Real-Time Bidding

-- of DSP Perspectives

賴旭昭 Hsu-Chao Lai

National Cheng Kung University

國立成功大學
National Cheng Kung University

About Me

- Hsu-Chao Lai 賴旭昭
- AI Scientist at NetDB Lab
- Ph.D. of NYCU CS
- Specialized in
  - Recommender systems
  - Real-time bidding
  - AI stock trading
- hclai@netdb.csie.ncku.edu.tw

Outline

- Introduction
- Bidding Strategy
  - Second-Price Auction
  - Bid Landscape
  - CTR Prediction
  - Budget Pacing
  - Cost Efficiency
- First-Price Auction
  - Bid Shading
- Takeaways

Display Ad

Wednesday, May 21, 2025

Today's Paper

The New York Times

S&P 500 -1.61% ↓

U.S. | World | Business | Arts | Lifestyle | Opinion | Audio | Games | Cooking | Wirecutter | The Athletic

Trump Lectures South African President in Televised Oval Office Confrontation

President Trump presented what he said was evidence of racial persecution of white South Africans. The country's president tried to correct the record.

5 MIN READ

Trump Says the U.S. Is Close to Brokering Peace Between Congo and Rwanda

2 MIN READ

Eric Lee/The New York Times

Heather Willensky for The New York Times

The 22 Best Pizza Places in New York Right Now

Some of the city's most famous pies didn't make the list, while some unexpected spots delivered superlative examples of the form.

10 MIN READ

Display Ad

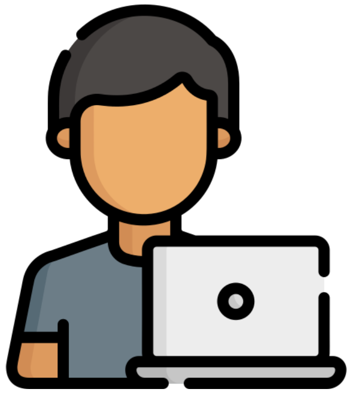
Advertisers

Target →

"Age 20"
"Male"
"Pokémon"
"Pizza"
"Travel"
…

Matching

← **Attribute**



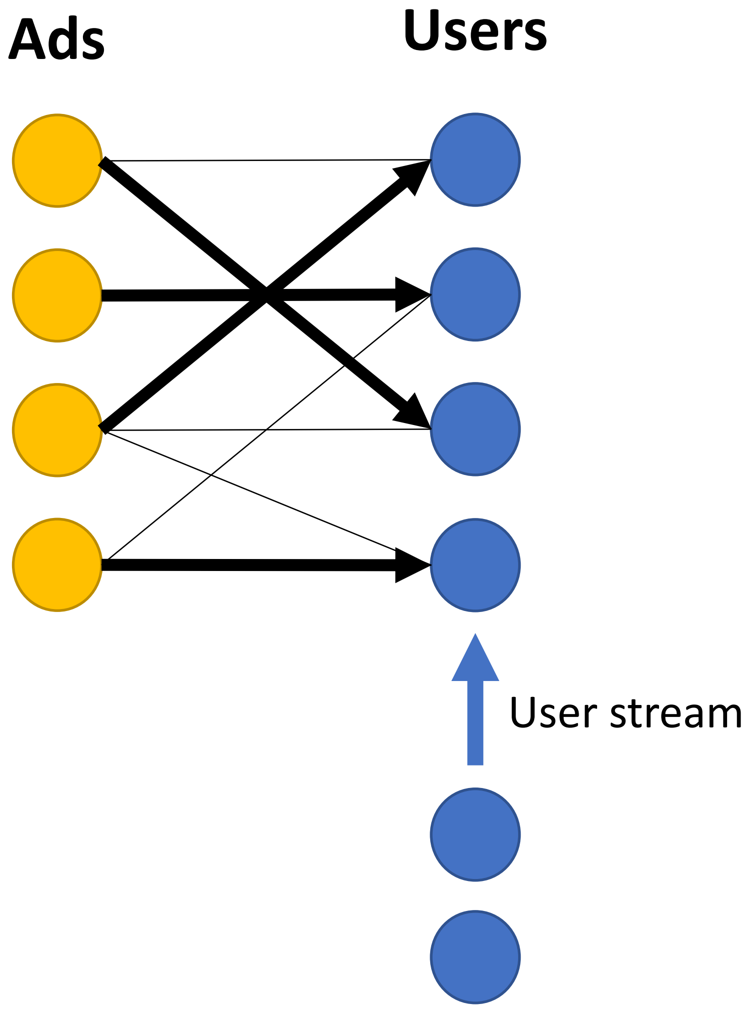
Internet Users

Publishers (websites) sell their columns to the advertisers

Online Matching

- **Problem:** find a matching between ads and the user stream
- **Constraint:** budget of each ad (campaign)
- **Goal:** maximize platform revenues
  - Cost Per Million Impression (CPM)
  - **Cost Per Click (CPC)**
  - Cost Per Action/Conversion (CPA)

- GREEDY yields 1/2 competitive ratio to OPT
- BALANCE yields 1-1/e≈0.63



Real-Time Bidding (RTB)

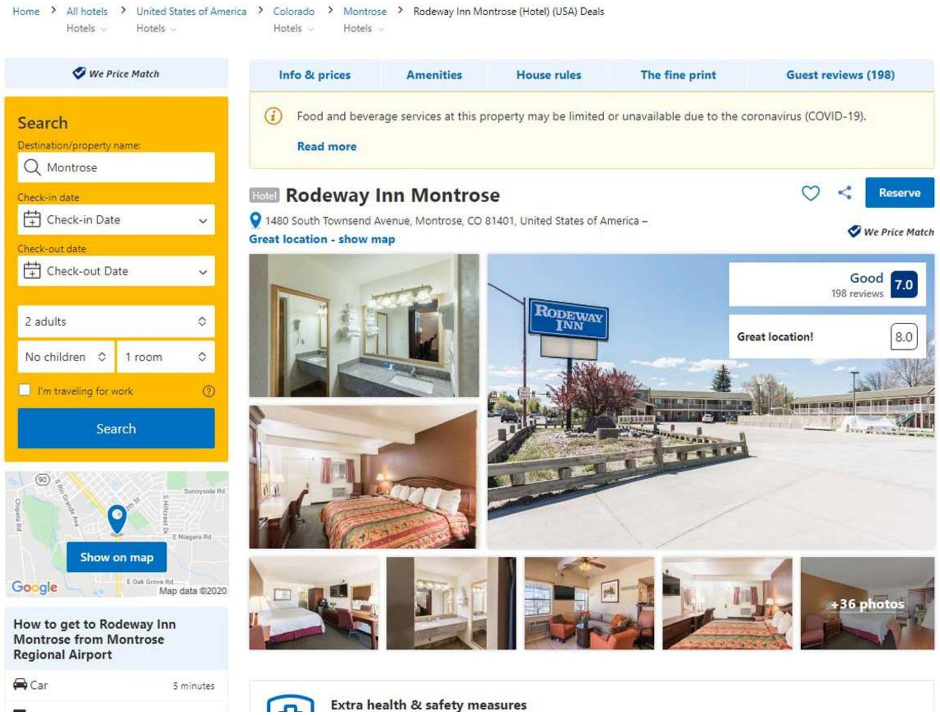
- Difficult to manage users, publishers, and ads explosions

- 2007-2008
- Advertisers **buy individual impressions** via **real-time auctions**
  - rather than purchasing bulk inventory in advance

