LinkedIn

Rise of the Machines: Removing the Human-in-the-loop

Aug 12, 2020

Viral Gupta

Tech Lead, Comms AI

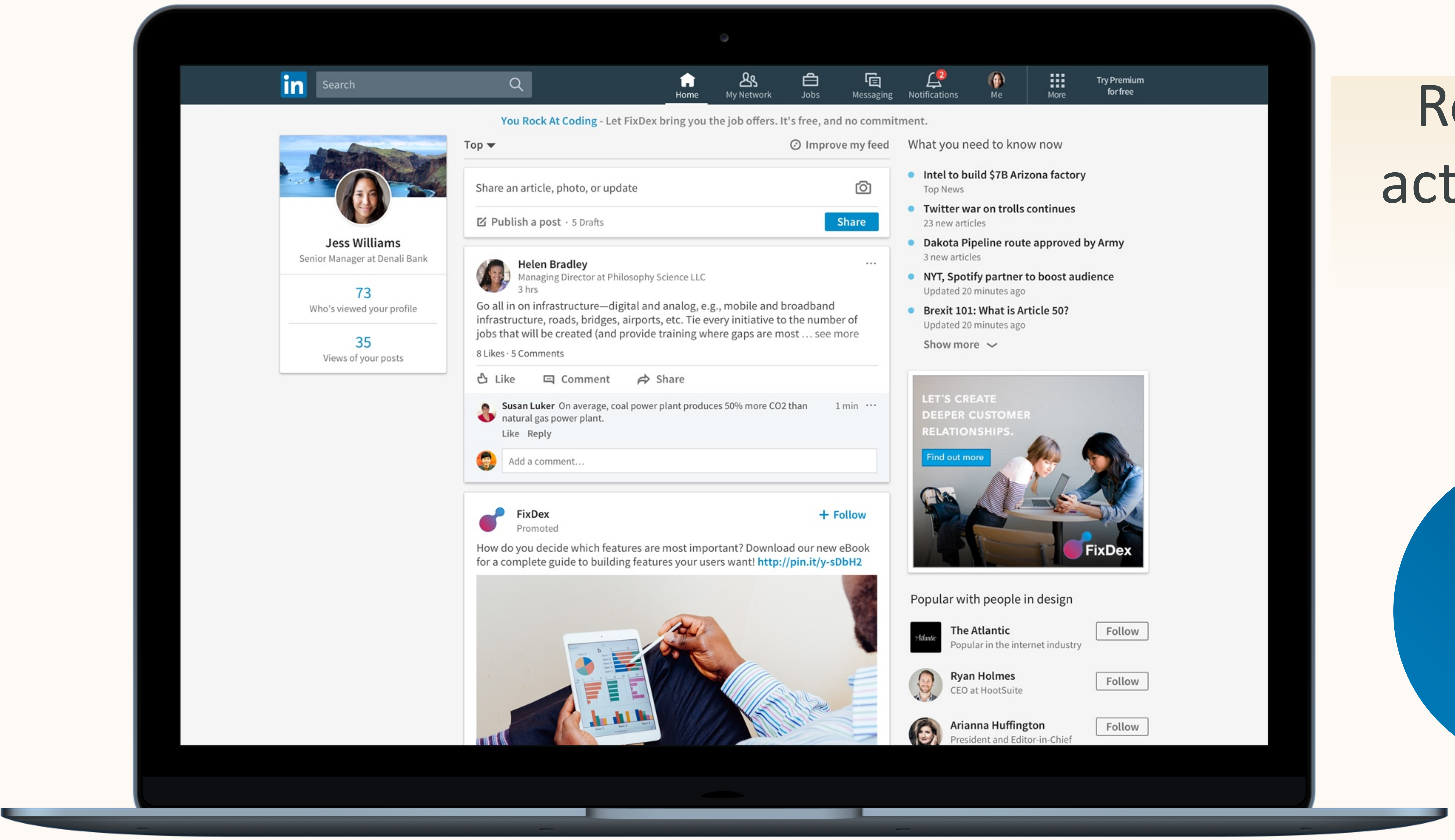
Yunbo Ouyang

Sr. Engineer, AI Foundation

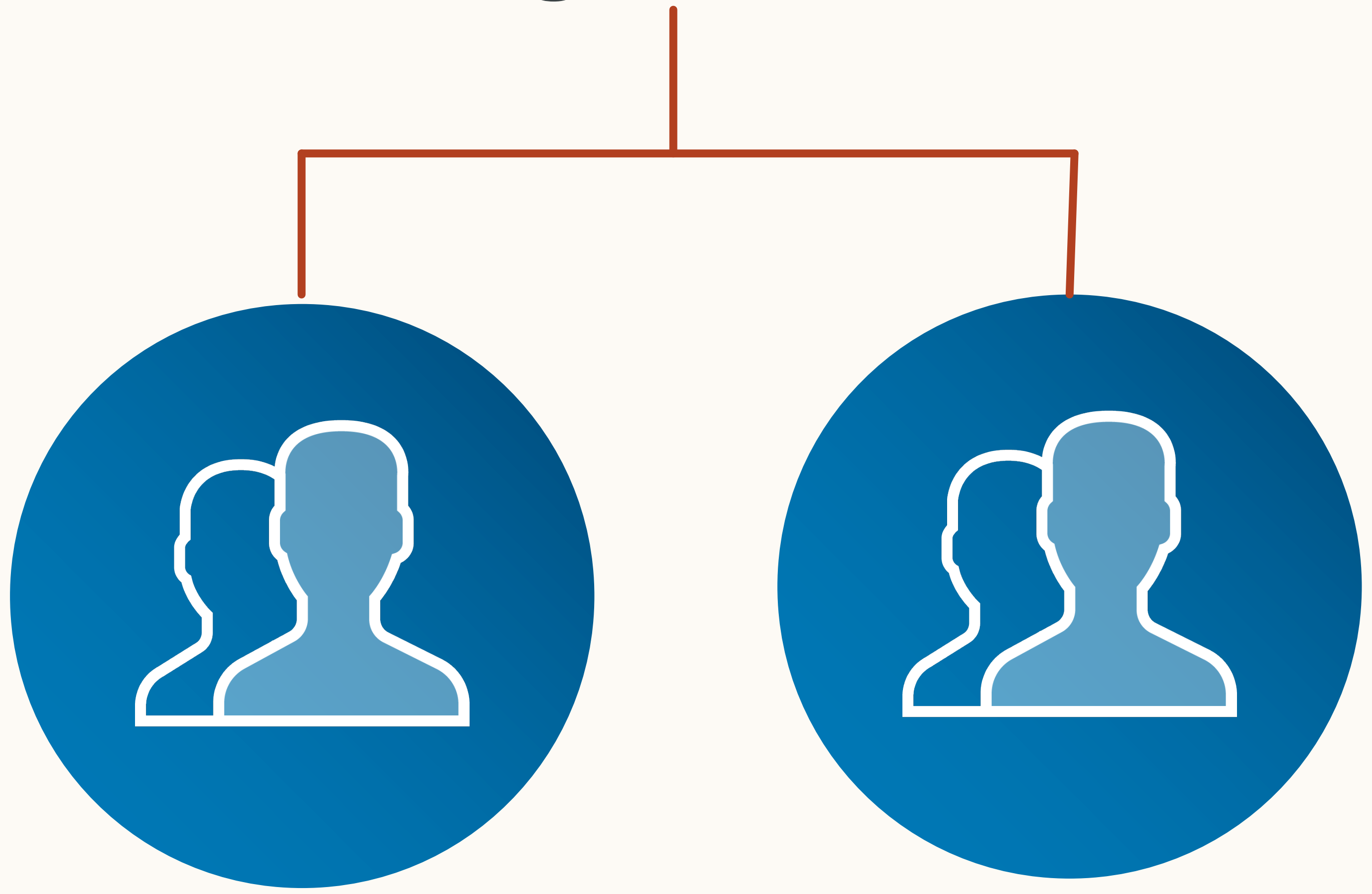
Agenda

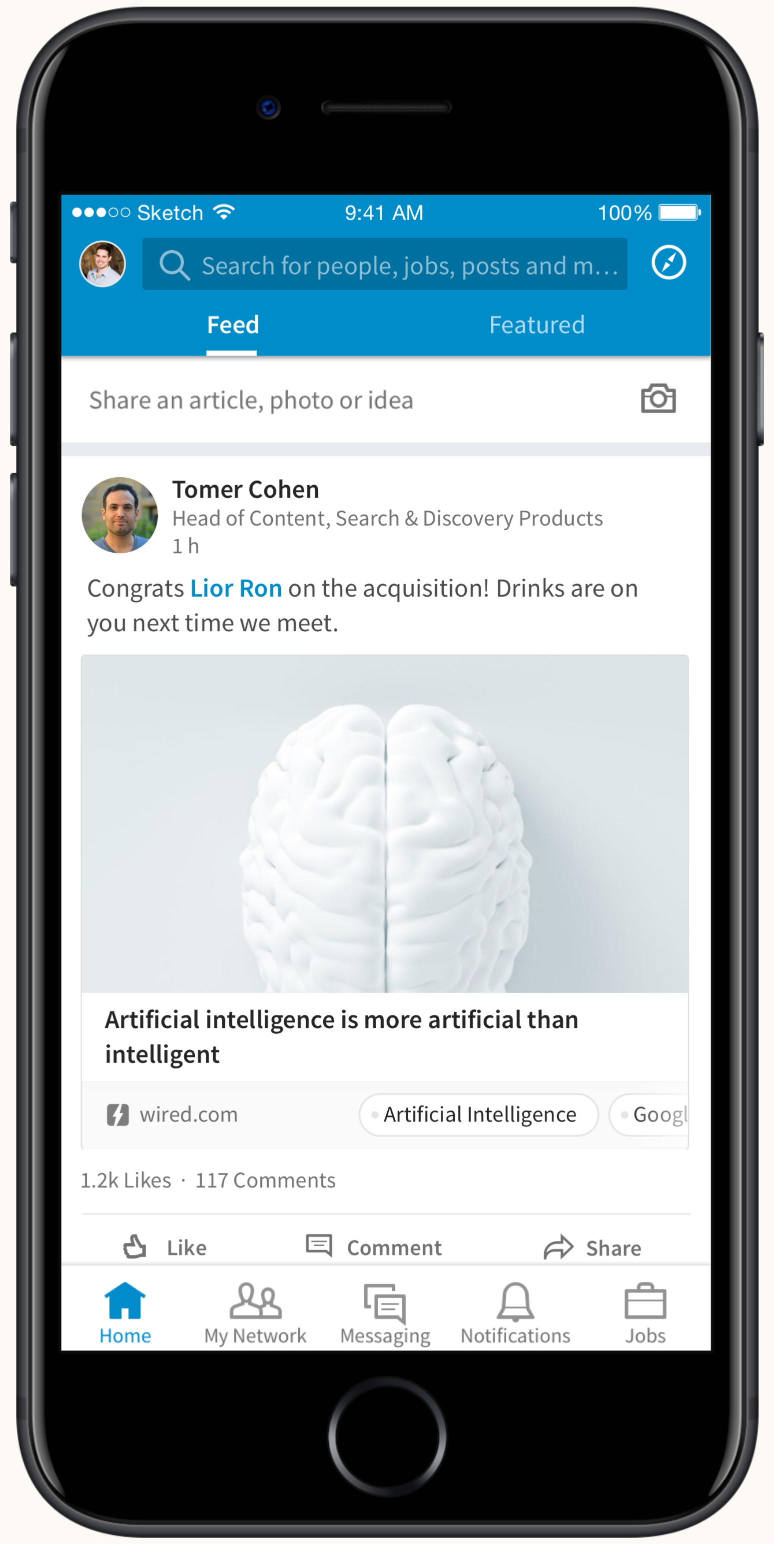
1. Problem Setup
   LinkedIn Notifications
2. Reformulation as a Black-Box Optimization
3. Explore-Exploit Algorithm
   Thompson Sampling
4. Infrastructure
5. Results: Notification

LinkedIn Connects the World's Professionals



Remain updated about the activities of their connections through newsfeed



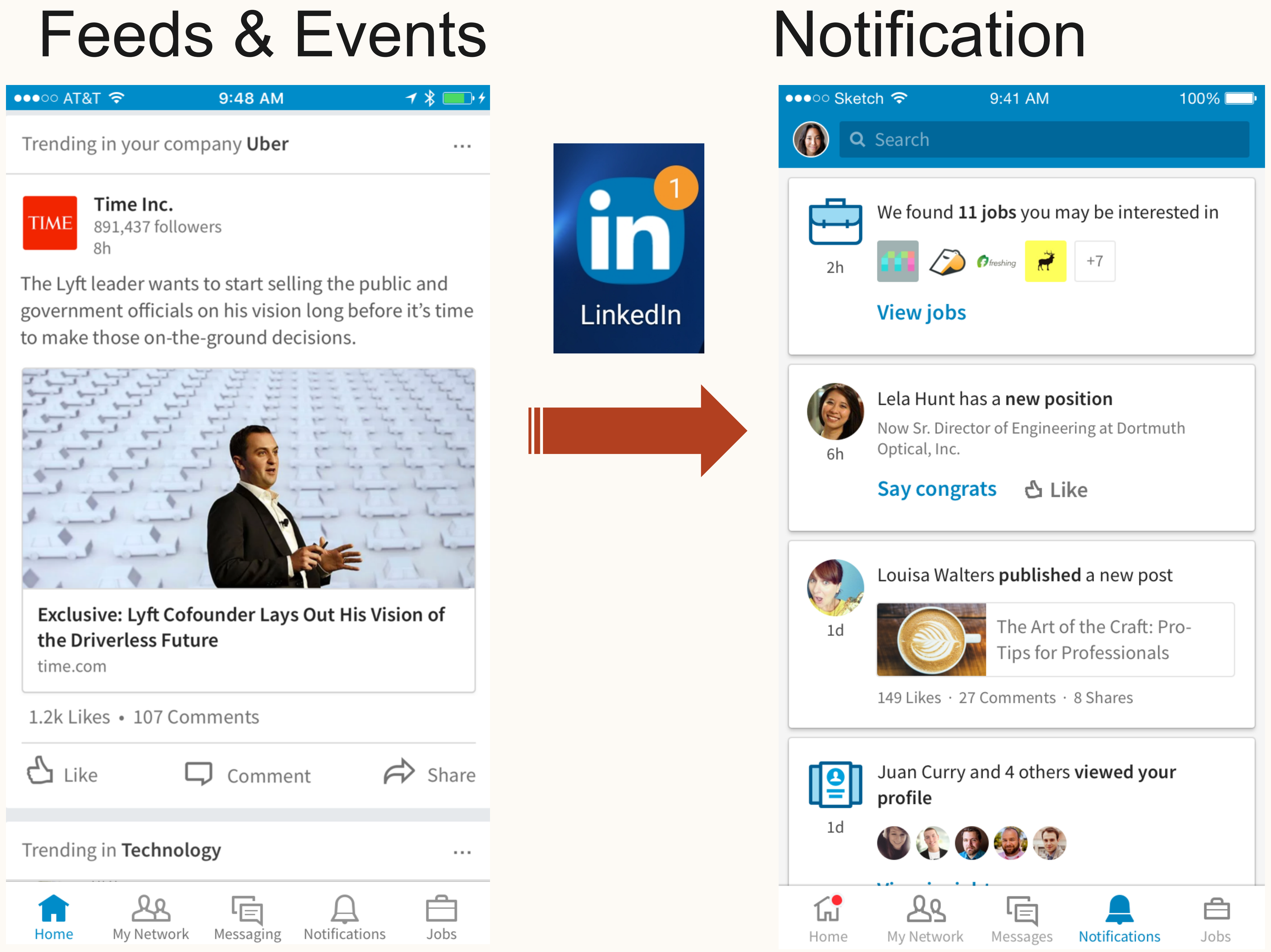


LinkedIn

Activity Based Notifications

Non-transactional messages, time-sensitive content

Goal: drive member engagement while creating delightful experiences



Mobile App Uses Notifications to Inform

Important Metrics

Sessions

Sessions where a member engaged on the platform.

Clicks Actions

Members clicks, liked, shared or commented on an item.

Send Volume

Notifications send to the members.

Ranking Function

- m – Member, u - Item

$$S(m,u) \coloneqq P_{Click}(m,u) + \; x_{\alpha}\, P_{Visit}(m,u) > x_{th}$$

- The weight vector $x = (x_{\alpha}, x_{th})$ controls the balance between the business metrics: Sessions, Clicks, Send Volume.
- A Sample Business Strategy is

$$\begin{aligned} &Maximize. \quad && Sessions(x) \\ &s.t. && Clicks(x) > c_{Clicks} \\ & && Send\ Volume(x) < \; c_{Send\ Volume} \end{aligned}$$

Major Challenges

- The optimal value of x (tuning parameters) changes over time
- Example of changes
    - New content types are added
    - Score distribution changes (Feature drift, updated models, etc.)

- With every change engineers would manually find the optimal x
    - Run multiple A/B tests
- Not the best use of engineering time

Reformulation into a Black-Box Optimization Problem

Modeling The Metrics

- $Y_{i,j}^{k}(x) \in \{0,1\}$ denotes if the i-th member during the j-th notification which was served by parameter x, did action k or not. Here k = Session, Click.

