Large Scale Machine Learning in the Real World

LÉON BOTTOU

PART 1 – PAID SEARCH
1. Ad Placement
Advertisement primer

Customer “funnel”

Advertisement opportunities

Sale!

Paid search

Display ads

Online

Print

Direct mail

TV

Advertising Revenue Market Share by Media, 2005-2011 (In $B)
Paid search

- The most “effective” online ads are those displayed on search engines.
- How to choose which ads to display where?
In general, **Paid search** advertisers pay when the user clicks on their ad.

(there are other payment models, per impression, per action, etc.)
The game

- Reveals interests with a search query

Users → Publisher

Publisher → Ads & Bids

Publisher → Prices

Advertiser

2
The game

- Computes search results
- Determines which ads to display (and where!)
- Determines price per click
The game

- May click on a relevant ad and jump to the advertiser site
- Triggering a payment from the advertiser to the publisher.

**User**

4

**Queries**

**Publisher**

**Advertiser**

**Ads & Bids**

**Ads**

**Prices**

**Clicks**
## Self interest

| **User** | Expects results that satisfy her interests  
Possibly by initiating business with an advertiser  
Future engagement depends on her satisfaction... |
| --- | --- |
| **Advertiser** | Expects to receive potential customers  
Expects to recover clicks costs from resulting business  
Return on investment impacts future ads and bids... |
| **Publisher** | Expects click money  
Learns which ads work from past data.  
In order to preserve future gains, publishers must ensure the continued satisfaction of users and advertisers. (this changes everything!) |
Second order effects

Users

Publisher

Advertisers

Queries

Ads

& Bids

Prices

Clicks (and consequences)

USER FEEDBACK LOOP

ADVERTISER FEEDBACK LOOP
Auctions

Setup
• Seller has an object to sell.
• Each buyer values the object differently.
• Each buyer knows the auction mechanism and places a bid.
• Auction mechanism determines who gets the object and how much he pays.

Notes
- The auction outcomes are functions of the bids.
- Buyer bids according to his value and his beliefs about other buyers values.
- The value of whoever gets the object is the size of the pie.
- The payment from the buyer to the seller then splits the pie.

Which mechanism works best for the seller?
Auctions

**A first auction mechanism**

“The highest bidder receives the object and pays his bid.”

- Buyers should bid less than their value.
  - If they bid their value, their surplus is zero in all cases.
  - If they bid more, they may get the object with a negative surplus.
  - If they bid less, they trade a chance to lose the object for a chance to pay less.

- The object may not go to the buyer who values it most.
  - The expected pie is smaller and the expected buyer surplus is larger. This cannot be good for the seller.

- The object may sell for less than the seller’s value.
  - Can use a *reserve price*, that is, an additional bid entered by the seller.
Auctions

A second auction mechanism
“The highest bidder receives the object and pays the second highest bid.”

- Buyers now should bid their value ("truthful mechanism")
  - Overbidding buyers may get the object with negative surplus.
  - Underbidding buyers will not pay less if they get the object. On the other hand, they may see the object sold to another buyer for less than their value, losing the opportunity to have a positive surplus.

  *Unless a buyer is certain that no other buyer will bid above a level smaller than his value, the buyer best interest is to bid his value, regardless of his exact beliefs.*

- The object always goes to the buyer who values it most.
- The object may still sell for less than the seller’s value.
Auctions

**A third auction mechanism**

“The seller announces a reserve price which works like an additional bid.
- If the highest bid is the reserve price, the seller keeps the object.
- Otherwise the highest bidder receives the object and pays the second highest bid.”

- Buyers should still bid their value (“truthful mechanism”)

- But the seller should set a reserve price that is higher than his value!
  - He trades the risk of not selling for the chance to get more than his value.
  - Therefore the object may not sell even though a buyer values it more than the seller. This in fact makes the pie smaller in a manner that benefits the seller.