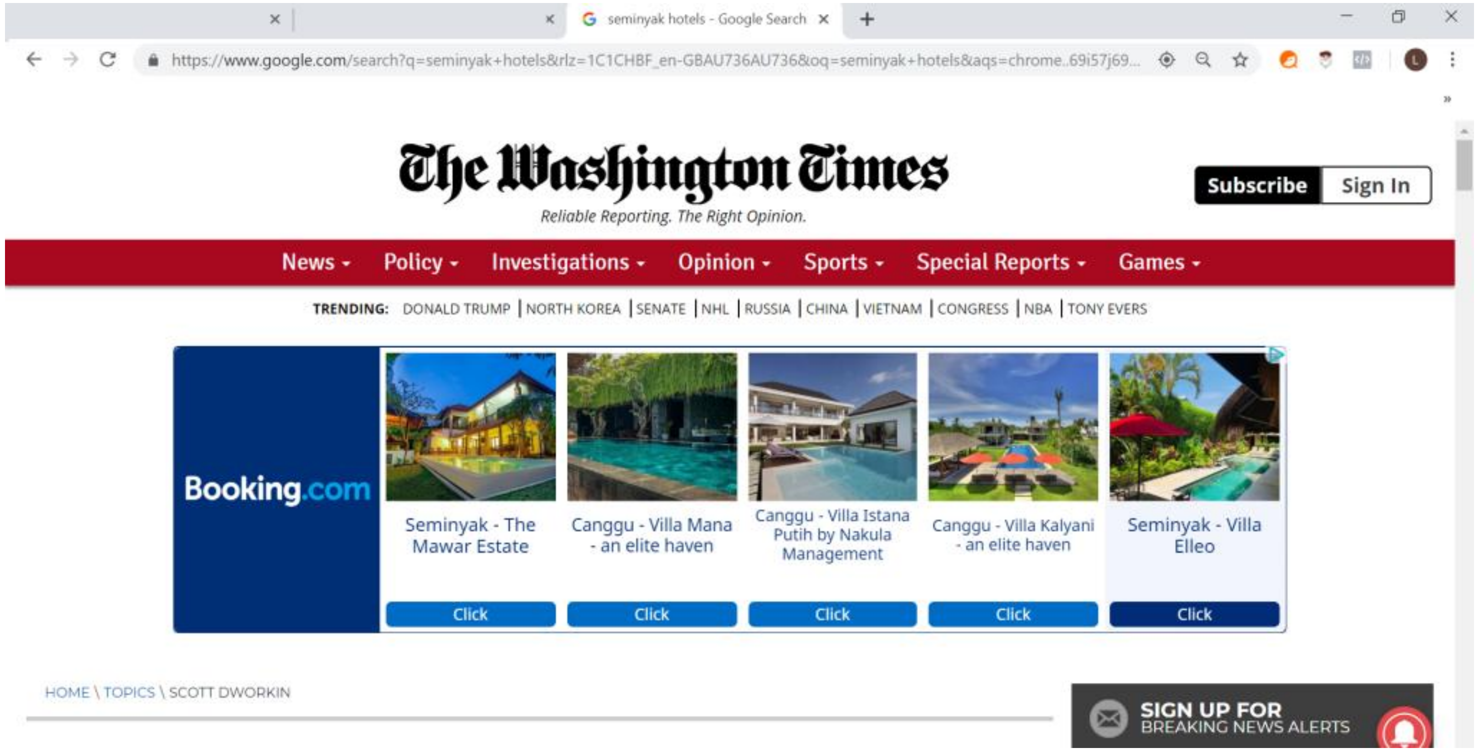
- Enable **behavioral targeting**
  - target users more precisely and optimize their ads in real time
  - Cookie!

Example of RTB

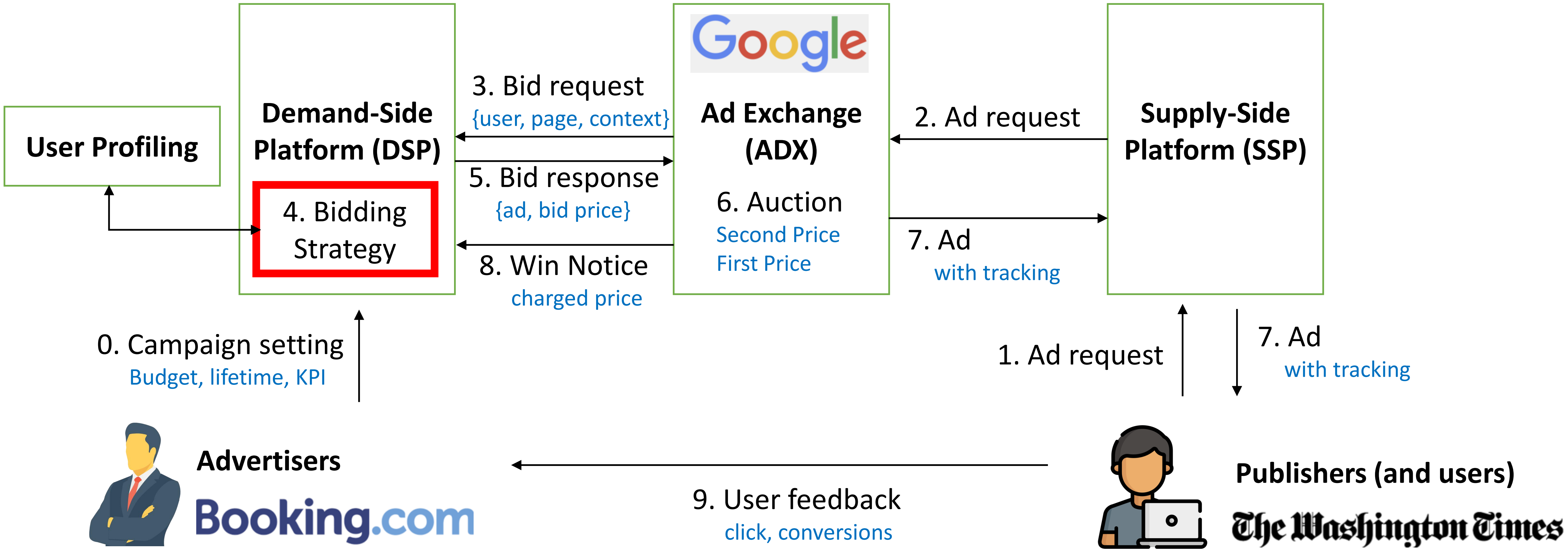
Search on Booking.com



Behavioral Targeting



RTB Architecture and Mechanism



RTB: Tremendous Data Volume

Daily Request Volume				
Country	**RTB**		**Stock Market**	
Taiwan	**Tenmax**	1B	**TAIEX**	1.7~3M
US	**Total**	>100B	**Total**	80~100M
China	**Total**	>100B	**Shanghai**	71M

Query Per Second	
Turn DSP	1.6M
Google Query	40K

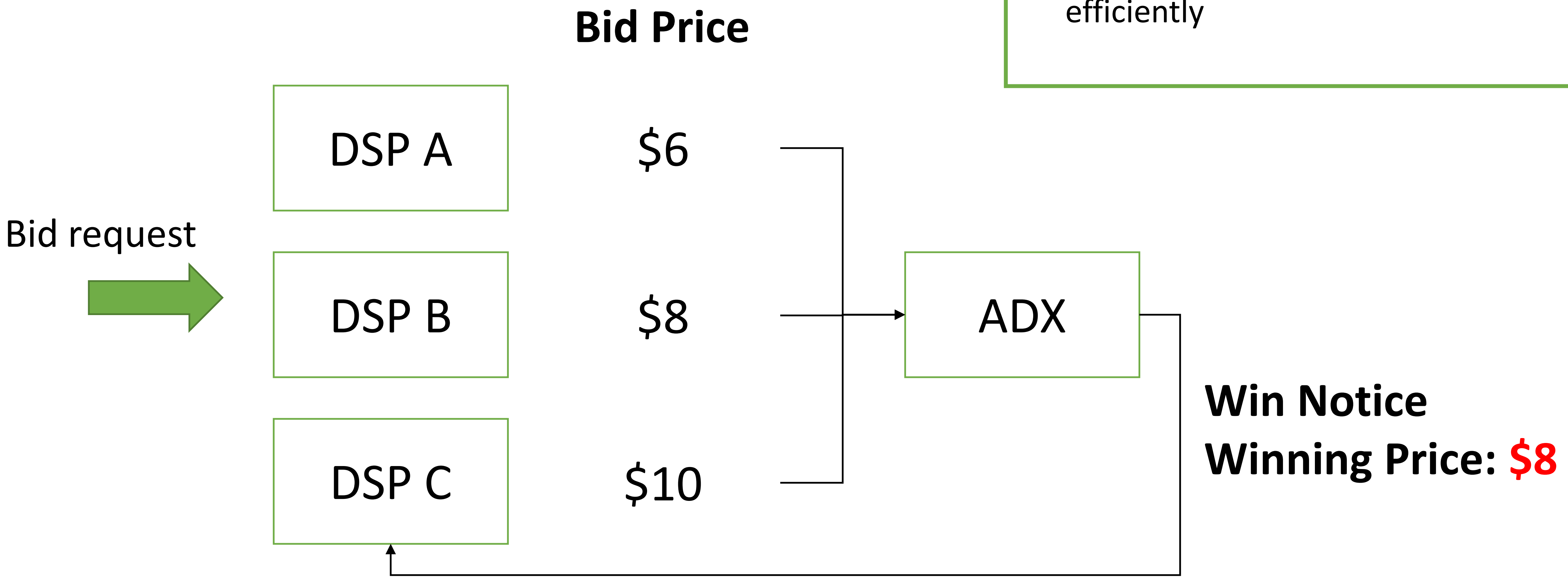
- RTB needs not only fast (<100ms) but also **precise** ← **Trade-off!**

Outline

- Introduction
- Bidding Strategy
    - Second-Price Auction
    - Bid Landscape
    - CTR Prediction
    - Budget Pacing
    - Cost Efficiency
- First-Price Auction
    - Bid Shading
- Takeaways

