

Advertising Engines

A Guide to Web Research: Lecture 1

Yury Lifshits

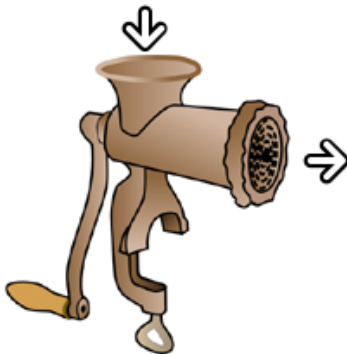
Steklov Institute of Mathematics at St.Petersburg

Stuttgart, Spring 2007

Microsoft® adCenter

Google™
AdWords

Google™
AdSense



**new algorithmic
problems
new models
and notions**

Talk Objective

Industrial solutions

- Google AdWords
- Google AdSense
- Yahoo! SearchMarketing
- Microsoft adCenter
- Amazon recommendations
- Coming soon: personalized ads for webmail, social networks, blogging platforms, phones, computer games, supermarket bills etc.

Talk Objective

Industrial solutions

- Google AdWords
- Google AdSense
- Yahoo! SearchMarketing
- Microsoft adCenter
- Amazon recommendations
- Coming soon: personalized ads for webmail, social networks, blogging platforms, phones, computer games, supermarket bills etc.

Today we show

- (1) single model for distributing personalized ads
- (2) open algorithmic problems motivated by such systems

Outline

- 1 Architecture of Advertising Engines
 - Component 1: Event
 - Component 2: Advertiser
 - Component 3: Advertising Engine

Outline

- 1 Architecture of Advertising Engines
 - Component 1: Event
 - Component 2: Advertiser
 - Component 3: Advertising Engine
- 2 Algorithmic Challenges
 - Target optimization
 - Click Volume
 - AdRank Computing
 - Ad Coverings

Part I: Architecture of Advertising Engines

Example: Sponsored Search

Stuttgart house

SEARCH
ENGINE

Advertisement

Advertisement

1

Algorithmic

Advertisement

2

results

Advertisement

3

Example: Context Ads

STUTTGART ESTATE AGENCY

Advertisement

Main

Advertisement

content

Advertisement

Three Components: Event, Advertiser, Engine

Event



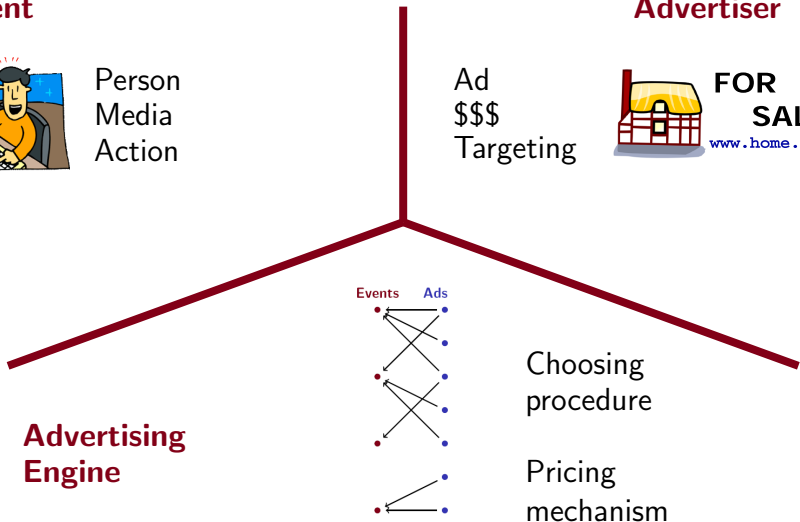
Person
Media
Action

Advertiser

Ad
\$\$\$
Targeting



**FOR
SALE**
www.home.org



Component 1: Event



Collect all available information:

