June 2020

Multi-party Computation to Train a Prediction Model for Advertisement Selection

Anonymous

Multi-party Computation to Train a Prediction Model for Advertisement Selection

ABSTRACT

Online advertising networks utilize prediction models to determine the likelihood of an advertisement delivered to a user meeting a particular objective, e.g., making a purchase from the advertiser. Such models are trained based on historical information about users that clicked the ad logged by the advertising network and purchase data obtained from the advertiser. This mechanism is at odds with privacy since the advertising network receives user data from the advertiser. This disclosure describes the use of multi-party computation techniques to train a predictor. An MPC exchange is performed in which labels from the advertiser and a feature vector of user features from the ad network are utilized to train a prediction model. Partial models are obtained for each advertiser and are combined to obtain a trained predictor. User privacy is thus preserved since no user data from the advertiser is revealed to the advertising network and vice versa.

KEYWORDS

- Multi-party computation (MPC)
- Privacy-preserving computation
- Ad optimization
- Differential privacy
- Conversion rate
- Conversion predictor
- Online advertising
- Machine learning
- Differential privacy

BACKGROUND

Advertisers place online ads through an advertising network in various contexts, e.g., websites, social media applications, messaging applications, image sharing applications, etc. Advertisers have an objective associated with their advertisements. An example objective is that the advertisement should drive people to purchase things from the advertiser's website. The effectiveness of a particular advertisement of this type can be determined based on the number of the users that clicked the advertisement that chose to purchase a product from the advertiser's website, which can be termed as conversion rate.

