# RL in the industry(Applications of bandits and recommendation systems)

Nicolas Le Roux

Google Brain

Presenter: Tianlu Wang

# Outline

# Two main components in RL

- Multi-step episodes
- **Reward evaluation and maximization**(in one episode)

# Retargeting in online advertising

- A user lands on a webpage
- Website contacts an ad-exchange
- Ad-exchange contacts the retargeter
- Its an **auction**: each competitor tells how much it bids
- Highest bidder wins the right to display an ad

# Details of auction

- Real-time bidding (RTB)
- 2nd -price auction: the highest bidder wins but pays the second highest bid
- Optimal strategy: bid the expected gain
- E[gain] = price per click (CPC) * **Probability(click)(CTR)**

# Outline

# Classifier

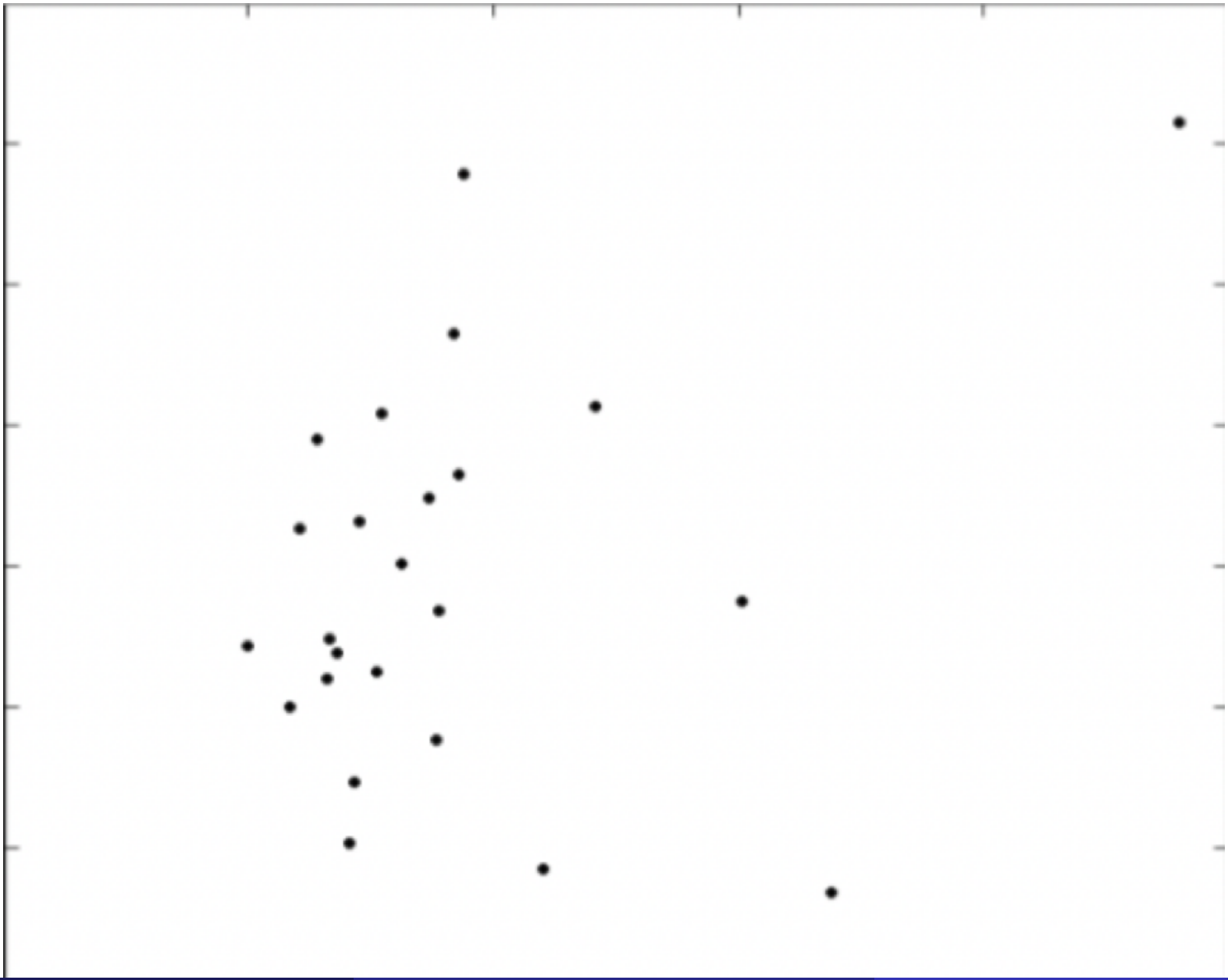
- We wish to estimate the probability of click
- We have access to labelled data (for won auctions)
- X: information about the user
- Y: click / no click
- First reaction is to build a classifier for this

# Pick a better model

- Two-arm bandit: system A (current) vs. B (new)
- Split the population for some period of time
- Choose the system with the best average reward

# Outline

# Test error vs. true revenue



# Because of implicit assumptions

- The log-loss is a good proxy for the revenue?:
  - people only care about what is the optimal when they define the loss function but we should also consider how much we need to pay if we make wrong decision(white board)
- The input distribution is the same?:
  - Labelled data is on the won auctions
  - The bidding algorithm impacts input distribution
  - The best model can change

# Simpson's paradox

| CTR | Top banner | Side banner |
|---|---|---|
| Overall | 60/9000(0.67%) | 50/7000(0.71%) |
| High-value-users | 48/8000(0.6%) | 2/1000(0.2%) |
| Low-value-users | 12/1000(1.2%) | 48/6000(0.8%) |

# Outline

# Possible methods

- Add as many variables as possible in the model: Too complicated
- Run online A/B tests: Smaller companies take longer to collect data
- Exploration(demonstration):
  - Exploration converges to the optimum when the model is well-specified!
  - It almost never is.(In most cases, bandit algorithms don't work)
- Perform counterfactual analyses:What would have happened if we had taken another decision?