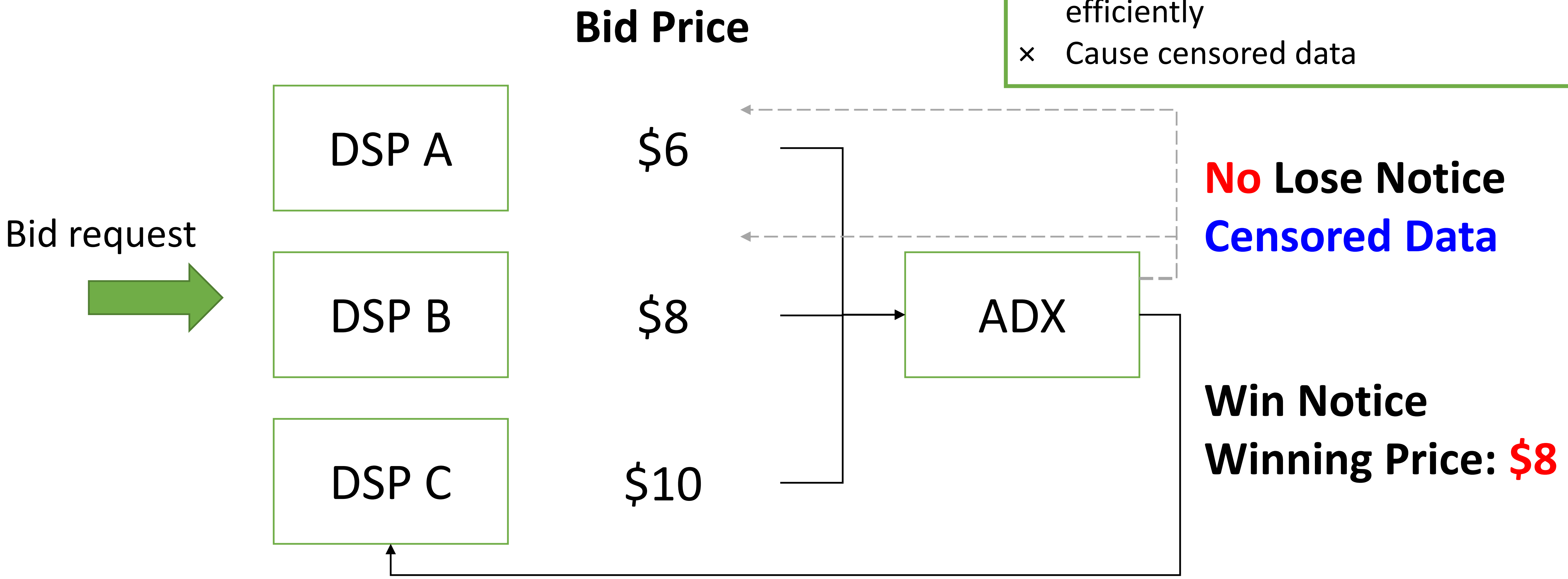
Second-Price Auction (SPA)

- ✓ Encourage DSPs bid they truly willing to pay
- ✓ Avoid Overpayment
- ✓ Allocate ads to who values it most efficiently



Censored Data

- ✓ Encourage DSPs bid they truly willing to pay
- ✓ Avoid Overpayment
- ✓ Allocate ads to who values it most efficiently
- × Cause censored data



Data Format

Bid Request Features	Bid price	Win	Win Price	Click
{banner, 1200x700, nytimes.com, 24, male}	5	**1**	**4**	1
{native, 800x800, facebook.com, 26, female}	4	**1**	**3**	NA
{video, 1200x1200, udn.com, 36, male}	3	**0**	**NA**	NA
{pop-up video, 1260x1260, pixnet.com, 42, male}	1	**0**	**NA**	NA

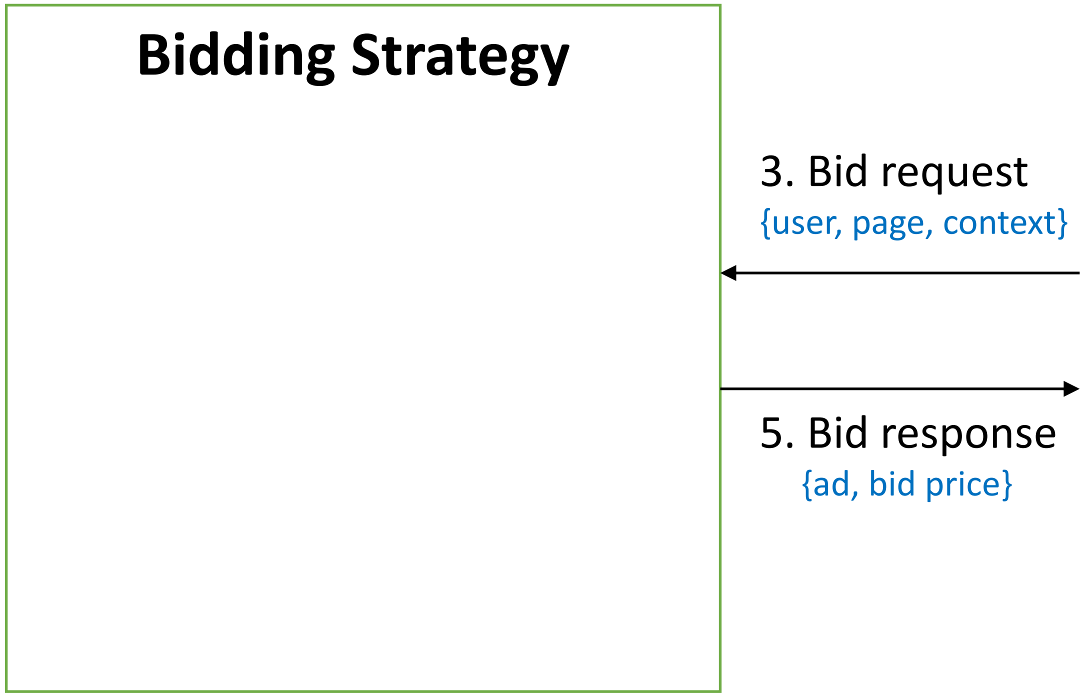
- **Bid Request Feature**: high-dimensional sparse binary vectors
  - Publisher information
  - Audience information
  - Impression details
  - Content and context
  - Targeting audiences
  - Etc…

Bidding Strategy

- **<u>Given</u>** a stream of bid requests and their features, ad features (budget, ad content, target audience), etc
- **<u>Determine</u>** bid prices for each bid request
- **<u>Such that</u>**

$$\max \mathrm{KPI} \quad \boxed{\#clicks}$$

$$s.t. \sum cost \le Budget$$



Bid price = Click-Through Rate (CTR)*Cost-Per-Click (CPC)

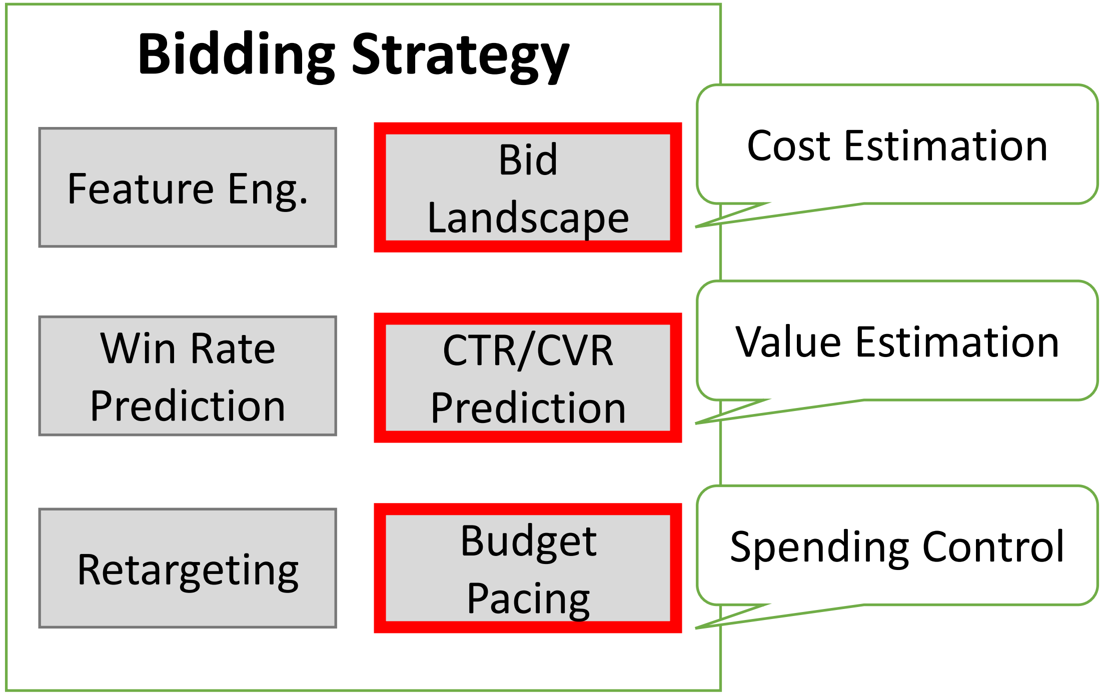
Bidding Strategy

- **Given** a stream of bid requests and their features, ad features (budget, ad content, target audience), etc
- **Determine** bid prices for each bid request
- **Such that**

$$\max \mathrm{KPI}$$

#clicks

$$s.t. \sum cost \leq Budget$$



Bid price = Click-Through Rate (CTR)*Cost-Per-Click (CPC)