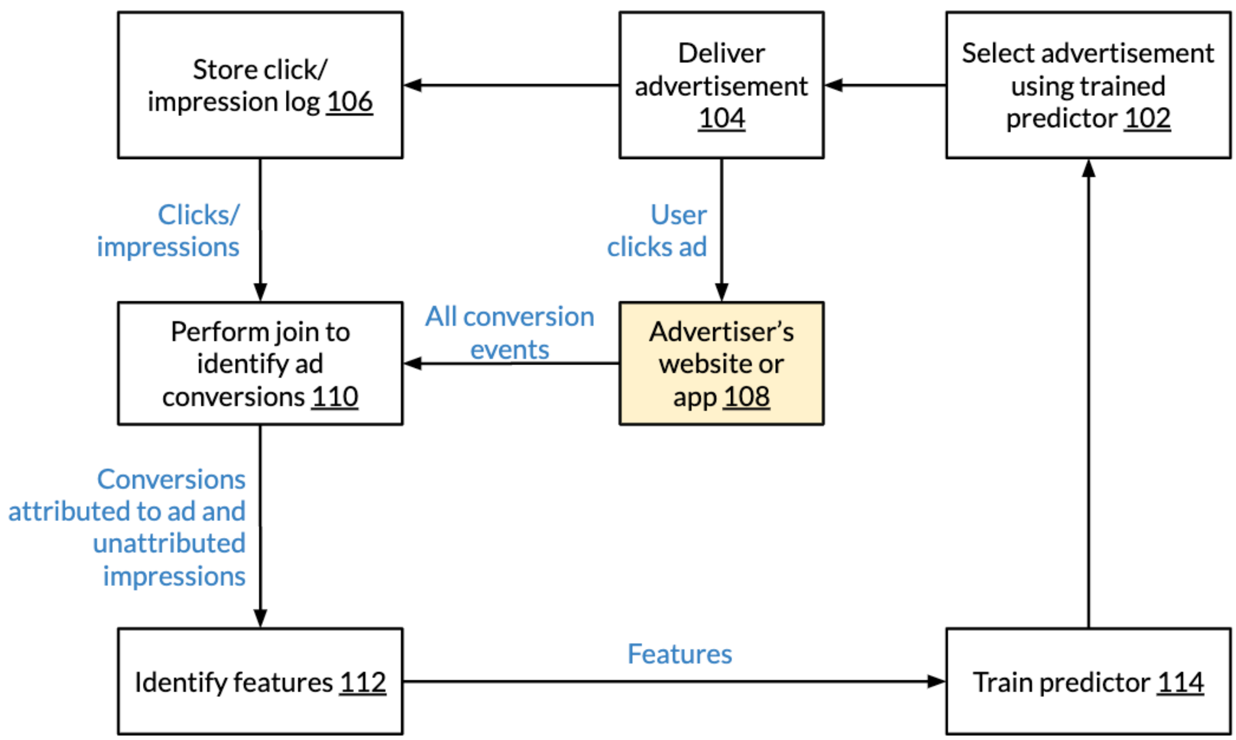

Fig. 1: Training and using an ad conversion predictor using data from advertiser

Fig. 1 illustrates an example process to train an advertisement conversion predictor using data obtained from an advertiser's website or app. An advertisement is selected (102), e.g., using the trained predictor if available, or by other techniques if no trained predictor is available. The selected advertisement is delivered to the user (104), e.g., via a website or app that the user visits.

A click/impression log is stored (106) that indicates an ad impression event (ad was shown to the user) and an ad click event (the user clicked on the ad). When the user clicks on the advertisement, the user is redirected to the advertiser's website or app (108).

The advertiser obtains data regarding purchase activity on their website (108) and sends conversion events back to the advertising network. For example, such data can include user identifiers for users that made a purchase. The advertising network then uses such data in combination with the click/impression log, e.g., by performing a join, to identify users that were associated with the ad conversions (110). Features of such users are identified (112) and are utilized to train an estimated conversion rate predictor (114). For example, the predictor can be implemented using machine learning techniques. The trained predictor generates subsequent predictions of how likely a user to whom an ad is to be served is to convert, e.g., to make a purchase from the advertiser's website. Thus, the trained predictor can be used in advertisement delivery such that an advertisement is shown only to people that are determined to convert, e.g., likely to make a purchase after seeing the advertisement.

As can be seen, training of the predictor is based on data regarding users that saw the advertisement (obtained by the advertising network) and users that made a purchase (obtained from the advertiser). The set of users that saw the ad and made a purchase serves as positive samples, while the set of users that saw the ad but did not make a purchase serves as negative samples. However, training the predictor in this manner necessitates that the advertising network receive user data, e.g., purchase data, even when the advertising network only needs the data for the specific purpose of training the predictor. Such a mechanism is not feasible when user data is not available for some reason, e.g., user's privacy preferences and/or rules.

