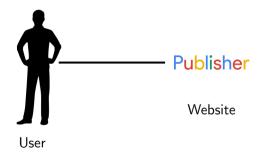
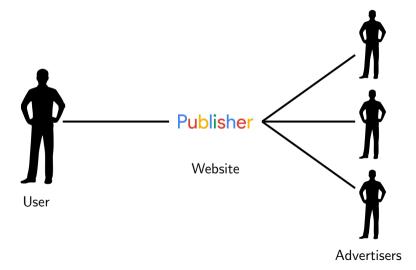
Online Learning Applications

Part 6: Online advertising and auctions

Introduction to online advertisement

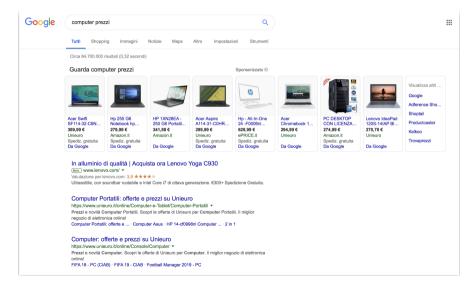




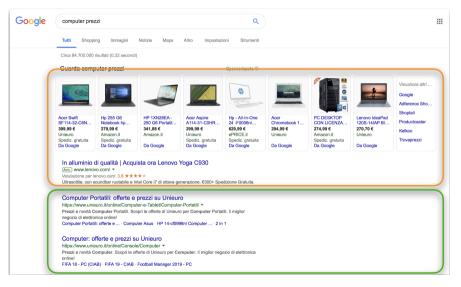


- Advertisers want to show ads to users visiting a web page
- An user visits the web page
- The **publisher** that handles the ad **slots** on the web page assigns each slot to an advertiser

Example



Example



Ads

Results

Interaction

The interaction among the publisher and the advertisers goes as follows:

Interaction

- An user arrives
- 2 Each advertiser bids
- The publisher assigns each slot to an advertiser based on bids
- 4 The advertisers pay the publisher if their ad is clicked

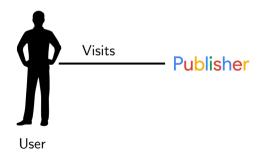
Both advertisers and publisher have a problem to deal with.

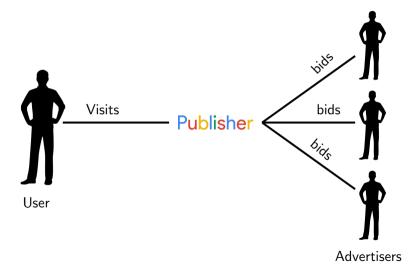
Advertiser's problem

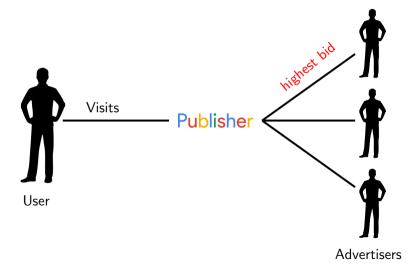
Choose how to bid.

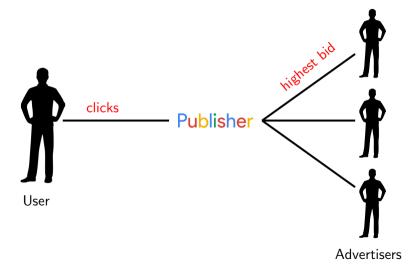
Publisher's problem

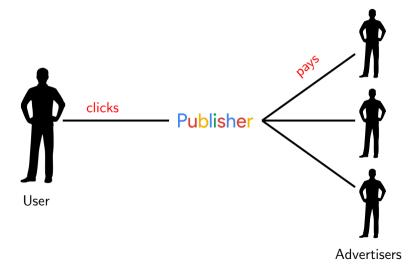
Choose how to allocate slots and set payments.



