Genie: An Open Box Counterfactual Policy Estimator for Optimizing Sponsored Search Marketplace

Murat Ali Bayir

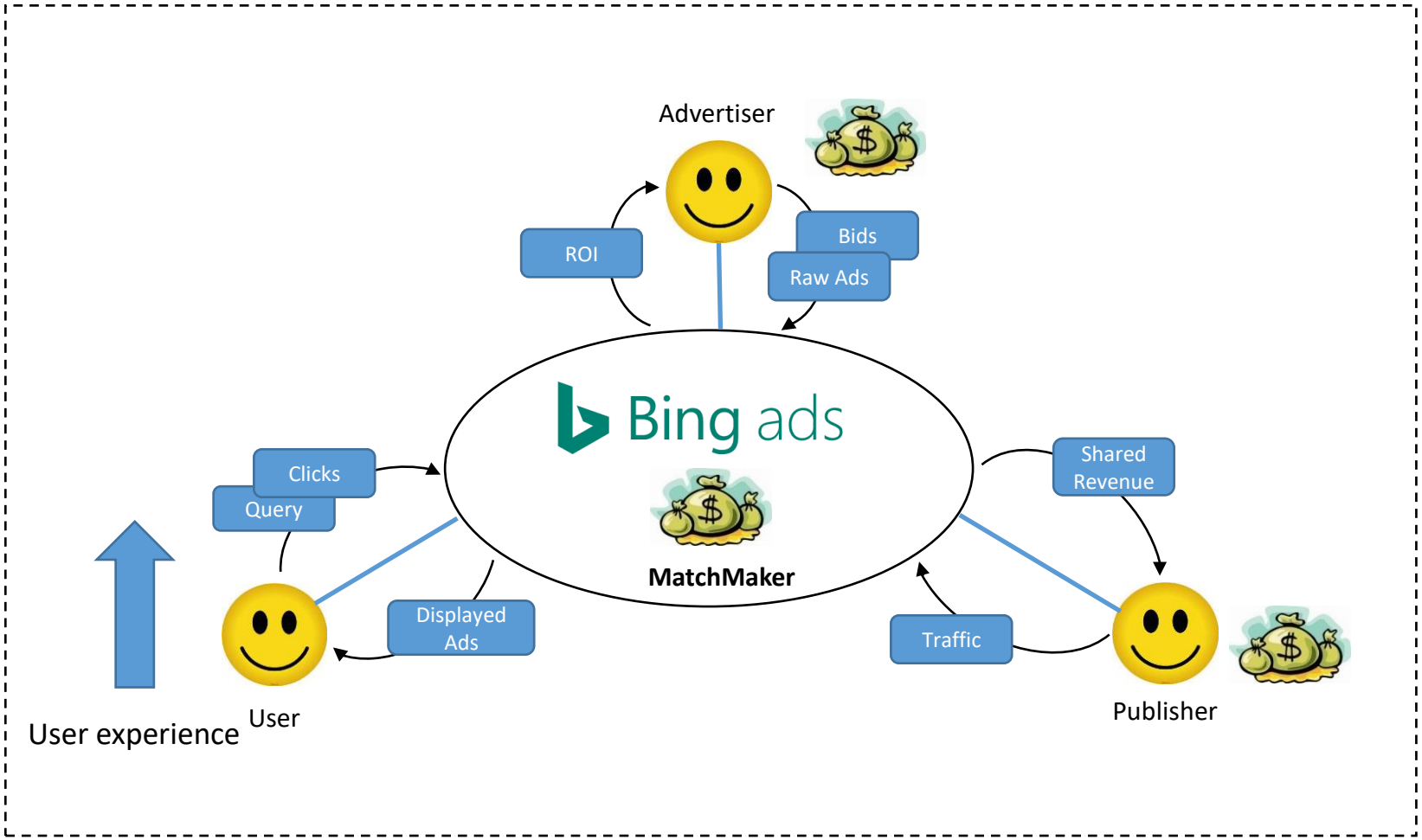
Joint work with Mingsen Xu, Yaojia Zhu and Yifan Shi

February 2019, WSDM 2019

Outline

- ❑ Sponsored Search Optimization
- ❑ Genie Open Box Counterfactual Policy Estimator
- ❑ Experimental Results
- ❑ Challenges and Lessons Learned

Overview of Ads Marketplace



How do we Optimize Marketplace?

- Optimizing Marketplace requires **measuring KPI Impact of any modification to the System.**
- Need a counterfactual estimation system to answer “What If Questions?”