Outline

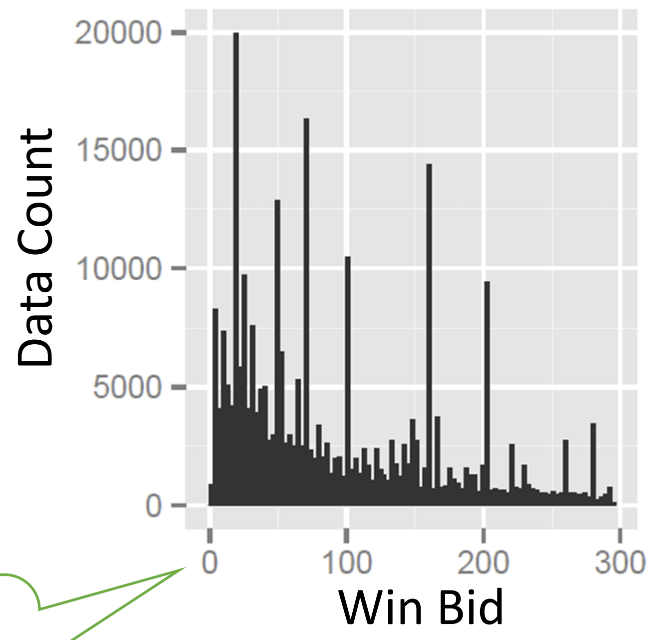
- Introduction
- Bidding Strategy
  - Second-Price Auction
  - Bid Landscape
  - CTR Prediction
  - Budget Pacing
  - Cost Efficiency
- First-Price Auction
  - Bid Shading
- Takeaways

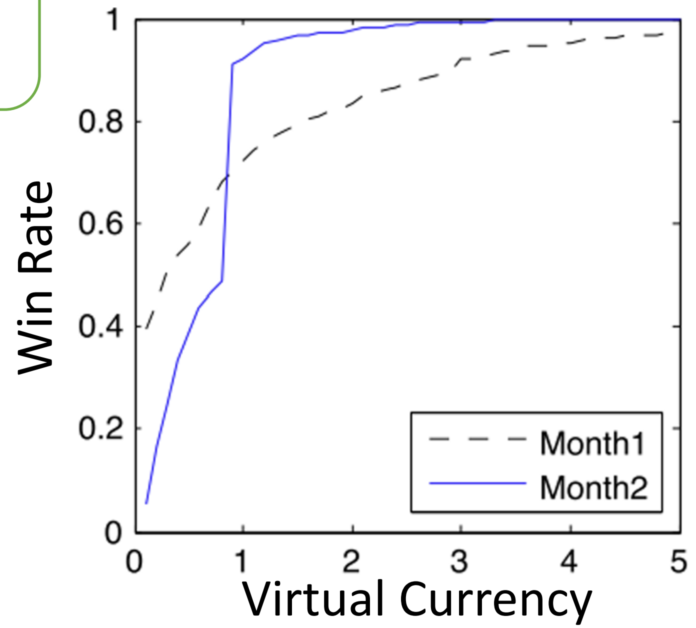
Bid Landscape

- The distribution (or analysis) of **bid prices** and **outcomes** within an online advertising auction environment
- **Goal**: Depicts market competition

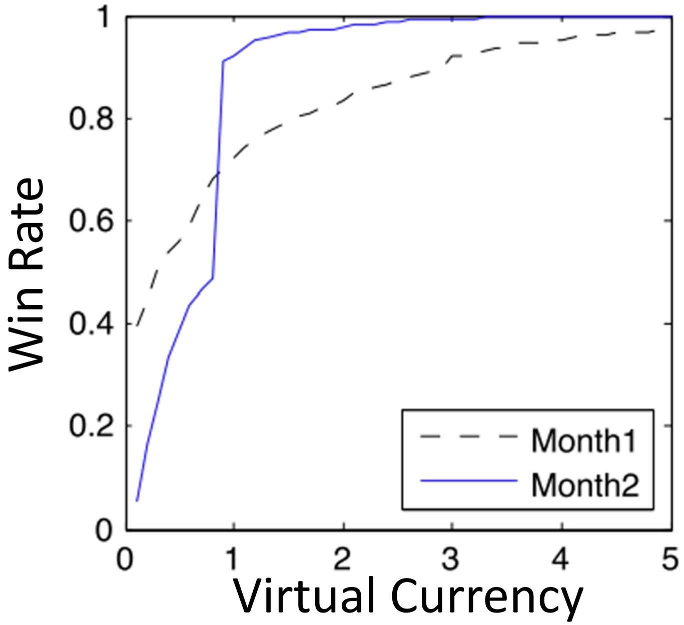
- Win rate: $w(b) = \int_{z=0}^{b} p(z)dz$
- Expected cost: $c(b) = \frac{\int_{z=0}^{b} zp(z)dz}{\int_{z=0}^{b} p(z)dz}$

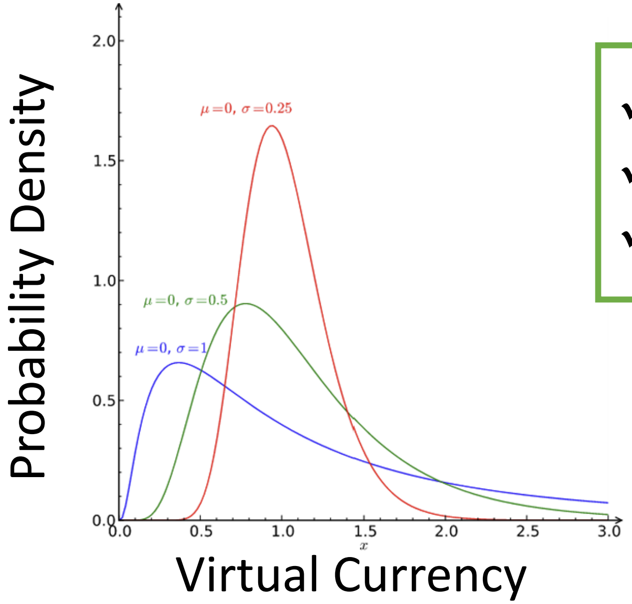
How can DSP collect so many bid prices for one bid request?





Bid Landscape





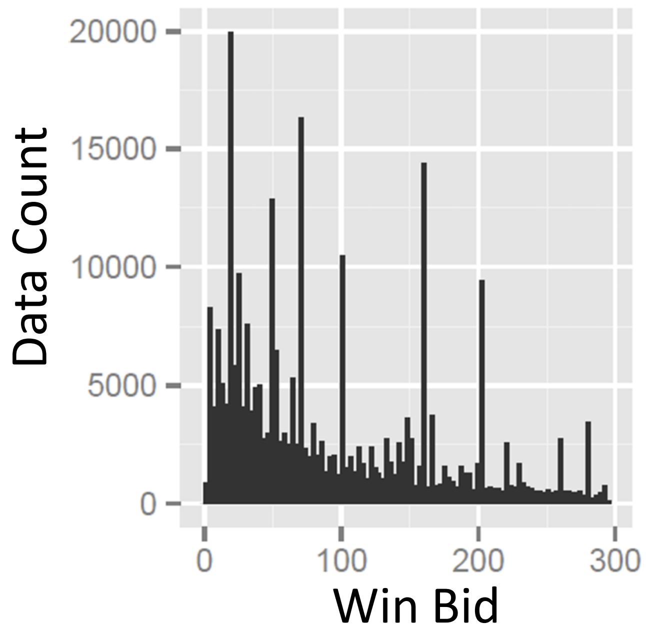
- ✓ Long-tail
- ✓ Non-negative bid price
- ✓ Normality

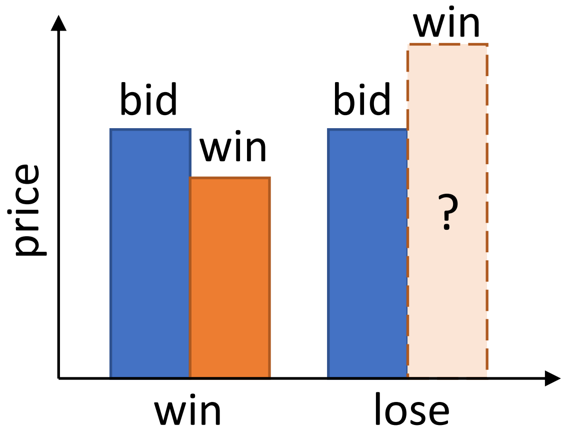
- Log-normal distribution (PDF)

$$f(b;\mu,\sigma) = \frac{1}{x\sigma\sqrt{2\pi}} e^{\frac{-(\ln b-\mu)^2}{2\sigma^2}}, b > 0$$

Underestimation of Bid Landscape

- Using win bid **only** is **over-optimistic**
  - Ignore bid prices you lost
- For DSP, impossible to know empirical win prices due to **censorship**!
- **Survival Analysis** in Medical/Financial domains
  - Study not-yet-happening events
  - Patient death in the future (death vs. time)
  - Win bid at a higher price (win vs. price)





Mixture Model with Censored Data

- For winning data, use **PDF $\boldsymbol{\phi}$** of standard normal distribution

$$\hat{z}_i = \beta x_i + \varepsilon; \;\; \max_{\beta_w} \sum_{i \in W} \underline{\log(\phi(\frac{z_i - \beta_w x_i}{\sigma}))}$$

Log-likelihood

- For censored (losing) data, use **CDF $\boldsymbol{\varphi}$** of standard normal distribution
  - φ gives *the probability your bid is not enough to win*

Why CDF but not PDF?

$$\underline{P(b_i < \hat{z}_i)} = \varphi(\frac{\beta_l x_i - b_i}{\sigma})$$

The probability that your bid b_i is less than the (unobserved) true winning price $\hat{z}_i$.