- Person: What do we know about him/her?
 - Age, geographic location, previous actions, interests etc
- Media: What is situated around the ad placement?
 - Content and typical audience of website, tv program, newspaper
- Action: Current relations between person and media?
 - Current search query, purchasing a book, signing up to a service

Component 2: Advertiser



Setting new campaign:

- Ad: What will be displayed?
 - Text, image, video, hyperlink, phone number, advertiser's website
- \$\$\$: Size of campaign?
 - Monthly/daily budget, maximal admissible price (bid) for click/impression
- Targeting: Who is target audience?
 - Location, specific query keywords, category of landing page

Targeting in general: any subset of event space $P \times M \times A$

Component 3: Advertising Engine

Basic routine of advertising engine:

- 1 Get all available info about current event

Component 3: Advertising Engine

Basic routine of advertising engine:

- 1 Get all available info about current event
- 2 Keep only ads that include this event to their target

Component 3: Advertising Engine

Basic routine of advertising engine:

- 1 Get all available info about current event
- 2 Keep only ads that include this event to their target
- 3 Rank ads according to their bids and their relevance to the event

Component 3: Advertising Engine

Basic routine of advertising engine:

- 1 Get all available info about current event
- 2 Keep only ads that include this event to their target
- 3 Rank ads according to their bids and their relevance to the event
- 4 Display (several) best ones

Component 3: Advertising Engine

Basic routine of advertising engine:

- 1 Get all available info about current event
- 2 Keep only ads that include this event to their target
- 3 Rank ads according to their bids and their relevance to the event
- 4 Display (several) best ones
- 5 In case of click compute discount (actual price for advertiser)

Component 3: Advertising Engine

Basic routine of advertising engine:

- 1 Get all available info about current event
- 2 Keep only ads that include this event to their target
- 3 Rank ads according to their bids and their relevance to the event
- 4 Display (several) best ones
- 5 In case of click compute discount (actual price for advertiser)

Component 3: Advertising Engine

Events

Ads

Basic routine of advertising engine:

- 1 Get all available info about current event
- 2 Keep only ads that include this event to their target
- 3 Rank ads according to their bids and their relevance to the event
- 4 Display (several) best ones
- 5 In case of click compute discount (actual price for advertiser)

Component 3: Advertising Engine

Events



Ads



Basic routine of advertising engine:

- 1 Get all available info about current event
- 2 Keep only ads that include this event to their target
- 3 Rank ads according to their bids and their relevance to the event
- 4 Display (several) best ones
- 5 In case of click compute discount (actual price for advertiser)



Component 3: Advertising Engine

Events Ads



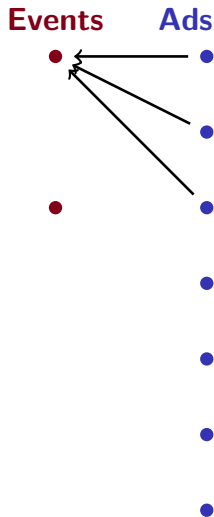
Basic routine of advertising engine:

- 1 Get all available info about current event
- 2 Keep only ads that include this event to their target
- 3 Rank ads according to their bids and their relevance to the event
- 4 Display (several) best ones
- 5 In case of click compute discount (actual price for advertiser)

Component 3: Advertising Engine

Basic routine of advertising engine:

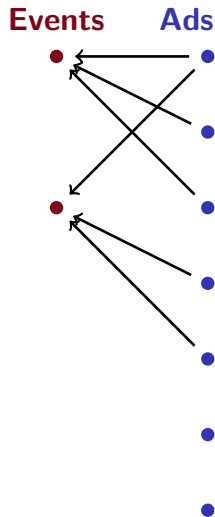
- 1 Get all available info about current event
- 2 Keep only ads that include this event to their target
- 3 Rank ads according to their bids and their relevance to the event
- 4 Display (several) best ones
- 5 In case of click compute discount (actual price for advertiser)



Component 3: Advertising Engine

Basic routine of advertising engine:

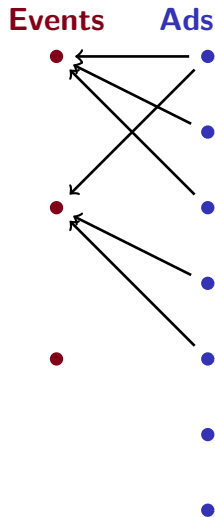
- 1 Get all available info about current event
- 2 Keep only ads that include this event to their target
- 3 Rank ads according to their bids and their relevance to the event
- 4 Display (several) best ones
- 5 In case of click compute discount (actual price for advertiser)



Component 3: Advertising Engine

Basic routine of advertising engine:

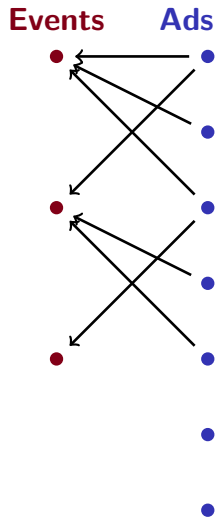
- 1 Get all available info about current event
- 2 Keep only ads that include this event to their target
- 3 Rank ads according to their bids and their relevance to the event
- 4 Display (several) best ones
- 5 In case of click compute discount (actual price for advertiser)



Component 3: Advertising Engine

Basic routine of advertising engine:

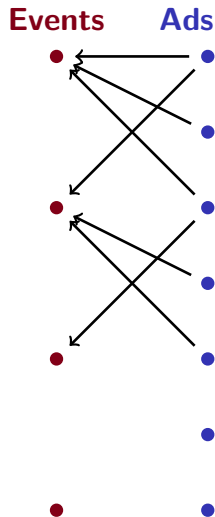
- 1 Get all available info about current event
- 2 Keep only ads that include this event to their target
- 3 Rank ads according to their bids and their relevance to the event
- 4 Display (several) best ones
- 5 In case of click compute discount (actual price for advertiser)



Component 3: Advertising Engine

Basic routine of advertising engine:

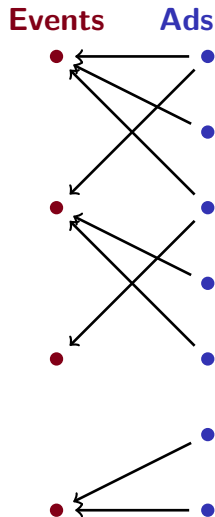
- 1 Get all available info about current event
- 2 Keep only ads that include this event to their target
- 3 Rank ads according to their bids and their relevance to the event
- 4 Display (several) best ones
- 5 In case of click compute discount (actual price for advertiser)



Component 3: Advertising Engine

Basic routine of advertising engine:

- 1 Get all available info about current event
- 2 Keep only ads that include this event to their target
- 3 Rank ads according to their bids and their relevance to the event
- 4 Display (several) best ones
- 5 In case of click compute discount (actual price for advertiser)



Objectives

- User
 - Maintain privacy
 - Receive only relevant ads

Objectives

- User
 - Maintain privacy
 - Receive only relevant ads
- Advertiser
 - Cheap clicks
 - Get “relevant” clicks (high conversion rate)
 - Transparent pricing and targeting mechanisms

Objectives

- User
 - Maintain privacy
 - Receive only relevant ads
- Advertiser
 - Cheap clicks
 - Get “relevant” clicks (high conversion rate)
 - Transparent pricing and targeting mechanisms
- Advertising Engine
 - Organize enough relevant clicks for any budget and any target
 - Keep prices high
 - Keep users/advertisers happy

Objectives

- User
 - Maintain privacy
 - Receive only relevant ads
- Advertiser
 - Cheap clicks
 - Get “relevant” clicks (high conversion rate)
 - Transparent pricing and targeting mechanisms
- Advertising Engine
 - Organize enough relevant clicks for any budget and any target
 - Keep prices high
 - Keep users/advertisers happy

More objectives?