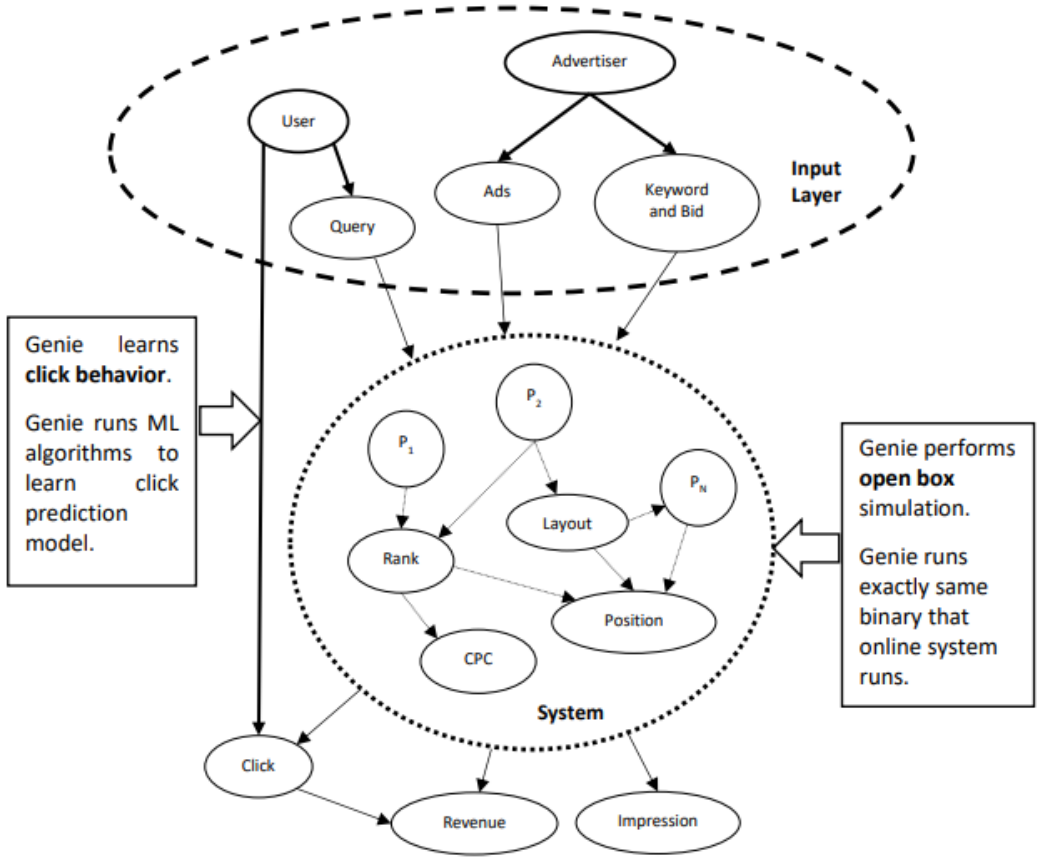
Existing Approaches

Method	Basic Idea	Pros	Cons
A/B Testing	- Run Two online experiment. - Deploy modification to real traffic run as treatment. - Compare with control traffic.	- Accurate - Can Measure many type of modification	- E2E Deployment - Risk in Real Traffic - Limited Parameter Space and policy Combinations.
Observational	- Run an online randomized experiment - Collect randomized data from logs. - Run Offline Training to generate new model or activate policy.	- Large Parameter Space. - Quick updates and efficiency.	- E2E Deployment. - Real Traffic for randomized experiment. - Cold Start problem.

Outline

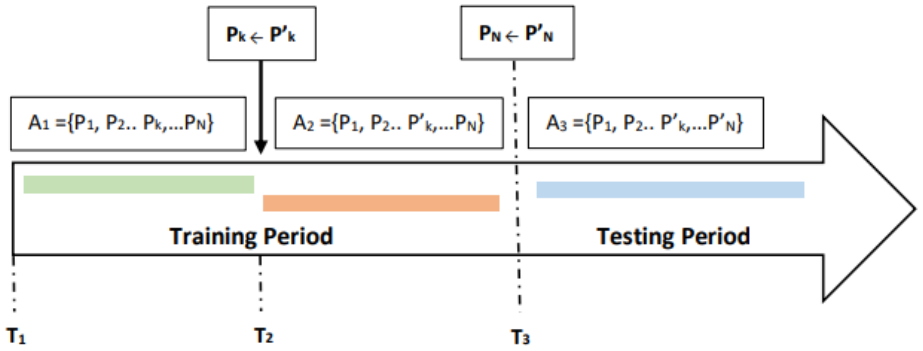
- ❑ Sponsored Search Optimization
- ❑ Genie Open Box Counterfactual Policy Estimator
- ❑ Experimental Results
- ❑ Challenges and Lessons Learned

Idea of Open Box Simulation



Simplified view of Causal Graph for Sponsored Search

Why Open Box?

