RL in the industry

Nicolas Le Roux

Google Brain

5/7/17

Disclaimer

- This talk is about my own experience
- It will be focused on online advertising
- This is by no means limited to that industry

Two components in RL

- Multi-step episodes
- Reward evaluation and maximization

Two components in RL

- Multi-step episodes
- **Reward evaluation and maximization**

Retargeting: how it works

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

Details of the auction

- Real-time bidding (RTB)
- 2^{nd}-price auction: winner pays the second highest bid
- Optimal strategy: bid the expected gain
- $\mathbb{E}$[gain] = price per click (CPC) * $\mathbb{P}$(click) (CTR)

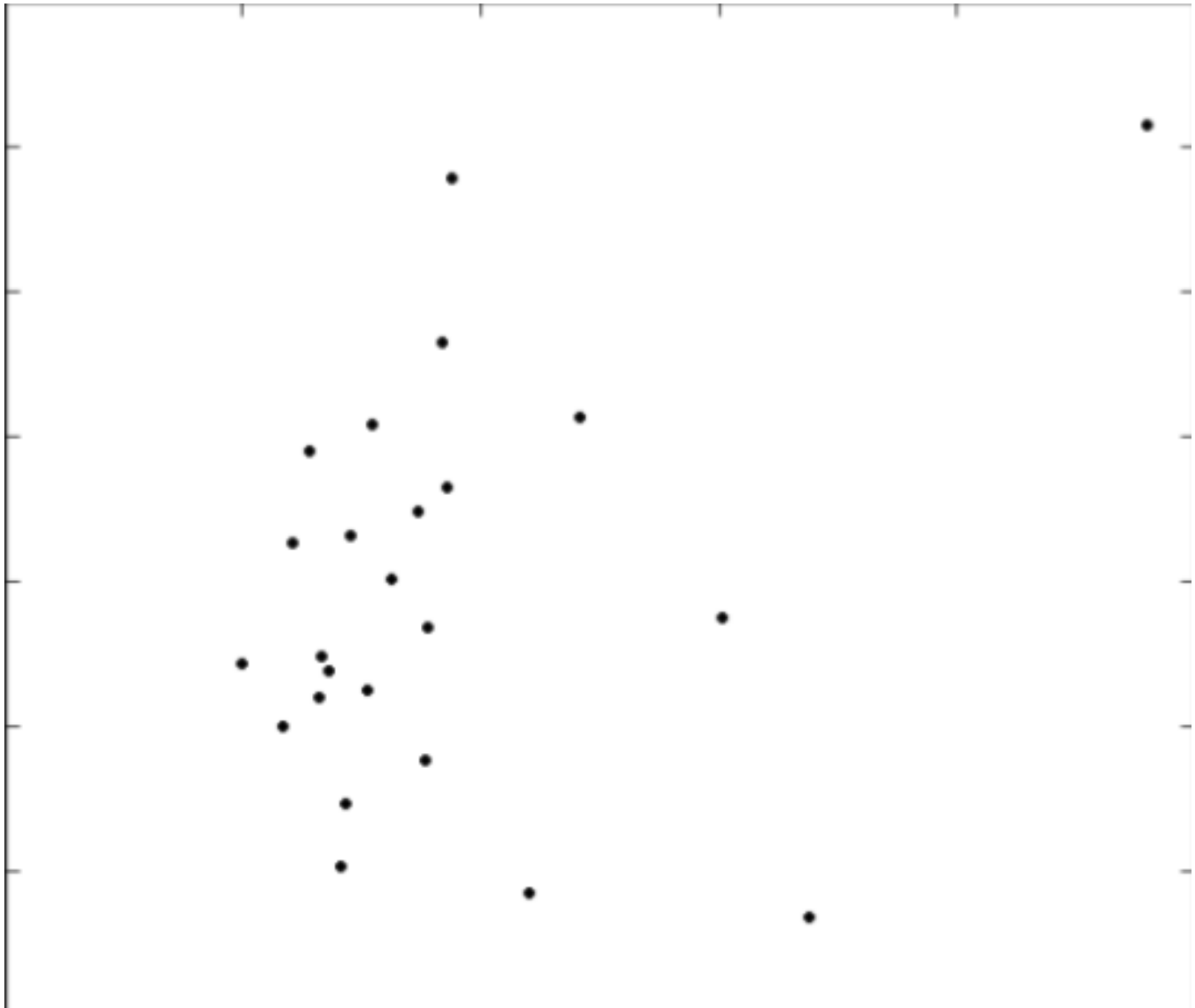
Finding a bidding strategy

- 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

A/B testing

- 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

RMSE vs. true revenue



Implicit assumptions

1. The log-loss is a good proxy for the revenue
2. The input distribution is the same

Quality of the proxy

Demonstration

Quality of the proxy

Demonstration

- *“How will my system be used?”*

Quality of the proxy

Demonstration

- *“How will my system be used?”*
- Actual rewards must drive the evaluation
- There is more to a loss function than its optimum

Is the input distribution the same?