- Under mild assumptions, this is the optimal mechanism for the seller.
  - For the correct value of the reserve price, of course...
Ad placement auctions

**Mapping auction theory to ad placement**
- The publisher is the seller (he receives bids)
- The advertisers are the buyers (they place bids)
- What about click decisions made by the user?
- What is the “object” exactly?

**Click probabilities**
The click probabilities \((q_1, \ldots, q_k)\) of the eligible ads \((a_1, \ldots, a_k)\) depend
- on the context \(x\), that is, the query, the user, the session, the weather...
- on the ad messages themselves \((a_1, \ldots, a_k)\),
- on the positions \((p_1, \ldots, p_k)\) chosen by the publisher,
- but do not depend on the bids \((b_1, \ldots, b_k)\).
Ad placement auctions

One of the many ways to view ad placement auctions...

• The auction mechanism specifies
  o The probability that each competing advertiser gets a click (the object).
  o The expected price paid by each competing advertiser.

• There is an optimal mechanism (Myerson, 1981).
  o The placement \((p_1, \ldots, p_k)\) maximizes \(\sum_i b_i \times q_i(x, a, p)\) subject to reserves.
  o The prices are determined by the Vickrey-Clarke-Groves (VCG) rule, a nontrivial generalization of the second price rule.

• See also (Varian, 2007; Edelman et al., 2007)
Optimal auctions?

- Many queries are targeted by a single advertiser.
  - When there is only one buyer, this is not an auction!

- The optimal auction theory is valid for a single auction.
  - The optimal auction might leave the buyer quite unhappy
    This is not going to work if we deal again and again with the same buyer...

- Advertisers place a single bid for multiple auctions.
  - An ad can be eligible for a lot of different queries.
  - The Bing/Yahoo engine serves hundreds of millions of queries per day.
    The most active advertisers change their bids every 15 minutes.

- Placement decisions impact the future behavior of users.
  - Some advertisers try to cheat the users by directing them to spam sites.
    This is not good for the long term revenue of the publisher.
How it really works

The following mechanism is the result of history. This is what the advertisers expect. Changing it is hard!

1. Publisher selects eligible ads \((a_1, \ldots, a_k)\) for the query \(x\).
2. Publisher computes click scores \(q_i\) and rank scores \(r_i\)
   \[
   q_i(x, a_i, p_i) = \gamma(x, p_i) \times \beta(x, a_i) \quad r_i(x, a_i) = b_i \times \beta(x, a_i)
   \]
3. Publisher greedily assigns ads with the largest rank scores to the best available positions, until reaching a predefined reserve score
4. Generalized second price (GSP): advertiser pays the smallest bid that would have guaranteed the same placement.
The ugly truth

2. Publisher computes click scores \( q_i \) and rank scores \( r_i \)

\[
q_i(x, a_i, p_i) = \gamma(x, p_i) \times \beta(x, a_i) \\

r_i(x, a_i) = b_i \times \beta(x, a_i)
\]

No longer a pure click probability. Secret ingredients attempt to represent user satisfaction.


The auction is not truthful because GSP is not VCG. Furthermore, additional ingredients give discounts for certain auctions.

I do not understand the combined effects of all these adjustments. I have never met anyone who could explain them to me.
The plumbing

Search engine

Real time ad placement engine

Selection → Scores → Auction

Ads (≈10⁹)

Models (GB)

Params (100s)

Offline computing platform

Accounting → Training → Experiments

Logs (TB/day)
2. Experimentation
Decision making

How to make sound decisions about such a system?

- Should we use a different click score model?
- Should we show more or less ads above the search results?
- Should we select eligible ads more or less aggressively?

Theoretical framework is neither complete nor accurate.

We need to experiment!
A/B Testing

How to compare two ad placement engine variants?

1. Implement both variants

2. **Randomly split traffic** in two groups (also called “flights”)
   - Place treatment flights ads using the variant under investigation.
   - Place control flights ads using the normal placement engine.

3. Run for **some time** and measure **performance metrics**.
Performance metrics

First order performance metrics

- Average number of ads shown per page
- Average number of mainline ads per page
- Average number of ad clicks per page
- Average revenue per page (RPM)

Should we just optimize RPM?