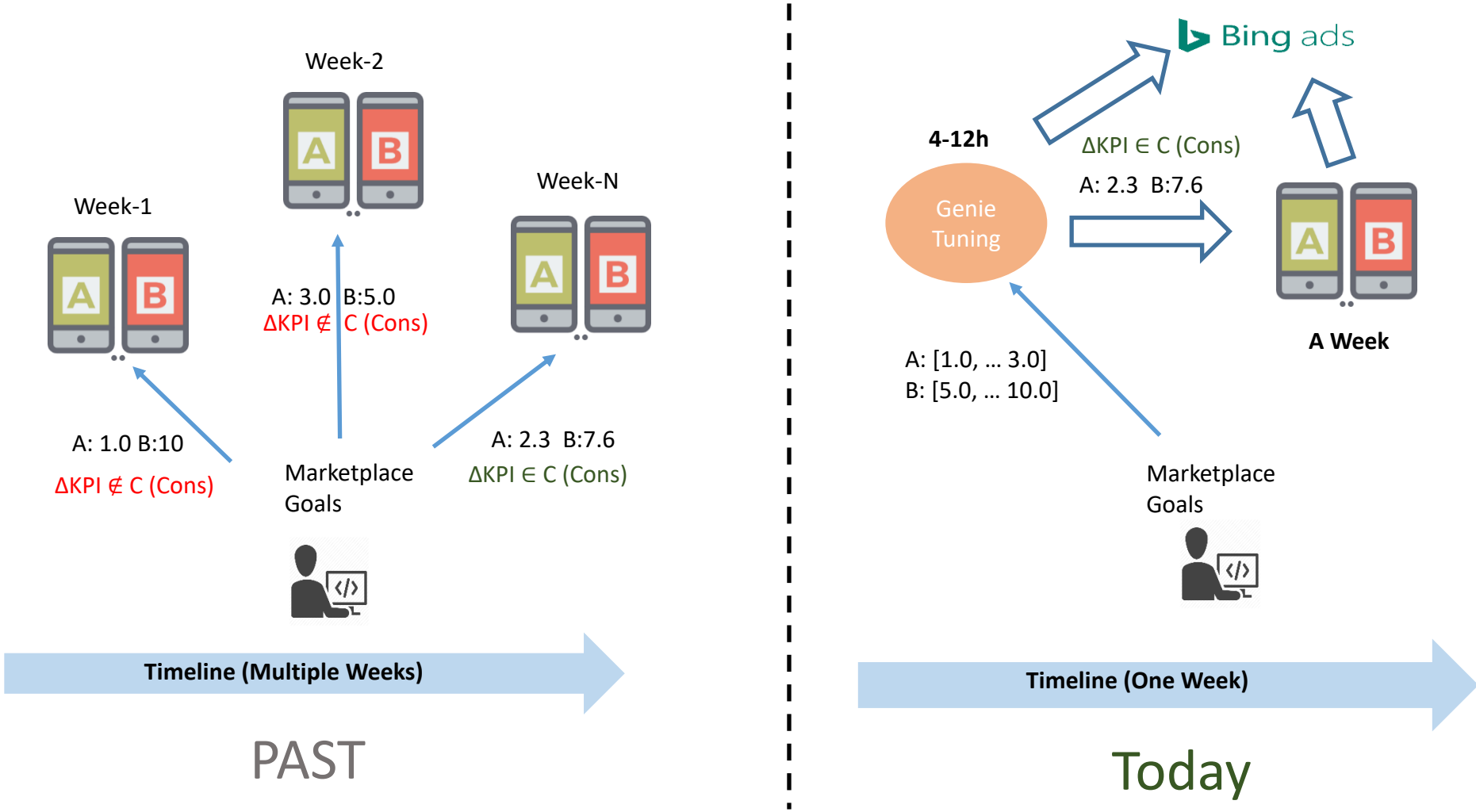


Policy-Estimation/Tuning Job with Biased Data

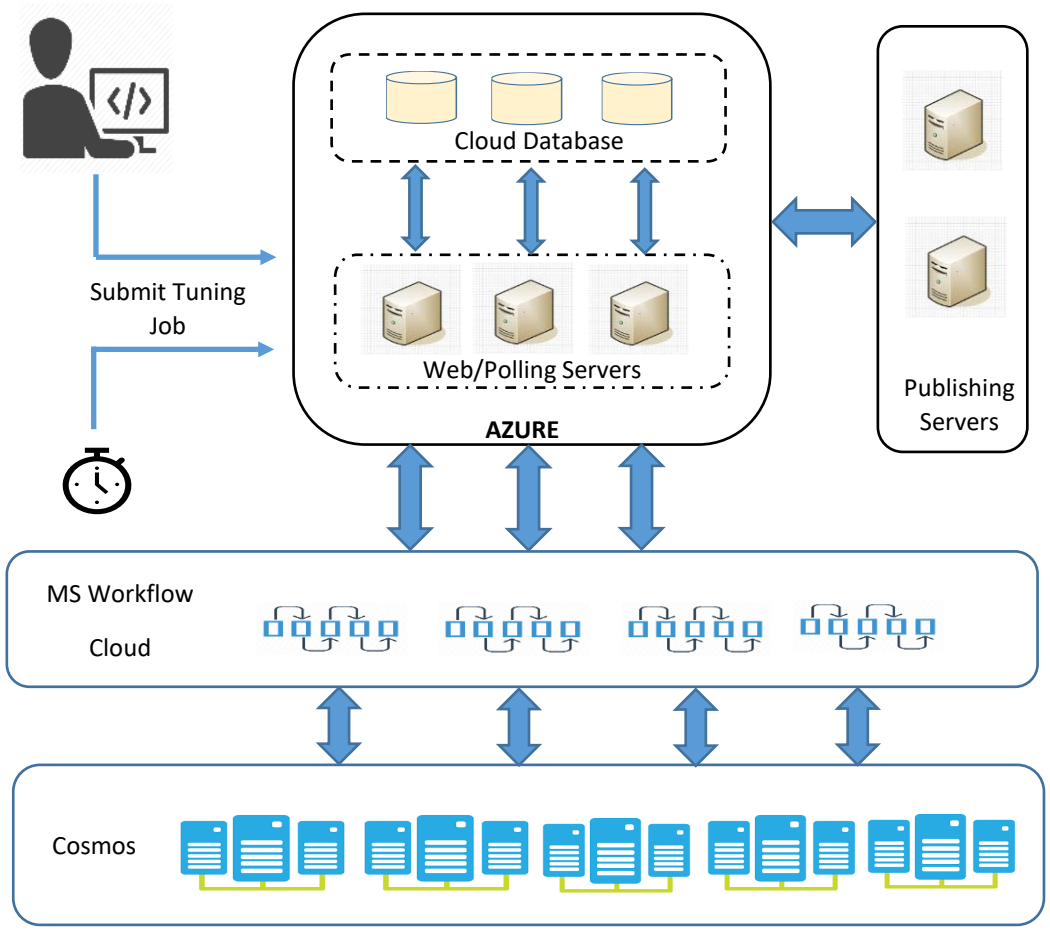
Actual Settings	[T1, T2) : $\{P_1, P_2, \dots P_k, \dots, P_N\}$ [T2, T3) : $\{P_1, P_2, \dots P_k', \dots, P_N\}$
Baseline For Open Box Estimator	[T1, T3) : $\{P_1, P_2, \dots P_k', \dots, P_N\}$
Counterfactual Setting	[T1, T3) : $\{P_1, P_2, \dots P_k', \dots, P_N'\}$

- Enables to get rid of the negative impact of historical policy interactions. (Bing Ads PC traffic slice is running more than 200 experiment at the same time.)
- Provides using higher volume of historical data. One sampling point could represent multiple settings/policies. No limitations due to sample size.
- No Randomization cost and minimize the risk for Real Experiment!
  - Cold Start problem can bite performance or delay shipping even for auto tuning.
- Can be leveraged when randomized experiment or A/B Testing is not appropriate.
  - Bid vs Traffic/Click Estimation Recommendation.

Tuning/Policy-Estimation Cycle with Genie



Genie Architecture and Platforms

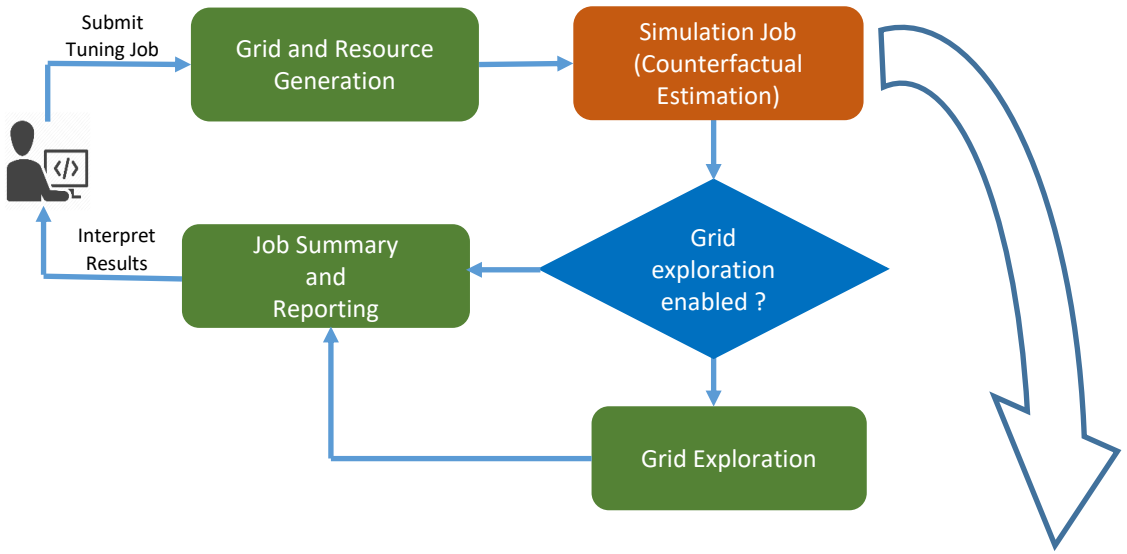


- Runs on Azure Cloud
- Submit and monitor jobs
- Download job results
- Provide REST API
- Talks with Publishing Servers