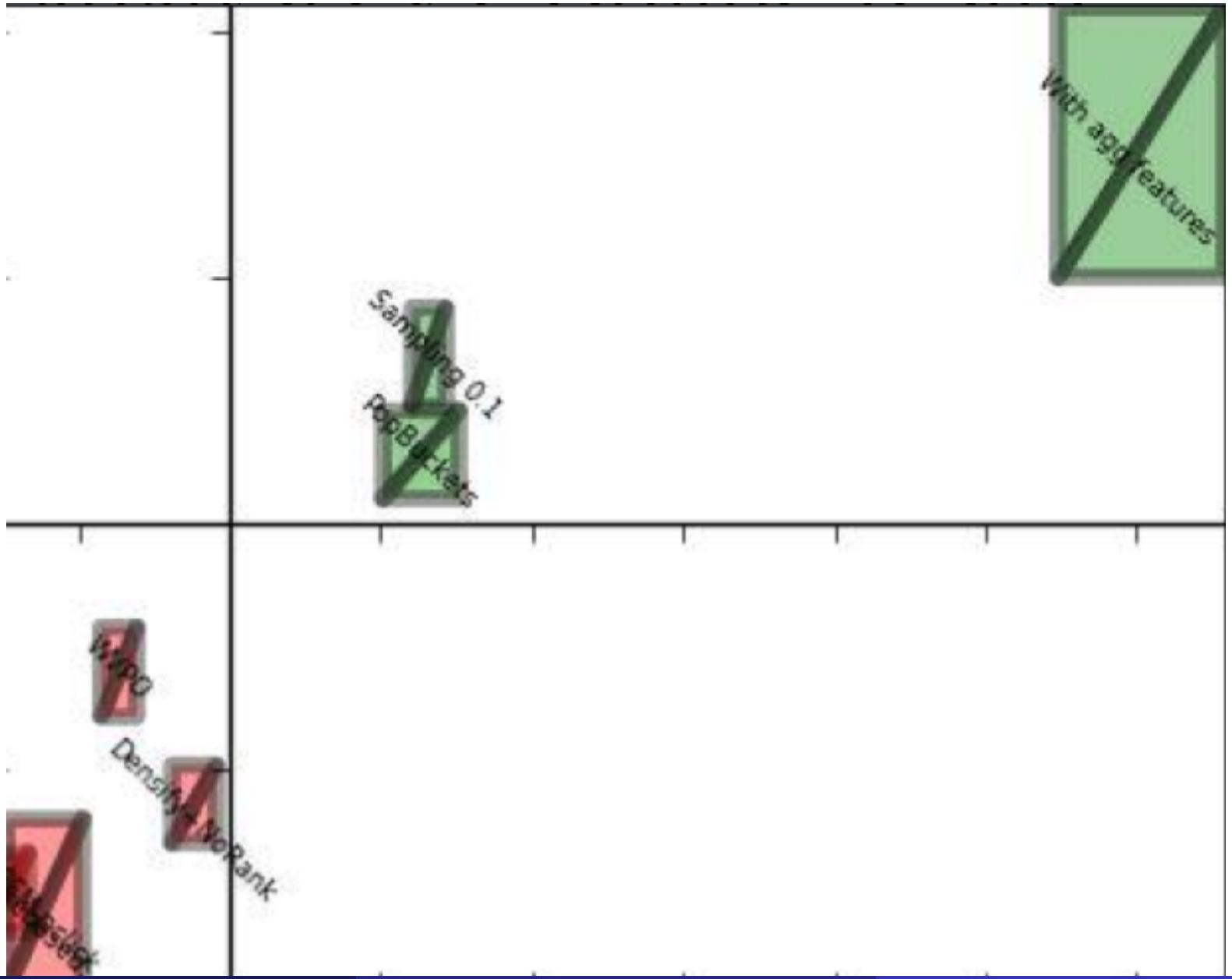
# Outline

# RL

- Current distribution over actions: $p(a|s)$
- Expected value of new distribution $q(a|s)$?
-

$$
\begin{aligned}
G(q) &= \int_s \int_a p(s) q(a|s) r(a,s) da ds \\
&= \int_s \int_a p(s) \frac{q(a|s)}{p(a|s)} p(a|s) r(a,s) da ds \qquad (1) \\
&\approx \frac{1}{N} \Sigma_i \frac{q(a_i|s_i)}{p(a_i|s_i)} r_i
\end{aligned}
$$

# Estimated reward vs. Real reward



# Outline

# Evaluation is underrated

- Importance sampling allows us to evaluate q
- We may now optimize over q
- Rolling out a new policy is expensive
- How to optimize with few updates?

# Summary

- Robustness and efficiency are critical
- This includes pipeline efficiency
- Improving the model is useless w/o good reward
- RL deals with tangible quantities.