- We model this data as follows

$$\sum_{i}\sum_{j} Y_{i,j}(x) \sim \text{Gaussian}\left(f(x), \sigma^2\right)$$

- Assume a Gaussian process prior on each of the latent function f_k .

$$f_k(x) \sim \text{N}\left(0, K_{RBF}(x,x)\right)$$

Reformulation

We approximate each of the metrics as:

$$Sessions(x) = f_{Sess}(x)$$
$$Clicks(x) = f_{Clicks}(x)$$
$$Send\ Volume(x) = f_{S.V}\ (x)$$

The original optimization problem can be written through this parametrization.

$$\begin{aligned} &Maximize \quad f_{Sess}(x) \\ &s.t. \quad f_{Clicks}(x) > c_{Clicks} \\ &\qquad\quad f_{SV}(x) < c_{SV} \end{aligned}$$

→

$$\begin{aligned} &Maximize\ \ f(x) + \lambda_1(c_{Clicks} - f_{Clicks}(x)) - \lambda_2(f_{SV} - c_{sv}) \\ &\qquad x \in X \end{aligned}$$

Benefit: The last problem can now be solved using techniques from the literature of Bayesian Optimization.

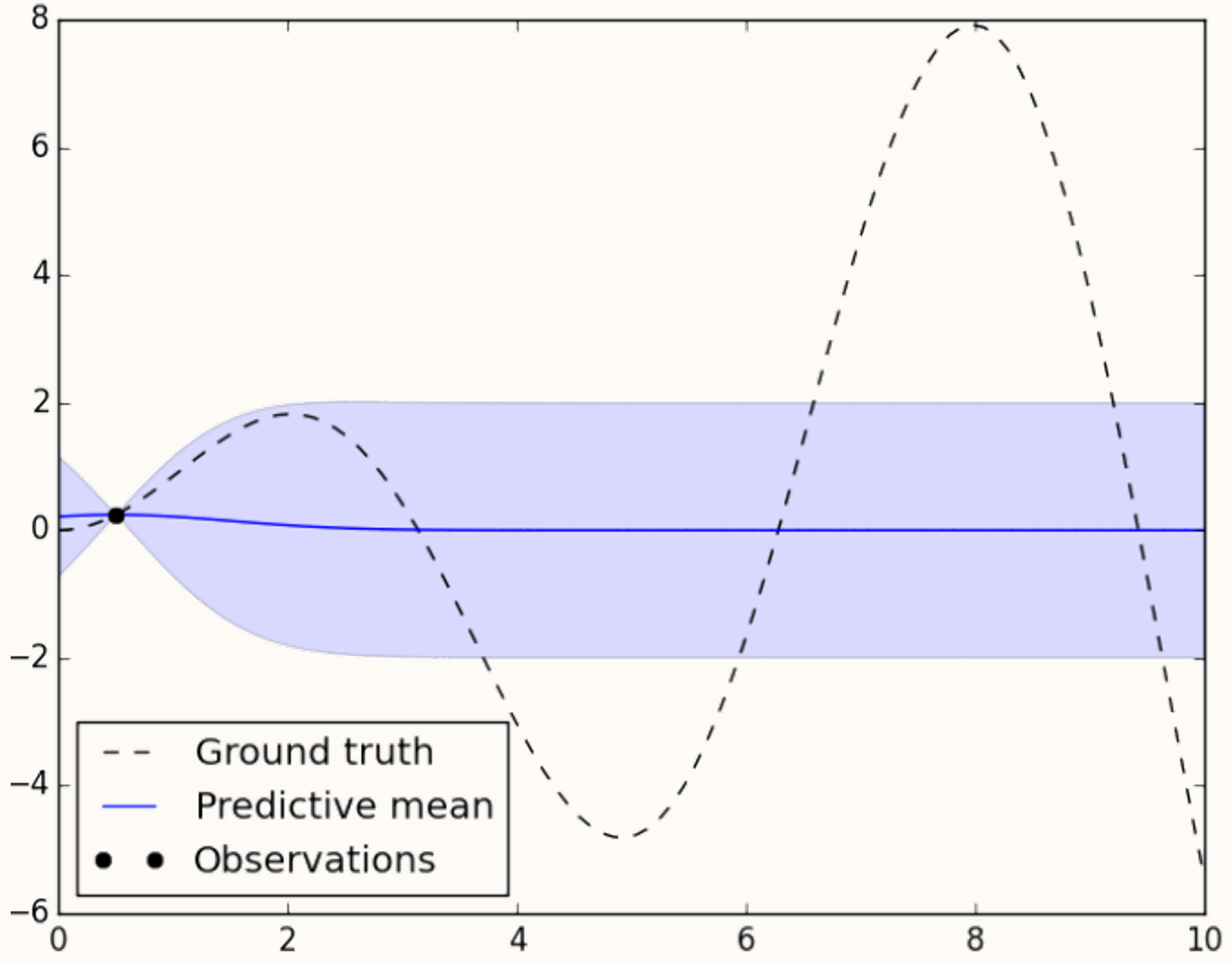
Explore-Exploit Algorithms

Bayesian Optimization

A Quick Crash Course

- Explore-Exploit scheme to solve

$$\underset{x \in X}{Maximize} \quad f(x)$$



Bayesian Optimization

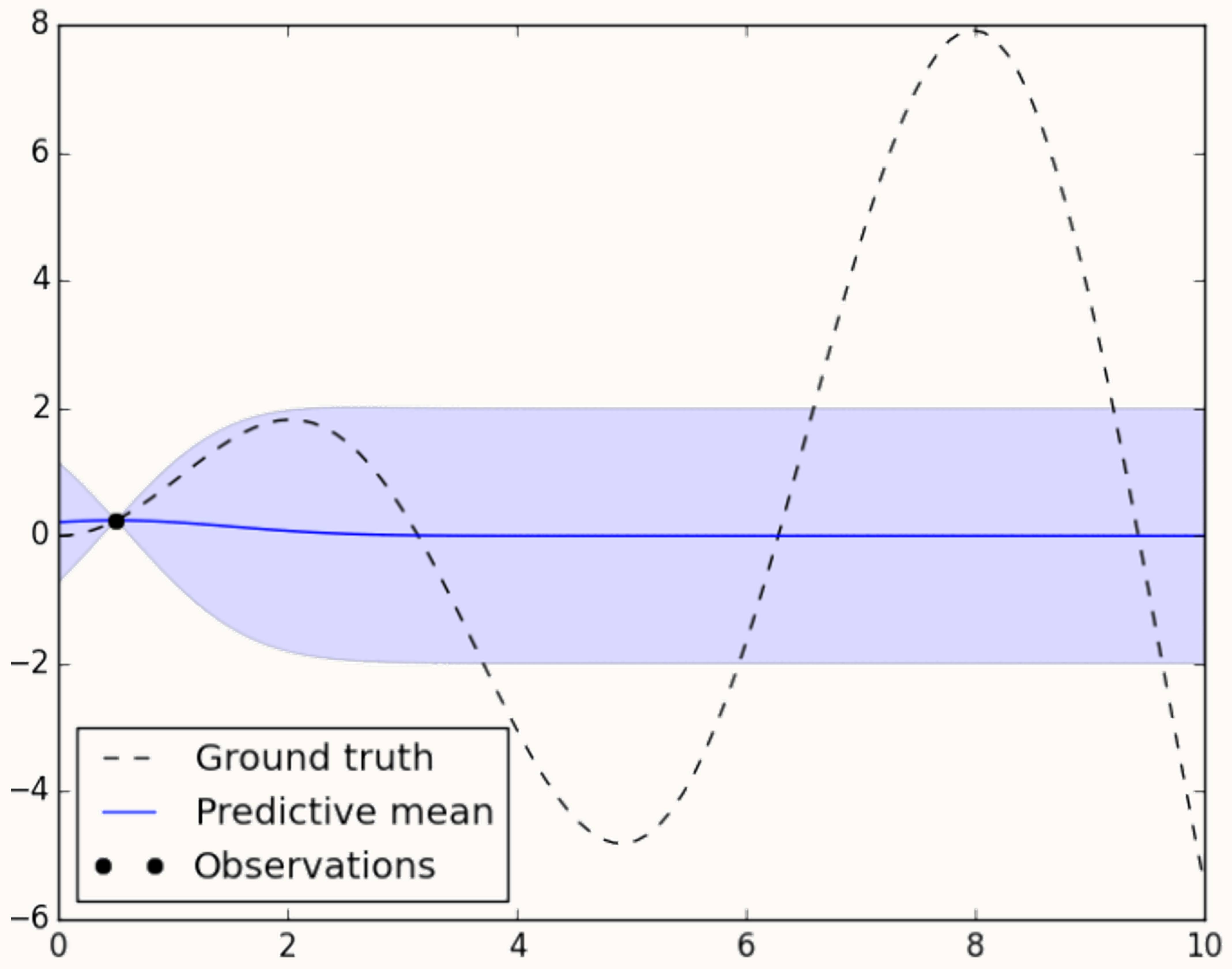
A Quick Crash Course

- Explore-Exploit scheme to solve

$$\underset{x \in X}{Maximize} \quad f(x)$$

- Assume a Gaussian Process prior on $f(x)$.