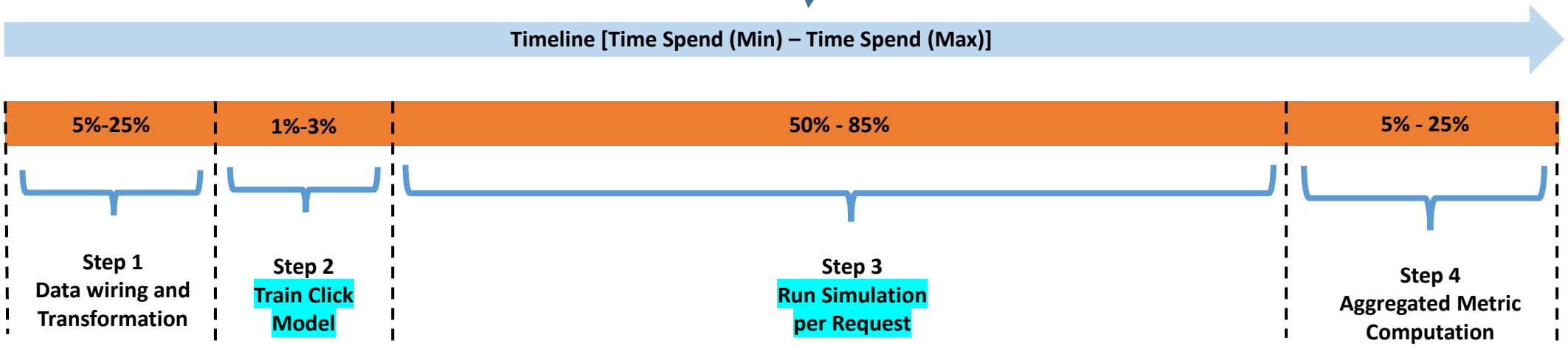
- Runs on AEther Cloud
- Executes different workflows based on scenario.

- Runs on Cosmos Cloud
- Do the heavy lifting.
- Main Training and KPI computation Jobs.

Genie Tuning/Policy-Estimation Workflow

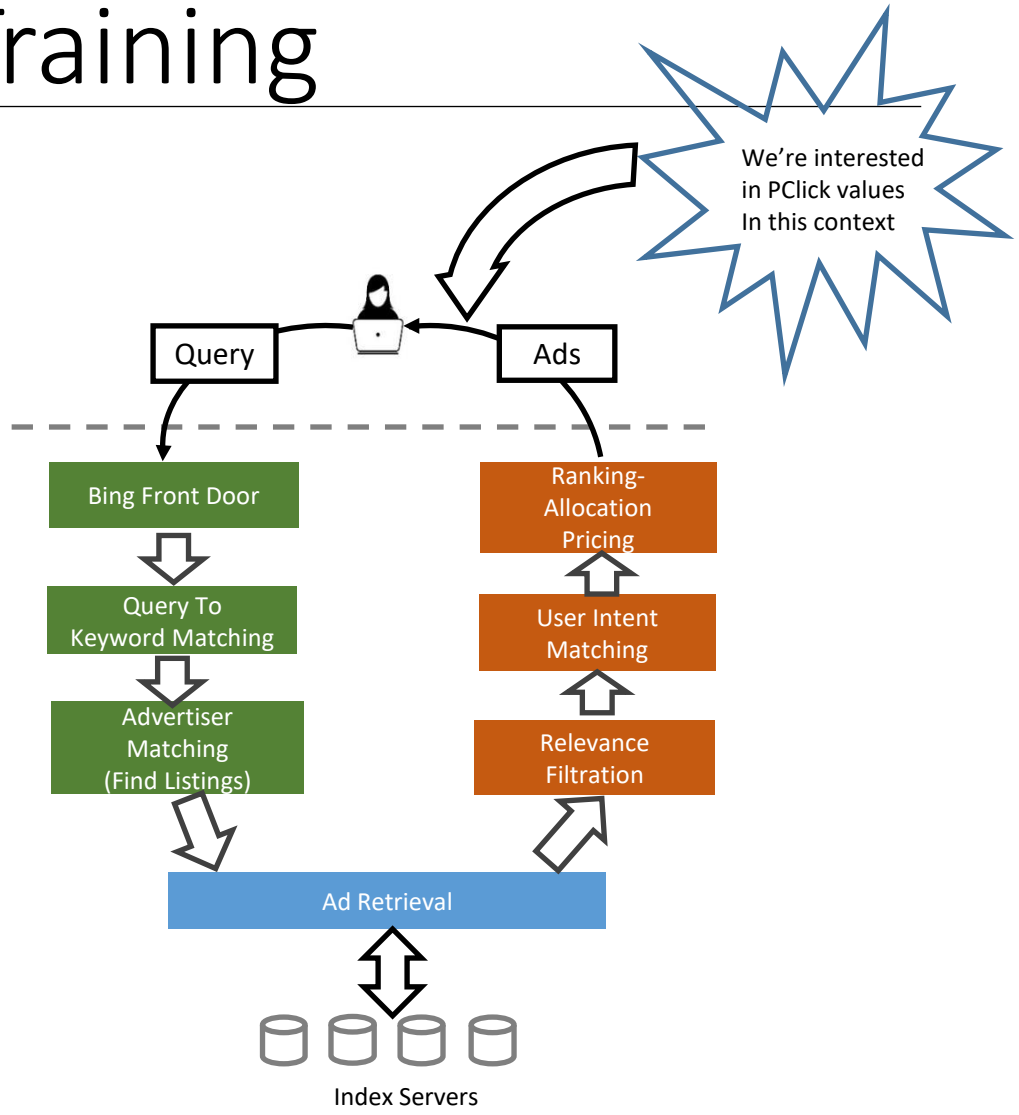


- Most Typical Workflow for Counterfactual Estimation.
  - Used for Hyperparameter Tuning
  - New Policy Evaluation and Comparison
- 3 Mandatory + 1 Optional Step
- Main Step is the **Simulation Job** that performs Counterfactual Estimation.
- **Train Click Calibration and Run Simulation per Request** are most important steps.