DESCRIPTION

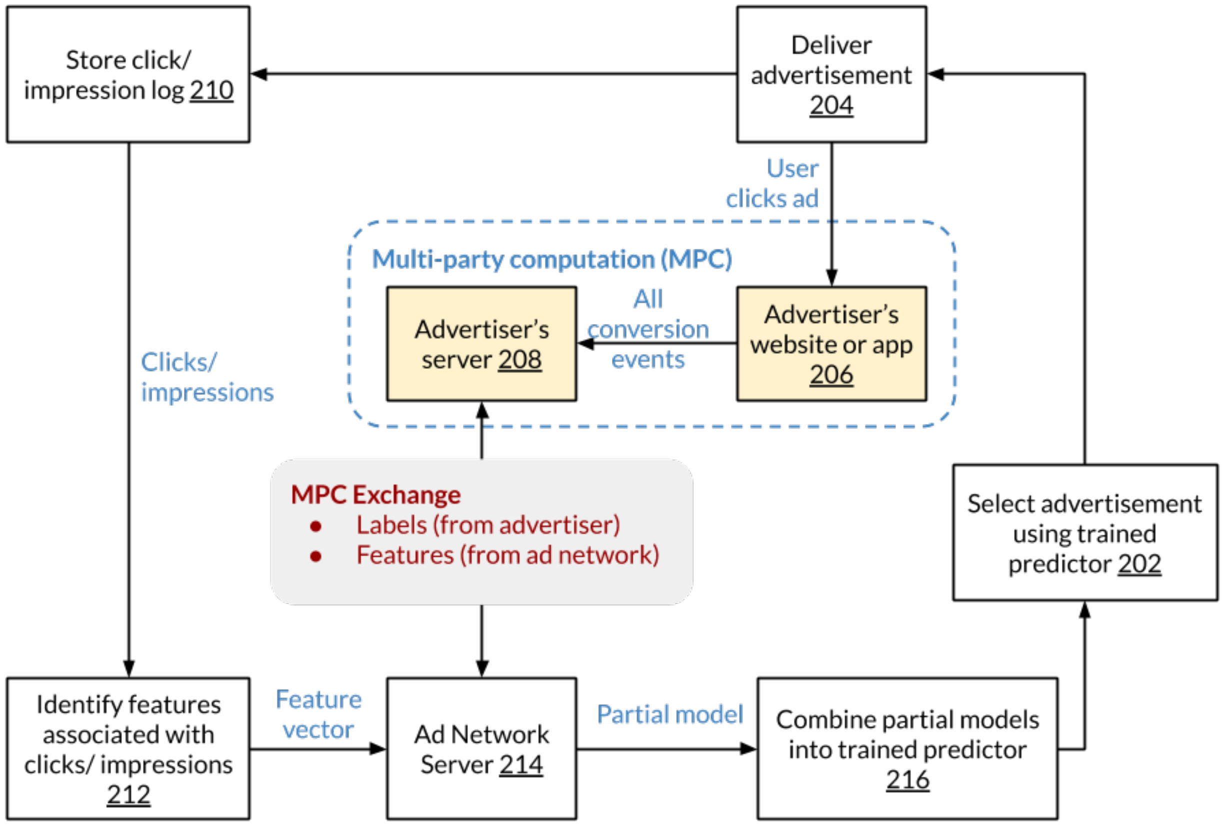

Fig. 2: Use of multi-party computation for training a predictor without revealing user information to counterparty

Fig. 2 illustrates an example process to train an advertisement conversion predictor without access to data obtained from an advertiser's website or app. An advertisement is selected (202), e.g., using the trained predictor if available, or by other techniques if no trained predictor is available. The selected advertisement is delivered to the user (204), e.g., via a website or app that the user visits. When the user clicks on the advertisement, the user is redirected to the advertiser's website or app (206). A click/impression log is stored (210) that indicates an ad impression event (ad was shown to the user) and an ad click event (the user clicked on the ad).

The advertiser retrieves data regarding purchase activity on their website and provides that data to a server (208) controlled by the advertiser. For example, such data can include user identifiers for users that made a purchase.

The advertising network matches the click/impression log to identify users that were associated with the ad conversions and identifies features of such users (212). These features are provided to an ad network server (214).

The advertiser's server (208) and the ad network server (214) engage in a multi-party computation (MPC) exchange. The advertiser's server utilizes labeled data, e.g., identifiers of users that made a purchase and other users that did not make a purchase. For example, in the MPC exchange, the advertiser's server can send a set of user identifiers (e.g., email addresses) that made a purchase via the advertiser's website or app, while the ad network server can provide a feature vector of user features (e.g., a dictionary of email address->feature vector) of users that clicked the ad, keyed by email address. Within the MPC exchange, the data is joined in a meaningful way to train a usable, privacy-preserving derivative, e.g., to obtain a simple linear regression model (or other model). After training, the model is provided to the ad network server (214). Performing a multi-party computation ensures that the model is trained without the labels being revealed to the ad network or the user features being revealed to the advertiser during the MPC exchange.

Model training is performed via a separate MPC exchange with each advertiser, and multiple partial models are obtained. The models are then combined (216) into a single trained predictor that can rank advertisements and is usable to select advertisements to deliver to users. For example, federated learning techniques are utilized to combine the partial models trained in respective MPC exchange with each advertiser.

By adding sufficient noise to the feature vectors that go into the training the model, it can be ensured that individual features cannot be traced back to individual users for whom the ad detected clicks. The trained model essentially enables a better estimation of likelihood of conversion for different users based on respective feature vectors descriptive of each user. Thus, the trained model can optimize ad delivery, while maintaining user privacy for both the advertiser and the ad network.

CONCLUSION

This disclosure describes the use of multi-party computation techniques to train a predictor. An MPC exchange is performed in which labels from the advertiser and a feature vector of user features from the ad network are utilized to train a prediction model. Partial models are obtained for each advertiser and are combined to obtain a trained predictor. User privacy is thus preserved since no user data from the advertiser is revealed to the advertising network and vice versa.

REFERENCES

1. SCALE-MAMBA Software