Showing lots of mainline ads improves RPM.
Users would quickly go away!

Increasing the reserve prices also improves RPM.
Advertisers would quickly go away!
Performance metrics

First order performance metrics
- Average number of ads shown per page
- Average number of mainline ads per page
- Average number of ad clicks per page
- Average revenue per page (RPM)
- Average relevance score estimated by human labelers
- Average number of bid-weighted ad clicks per page
- ...

Monitor heuristic indicators of user fatigue

Monitor heuristic indicators of advertiser value
Splitting traffic

**Long term user feedback experiments**

Measure actual user fatigue instead of heuristic indicators.

- Randomly split users into treatment and control groups.
- Wait a couple months and compare performance metric.
- This comparison reveals second order user effects...

**Long term advertiser feedback experiments**

- Randomly split advertisers into treatment and control groups
- Which version of the ad placement engine should we run when an auction involves advertisers from both groups?
Significance

Central Limit Theorem

\[ \hat{Y} = \frac{1}{n} \sum y_i \]

\[ (Y - \hat{Y}) \sim \mathcal{N} \left( 0, \frac{\sigma}{\sqrt{n}} \right) \]
Variance reduction

Hourly average click yield for treatment and control

\[
Y - \frac{1}{n} \sum y_i \sim \mathcal{N}(0, \frac{\sigma}{\sqrt{n}})
\]

Daily effects increases the variance of both treatment and control.

Daily effects affect treatment and control in similar ways! Can we subtract them?
Variance reduction

• Treatment estimate
  \[ Y^* \approx \hat{Y}^* = \frac{1}{|T|} \sum_{i \in T} y_i \]

• Control estimate
  \[ Y \approx \hat{Y} = \frac{1}{|C|} \sum_{i \in C} y_i \]

• Predictor \( \zeta(X) \) tries to estimate \( Y \) on the basis of solely the context \( X \).

• Then
  \[ Y^* - Y = \left( Y^* - \zeta(X) \right) - \left( Y - \zeta(X) \right) \]
  \[ \approx \frac{1}{|T|} \sum_{i \in T} (y_i - \zeta(x_i)) - \frac{1}{|C|} \sum_{i \in C} (y_i - \zeta(x_i)) \]

This is true regardless of the predictor quality.
But if it is any good, \( \text{var}[Y - \zeta(X)] < \text{var}[Y] \), and
Problems with A/B testing

• No single decision criterion →
  o Because of complex second order effects.

• Requires full implementation of treatment.

• Must wait two weeks for significant results.
  o Impractical for the early development of new ideas.
  o Cannot drive learning algorithms.

• Experimentation is limited by total traffic.
  o Hundreds of experiments are running at the same time.
  o Overlapped experiments.
3. Learning
End-to-end learning

Ideally we should train at the system level

“Train all aspects of the system to maximize a well defined objective function.”

Requirements

• The objective function **must** be correct.
  → we get what we ask for (only that and all of it!)
  → hard to debug

• Learning impacts global system architecture.
  → this point is difficult to make.
Learning as a component

Train a component of the system

• Convenient to manage teams.
• Example: train a click prediction module.

“ If the click prediction guys produce good probability estimates, the auction guys will know what to do...”

• Reproducing a software development pattern.

Machine learning is not like software development

• Machine learning is about the estimation error. How will they affect the other components of the system?
Click prediction and auctions

Optimal placement
(according to auction theory)

- Select eligible ads \((a_1 \ldots a_k)\)
- Obtain exact click probabilities \(q_i(x, a, p)\)
- Maximize purported value

\[
\max_p \sum_i b_i \times q_i(x, a, p)
\]

subject to reserve constraint

\[
b_i \times q_i(x, a, p) \geq R
\]

for all displayed ads.