Click Model Training

- **Goal:** Learn Click Predict Model to *calibrate PClick scores* in replay/counterfactual page allocations.
- **Inputs:**
  - **Logs:** Extract Impression and Page Level Features with Click/No-Click signals.
  - **Tuning XML**: Features and types of model to train.
- **Outputs:**
  - Trained model on Cosmos
  - The PClick Score here is special one that is used for Offline metric Computation! Not the one used in online system.
    - P(Click | Displayed Context)
    - Different problem than Pclick Scores used in Online.
- **Genie currently supports two models:**
  - Bayesian Probit (Full Support)
    - **Around %95 of scenarios.**
  - GTB: (Evaluation Only in Genie. Training is outside the Genie)



Simplified Lifecycle of Query in Bing Ads

Bayesian Probit

- **Probit Model:** Special regression y can have only two value.
  - In our case y $\in$ {-1, +1} and model produces P in [0,1].
  - Supports only categorical features!
  - Each feature j is divided into finite number of M_j bins (E1)
  - Only single bin is activated for each feature (E1)
  - Each data point corresponds to sparse vector that has many zeros and L ones if there are L features. (E2)
  - Each bin is represented by weight which comes from Gaussian Distribution (E3)
  - (E4) is the sampling Distribution for Probit Model (likelihood):
  - Prior for a given weight vector W is given in (E5).
- **Training:** The Bayesian Inference for the posterior in the form of (E6) does not have any closed form solution.

$$x_{ij} = \begin{bmatrix} x_{ij(1)} \\ \dots \\ x_{ij(M_j)} \end{bmatrix} and \sum_{k=1}^{k=M_j} x_{ij(k)} = 1 \quad \text{E1}$$

$$x_i = \begin{bmatrix} x_{i1(1)} \\ \dots \\ x_{iL(M_L)} \end{bmatrix}, \sum_{j=1}^{j=L} \sum_{k=1}^{k=M_j} x_{ij(k)} = L \quad \text{E2}$$

$$\mu = \begin{bmatrix} \mu_{11} \\ \dots \\ \mu_{L(M_L)} \end{bmatrix}, \sigma^2 = \begin{bmatrix} \sigma^2_{11} \\ \dots \\ \sigma^2_{L(M_L)} \end{bmatrix} \quad \text{E3}$$

$$p(y|\boldsymbol{x}, \boldsymbol{w}) := \Phi\left(\frac{y \cdot \boldsymbol{w}^T \boldsymbol{x}}{\beta}\right) \quad \text{E4}$$

$$p(\boldsymbol{w}) = \prod_{i=1}^{N} \prod_{j=1}^{M_i} \mathcal{N}\left(w_{i,j}; \mu_{i,j}, \sigma^2_{i,j}\right) \quad \text{E5}$$

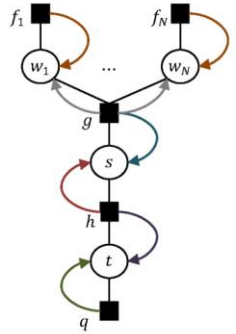
$$p(\boldsymbol{w}|\boldsymbol{x}, y) \propto p(y|\boldsymbol{x}, \boldsymbol{w}) \cdot p(\boldsymbol{w}) \quad \text{E6}$$

Bayesian Probit

- **Training:**
  - Generate Factor Graph Model for Bayesian Probit with message flow.
    - Each f_k sample from w_k and s is linear combination of weights $s = x^T w$.
    - Then add zero gaussian noise to get t from s.
    - The sign(t) is the click or no click signal.
  - Use stochastic message passing algorithm to optimize weights for each single data point by using **expectation propagation [1]**.
- **Training Algorithm**
  - For each Impression data point.
    - Find matched bins for each feature
    - Compute the total variance and mean using gaussian of matched bins. (E1)
    - For each matched bin:
      - Update the mean and variance. E(2-3) [2]
- **Evaluation:**
  - Total mean and variance are computed for new data X (E1)
  - The cumulative distribution on total mean over square root of variance.

$$p(\boldsymbol{w}|\boldsymbol{x}, y) \propto p(y|\boldsymbol{x}, \boldsymbol{w}) \cdot p(\boldsymbol{w})$$

$$p(y \mid t) \cdot p(t \mid s) \cdot p(s \mid \boldsymbol{x}, \boldsymbol{w}) \cdot p(\boldsymbol{w})$$



(Total Mean/Variance) $\Sigma^2 := \beta^2 + \boldsymbol{x}^T \boldsymbol{\sigma}^2 \quad \mu = \boldsymbol{x}^T \boldsymbol{\mu}$ E1

(Mean Update) $\tilde{\mu}_{i,j} = \mu_{i,j} + y x_{i,j} \cdot \frac{\sigma_{i,j}^2}{\Sigma} \cdot v\left(\frac{y \cdot \boldsymbol{x}^T \boldsymbol{\mu}}{\Sigma}\right)$ E2

(Variance Update) $\tilde{\sigma}_{i,j}^2 \leftarrow \sigma_{i,j}^2 \cdot \left[1 - x_{i,j} \cdot \frac{\sigma_{i,j}^2}{\Sigma^2} \cdot w\left(\frac{y \cdot \boldsymbol{x}^T \boldsymbol{\mu}}{\Sigma}\right)\right]$. E3

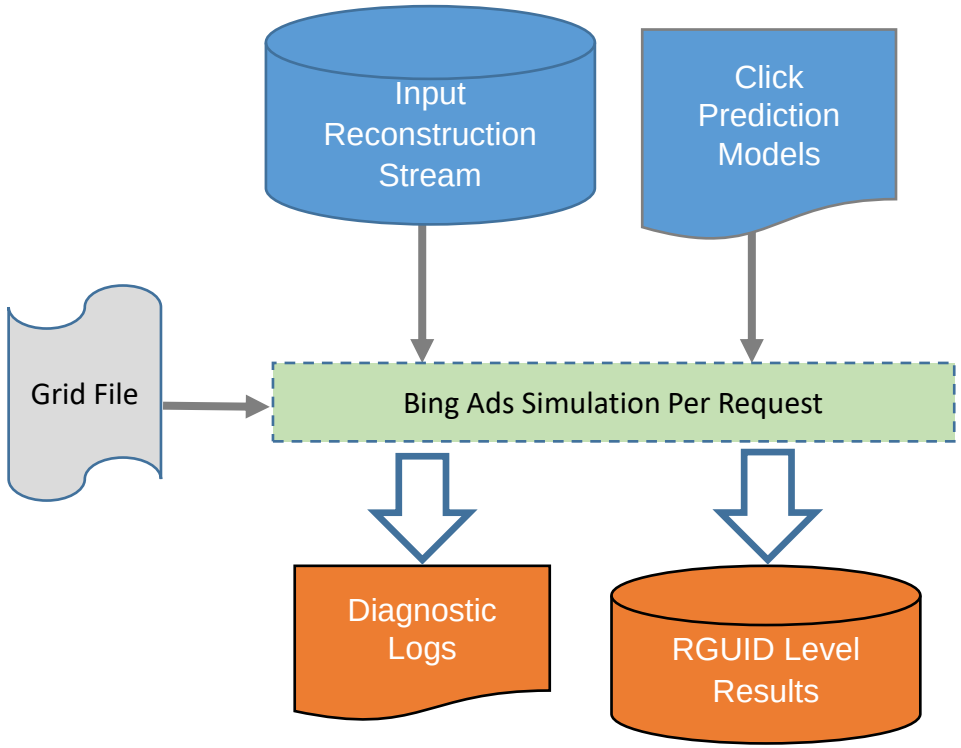
P(y|x) = $\Phi\left(\mu / \sqrt{\sigma^2 + \beta^2}\right)$