Mixture Model with Censored Data

- Overall objective

$$\max_{\beta_w,\beta_l} \sum_{i\in W} \underbrace{\log(\phi(\frac{z_i - \beta_w x_i}{\sigma}))}_{\text{maximize preciseness}} + \sum_{i\in L} \underbrace{\log(\varphi(\frac{\beta_l x_i - b_i}{\sigma}))}_{\text{maximize prob. of } b_i < \hat{z}_i}$$

$\Rightarrow$ raise $\hat{z}_i$

- Win price prediction

$$\hat{z}_i = \left[P(z_i < b_i)\beta_w + \left(1 - P(z_i < b_i)\right)\beta_l\right]x_i,$$

where $P(z_i < b_i) = \underbrace{\frac{1}{1+\exp(-\beta_{logit}x_i)}}_{}$

Additional logistic regression to classify if the price is winnable, independent to φ

Isn't this weird? Are separated β_w and β_l necessary?

Mixture Model with Censored Data

- Overall objective

$$\max_{\beta} \sum_{i \in W} \log(\phi(\frac{z_i - \beta x_i}{\sigma})) + \sum_{i \in L} \log(\varphi(\frac{\beta x_i - b_i}{\sigma}))$$

- Win price prediction

$$\widehat{z_i} = \beta x_i$$

Outline

- Introduction
- Bidding Strategy
  - Second-Price Auction
  - Bid Landscape
  - CTR Prediction
  - Budget Pacing
  - Cost Efficiency
- First-Price Auction
  - Bid Shading
- Takeaways

Optimal Bid Price

- Reward for a given bid: $R(b) = \int_0^b (\underline{v} - z)\underline{p(z)}dz$

True value of this request

Win rate at bid price = z

- Optimal bid price $b^* = \max_b R(b)$

$\Rightarrow \frac{\partial R}{\partial b} = (v - b)p(b) = 0$

$\Rightarrow b^* = v$

$\Rightarrow$ *The optimal bid price is exactly the value we believe*

Click-Based Value

- Strategy: bid $\begin{cases} value\ of\ click, & if\ clicked \\ 0, & otherwise \end{cases}$
- Average value of click: $(value\ of\ click) \times \frac{\#click}{\#impression}$
- Click as KPI: $\boldsymbol{CPC \times CTR}$

Click-Through-Rate (CTR):

1. #click/#impression
2. $P(click \mid x)$

Cost-Per-Click (CPC):

1. Optimal reward $b^* = v$
2. Manually-set maximum bid
3. Total spend/Total (expected) clicks

Accurate CTR predictions
=> Precise data-driven bidding strategy

CTR Prediction

- **Given**: n impressions consisting of feature vector x and click label $y \in \{0,1\}$
- **Goal**: train a CTR prediction model f such that $\hat{y} = f(x)$

Logistic regression

- **Challenges**:
  - High-dimensional and extremely sparse feature space
  - Feature interactions
  - Imbalance label (#clicked/#non-clicked≈0.001)

Field-Aware Features

- **Bid Request Feature**: high-dimensional sparse binary vectors
  - Publisher information
  - Audience information
  - Impression details
  - Content and context
  - Targeting audiences
  - Etc…

M fields

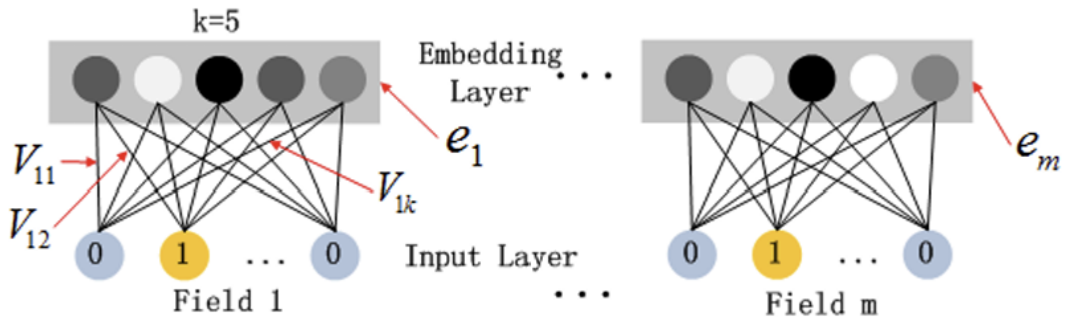
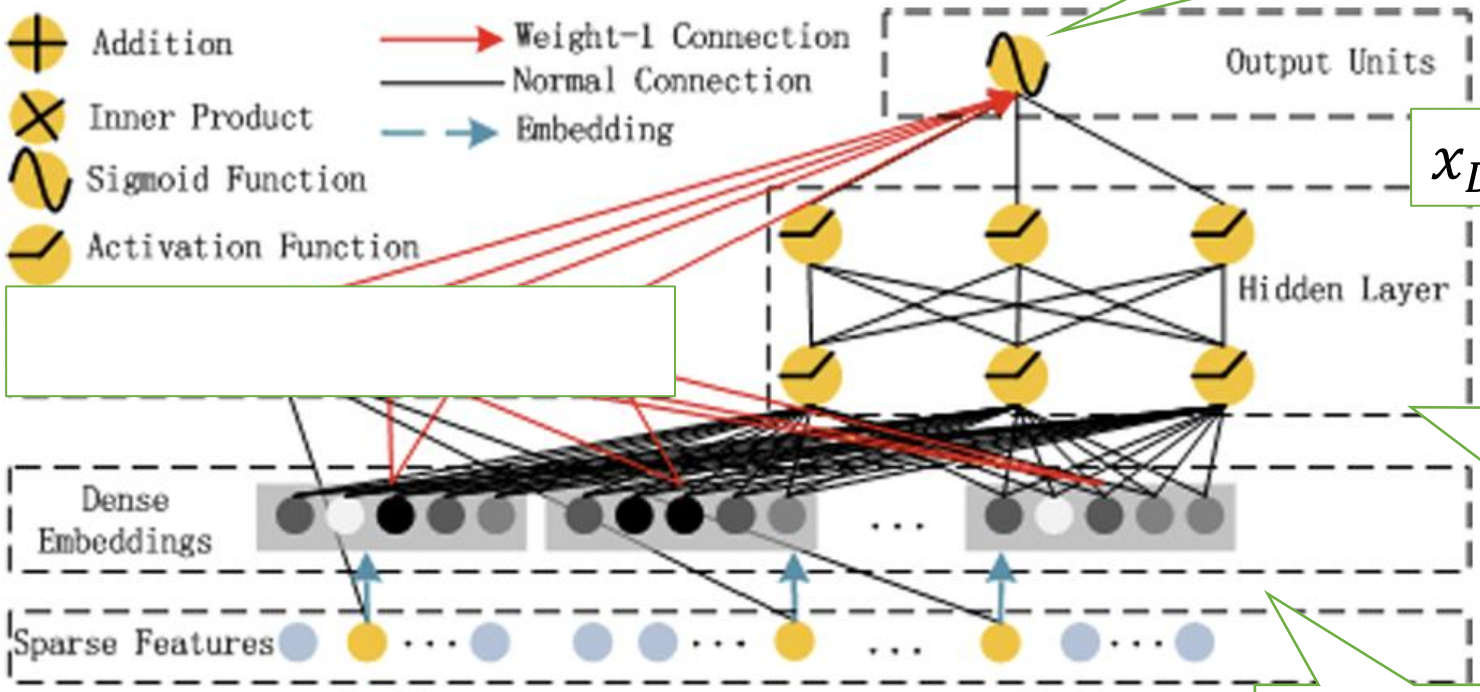


Figure 4: The structure of the embedding layer

- **Dimension reduction** for each field => **field-aware** features
  - PCA, LDA, etc
  - Embedding in deep models