- Start with uniform sample get $(x, f(x))$
- Estimate the mean function and covariance kernel



Bayesian Optimization

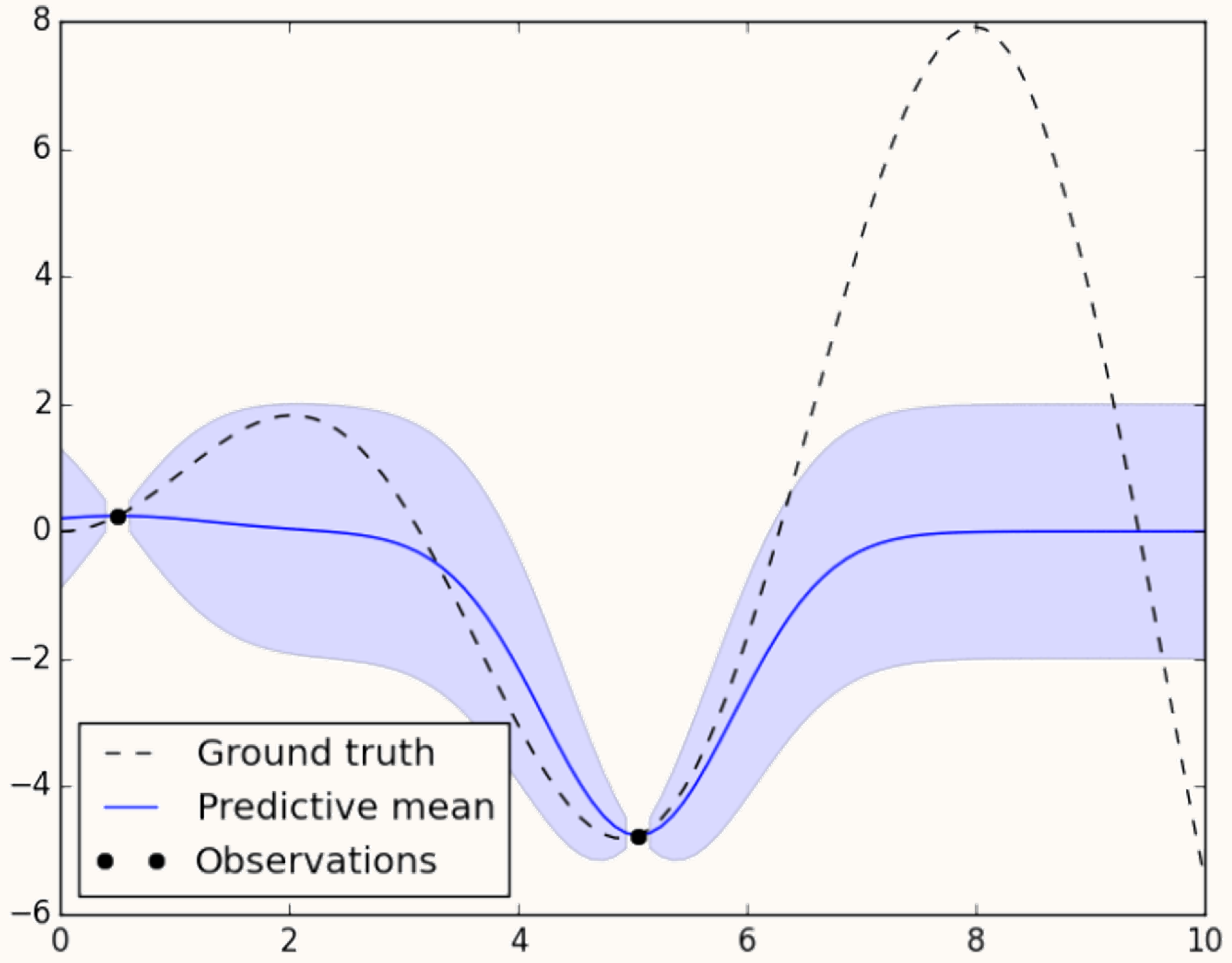
A Quick Crash Course

- Explore-Exploit scheme to solve

$$\underset{x \in X}{Maximize} \quad f(x)$$

- Assume a Gaussian Process prior on $f(x)$.

- Start with uniform sample get $(x, f(x))$
- Estimate the mean function and covariance kernel
- Draw the next sample x which maximizes an "*acquisition function*" or predictive posterior.
- Continue the process.



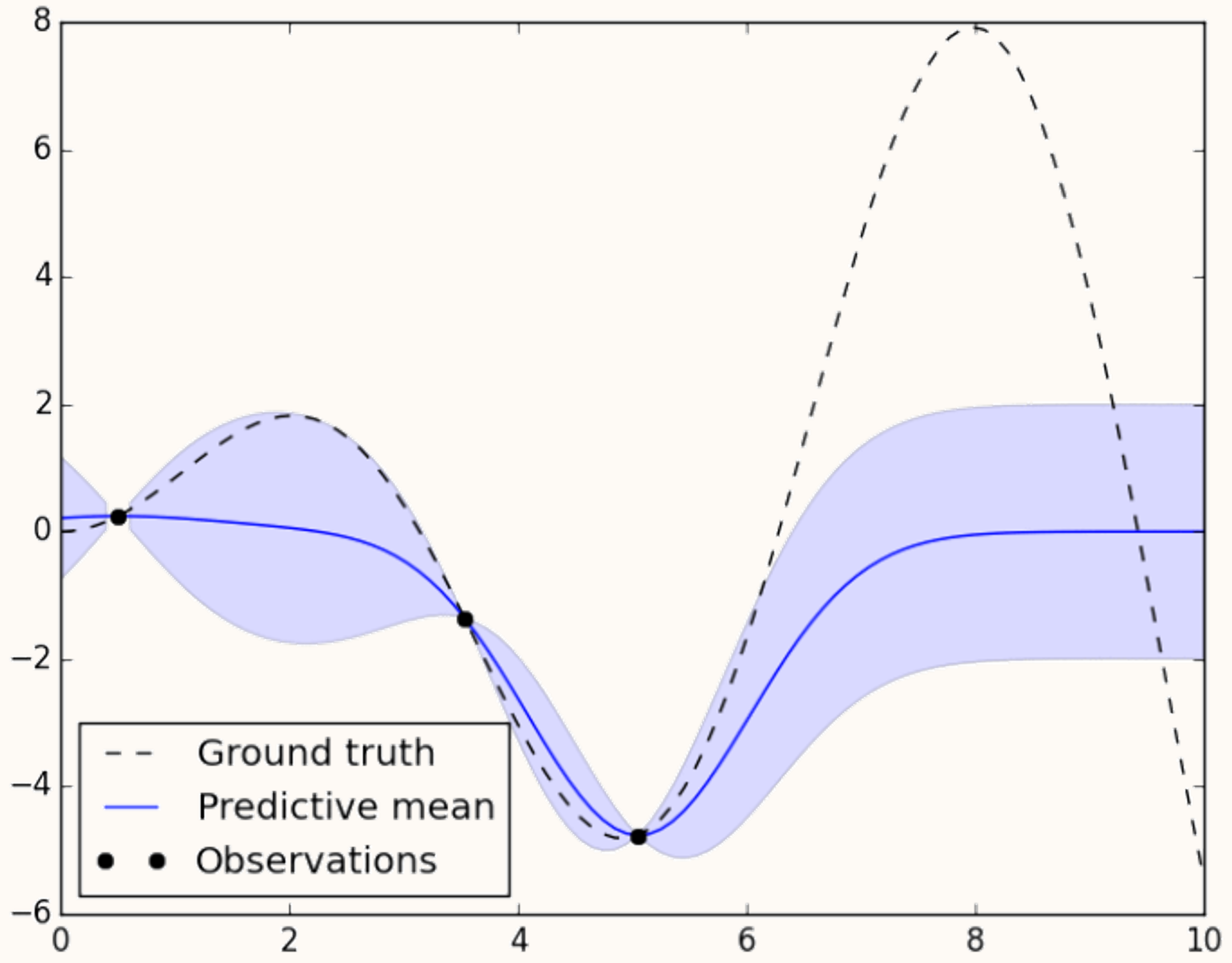
Bayesian Optimization

A Quick Crash Course

- Explore-Exploit scheme to solve

$$\underset{x \in X}{Maximize} \quad f(x)$$

- Assume a Gaussian Process prior on $f(x)$.
- Start with uniform sample get $(x, f(x))$
- Estimate the mean function and covariance kernel
- Draw the next sample x which maximizes an "*acquisition function*" or predictive posterior.
- Continue the process.