$$\Phi(x) = \int_{-\infty}^{x} N(x, 0, 1) dx = \int_{-\infty}^{x} \frac{e^{-x^2/2}}{\sqrt{2\pi}} dx$$

[1] Tom Minka. A family of algorithms for approximate Bayesian inference. PhD thesis, MIT. 2001

[2] Thore Graepel, Joaquin Quiñonero Candela, Thomas Borchert, Ralf Herbrich: Web-Scale Bayesian Click-Through rate Prediction for Sponsored Search Advertising in Microsoft's Bing Search Engine. ICML 2010: 13-20

Bing Ads Simulation per RGUID



- **Goal:** Performs counterfactual estimation in request level.
  - Inputs:
    - **Input Reconstruction:** Contains all of the data that is needed to call online libraries.
    - **Grid File**: Contains Counterfactual Settings
    - **Click Prediction Models:** Contains click prediction models
  - Outputs:
    - **RGUID Level Results**. Contains Request level KPI results for Request X CF-Setting pairs.
    - **Diagnostic Logs:** Contains error logs like replay/simulation problems.

Bing Ads Simulation per RGUID

Algorithm 1: Simulation Algorithm

Input: Auction Data: A, Grid: G, Click Model: C
Output: List of (Setting, KPI) pairs as $KPIs$

$M \leftarrow G$.GenerateModifiers(A)
$KPIs \leftarrow \{\}$ // Initialize the output.
foreach $M_i \in M$ **do**
  // Modify the input and get (restorer, setting)
  $(R_i, S_i) \leftarrow M_i$.Modify($A$)
  $P_i \leftarrow$ OnlineLibrary(A) // Create i^{th} page allocation
  C.Predict(P_i) // Adjust click probabilities
  $KPI_i \leftarrow$ GetKPI(P_i)
  $KPIs \leftarrow KPIs \cup (S_i, KPI_i)$
  R_i.Restore(A) // Restore the input to original value
end

- Each counterfactual is converted into **Modifier and Restorers**.
- Modifiers modifies the simulation input in place and returns a restorer.
- Restorer restores the input to the original value.
- Each Inner loop do the following:
  - Modify Input
  - Call Online Library
  - Calibrate Page Assignment
  - Compute KPI
  - Restore Inputs.

Grid Exploration

- Optional step and executed as a single box in Microsoft Workflow Execution Cluster.
- **Goal:** Explore the Hyperparameter -> KPI space and find better point than the most optimal point that comes from initial Genie job.
- Improves the Job efficiency without running large grid space.

Algorithm 3: Optimization Algorithm

Input: Hyperparameters: X, KPI deltas: ΔY, Batches: B
Input: Population size P, Solution size: k, Objective: F_{obj}
Output: Best solution set: X' and $\Delta Y'$

```
Model = Train(X, Y)
X' ← X, ΔY' ← ΔY
for i ← 1 to B do
    X_i ← Explore(X', P) // Create new solution set
    ΔY_i ← Model.Predict(X_i)
    // Select best P solution from current union previous.
    ΔY', X' ← Select(X' ∪ X_i, ΔY_i ∪ ΔY', P)
end
// Select the top k solution among ΔY'
(X', ΔY') ← Top(X', ΔY', k, F_obj)
```

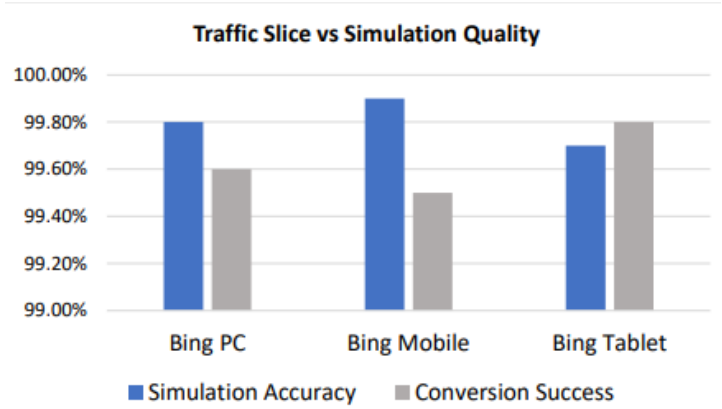
- First learn a linear mode from hyperparameters to KPI Deltas X -> ΔY
- Then use iterative algorithm similar to hill climbing to explore the space to find better operating points!
- Works pretty good for interpolation.
- Very fast!

Outline

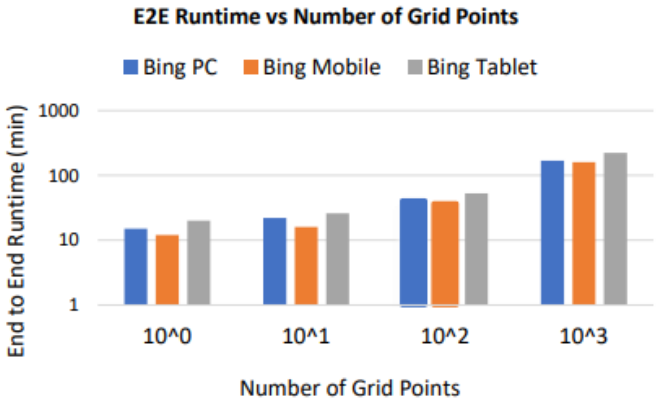
- ❑ Sponsored Search Optimization
- ❑ Genie Open Box Counterfactual Policy Estimator
- ❑ Experimental Results
- ❑ Challenges and Lessons Learned

Simulation Quality and Runtime Performance

Jan 2018



Jan 2018



Accurate and scalable Simulator is the most important component of open box simulation!