Order-1 Prediction

$y = \sigma(x_{DNN})$



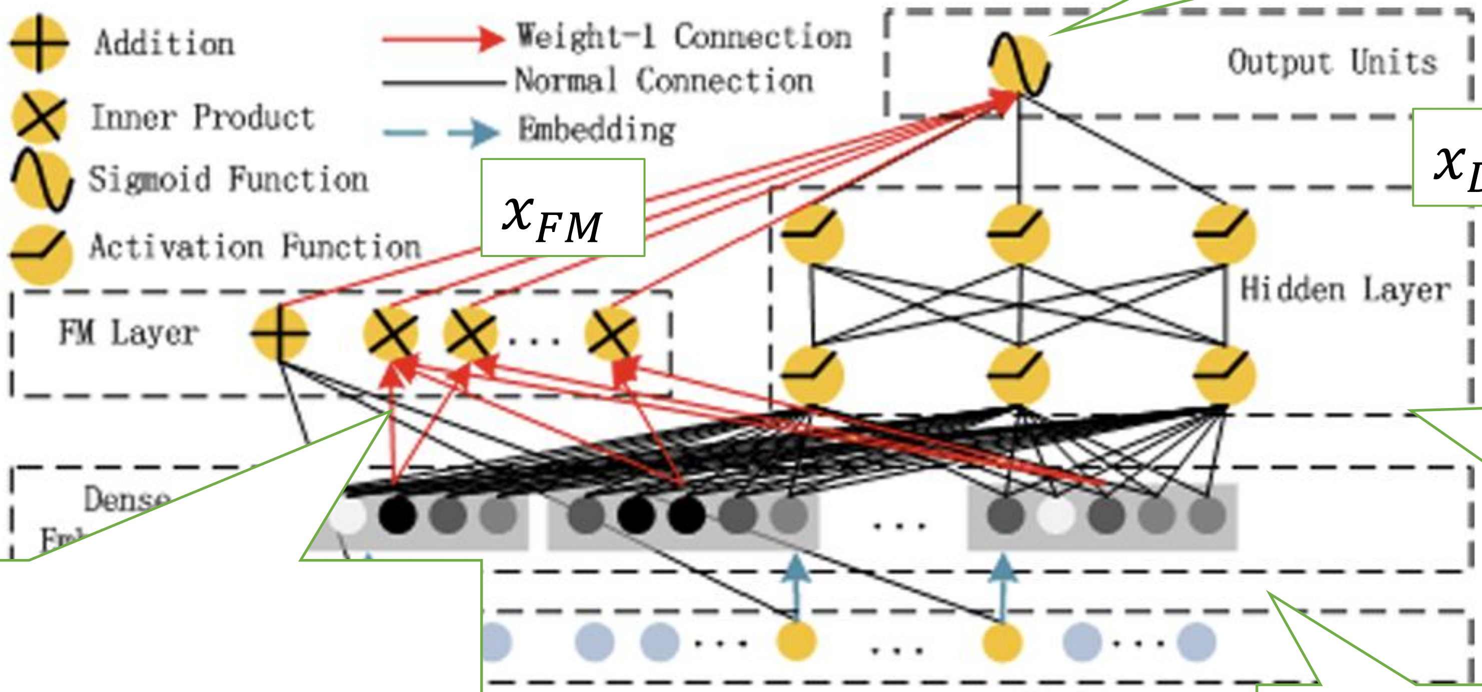
x_{DNN}

Order-1 feature activation

Dimension reduction

Order-2 Prediction

$$y = \sigma(x_{FM} + x_{DNN})$$



x_{DNN}

x_{FM}

Order-1 feature activation

Dimension reduction

Order-2 feature:

$x_{FM} = \langle w, x \rangle + \sum_{i=1}^{d} \sum_{i=j+1}^{d} \langle V_i, V_j \rangle x_i x_j$,

where V_i, V_j are latent vectors

Label Imbalance Issue

- Imbalance label (#clicked/#non-clicked≈0.001)
- Always predict non-click yields 99.9% accuracy

- Downsample #non-clicked
- #clicked augmentation (upsample)
- Upweight importance of clicked samples
- Etc…

Outline

- Introduction
- Bidding Strategy
    - Second-Price Auction
    - Bid Landscape
    - CTR Prediction
    - Budget Pacing
    - Cost Efficiency
- First-Price Auction
    - Bid Shading
- Takeaways

What We Have Now

- Win price prediction: $\widehat{z_i} = \beta x_i$
- Value prediction: $\widehat{v_i} = CTR \times CPC$

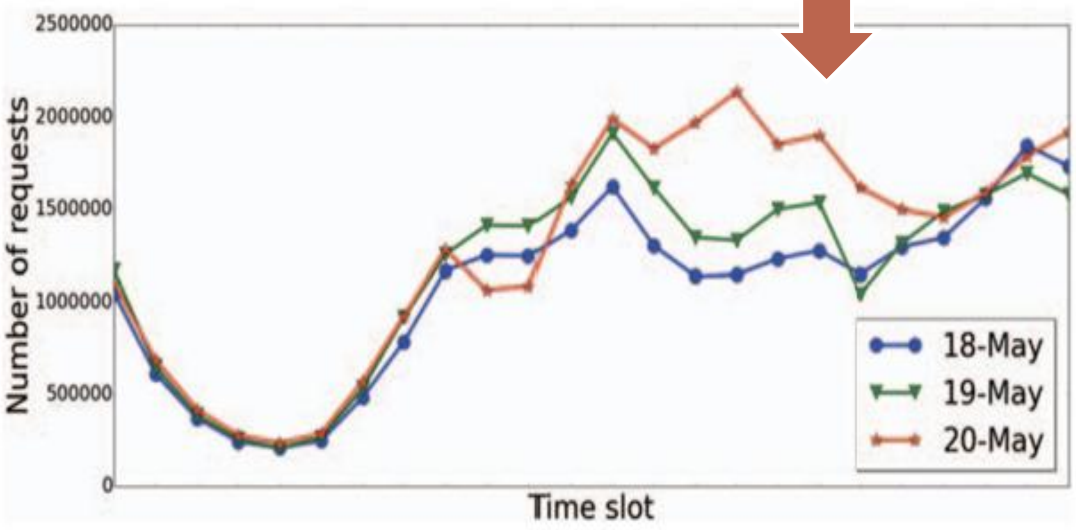
- **Naïve bidding strategy:**
  - If $\widehat{v_i} > \widehat{z_i}$: offer bid price $\mathrm{b_i} = \widehat{z_i} + 1$
  - Else: offer bid price $\mathrm{b}_i = \{\widehat{v_i}, 0\}$

This strategy prevents us from overpayment

Issues

- **Naïve bidding strategy:**
  - If $\widehat{v}_i > \widehat{z}_i$: offer bid price $\mathrm{b_i} = \widehat{z}_i + 1$
  - Else: offer bid price $\mathrm{b}_i = \{\widehat{v}_i, 0\}$

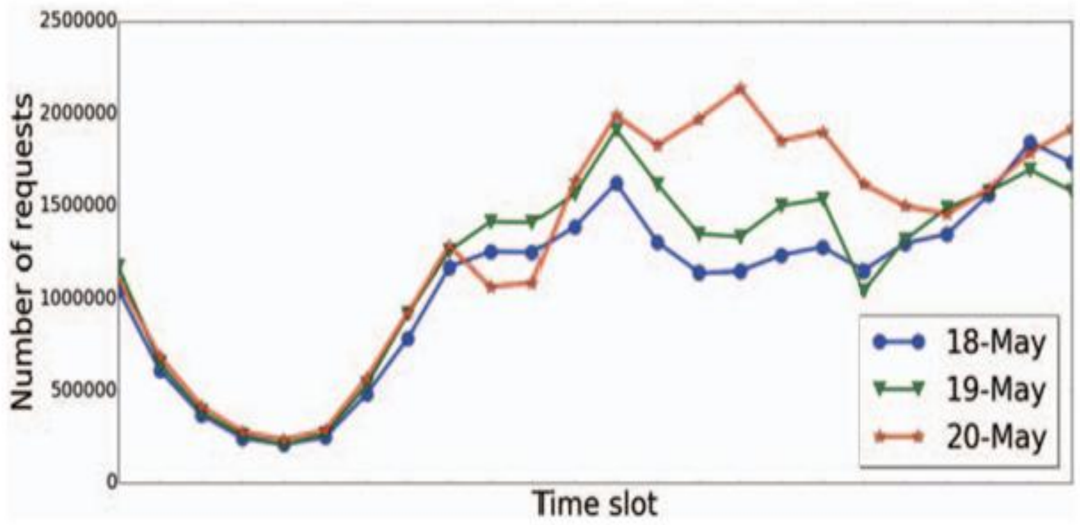
Market fluctuation



- Suppose the lifetime is one week long
  - Naïve method may spend all the budget on the first day
  - No ad delivery in the remaining six days
- Sensitive to abnormal traffic patterns (e.g., market fluctuations)

- Advertisers prefer **smooth delivery over lifetime**
- Need to **foresee future revenue**

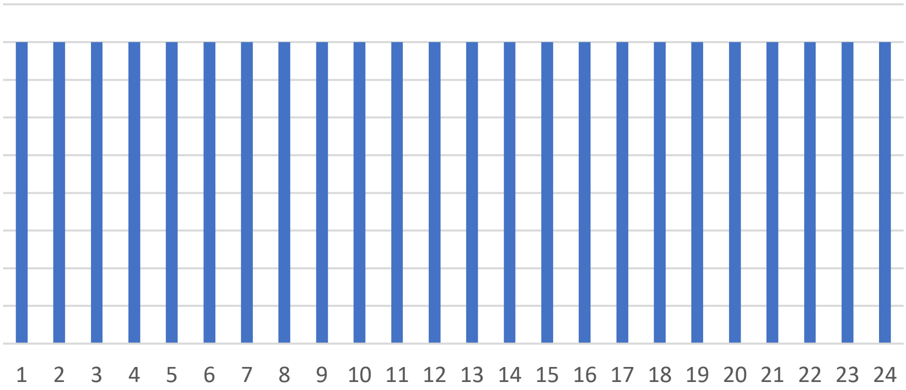
Budget Over Time



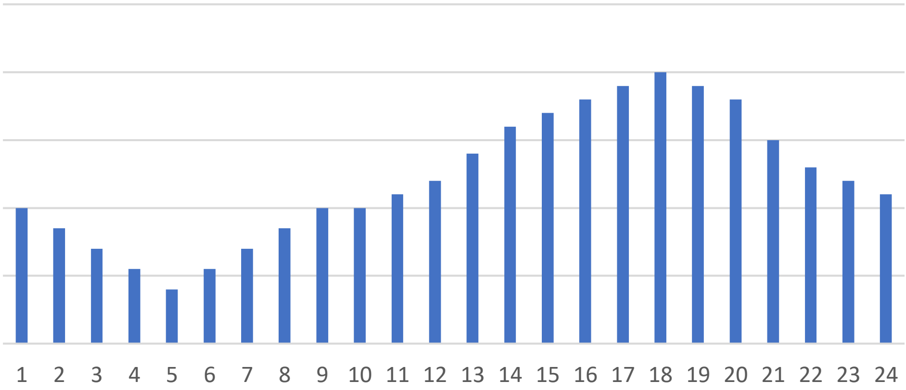
For $B_t \in \{B_1, B_2, \dots, B_T\}$
Run naïve bidding strategy