How it really works
(with countless variations)

- Select eligible ads \((a_1 \ldots a_k)\)
- Estimate click probabilities

\[
q_i(x, a_i, p_i) = \gamma(x, p_i) \times \beta(x, a_i)
\]

- Rank ads with rank-score

\[
b_i \times \beta(x, a_i)
\]

- Greedily fill the best positions while reserve constraint is satisfied.

\[
b_i \gamma(x, p_i) \beta(x, a_i) \geq R
\]
Log-loss linear model

Basic probability estimation model

\[ f(z) = \sum_j w[j] z[j] \]

Train with

\[
\min_w \frac{1}{n} \sum \log(1 + e^{-y_t f(z_t)}) + \lambda \Omega(w)
\]

Estimate probabilities with

\[
q_i(x, a_i, p_i) = s(f(z)) = \frac{1}{1 + e^{-f(z)}}
\]
Auction constraints

Position model and ad model

\[ q_i(x, a_i, p_i) = S(w_1z_1 + \cdots + w_{d-1}z_{d-1} + w_dz_d) \]

\[ \gamma(x, p_i) \times \beta(x, a_i) \]

• This is not exactly \( \gamma(x, p_i) \times \beta(x, a_i) \) but this is monotonic

Forbidden features

• Click probability does not depend on bid.
  Otherwise auction theory result do not hold.
Crossing and coding features

- **\( p_i \):**
  - Layout
  - Position

- **\( x \):**
  - Query ID
  - Query text
  - Date, location
  - Session history
  - ...

- **\( a_i \):**
  - Ad ID
  - Ad text
  - Ad keywords
  - Advertiser ID
  - Bid

**Cross discrete features** → **Code discrete features** → **Statistical Model**

Cross discrete features

Code discrete features

Code discrete features

(001000 or hash)
Better click prediction ≠ better system

Which features help click prediction the most?
1. The position features (mainline vs. sidebar)
2. The context features (date, user, session history)
3. The advertiser features (some advertisers target better)
4. Finally the features crossing ad and query.

• 1 and 2 do not help picking good ads.
• 3 does not help picking ads relevant to the query.
• Only 4 makes the system markedly better.

➔ Optimizing click prediction without understanding the full system leads to selecting useless features.
Auction constraints

Using features that violate the auction theory constraints

• We know that auction theory does not model ad placement very well.
• A/B testing may show the system working better this way.
• We make the system harder to understand
  → Instant rewards compromise innovation speed
    (see previous slide.)
• We try to do end-to-end training by hand
  → Worst way to do it.
    (we get most of the problems and none of the benefits.)
Large-scale or not large-scale?

Discrete features can take many distinct values

- Query ID: \(~10^9\) distinct values
- (Query ID, Ad ID): \(~10^{10}\) effective distinct values
- User ID: \(~10^8\) distinct values

Discrete features are Zipf-distributed

- “Feature values appear with frequencies inversely proportional to their rank in the frequency table.”
- Lots of training examples for a few “head” values.
- Very few examples for many “tail” values.
- Our ability to model the tail is very limited. Even a linear model may be overkill in the tail.
Counting

Estimating a probability from a single discrete feature

• Training data \{ (f_1, y_1) \ldots (f_n, y_n) \}

• Maximum likelihood solution

\[ P(Y = 1|F = r) \approx \frac{\sum_i \mathbb{I}\{f_i = r\} \mathbb{I}\{y_i = 1\}}{\sum_i \mathbb{I}\{f_i = r\}} \]

• Laplacian smoothing is useful in the tail and harmless in the head

\[ P(Y = 1|F = r) \approx \frac{\varepsilon p + \sum_i \mathbb{I}\{f_i = r\} \mathbb{I}\{y_i = 1\}}{\varepsilon + \sum_i \mathbb{I}\{f_i = r\}} \]

• Accumulating counts with map-reduce is easy.
Leveraging counts

Position features

Code position features

Code head values only

Features for \( \beta(x, a_i) \)

Single feature counts for \( \beta(x, a_i) \)

Single feature counts for \( \beta(x, a_i) \)

Single feature counts for \( \beta(x, a_i) \)

Small combiner trained with log loss

log counts

log counts

log counts
Leveraging counts

Each single feature predictor produces two log counts

- \( \log \sum_i \mathbb{1}\{f_i = r\} \mathbb{1}\{y_i = 1\} \)
- \( \log \sum_i \mathbb{1}\{f_i = r\} \gamma_i(x, p_i) \quad \leftarrow \text{account for position model} \)

Training

- Pre-compute a small position model.
- Accumulate counts using map-reduce.
- The small combiner (~\(10^4\) params instead of \(10^{10}\)) trains in seconds.