Part II: Algorithmic Challenges

Part II: Algorithmic Challenges

Disclaimer: my style is

- 1 At first, think independently (e.g. pose new problems)
- 2 **Only after that** look into literature

Hence, the following problems might be already known and heavily studied!

Advertiser sets target audience. **Advertising engine should help:**

- Some potentially interested people are missed
- Exclude people who will be offended by this ad
- Proper setting of target audience is difficult
- Advertising engine knows much more about event space

Events are vectors

Advertiser provides some **sample** events $S = \{e_1, \dots, e_k\}$ from the target

Advertising engine produces an effective membership procedure for **optimized target** \bar{S}

Events are vectors

Advertiser provides some **sample** events $S = \{e_1, \dots, e_k\}$ from the target

Advertising engine produces an effective membership procedure for **optimized target** \bar{S}

How to define **optimized target**?

Let $B(e, r)$ be the ball in event space with center e and radius r

$$\text{New target} = \bigcup_k B(e_k, r)$$

Let $B(e, r)$ be the ball in event space with center e and radius r

$$\text{New target} = \bigcup_k B(e_k, r)$$

Effective membership for e_{new} :

- 1 Find nearest representative $e_i \in S$
- 2 Check whether $Dist(e_{new}, e_i) < r$

Let $B(e, r)$ be the ball in event space with center e and radius r

$$\text{New target} = \bigcup_k B(e_k, r)$$

Effective membership for e_{new} :

- 1 Find nearest representative $e_i \in S$
- 2 Check whether $\text{Dist}(e_{new}, e_i) < r$



Let $B(e, r)$ be the ball in event space with center e and radius r

$$\text{New target} = \bigcup_k B(e_k, r)$$

Effective membership for e_{new} :

- 1 Find nearest representative $e_i \in S$
- 2 Check whether $\text{Dist}(e_{new}, e_i) < r$

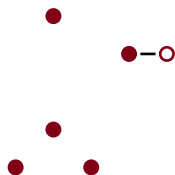


Let $B(e, r)$ be the ball in event space with center e and radius r

$$\text{New target} = \bigcup_k B(e_k, r)$$

Effective membership for e_{new} :

- 1 Find nearest representative $e_i \in S$
- 2 Check whether $\text{Dist}(e_{\text{new}}, e_i) < r$

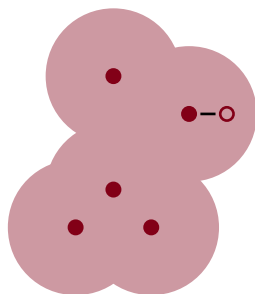


Let $B(e, r)$ be the ball in event space with center e and radius r

$$\text{New target} = \bigcup_k B(e_k, r)$$

Effective membership for e_{new} :

- 1 Find nearest representative $e_i \in S$
- 2 Check whether $\text{Dist}(e_{new}, e_i) < r$



- Other definitions for optimized target?
- Exploiting historical information for target optimization
- Target construction based on advertisement content

Assume we show the same ad at all events. Then average daily amount of clicks is **click volume** for the given ad

Assume we show the same ad at all events. Then average daily amount of clicks is **click volume** for the given ad

Motivation

- Understand how much can we sell
- Evaluate the effectiveness of current engines
- The first step towards recognizing interested audience
- Use different strategies for (supply < demand) and (supply > demand)

Assume we show the same ad at all events. Then average daily amount of clicks is **click volume** for the given ad

Motivation

- Understand how much can we sell
- Evaluate the effectiveness of current engines
- The first step towards recognizing interested audience
- Use different strategies for (supply < demand) and (supply > demand)



Assume we show the same ad at all events. Then average daily amount of clicks is **click volume** for the given ad