※ traffic prediction steps in

Might not spend all the budget
⇒ Revenue loss



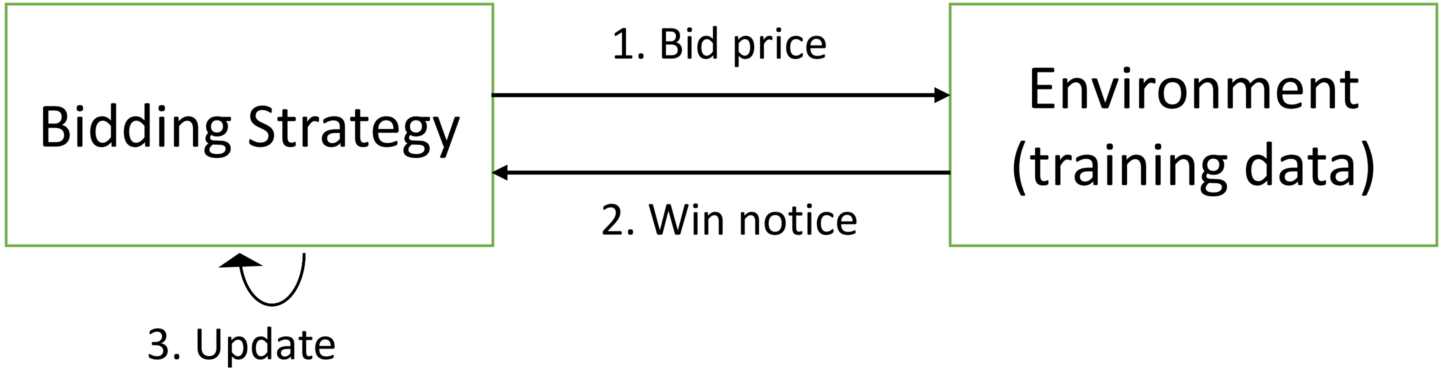
Uniform Distribution



Traffic-dependent Distribution

Deep Q-Network (DQN)

- Reinforcement learning (RL)
  - An intelligent brute-force approach
  - Learn complex and dynamic relations through experience
  - DQN is often used in RTB due to efficiency issue



Elements (MDP Process)

- **State $\boldsymbol{S}$**: {feature, pCTR, pWinPrice, remaining budget, time left, ….}
- **Action $\boldsymbol{A}$**: bid price or bid price adjustment (e.g., $ratio * constant\ price$)
- **Reward $\boldsymbol{R}$**: utilities earned per bid request won
  - Click, impression, revenue, …
- **Transition $\boldsymbol{T = P(s'|s, a)}$**: given current state s, select action a, transit to state s', and gain reward r

What Is DQN Optimizing

- DQN minimizes the difference between the predicted Q-value (the expected return for a state-action pair) and the target Q-value
- $L(\underbrace{\theta_i}_{\text{The i-th round DQN}}) = E_{s,a,r,s'\sim\rho}[(\underbrace{y_i}_{\text{Target}} - \underbrace{Q(s,a;\theta_i)}_{\text{Predicted Q-value by current DQN}})^2]$
- $y_i = \underbrace{r}_{\text{Immediate reward from transiting from s to s' with action a}} + \gamma \cdot \underbrace{\max_{a'} Q(s',a';\theta_{i-1})}_{\text{The maximum Q-value of the next state-action pair, estimated by last-iteration DQN} \Rightarrow \text{Future benefit!}}$

γ is the discount factor of future benefit, commonly use {0.9, 0.95, 0.99}

DQN Output

- Given the current state s

- Predict Q-value on all action candidates $a \in A$

- Output the action with highest Q-value
- $a^* = arg \max_{a \in A} Q(s, a)$

s= {…,remaining budget, time left, ….}

DQN: if I don't raise bid price, reward will be low => bid price raised

Outline

- Introduction
- Bidding Strategy
  - Second-Price Auction
  - Bid Landscape
  - CTR Prediction
  - Budget Pacing
  - Cost Efficiency
- First-Price Auction
  - Bid Shading
- Takeaways

Cost Efficiency for Clicks

- Cost-performance ratio
- Efficiency: CTR-WP ratio

	High predicted WP	Low predicted WP
High predicted CTR	Marginal case, considered dependently	Good investment, considered first
Low predicted CTR	Bad investment, considered last	Marginal case, considered dependently

*WP=win price

Observation

Threshold algorithm in online stochastic knapsack problem

- If we know win price for each bid request
- Maybe we can purchase cheap requests (w_1) first, and move on to the next level (w_2), so on and so forth

Win price (ascending order)	w_1	w_2	…	w_k
Amount	n_1	n_2	…	n_k
Total cost	$w_1 \cdot n_1$	$w_2 \cdot n_2$	…	$w_k \cdot n_k$

Find **maximal** win price upper bound bnd from training data such that $\sum_{i=1}^{bnd} w_i \cdot n_i \leq Budget$

Prioritize
{low WP+low CTR}U{low WP+high CTR}

Ideal Scenario

- **Oracle** click and win price predictors: $I(x)$ and $W(x)$
- PDF of win prices with click: Ω
  - Average and standard deviation in training data
- Number of impressions with click: N_{train}^{clk} and N_{test}^{clk}
  - $N_{test}^{clk} = N_{train}^{clk} \cdot \frac{T_{test}}{T_{train}}$

- Allocate budget with: $\int_{w=0}^{bnd} w \cdot N_{test}^{clk} \cdot \Omega(w)dw = Budget$
- Find **maximal** win price upper bound bnd from training data

Strategy for Ideal Scenario

IdealBidPrice(x, B_{cur}, bnd)

1. If $I(x) = 1$ and $W(x) \le \min(B_{cur}, bnd)$
2. return $W(x) + \delta$
3. Else return 0

- Offer bid price $W(x) + \delta$ only if it will be clicked and affordable
- δ is a small lift

IdealEfficiencyStrategy(B, bnd)

1. $B_{cur} = B$
2. For each request:
3. b = IdealBidPrice(x, B_{cur}, bnd)
4. if $b > 0$:
5. Offer b to ADX
6. $B_{cur} = B_{cur} - b$
7. Else break

Practical Scenario

- Predicted CTR: $pCTR$
- Predicted win price: pWP
- Bid efficiency: $\rho(x) = pCTR(x)/pWP(x)$

IdealBidPrice(x, B_{cur}, bnd)

1. If $I(x) = 1$ and $W(x) \leq \min(B_{cur}, bnd)$
2. return $W(x) + \delta$
3. Else return 0

PracticalBidPrice ($x, B_{cur}, \rho_{\text{cut}}$)

1. If $\rho(x) > \rho_{\text{cut}}$ and $pWP(x) \leq B_{cur}$
2. return $pWP(x) + \delta$
3. Else return 0

ρ_{cut} is a predefined efficiency cut-off function

Practical Scenario

IdealEfficiencyStrategy(B, bnd)

1. $B_{cur} = B$
2. For each request:
3. b = IdealBidPrice(x, B_{cur}, bnd)
4. if $b > 0$:
5. Offer b to ADX
6. $B_{cur} = B_{cur} - b$
7. Else break

PracticalEfficiencyStrategy(B, ρ_{cut})

1. $B_{cur} = B$
2. For each request:
3. b = PracticalBidPrice($x, B_{cur}, \rho_{\text{cut}}$)
4. if $b > 0$:
5. Offer b to ADX
6. if win: $B_{cur} = B_{cur} -$ real win price
7. Else break

How to get ρ_{cut} and δ?