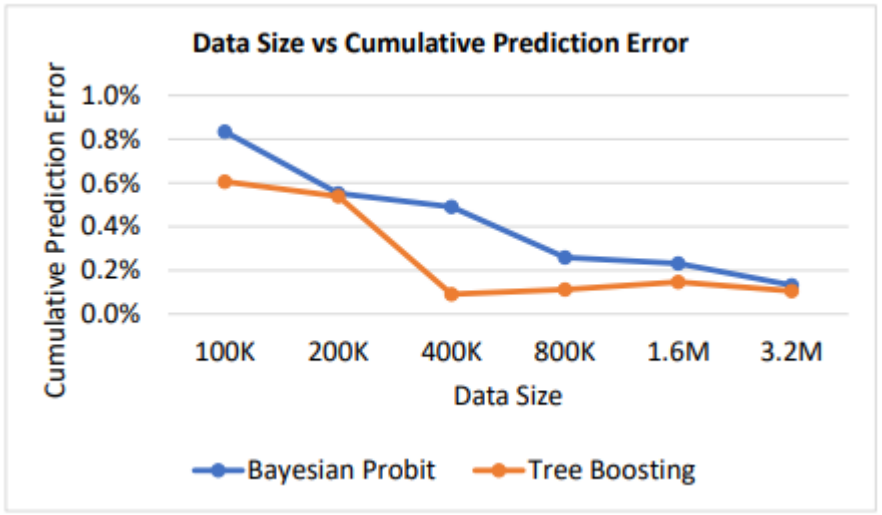
Simulation Quality Metrics

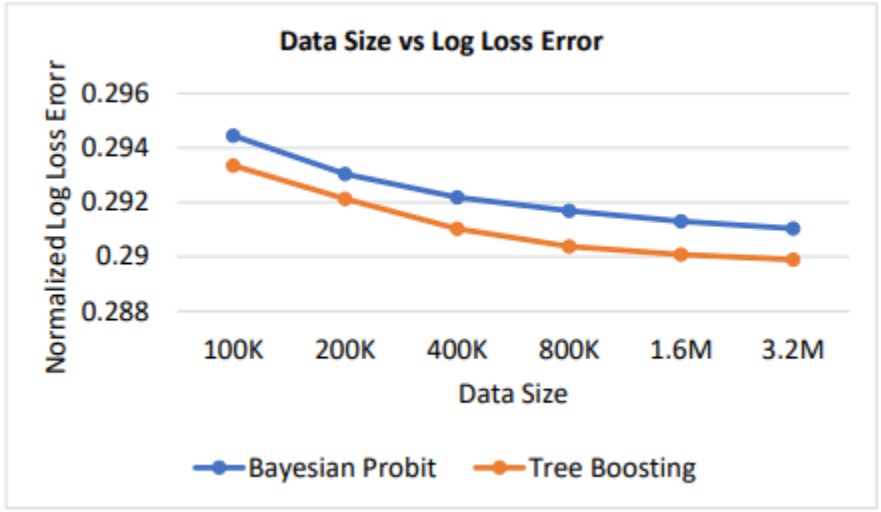
Genie JobId	AEther Url	Start Date	End Date	AB Ids	Alert Name	Alert Value	Alert Threshold
479653ea-5b9b-4194-9aed-5ae1d856c483	Genie_Default__166	2018-09-16	2018-09-16	166	ReRunError	0.078%	2%
479653ea-5b9b-4194-9aed-5ae1d856c483	Genie_Default__166	2018-09-16	2018-09-16	166	ReconstructionError	0.427%	2%
479653ea-5b9b-4194-9aed-5ae1d856c483	Genie_Default__166	2018-09-16	2018-09-16	166	SimulationError	0.007%	2%

Scope Job Performance Metrics

Genie JobId	Job Name	Scope Job Id	Job Details		
479653ea-5b9b-4194-9aed-5ae1d856c483	Genie_Default__166	9ed1d63b-c942-4286-9383-01a17bbbcb25	AuctionSimulationProcessor	Average Execution Time per vertex	00.00:03:01
				Total Vertices	2000

