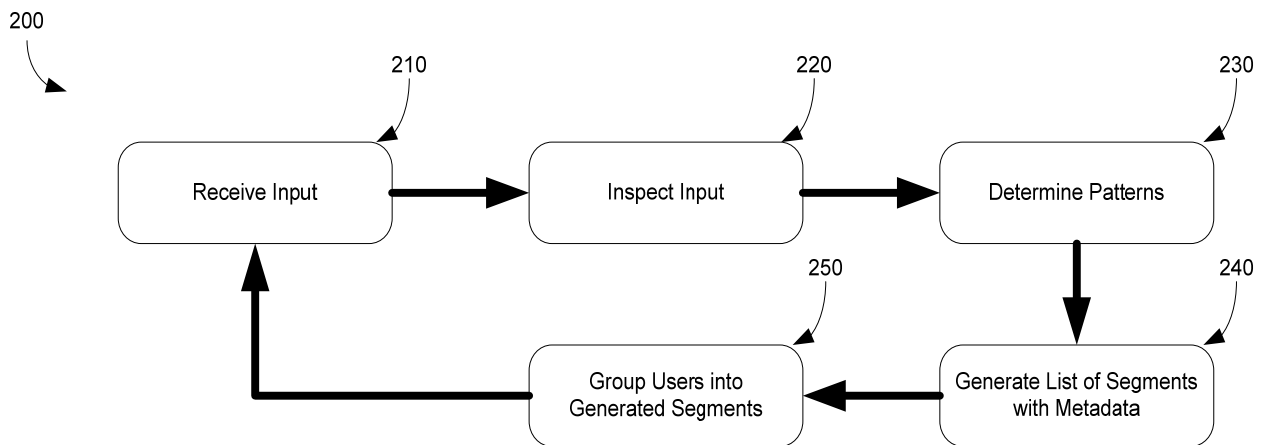


Figure 2



Abstract

The present disclosure provides systems and methods for automating the entire process of segmentation analytics for users of an application. More particularly, input data is received by a machine learning algorithm. The algorithm inspects the data to determine patterns. Upon discovering patterns of event information the algorithm generates a list of segmentations with applicable characteristics. An application's users are grouped together, matching user characteristics with those shared by a particular segment. Over time the algorithm learns from an increased amount of data, generating more accurate and comprehensive user segmentations with time and increased use. Keywords associated with the present disclosure include: machine learning; neural network; events; users; sessions; segmentation; population; event based machine learning.