Hyperparameters

- Win price lift δ: ideal=1
  - In practice: minimum δ s.t. $pWP(x) + \delta > P(x)$ for 95% of $\{x|I(x) = 1\}$

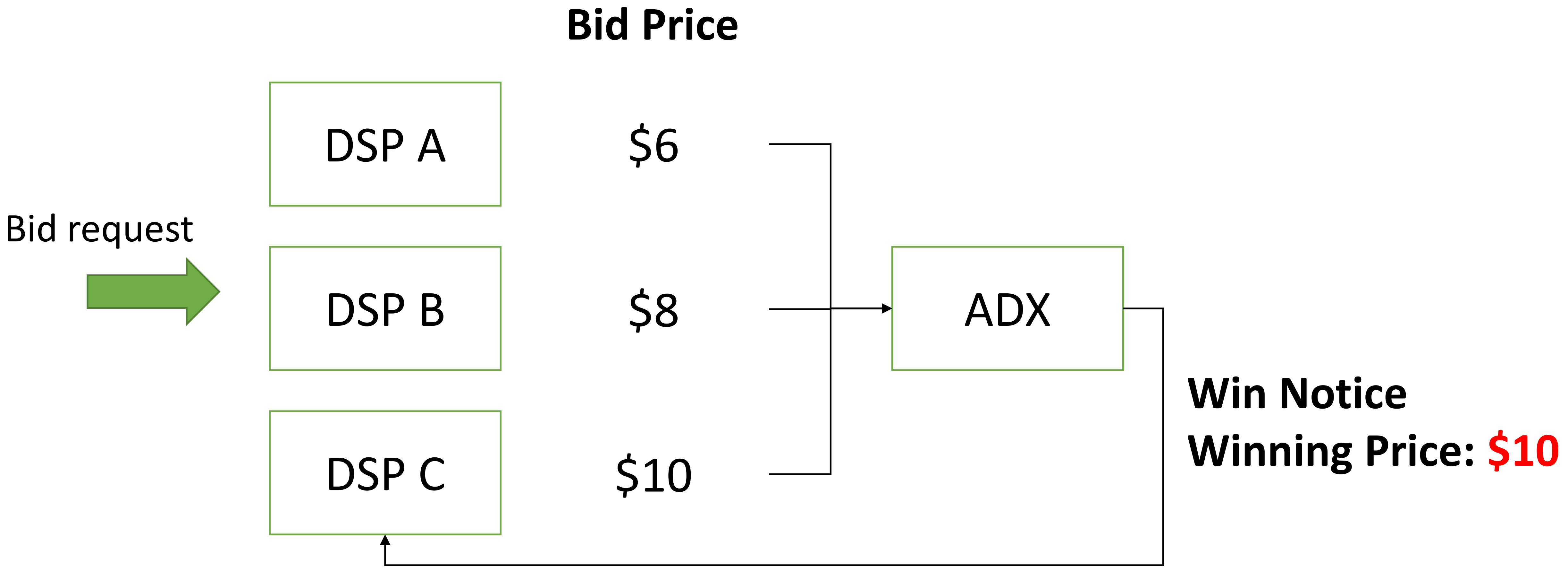
- After getting δ, search optimal ρ_{cut} in training data
  - Golden section search
  - Brute-force search
  - etc

Outline

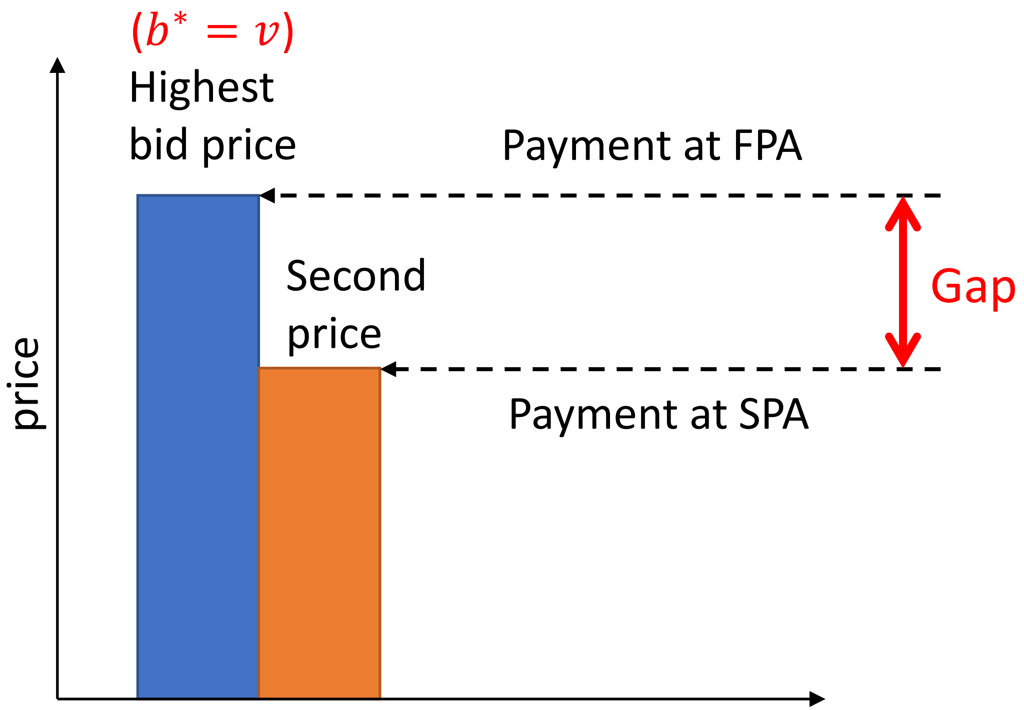
- Introduction
- Bidding Strategy
  - Second-Price Auction
  - Bid Landscape
  - CTR Prediction
  - Budget Pacing
  - Cost Efficiency
- First-Price Auction
  - Bid Shading
- Takeaways

First-Price Auction (FPA)

Google changed their policy since 2017



FPA vs. SPA



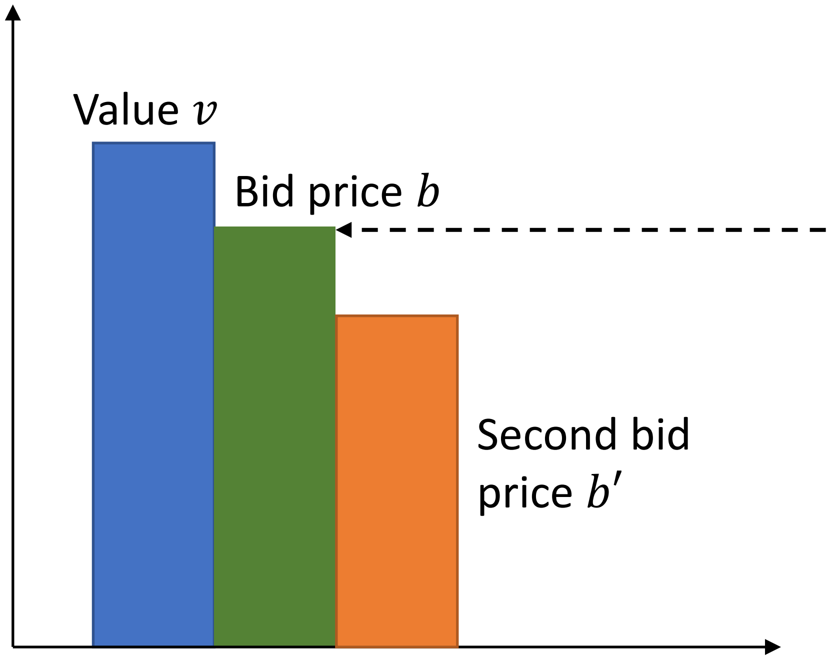
For SSP (publisher):

- Straightforward and transparent
- Better revenue estimation
- Potentially better revenue
- Less manipulation

For DSP (advertiser):

- Straightforward and transparent
- Overpayment
- Less revenue
- More competitive

Bid Shading



Bid shading: $\mathrm{b}' < \underbrace{\boldsymbol{b} = \boldsymbol{s}}_{\propto \text{ win rate}}\underbrace{(\boldsymbol{v}) < v}_{\text{surplus } v-b \ \propto \text{ expected return}}$

Intuitive solution: $b = \alpha \cdot v + (1-\alpha) \cdot \widehat{b'}$, s.t. $\alpha \in (0,1)$

Surplus Maximization

- $s(b; v, b') = (v - b)\boldsymbol{I}(b' < b) = \begin{cases} v - b\ , & if\ win \\ 0\ , & if\ lose \end{cases}$

- $b^* = \arg_{b \in (0,v)} \max E[s(b; v, b')]$
  $= \arg_{b \in (0,v)} \max (v - b)\underline{\Pr(b' < b|x)}$

Win price prediction!

How to get b'?

How to Get b'

- If we know the win price of every bid (non-censored):
  - $\min L = \sum_{\forall req} \log f(b'; x)$
  - f is the likelihood of win price prediction (PDF)

- If censored:
  - Recall survival analysis
  - $\min L = \sum_{i \in Win} \log f(b'; x) + \sum_{i \in Lose} \log F(b'; x)$
  - F is the CDF

- Now search $\mathrm{b} \in (0, v)$ to get b^*

Takeaways

- RTB heavily relies on CTR and win price predictions for intelligent bidding
- Trade-off between model complexity and accuracy
- Bid shading is critical in FPA to maximize surplus and avoid unnecessary costs

References and Further Reading

- A Collection of RTB Papers: [wnzhang/rtb-papers: A collection of research and survey papers of real-time bidding (RTB) based display advertising techniques.]