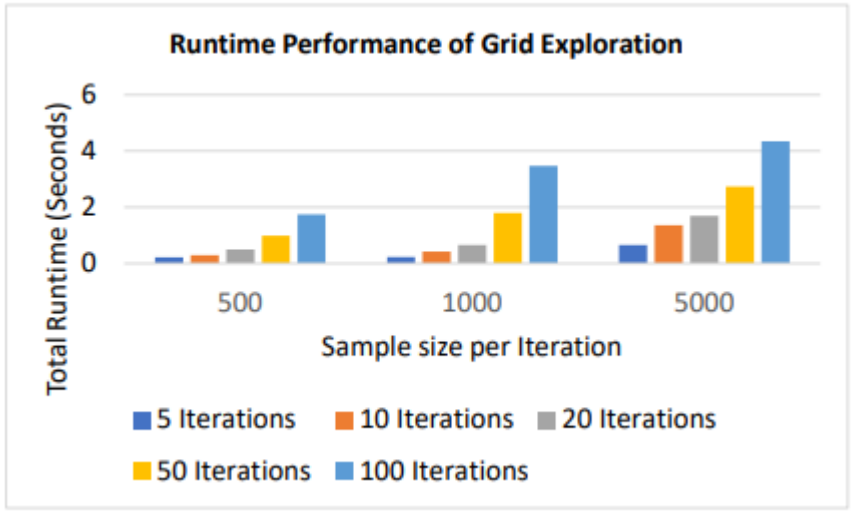
Click Prediction Performance

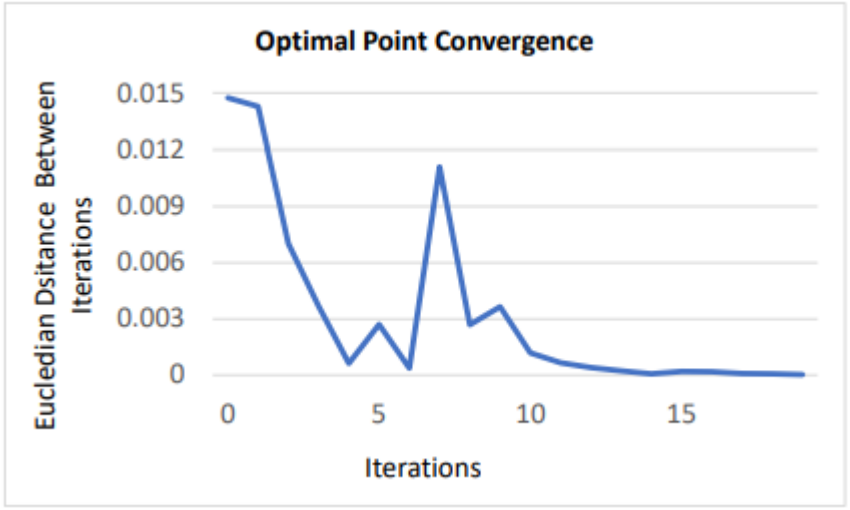




- Sampled from January 2018 Bing PC Traffic (Homogenous)
- Tree-Boosting requires separate training job and 10 min -> 2 hours of shipping cost back to cosmos cluster. Only preferred for certain cases of mixed traffic for now

Grid Exploration Performance





- From Week of January 2018 Bing PC Traffic
- Linear run time as sample size increases
- Converges after < 15 iterations. This is true for many other tuning scenarios.

Comparison with Importance Sampling

$$Y = \int_x yP(x) \approx \frac{1}{N}\sum_{i=1}^{N} y_i \qquad Y^* = \int_x y\frac{P^*(x)}{P(x)}P(x) \approx \frac{1}{N}\sum_{i=1}^{N}\frac{P^*(x_i)}{P(x_i)}y_i = \frac{1}{N}\sum_{i=1}^{N} w(x_i)y_i$$

- P(X) is a distribution for parameter set X. Each request has realized parameters randomly drawn from P(X).
- The counterfactual KPI Y* could be estimated for a given P*(X).
- Genie recently adapted separate production workflow for implementation of [3] below.

Method	RPM	MLIY	CY	CPC
IS (Historical)	1.27%	0.41%	0.39%	1.14%
Genie (Historical)	**1.16%**	**0.32%**	**0.37%**	**0.93%**
IS (Regression)	0.90%	0.36%	**0.24%**	0.98%
Genie (Regression)	**0.88%**	**0.25%**	0.27%	**0.66%**

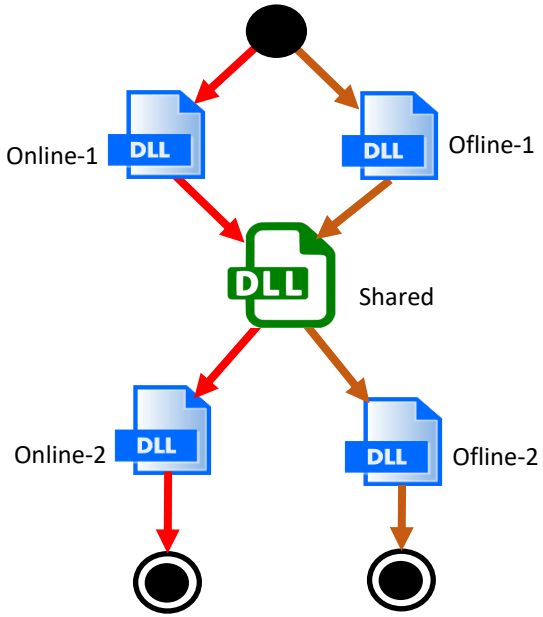
- Bing PC Experiment on 5 consecutive tuning time period during April to May 2018.
- Each cell corresponds to absolute delta compared to A/B testing.

[3] Léon Bottou, Jonas Peters, Joaquin Quiñonero Candela, Denis Xavier Charles, Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Y. Simard, Ed Snelson: Counterfactual reasoning and learning systems: the example of computational advertising. Journal of Machine Learning Research 14(1): 3207-3260 (2013)

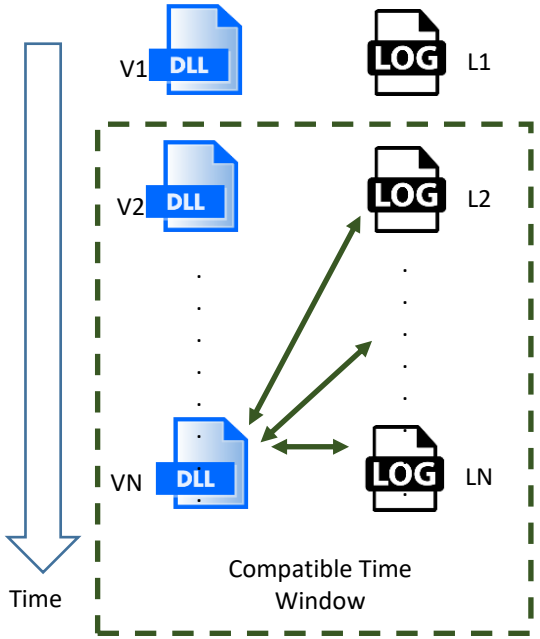
Outline

- ❑ Sponsored Search Optimization
- ❑ Genie Open Box Counterfactual Policy Estimator
- ❑ Experimental Results
- ❑ Challenges and Lessons Learned

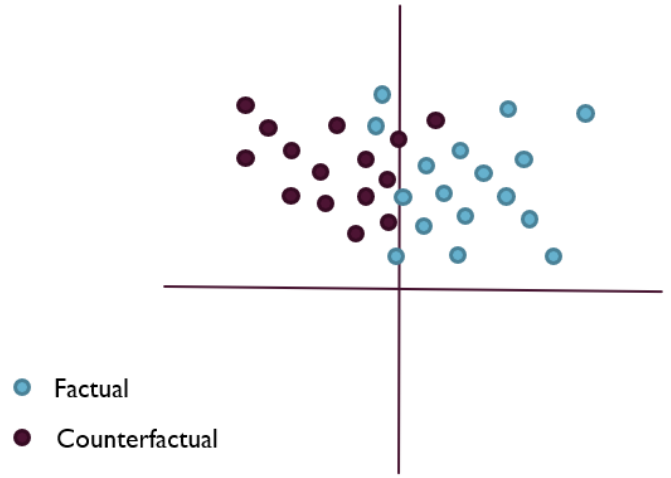
Challenges and Lessons Learned



Online vs Offline Parity



Backward Compatibility



Factual vs Counterfactual Feature Distribution

End of Presentation ☺

Questions?