- Labelled data is on the won auctions
- The bidding algorithm impacts input distribution

Is the input distribution 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%)

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%)

Dealing with confounding variables

- Add as many variables as possible in the model
- Run online A/B tests
- Exploration
- Perform counterfactual analyses

Exploring exploration

Demonstration

Exploring exploration

Demonstration

- Exploration converges to the optimum when the model is well-specified!

Exploring exploration

Demonstration

- Exploration converges to the optimum when the model is well-specified!
- It almost never is.

Misspecified model

Demonstration

- Misspecified model: tradeoffs are made
- Tradeoffs are based on input distribution
- This must be controlled

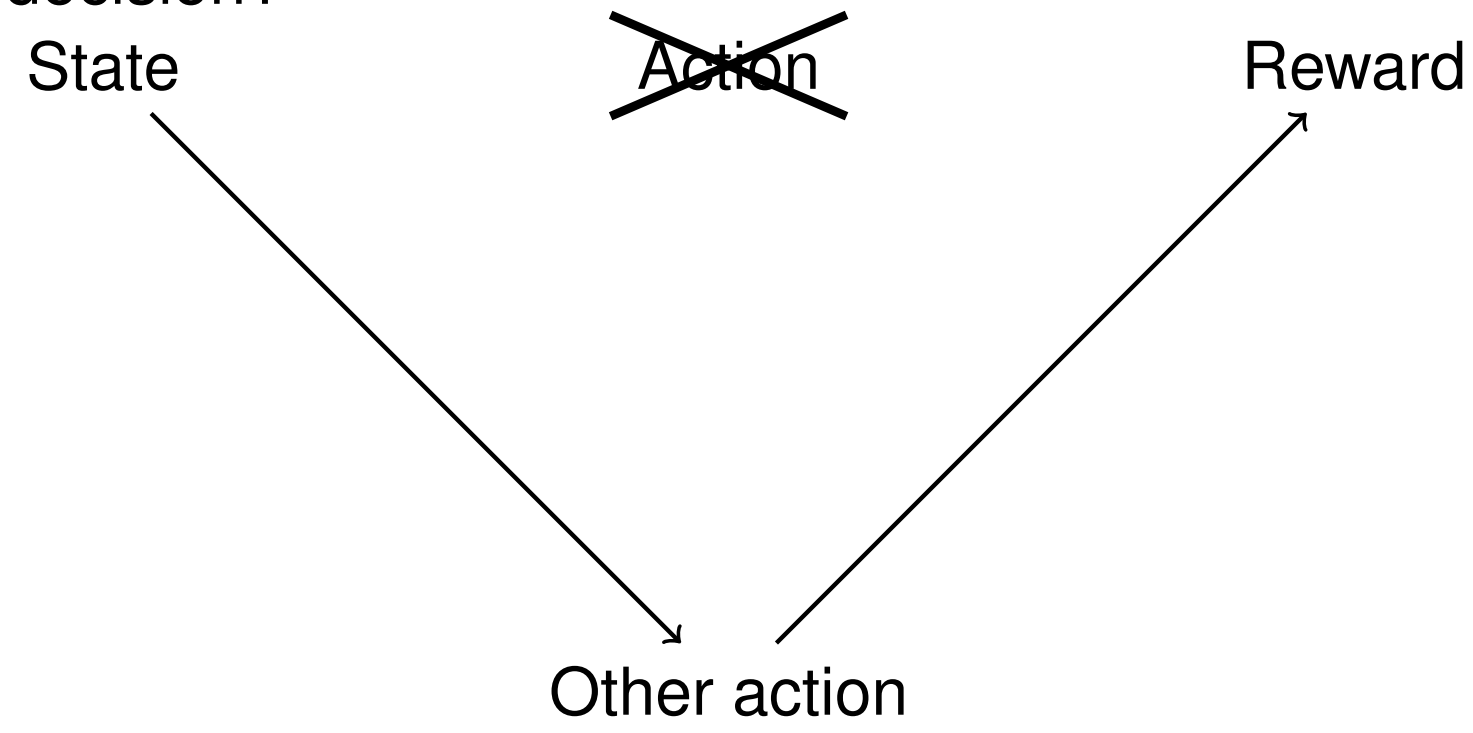
Counterfactual question

"What would have happened if we had taken another decision?"

State $\longrightarrow$ Action $\longrightarrow$ Reward

Counterfactual question

“What would have happened if we had taken another decision?”