Motivation

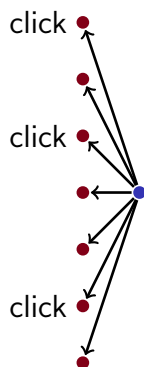
- Understand how much can we sell
- Evaluate the effectiveness of current engines
- The first step towards recognizing interested audience
- Use different strategies for (supply < demand) and (supply > demand)



Assume we show the same ad at all events. Then average daily amount of clicks is **click volume** for the given ad

Motivation

- Understand how much can we sell
- Evaluate the effectiveness of current engines
- The first step towards recognizing interested audience
- Use different strategies for (supply < demand) and (supply > demand)



Take daily history **event-ad-click**:

$$(e_1, a_1, b_1) \dots (e_n, a_n, b_n)$$

Use similarity-between-ads function S for computing click volume V :

$$V(a_{new}) = \sum S(a_{new}, a_i) \cdot b_i$$

Take daily history **event-ad-click**:

$$(e_1, a_1, b_1) \dots (e_n, a_n, b_n)$$

Use similarity-between-ads function S for computing click volume V :

$$V(a_{new}) = \sum S(a_{new}, a_i) \cdot b_i$$

Any comments/objections?

Problem: click volume is underestimated since not all chosen ads are similar to a_{new}

Problem: click volume is underestimated since not all chosen ads are similar to a_{new}

First step: prediction of click-through rate for a given **event-ad** pair

$$CTR(e, a_{new}) = \frac{\sum_{e_i=e} S(a_i, a_{new}) b_i}{\sum_{e_i=e} S(a_i, a_{new})}$$

Problem: click volume is underestimated since not all chosen ads are similar to a_{new}

First step: prediction of click-through rate for a given **event-ad** pair

$$CTR(e, a_{new}) = \frac{\sum_{e_i=e} S(a_i, a_{new}) b_i}{\sum_{e_i=e} S(a_i, a_{new})}$$

Second step: using click rates

$$V(a_{new}) = \sum_{1 \leq i \leq n} CTR(e_i, a_{new})$$

- Computing ad volume (the amount of advertisements that can get positive response at the given event)
- Fast algorithm for predicting click volume for all ads in the system
- Exploiting metric inside event space

Input: event e_{new} , set of all ads A . **Choosing-ads principles:**

- Take the most content-relevant
- Take the ones with best click-through rate
- Take ads with maximal bids

$$AdRank(e_{new}, a) = Bid(a) \cdot (ContRel(e_{new}, a) + CTR(e_{new}, a))$$

Actually, finding content-closest ads to the given input is just the nearest neighbor problem. **We need:**

- Data structure for A for fast computing of best $AdRank(e_{new}, a)$ values
- Accurate and fast prediction for $CTR(e_{new}, a)$

Actually, finding content-closest ads to the given input is just the nearest neighbor problem. **We need:**

- Data structure for A for fast computing of best $AdRank(e_{new}, a)$ values
- Accurate and fast prediction for $CTR(e_{new}, a)$

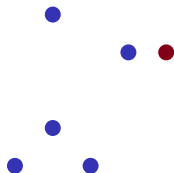
All ads



Actually, finding content-closest ads to the given input is just the nearest neighbor problem. **We need:**

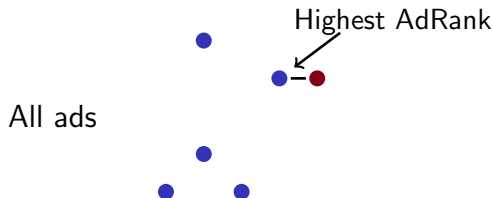
- Data structure for A for fast computing of best $AdRank(e_{new}, a)$ values
- Accurate and fast prediction for $CTR(e_{new}, a)$

All ads



Actually, finding content-closest ads to the given input is just the nearest neighbor problem. **We need:**

- Data structure for A for fast computing of best $AdRank(e_{new}, a)$ values
- Accurate and fast prediction for $CTR(e_{new}, a)$

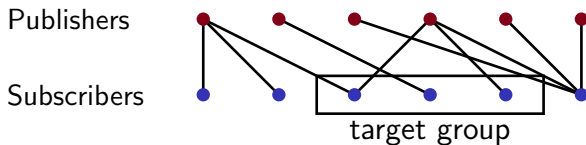


Consider any **publishers-subscribers** graph (say, RSS feeds):

- What is the minimal amount of placements to cover all (target) audience?
- Given fixed amount of placements how many subscribers can we cover twice?