Trading some modeling power for agility

- Empirically works as well as a full linear model.
- Easy to maintain, easy to experiment with.
The still ugly picture

• End-to-end training is desirable
  ➔ But we lack a clear objective function.
• Auction theory suggests to train click probability models
  ➔ We do not fully trust auction theory but we do it anyway...
• We can train click probability models efficiently
  ➔ Improving the click probability estimation does not necessarily improve the system.
• We need to discuss how we collect the training data
  ➔ This detail hides the most serious problem.
4. Loops
Toy example

Two queries

Q1: “cheap diamonds” (50% traffic)
Q2: “google” (50% traffic)

Three ads

A1: “cheap jewelry”
A2: “cheap automobiles”
A3: “engagement rings”

More simplifications

- We show only one ad per query
- All bids are equal to $1.
Toy example

True conditional click probabilities

<table>
<thead>
<tr>
<th></th>
<th>A1 (cheap jewelry)</th>
<th>A2 (cheap autos)</th>
<th>A3 (engagement rings)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1 (cheap diamonds)</td>
<td>7%</td>
<td>2%</td>
<td>9%</td>
</tr>
<tr>
<td>Q2 (google)</td>
<td>2%</td>
<td>2%</td>
<td>2%</td>
</tr>
</tbody>
</table>

Step 1: pick ads randomly.

\[ CTR = \frac{1}{2} \left( \frac{7 + 2 + 9}{3} + \frac{2 + 2 + 2}{3} \right) = 4\% \]
Step 2: estimate click probabilities

- Build a model based on a single Boolean feature:
  \( F \) : “query and ad have at least one word in common”

<table>
<thead>
<tr>
<th></th>
<th>A1 (cheap jewelry)</th>
<th>A2 (cheap autos)</th>
<th>A3 (engagement rings)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Q1</strong> (cheap diamonds)</td>
<td>7%</td>
<td>2%</td>
<td>9%</td>
</tr>
<tr>
<td><strong>Q2</strong> (google)</td>
<td>2%</td>
<td>2%</td>
<td>2%</td>
</tr>
</tbody>
</table>

\[
P(Click|F) = \frac{7 + 2}{2} = 4.5\%
\]

\[
P(Click|\neg F) = \frac{9 + 2 + 2 + 2}{4} = 3.75\%
\]
Toy example

Step 3: place ads according to estimated pclick.

Q1: show A1 or A2. (predicted pclick 4.5% > 3.75%)
Q2: show A1, A2, or A3. (predicted pclick 3.75%)

<table>
<thead>
<tr>
<th></th>
<th>A1 (cheap jewelry)</th>
<th>A2 (cheap autos)</th>
<th>A3 (engagement rings)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1 (cheap diamonds)</td>
<td>7%</td>
<td>2%</td>
<td>x 9% x</td>
</tr>
<tr>
<td>Q2 (google)</td>
<td>2%</td>
<td>2%</td>
<td>2%</td>
</tr>
</tbody>
</table>

\[
CTR = \frac{1}{2} \left( \frac{7 + 2}{2} + \frac{2 + 2 + 2}{3} \right) = 3.25\%
\]
## Toy example

### Step 4: re-estimate click probabilities with new data.

<table>
<thead>
<tr>
<th></th>
<th>A1 (cheap jewelry)</th>
<th>A2 (cheap autos)</th>
<th>A3 (engagement rings)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1 (cheap diamonds)</td>
<td>7%</td>
<td>2%</td>
<td>9%</td>
</tr>
<tr>
<td>Q2 (google)</td>
<td>2%</td>
<td>2%</td>
<td>2%</td>
</tr>
</tbody>
</table>

\[
P(\text{Click}|F) = \frac{7 + 2}{2} = 4.5% \\
P(\text{Click}|\neg F) = \frac{2 + 2 + 2}{3} = 2%
\]

- We keep selecting the same inferior ads. 😞
- Estimated click probabilities now seem more accurate. 😊
What is going wrong?