Bayesian Optimization

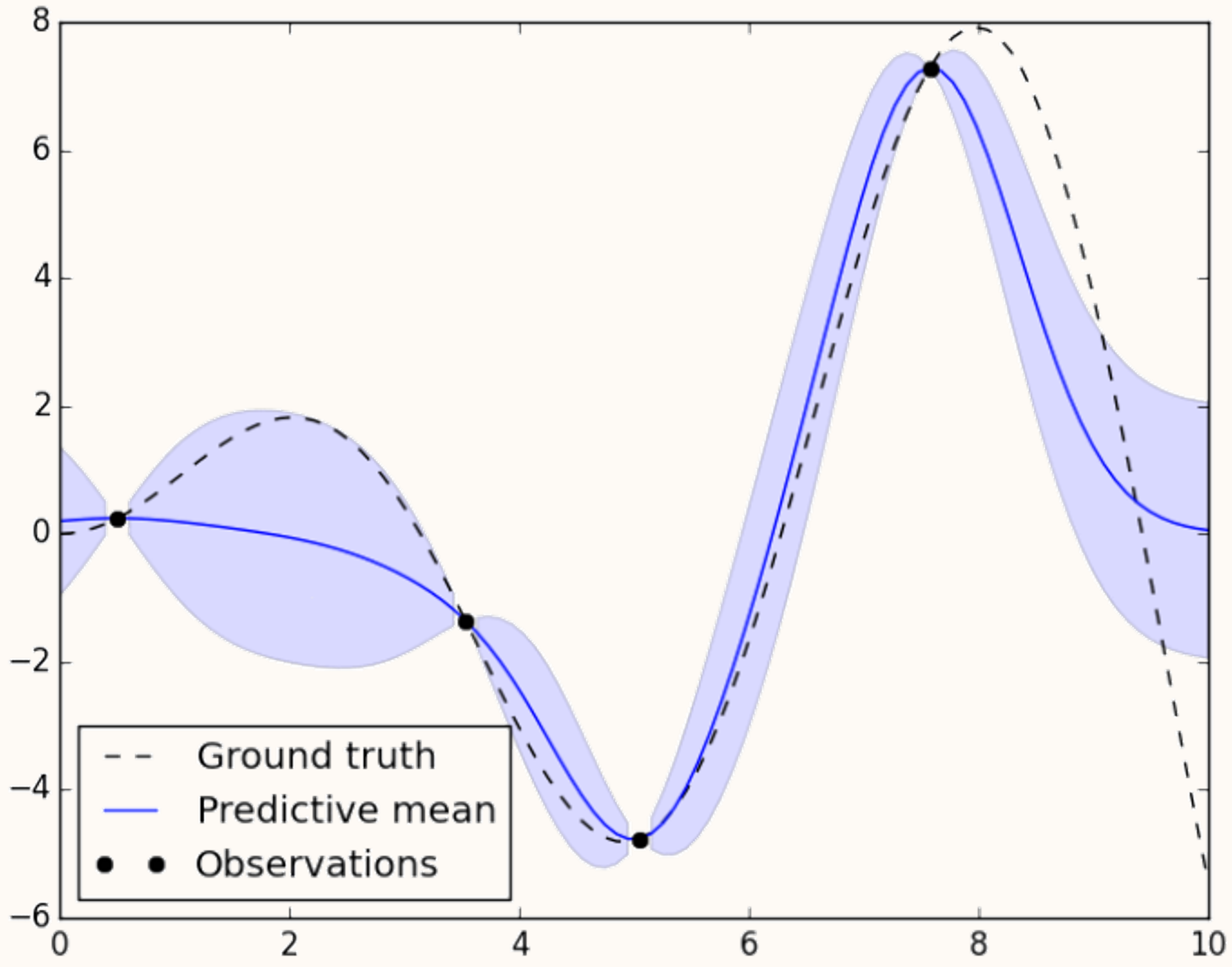
A Quick Crash Course

- Explore-Exploit scheme to solve

$$\underset{x \in X}{Maximize} \quad f(x)$$

- Assume a Gaussian Process prior on $f(x)$.

- Start with uniform sample get $(x, f(x))$
- Estimate the mean function and covariance kernel
- Draw the next sample x which maximizes an "*acquisition function*" or predictive posterior.
- Continue the process.



Bayesian Optimization

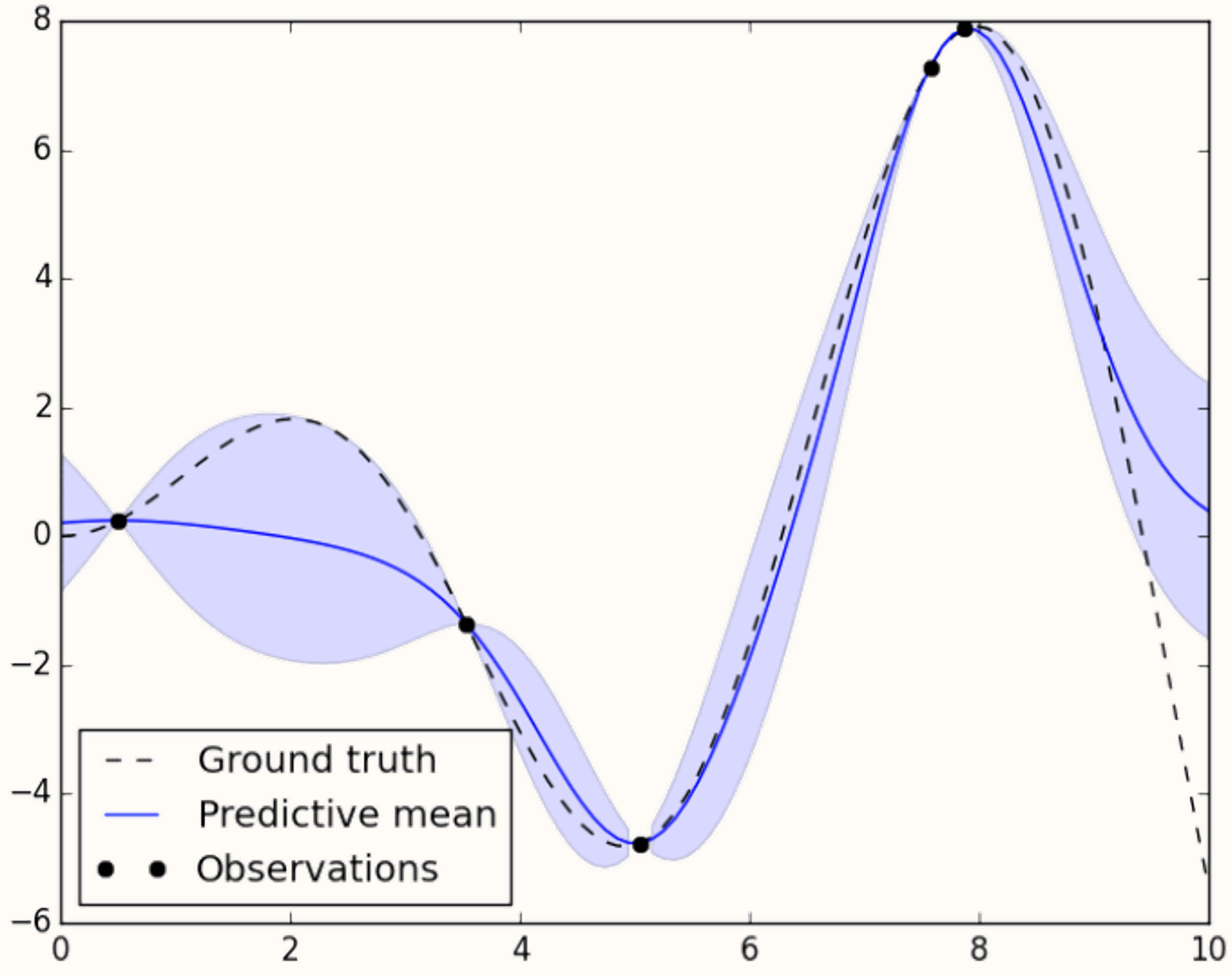
A Quick Crash Course

- Explore-Exploit scheme to solve

$$\underset{x \in X}{Maximize} \quad f(x)$$

- Assume a Gaussian Process prior on $f(x)$.

- Start with uniform sample get $(x, f(x))$
- Estimate the mean function and covariance kernel
- Draw the next sample x which maximizes an "*acquisition function*" or predictive posterior.
- Continue the process.