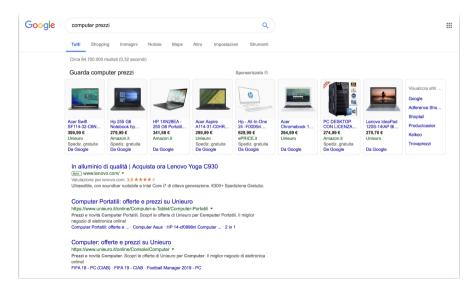




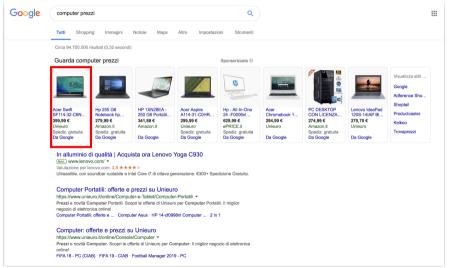




Slot prominence (some slots are better than others)

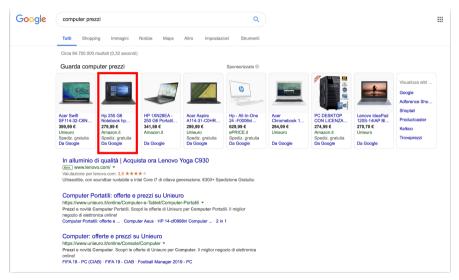


Slot prominence (some slots are better than others)



Best slot

Slot prominence (some slots are better than others)



Second-best slot

Publisher's perspective

How to set payments and allocation?

The utility of an advertiser a for visualizing an ad on a slot s is:

$$\lambda_s q_a v_a$$

- \bullet λ_s is the probability that the user observes the slot (**slot prominence**)
- q_a is the advertiser's click probability given the ad has been observed by the user
 (ad quality)
- v_a is the advertiser's value per click (ad value)
- \bullet λ_s q_a is the advertiser's click probability (click trough rate)
- \blacksquare q_a v_a is the **expected utility** of an advertiser if the slot is observed

How to allocate slots?

Suppose that the **publisher knows all the parameters**. The optimal allocation solves:

$$\max_{s(\cdot)} \sum_{a} \lambda_{s(a)} \ q_a \ v_a$$

where:

- The function $s(\cdot)$ assigns to each ad at most one slot
- If an ad is not allocated, $\lambda_{s(a)} = 0$

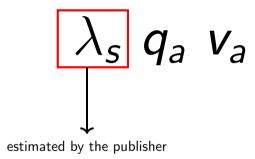
The optimal s():

- Assigns the best slot to the ad with largest $q_a v_a$
- \blacksquare assigns the second-best slot to the ad with second-largest $q_a v_a$

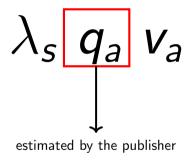
. . . .



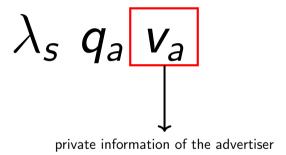


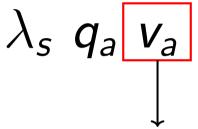












private information of the advertiser use the bid instead of the valuation

How to allocate slots?

The publisher chooses the allocation solving:

$$\max_{s(\cdot)} \sum_{a} \lambda_{s(a)} \ q_a \ b_a$$
,

where:

- The function $s(\cdot)$ assigns to each ad at most one slot
- If an ad is not allocated, $\lambda_{s(a)} = 0$

The optimal s():

- Assigns the best slot to the ad with largest $q_a b_a$
- \blacksquare assigns the second-best slot to the ad with second-largest q_ab_a

. . . .

Truthful auctions

Problem

The publisher uses bids as an estimation of the valuations.

Truthful auctions

Problem

The publisher uses bids as an estimation of the valuations.

Solution

Design an auction that incentivizes the advertisers to bid their valuations. This is called a **truthful auction**.

Second-price auctions

If there is only one slot, the publisher can use second-price auctions:

Second-price auction

- \triangleright The advertiser a_1 with larger q_ab_a wins the slot
- ightarrow The advertiser pays $p_{a_1}=rac{q_{a_2}b_{a_2}}{q_{a_1}}$ if the ad is clicked
- $\triangleright a_2$ is the advertiser with the second largest q_ab_a
- lacksquare The advertiser always pays $p_{a_1} \leq b_{a_1}$

Second-price auctions

If there is only one slot, the publisher can use second-price auctions:

Second-price auction

- \triangleright The advertiser a_1 with larger q_ab_a wins the slot
- ightarrow The advertiser pays $p_{a_1}=rac{q_{a_2}b_{a_2}}{q_{a_1}}$ if the ad is clicked
- \triangleright a_2 is the advertiser with the second largest $q_a b_a$
- lacksquare The advertiser always pays $p_{a_1} \leq b_{a_1}$
- \blacksquare The auction is truthful \to each advertiser is incentivized to report the true valuation

VCG Auctions

If there are multiple slots, the publisher can use a VCG (Vickrey-Clarke-Groves) auction.