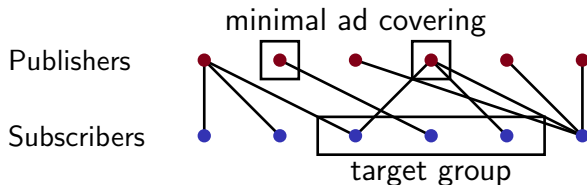
Consider any **publishers-subscribers** graph (say, RSS feeds):

- What is the minimal amount of placements to cover all (target) audience?
- Given fixed amount of placements how many subscribers can we cover twice?



Consider any **publishers-subscribers** graph (say, RSS feeds):

- What is the minimal amount of placements to cover all (target) audience?
- Given fixed amount of placements how many subscribers can we cover twice?



Other Directions in Advertising Engines

- Optimal ad distribution in case when interested audience is larger than budget
- Machine learning for advertising engines
- Weighted targeting (some events are preferable to others)
- Advertising engines for social networks
- Auction design for sponsored search
- Click fraud

Call for participation

Know a relevant reference?

Have an idea?

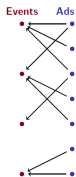
Find a mistake?

Solved one of these problems?

- Knock to my office 1.156
- Write to me yura@logic.pdmi.ras.ru
- Join our informal discussions
- Participate in writing roadmap-paper

Summary

Three components:

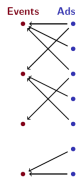


**FOR
SALE**

www.home.org

Summary

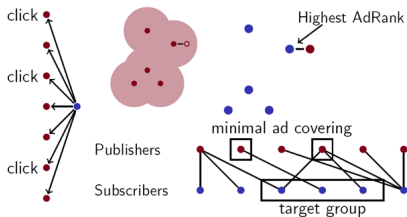
Three components:



**FOR
SALE**

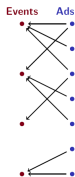
www.home.org

Four problems:



Summary

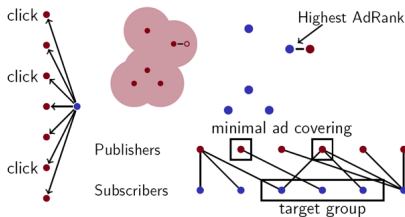
Three components:



**FOR
SALE**

www.home.org

Four problems:



Vielen Dank für Ihre Aufmerksamkeit! Fragen?

Sources

Course homepage <http://logic.pdmi.ras.ru/~yura/webguide.html>



Daniel C. Fain and Jan O. Pedersen

Sponsored Search: a Brief History

http://www.bus.ualberta.ca/kasdemir/ssa2/fain_pedersen.PDF



Alexander Tuzhilin

The Lanes Gifts v. Google Report

http://googleblog.blogspot.com/pdf/Tuzhilin_Report.pdf



Moira Regelson and Daniel C. Fain

Predicting ClickThrough Rate Using Keyword Clusters

http://www.bus.ualberta.ca/kasdemir/ssa2/regelson_fain.pdf



Juan Feng, Hemant K. Bhargava and David M. Pennock

Implementing Sponsored Search in Web Search Engines: Computational Evaluation of Alternative Mechanisms

<http://research.yahoo.com/node/338/2371>



Panel Discussion at SSA2

Models for Sponsored Search: What are the right questions?

<http://research.microsoft.com/~hartline/papers/panel-SSA-06.pdf>