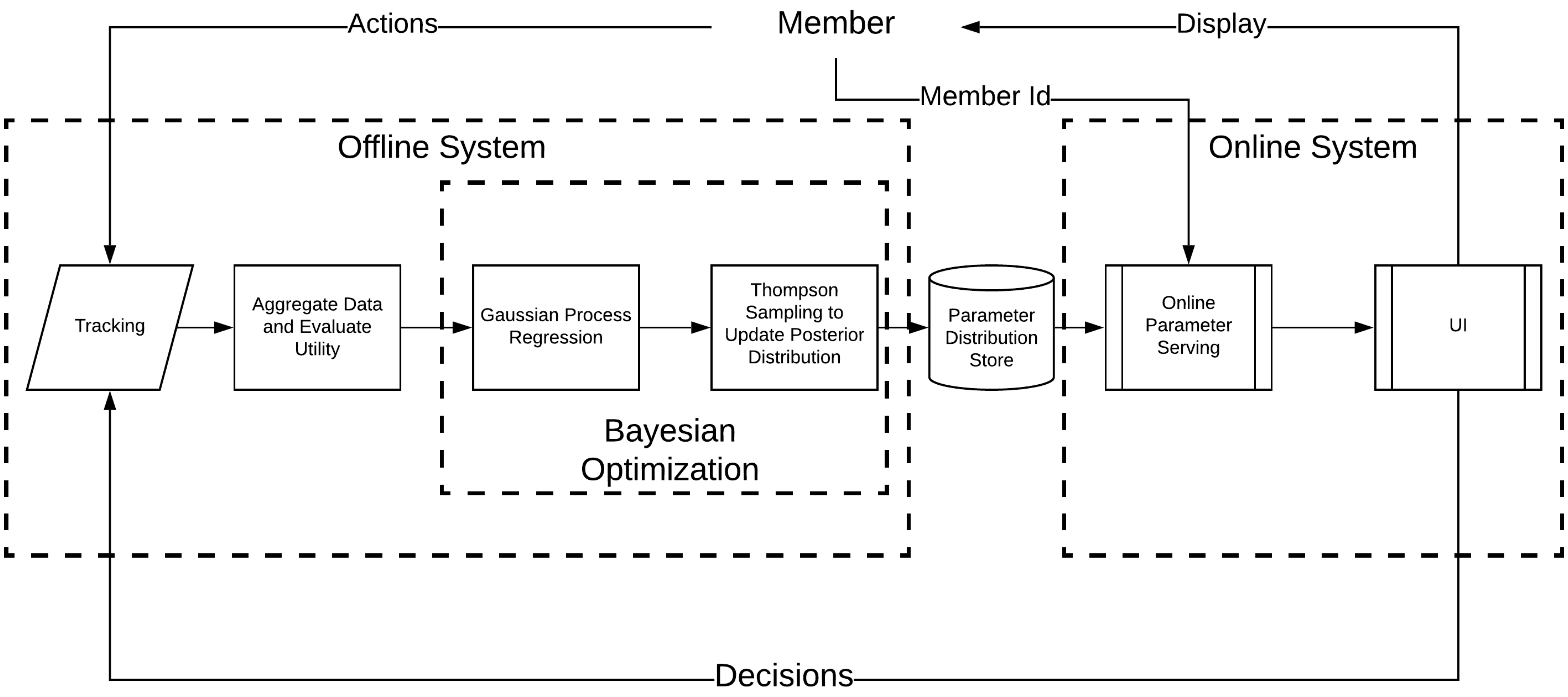
Thompson Sampling

- Consider a Gaussian Process Prior on each f_k, where k is Sessions, Clicks or Send Volume
- Observe the data $(x, f_k(x))$
- Obtain the posterior of each $f_k(x)$
- Sample from the posterior distribution
- Compute the Lagrangian for the overall objective function.
- Get the next distribution of hyperparameters by computing the probability of each hyper-parameter to be optimal.
- Continue this process till convergence.

$$
\begin{aligned}
&Maximize \quad f_{Sess}(x) \\
&s.t. \quad f_{Clicks}(x) > c_{Clicks} \\
&\qquad f_{SV}(x) < c_{SV}
\end{aligned}
$$

Infrastructure

System Architecture Overview



Offline System

The heart of the product

Tracking

- All member activities are tracked with the parameter of interest.
- ETL into HDFS for easy consumption.

Utility Evaluation

- Using the tracking data we generate $(x, f_k(x))$ for each function k.
- The data is kept in appropriate schema that is problem agnostic.

Bayesian Optimization

- The data and the problem specifications are input to this module.
- Using the data, we first estimate each of the posterior distributions of the latent functions using Gaussian Process Regression.
- Sample from those distributions to get distribution of the parameter x which maximizes the objective.

The Parameter Store and Online Serving

- Bayesian Optimization library generates
  - A set of potential parameters for serving in the next round $(x_1, x_2, \ldots, x_n)$
  - Serving probability $(p_1, p_2, \ldots, p_n)$ of each parameter such that $\sum_{i=1}^{n} p_i = 1$
- To determine the serving parameter for each member, first this member's id is mapped to [0,1] using a hashing function $\boldsymbol{h}$. If

$$\sum_{i=1}^{k} p_i < h(Id) \leq \sum_{i=1}^{k+1} p_i$$

  Then Notifications are served with parameter x_{k+1}

- The parameter store (depending on use cases) can contain
  - `<parameterValue, probability>` i.e. (x_i, p_i) or
  - `<memberId, parameterValue>`

Online System

Serving hundreds of millions of users

Parameter Sampling

- For each member m visiting LinkedIn,
- Depending on the parameter store, we either evaluate `<m, parameterValue>`
- Or we directly call the store to retrieve `<m,parameterValue>`

Online Serving

- Depending on the parameter value that is retrieved (say x), the member's notifications are scored according to the ranking function and served

$$S(m,u) \coloneqq P_{Click}(m,u) + \ x_{\alpha}\ P_{Visit}(m,u) > x_{th}$$

Practical Design Considerations

- Consistency in user experience.
  - Randomize at member level instead of session level.
- Offline Flow Frequency
  - Batch computation where we collect data for an hour and run the offline flow each hour to update the sampling distribution.
- Assume $(f_{Sessions}, f_{Clicks}, f_{SV})$ to be Independent
  - Works well in our setup. Joint modeling might reduce variance.
- Choice of Business Constraint Thresholds.
  - Chosen to allow for a small drop.

Results: Notification

Notification Optimization Problem Revisited: Tune InApp Threshold

- Tune InApp Threshold x_{th}

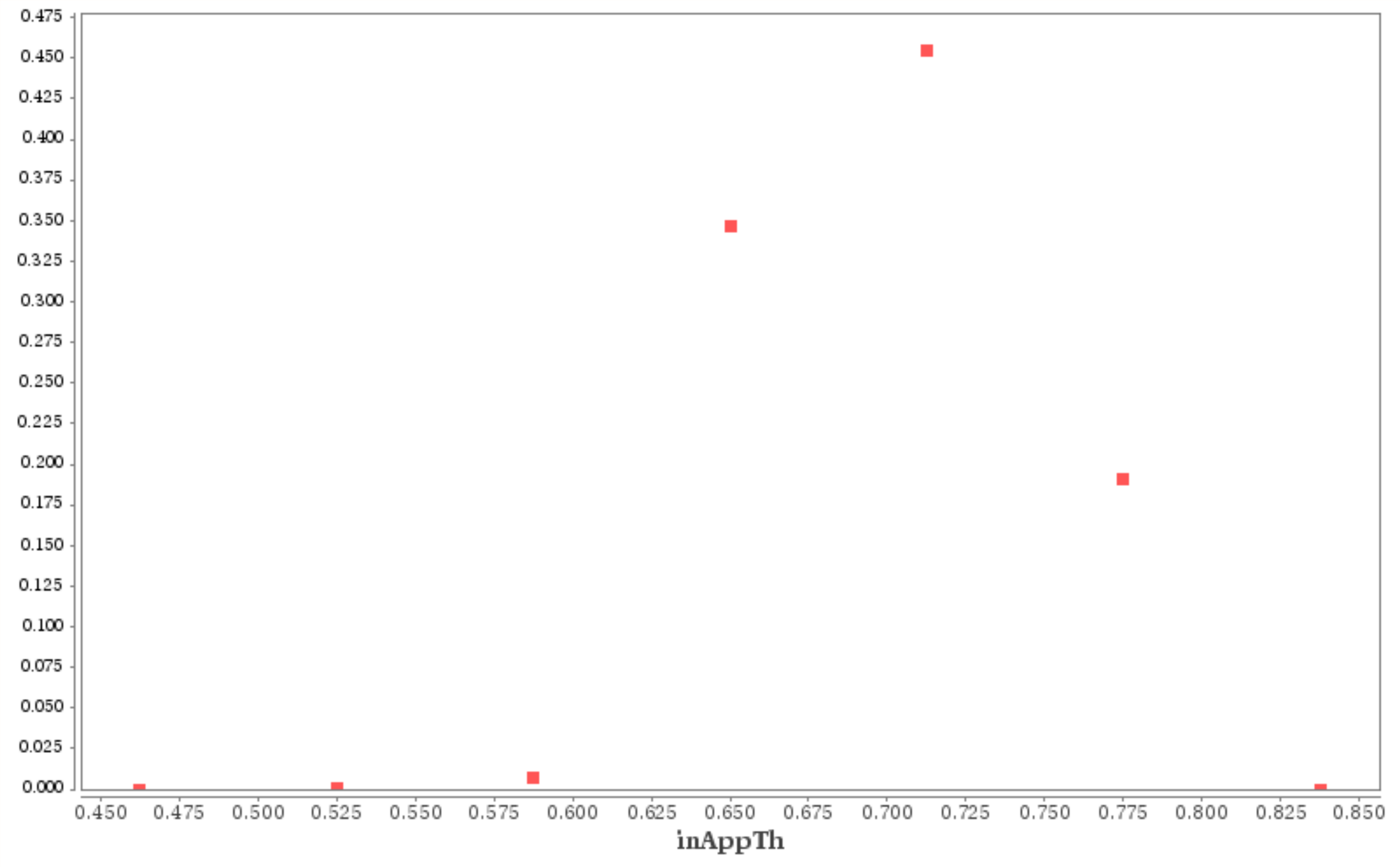
$$S(m,u) \coloneqq P_{Click}(m,u) + \ x_{\alpha}\ P_{Visit}(m,u) > x_{th}$$