- Current distribution over actions: $p(a|s)$

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

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

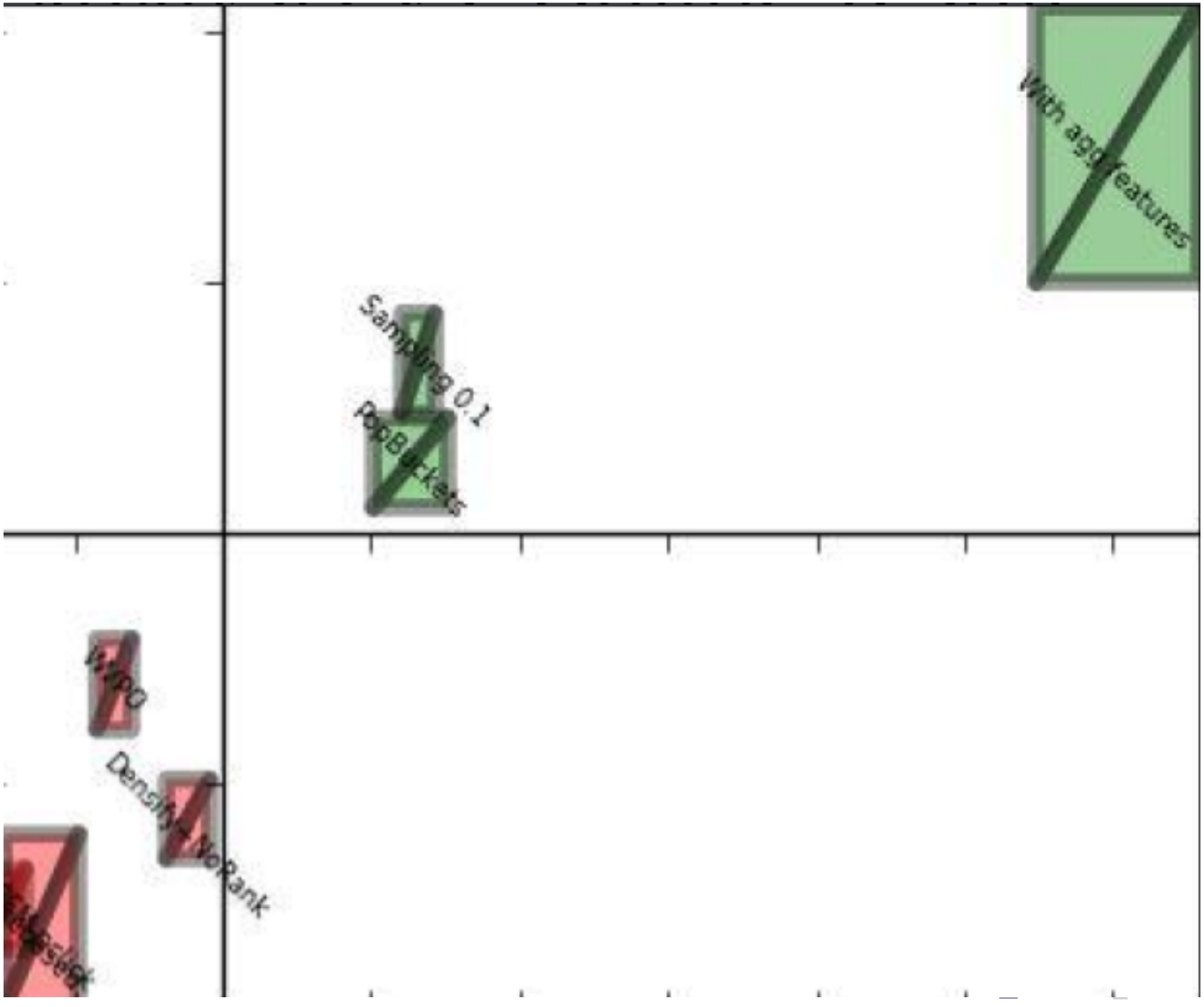
$$
\begin{aligned}
G(q) &= \int_s \int_a p(s) q(a|s) r(a,s)\ da\,ds \\
&= \int_s \int_a p(s) \frac{q(a|s)}{p(a|s)} p(a|s) r(a,s)\ da\,ds \\
&\approx \frac{1}{N} \sum_i \frac{q(a_i|s_i)}{p(a_i|s_i)} r_i \ .
\end{aligned}
$$

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

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

- This is *off-policy* policy evaluation.

Offline vs. online evaluation



From evaluation to optimization

- Importance sampling allows us to evaluate q
- We may now optimize over q

From evaluation to optimization

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

From evaluation to optimization

- 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?

Benefits of policy evaluation

- It is a better predictor
- It predicts *tangible* quantities
  - Constraint optimization becomes meaningful
- It takes other system components into account

Efficient policy optimization

- Optimizations are performed regularly
- They must be trouble-free
- Stochastic methods are rarely trouble-free

Efficient policy optimization

- Optimizations are performed regularly
- They must be trouble-free
- Stochastic methods are rarely trouble-free
- There is a need for robust optimization methods!

Other unanswered questions

- Inference time is critical
  - How to balance precise and fast inference?
- Rewards are of multiple form (clicks/sales/etc.)
  - How to combine them?

Executive summary

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

Thank you!

nlr@google.com