- Estimating Pclick using click data collected by showing random ads.

<table>
<thead>
<tr>
<th></th>
<th>A1</th>
<th>A2</th>
<th>A3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>7%</td>
<td>2%</td>
<td>9%</td>
</tr>
<tr>
<td>Q2</td>
<td>2%</td>
<td>2%</td>
<td>2%</td>
</tr>
</tbody>
</table>

1. Feature F identifies relevant ads using a narrow criterion.
2. Feature F misses a very good ad for query Q1.
3. Ads for query Q1 are ranked incorrectly.
4. P(Click|¬F) is pulled down by queries that do not click.

Adding a feature that singles out the case (Q1,A3)

- would improve the pclick estimation metric.
- would rank Q1 ads more adequately.
What is going wrong?

- Re-estimating P(click) using click data collected by showing ads suggested by the previous P(click) model.

<table>
<thead>
<tr>
<th></th>
<th>A1</th>
<th>A2</th>
<th>A3</th>
</tr>
</thead>
</table>
| Q1 | 7% | 2% | 9% | In this data, A3 is never shown for query Q1.  
|  | 2% | 2% | 2% |

Adding a (Q1,A3) feature
- would not improve the P(click) estimation on this data.
- would not help ranking (Q1,A3) higher.

Further feature engineering based on this data
- would always result in eliminating more options, e.g. (Q1,A2).
- would never result in recovering lost options, e.g. (Q1,A3).
We have created a black hole!

(Q,A) can be occasionally sucked by the black hole.
- All kinds of events can cause ads to disappear.
- Sometimes, advertisers spend extra money to displace competitors.

(Q,A) can be born in the black hole.
- Ads newly entered by advertisers
- Ads newly selected as eligible because of algorithmic improvements.

Exploration
- We should sometimes show ads that we would not normally show in order to train the click prediction model.
Learning feedback

- User
- Publisher
- Advertiser

Queries → Ads & Bids → LEARNING FEEDBACK LOOP → Prices → Clicks → Ads

Learning algorithm
Programmer feedback

Advertisements are places in the ad engine.

Hundreds working on the ad engine.
User and advertiser feedback

- **Users**
- **Publisher**
- **Advertisers**

**User Feedback Loop**
- Queries
- Ads
- Prices
- Clicks (and consequences)

**Advertiser Feedback Loop**
- Ads & Bids
The feedback loop problem

Shifting distributions

• Data is collected when the system operates in a certain way. The observed data follows a first distribution.
• Collected data is used to justify actions that change the operating point. Newly observed data then follows a second distribution.
• Correlations observed on data following the first distribution do not necessarily exist in the second distribution.

Often lead to vicious circles..
Explore/exploit trade-off

**Exploitation**
- Select actions that maximize our performance metrics.
- **Problem:** The system will settle into a regime that might not be well described by the past data. Other regimes may be more favorable.

**Exploration**
- Select actions that run the system in regimes that are not known, in the hope of discovering better ways to run it.
- **Problem:** We may not discover anything useful.
Coming next

**Next lecture**

- John  
  Classic approaches of the explore/exploit dilemma: Multi-armed bandits (MAB)
- John  
  Since MABs are not sufficient for the task: Contextual multi-armed bandits (CMAB).

**Second next lecture**

- Leon  
  Even CMABs are not sufficient for the full problem. Studying causation provides an alternative viewpoint with its strength and weaknesses.
- John, Leon  
  TBD.