- Optimization Problem

$$\begin{aligned} &Maximize. \quad && Sessions(x_{th}) \\ &s.t. && Clicks(x_{th}) > c_{Clicks} \\ & && Send\ Volume(x_{th}) < \ c_{Send\ Volume} \end{aligned}$$

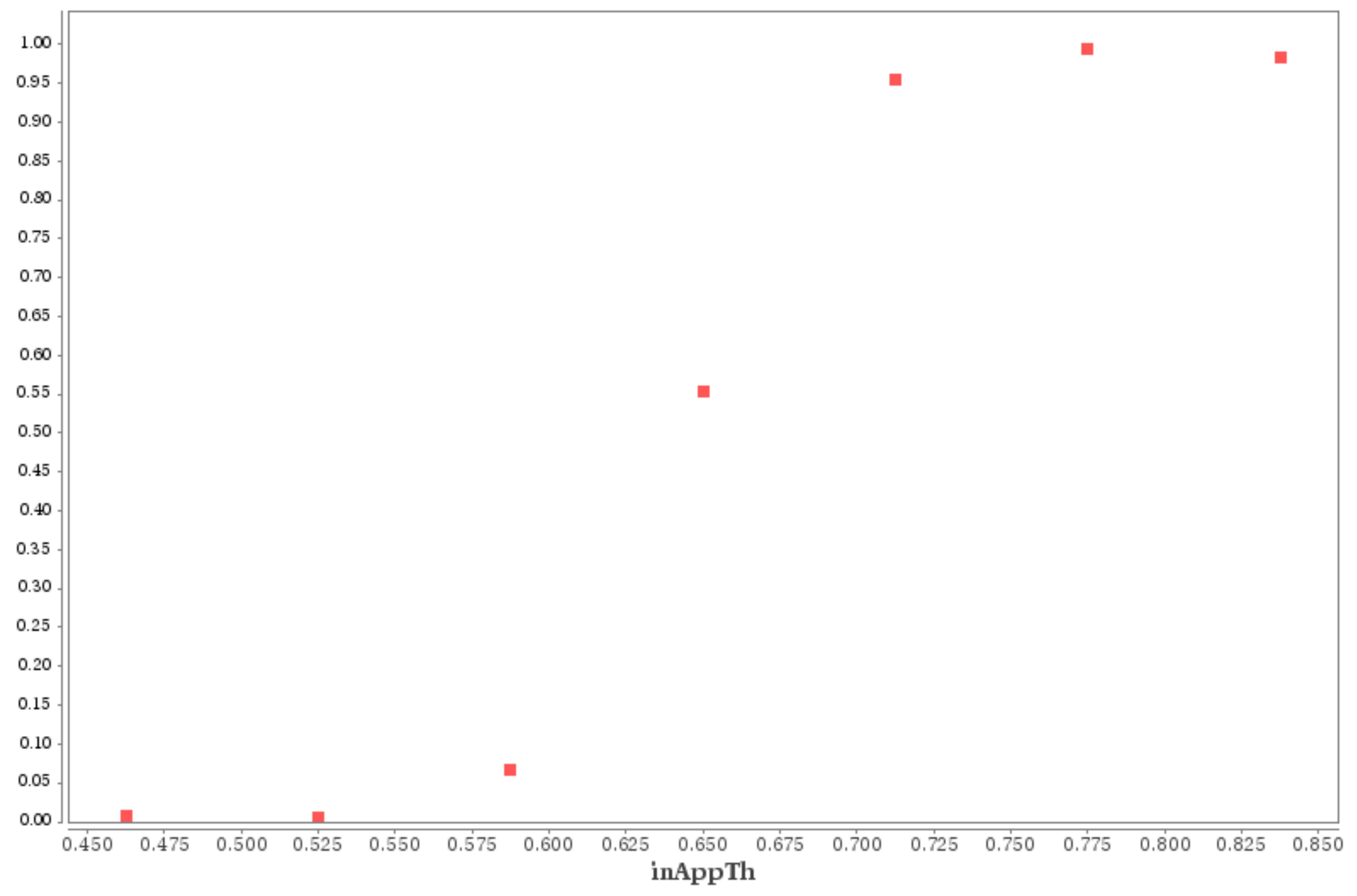
Serving Probability of InApp Threshold

- Serving probability distribution
  - The plot is different depending on whether the algorithm is in exploration stage or exploitation state.
  - Serving probability distribution is calculated via Thompson Sampling.



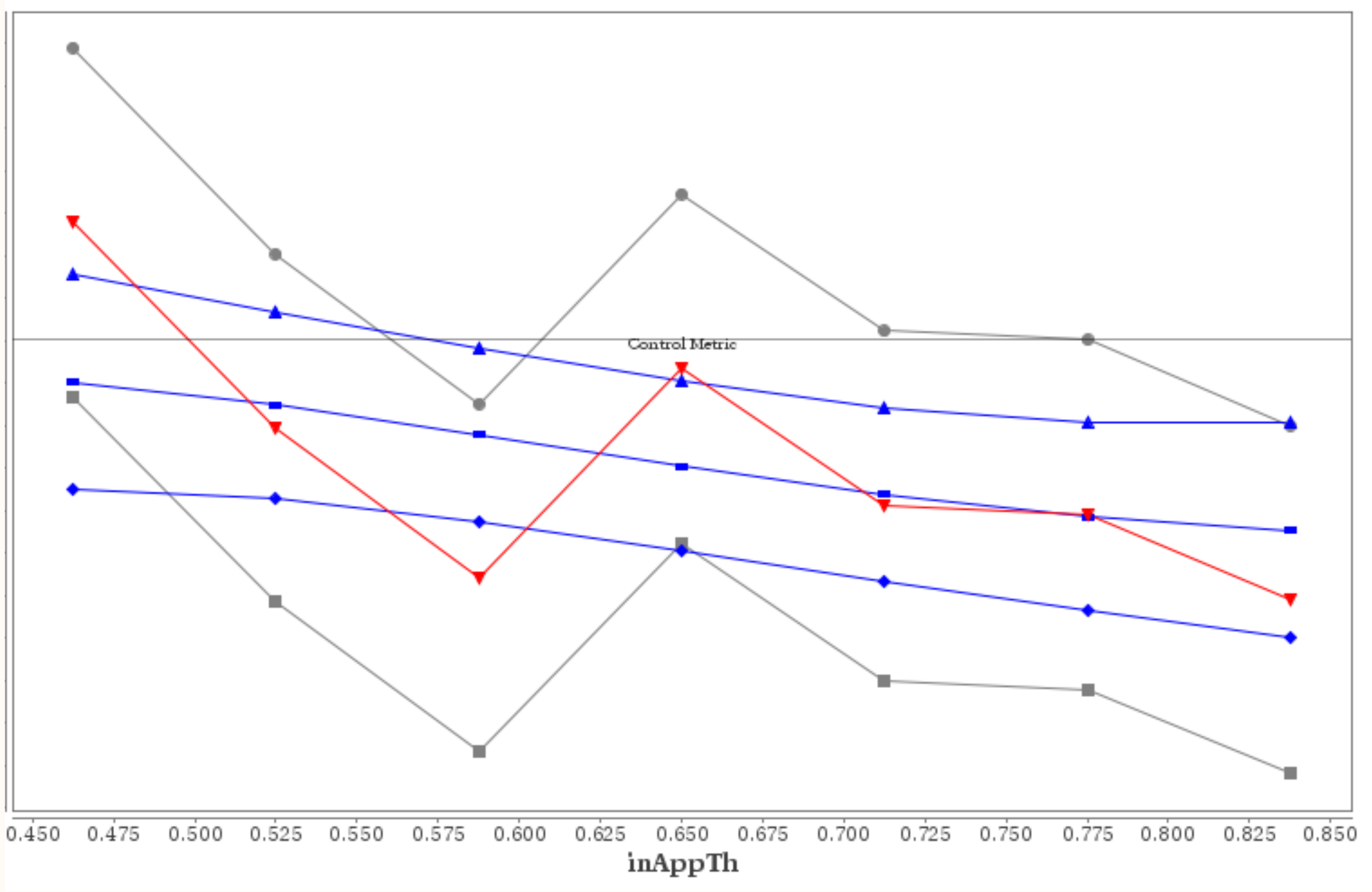
Probability of Feasibility of InApp Threshold

- Probability of constraint feasibility
  - Probability of constraint feasibility is the probability that each point satisfies all the constraints.