VCG auction

- \triangleright Recall that s() is the allocation function that assign slots to advertisers
- ▶ Recall that s() assigns slots in decreasing order of expected value
- \triangleright Each advertiser a pays $p_a = \frac{1}{\lambda_{-(a)}q_a}(Y_a X_a)$ if the ad is clicked
- $V_a = \sum_{a' \neq a} \lambda_{s(a')} q_{a'} b_{a'}$ is the utility of the other advertisers in the optimal allocation $V_a = \max_{s'(\cdot)} \sum_{a' \neq a} \lambda_{s'(a')} q_{a'} b_{a'}$ is the utility of the other advertisers if a does not take part to the auction

■ Each advertiser "pays" his externalizes on the other advertisers

VCG Auctions

If there are multiple slots, the publisher can use a VCG (Vickrey-Clarke-Groves) auction.

VCG auction

- \triangleright Recall that s() is the allocation function that assign slots to advertisers
- ▶ Recall that s() assigns slots in decreasing order of expected value
- \triangleright Each advertiser a pays $p_a = \frac{1}{\lambda_{-(a)}q_a}(Y_a X_a)$ if the ad is clicked
- $V_a = \sum_{a' \neq a} \lambda_{s(a')} q_{a'} b_{a'}$ is the utility of the other advertisers in the optimal allocation $V_a = \max_{s'(\cdot)} \sum_{a' \neq a} \lambda_{s'(a')} q_{a'} b_{a'}$ is the utility of the other advertisers if a does not take part to the auction
- Each advertiser "pays" his externalizes on the other advertisers
- lacktriangle The auction is truthful ightarrow each agent is incentivized to report the true valuation

If there is a single slot, VGC is equivalent to a second-price auction

First-price auctions

Truthful auctions have complex payments (see, e.g., VCG). Publishers also use non-truthful auctions:

- Non-truthful auctions are easier to understand
- With a single slot, the most common is the **first-price auction**

First-price auction

- \triangleright The advertiser a_1 with larger q_ab_a wins the slot
- \triangleright The advertiser pays $p_{a_1} = b_{a_1}$ if the ad is clicked
- The winning advertiser pays exactly what they bid

First-price auctions

Truthful auctions have complex payments (see, e.g., VCG). Publishers also use non-truthful auctions:

- Non-truthful auctions are easier to understand
- With a single slot, the most common is the **first-price auction**

First-price auction

- \triangleright The advertiser a_1 with larger q_ab_a wins the slot
- \triangleright The advertiser pays $p_{a_1} = b_{a_1}$ if the ad is clicked
- The winning advertiser pays exactly what they bid
- A The auction is not-truthful → the advertisers are incentivized to reduce the bid in order to decrease the payment

Non-truthful auctions

First-price auctions can be generalized to multiple slots.

Generalized first-price auction

 \triangleright Each advertiser a pays $p_a = b_a$ if the ad is clicked

Non-truthful auctions

First-price auctions can be generalized to multiple slots.

Generalized first-price auction

 \triangleright Each advertiser a pays $p_a = b_a$ if the ad is clicked

Other non-truthful auctions commonly used are a generalization of second-price auctions.

Generalized second-price auction

- \triangleright Recall that s() is the allocation function that assign slots to advertisers
- \triangleright Each advertiser a pays $p_a = \frac{q_{a+1}}{q_a} b_{a+1}$ if the ad is clicked
- $\triangleright a + 1$ is the ad assign to the slot just after s(a)

Advertiser's perspective

Bidding strategies

Now, we consider the advertiser's perspective

- The advertiser wants to maximize his utility
- An advertiser takes part to a an auction every time a new user visit the page:
 - ▶ If the auction is truthful, they simply bid their valuation
 - ▶ If the action is not truthful, they must learn how to bid

Learning to bid in non-truthful auctions

The problem of bidding is non-truthful auctions is an **online learning problem**:

- The advertiser chooses a bid at each auction $t \in [T]$
- Depending on the assumptions the setting can be adversarial or stochastic:
 - ▶ The setting is adversarial since the other bidders play unpredictably affecting the possibility of winning the auction and the payment (i.e., the reward)
 - ▶ the setting is **stochastic** if we assume that there are **lots of other bidders** and their joint behavior is stochastic

Bidding strategies

The goal of an advertiser is usually not unconstrained utility maximization.

