**Ad exchange game workshop**

**Tel-Aviv University 2016**

**Project report**

**Mor Chen**

**Ran Feder**

**Lee Hellow**

**Nofar Socher**

# **Introduction**

**Now days, advertisements are everywhere, and are one of the largest source of income sites and apps have.**

**As a result of the growing advertisement market, new technology called ad exchange was developed in order to find** efficient way for trading advertising opportunities.

Whenever a user visits a publisher's web page, the ad exchange conducts an auction for the ad among relevant ads from ad networks.

The winning ad is then displayed and the corresponding ad network is charged.

**In this workshop we have simulated advertising agent, which in each day of the game bids to win advertising campaign contracts, and submits a bidding strategy to the Ad Exchange, while the main purpose of the game is to gain the biggest profit.**

# **Game elements**

**The game has three main components:**

**Campaign Opportunity auction – Our agent bids to win advertising campaign contracts, we need to bid low enough so that the publishers would agree to pay us and get campaigns, but not too low so we could make profit from the campaign..**

**User Classification Service auction – Our agent bid to get the highest UCS level in order to get the best quality of matching, the higher our UCS level will be the more accurate information we get.**

**Bid bundle auction – Our agent bid to get as many impressions as he can in order to finish the campaign reach impressions and get high rating.**

## **Campaign Opportunity**

## **User Classification Service**

## **Bid Bundle**

Every day we build and send bid bundle for each of our active campaigns.

The Bid bundle is the bid we offer in order to win impressions. The more impressions we get, the faster we finish the campaign and raise our quality rating.

The bid bundle depends on many different parameters in the game, we chose what we thought to be the most important parameters, and built three different strategies based on them.

### **Bid parameters**

For each bid bundle we collect data in order to build our bid. The data parameters are:

- The average revenue our agent get for each impression, most of the time we will use this parameter as our higher bound for our bid.

- Factor of the ad type and device coefficients.

|  |  |  |
| --- | --- | --- |
|  | **Text** | **Video** |
| **Desktop** | 1 | coefVideo |
| **Mobile** | coefMobile | coefMobile\*coefVideo |

– Factor of how many days left for this campaign - the less days left, the more we want to get impressions to finish the campaign so we bid higher.

**if** (daysLeft == 1) {

daysLeftFactor = 1.175;

} **else** **if** (daysLeft == 2) {

daysLeftFactor = 1.1;

} **else** {

daysLeftFactor = 0.9 + (((**double**) (campaignLength - daysLeft + 1)) / (campaignLength \* 10));

}

– This parameter will tell us about our state in the game, we calculate the ratio between the campaign impressions state and days left state.

If the ratio is low meaning our progress is good, otherwise our progress is not that good and we need to be more aggressive and get more impressions

– Based on the User Population Probabilities table from the spec, this is a factor of how popular the campaign market segment is.

If the market segment is big, meaning there are a lot of potential impressions from this segment, so our bid will not change.

But if the segment is rare, meaning there are not many impressions from this campaign market segment so we bid higher for each impression.

– The bid random factor depends on the parameter. The parameter tells us about our progress in the game, if our progress is good then the is low, and the random factor will be lower, otherwise our progress is not that good and we give higher random factor to enlarge our bid and get more impressions

- Factor of how many campaign are running at the current day and have the same market segment as our campaign - the competition we have for each impression.

### **Bid Bundle strategies**

We decided to create three different strategies for each part of the game –

**Stable bid strategy**– we use this strategy at most of the game days, we build the bid as a function of all the data parameters we collected.

We calculate this strategy after we`ve build object of type BidBundleData and calculated all the relevant factors. As a part of this strategy we also calculate the bid Knn factor (see explanation in the next section).

**First days strategy** – at the first days of the game we decided, after running several simulations with only the stable bid strategy, to be more aggressive and bid higher to get impressions in order to finish campaigns and get high rating. We`ve learned that the first days are the most critical and if we don’t finish the first campaign, we will get a low rating and this will affect the entire game.

This strategy is based on calculating the stable strategy as described before, and then enlarge it by multiplying it with random number bigger then 1.

We will use this strategy at the first 12 days of the game.

**Last days strategy** – As a part of our strategy, we decided in this part to focus on two important points:

* Getting as much profit as possible.
* Giving less importance to the quality rating.

In order to perform this, our bid will be lower than usual and we aim to get impressions at lower cost.

We will use this strategy at the last 8 days of the game.

**Days 52 - 60**

**Days 0 - 12**

**Days 13 - 51**

### **Bid Knn algorithm**

As a part of our strategy, we wanted to learn from previous games in order to build better bid bundle, so we implemented the K nearest neighbor's algorithm.

K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions).

We collected data from previous games about our campaigns and bids, and based on this data we calculate the Knn factor which is the average bids from previous successful bid bundles.

The Knn algorithm most important factor is the distance function which decide if the observation is close enough to be considered or not. Our distance function is based on four parameters which we thought are the most significant to decide whether a campaign is relevant for us and can improve our bid or not- campaign impressions, market segment and budget. For each parameter we calculates the ratio between the history campaign and current campaign, and the total distance result is the sum of all three ratios. After calculating the campaigns distance we checked if the history campaign is good, meaning we got more than 0.5 impressions we bet for, and if so we added the history campaign to the similar campaigns list. The Knn factor is the average bids of all similar campaigns bids.

### Data collection

We collected and saved all the data we needed for the bid bundle in two different classes –

BidBundleData – in this class we saved all the parameters mentioned in the bid bundle parameters section in order to calculate the bid.

On each day and for each campaign query we create an object of this class and using the data we collected in the CampaignData class and GameData class we build the BidBundleData object.

CampaignBidBundleHistory – in this class we save data about old bid bundles.

For each bid bundle we create object of this class and link it to a list of object from class CampaignBidBundleHistory.

This object saves parameters like campaign budget, campaign reach impressions, bid results etc. and we use this information to calculate the bid Knn factor.