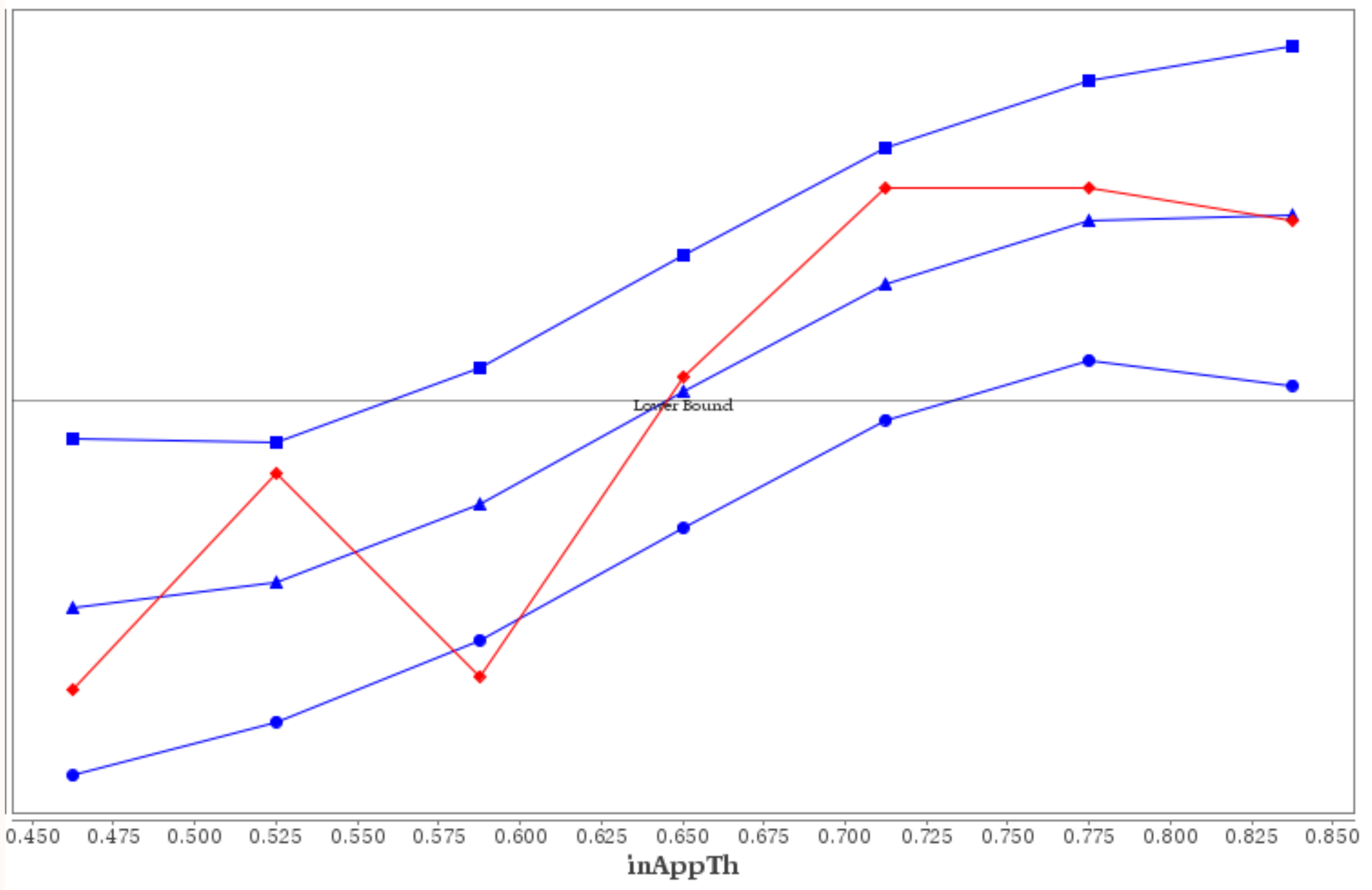
Function Fitting Plot for the Objective: Sessions vs. InApp Threshold

- Function Fitting
  - The red curve refers to observed metrics with optional grey lower and upper confidence bands
  - The blue curve refers to fitted metrics with lower and upper confidence bands
  - The horizontal line for the objective refers to the metric for the control model



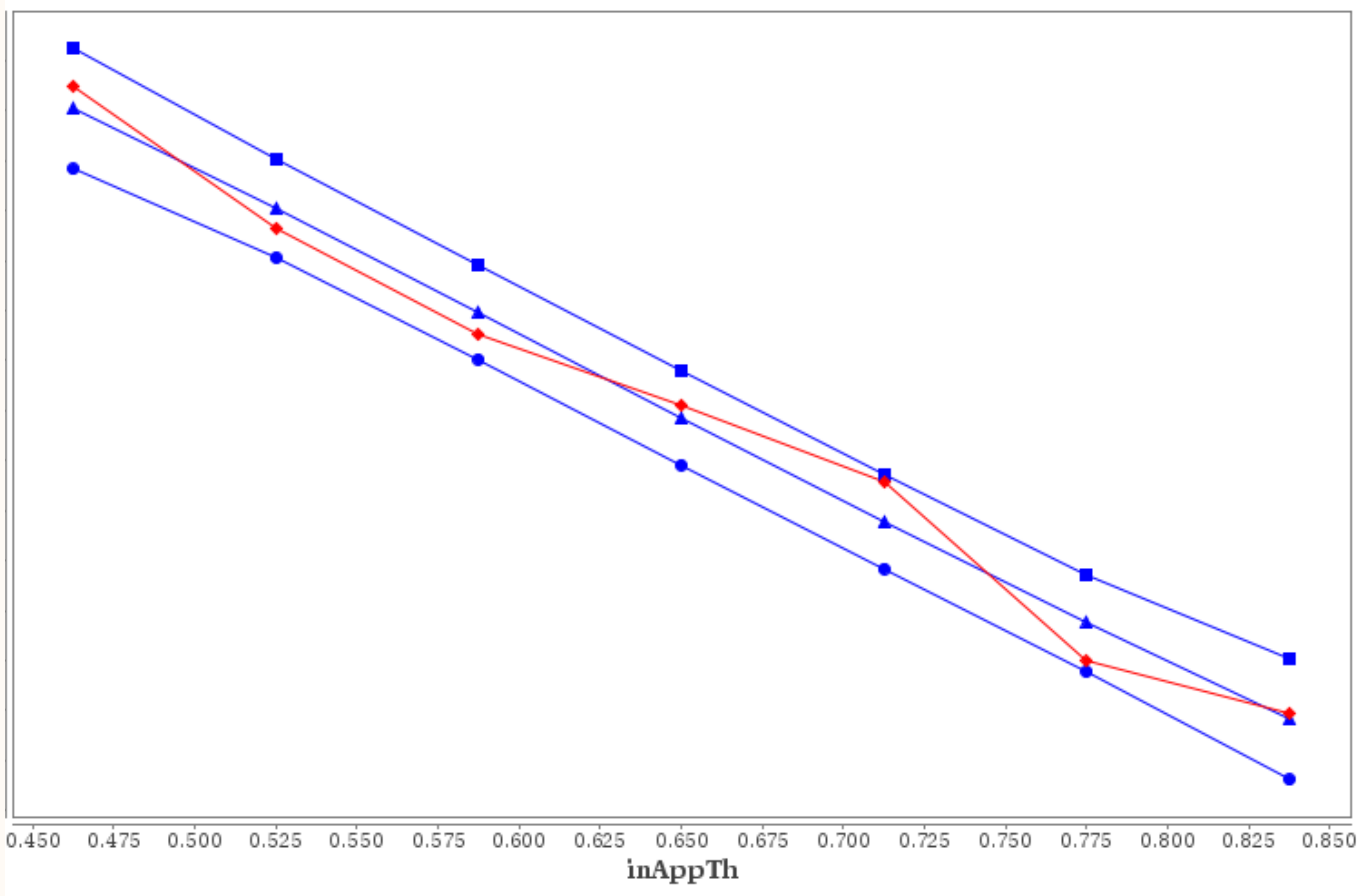
Function Fitting Plots for Constraints: Clicks vs. InApp Threshold

- Function Fitting
  - The red curve refers to observed metrics
  - The blue curve refers to fitted metrics with lower and upper confidence bands
  - The horizontal line for the constraint refers to the lower bound / upper bound for the constraint



Function Fitting Plots for Constraints: Send Volume vs. InApp Threshold

- Function Fitting
  - The red curve refers to observed metrics
  - The blue curve refers to fitted metrics with lower and upper confidence bands
  - The horizontal line for the constraint refers to the lower bound / upper bound for the constraint



Future Directions

Future Directions

- Add on other Explore-Exploit Algorithms
  - UCB (Upper Confidence Bound), EI (Expected Improvement)
- Multi-Task Gaussian Process
  - Offline metrics could provide valuable prior information for online metrics. Multi-Task Gaussian Process models the correlation between offline metrics and online metrics to achieve faster convergence.
- Problem Splitting
  - Optimal parameters for different cohorts (daily active users, weekly active users) might be different. Problem splitting targets on searching for parameters for various cohorts.

Key Takeaways

- Removes the human in the loop: Fully automatic process to find the optimal parameters.
- Multi-Drastically improves developer productivity.
- Can scale to multiple competing metrics.
- Very easy onboarding infra for multiple vertical teams. Currently used by Ads, Feed, Notifications, PYMK (People You May Know), etc.

Thank you

Linked in