Budget constraint

The advertisers assign a **budget** to a marketing campaign.

Problem

The advertiser must choose how to spent the budget over time.

Example of campaign:

- The advertiser tries to display the ad whenever a specific keyword is searched on a search engine.
- The advertiser stops bidding when the budget is depleted



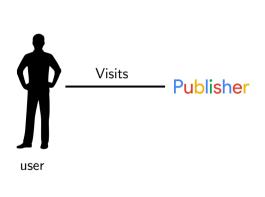
Publisher





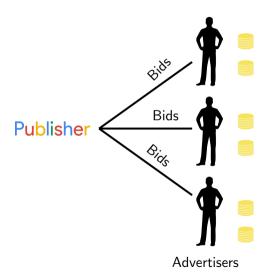
Publisher

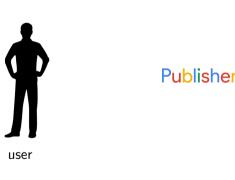


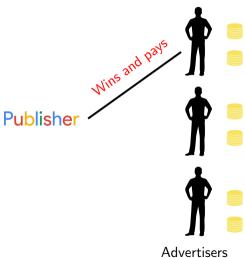










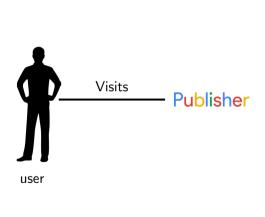




Publisher



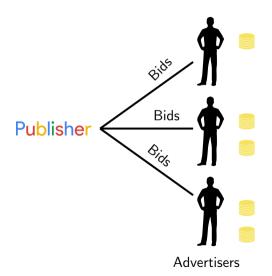
Advertisers



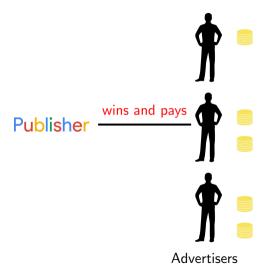








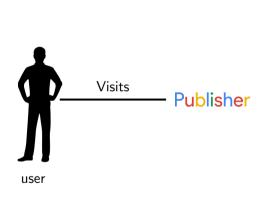






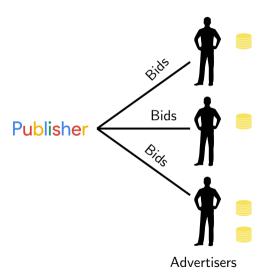
Publisher

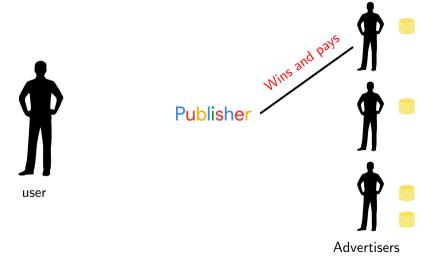














Publisher

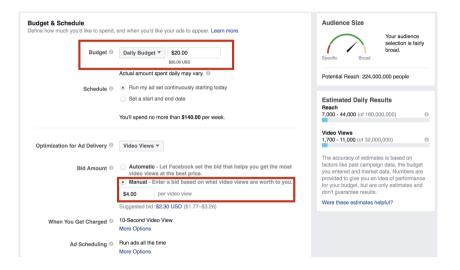




Delegation of the campaign management

- Handling this repeated bidding problem is complex
- Advertisers usually delegate this task to:
 - Ad networks
 - Publishers
 - Ad exchange
- The advertiser sets a small number of parameters and a **proxy bidder** handles the bidding.

Delegation of the campaign management



Proxy Bidders

Proxy bidders take in input a small number of parameter, handle the auctions, and report only aggregated results to the advertiser.

