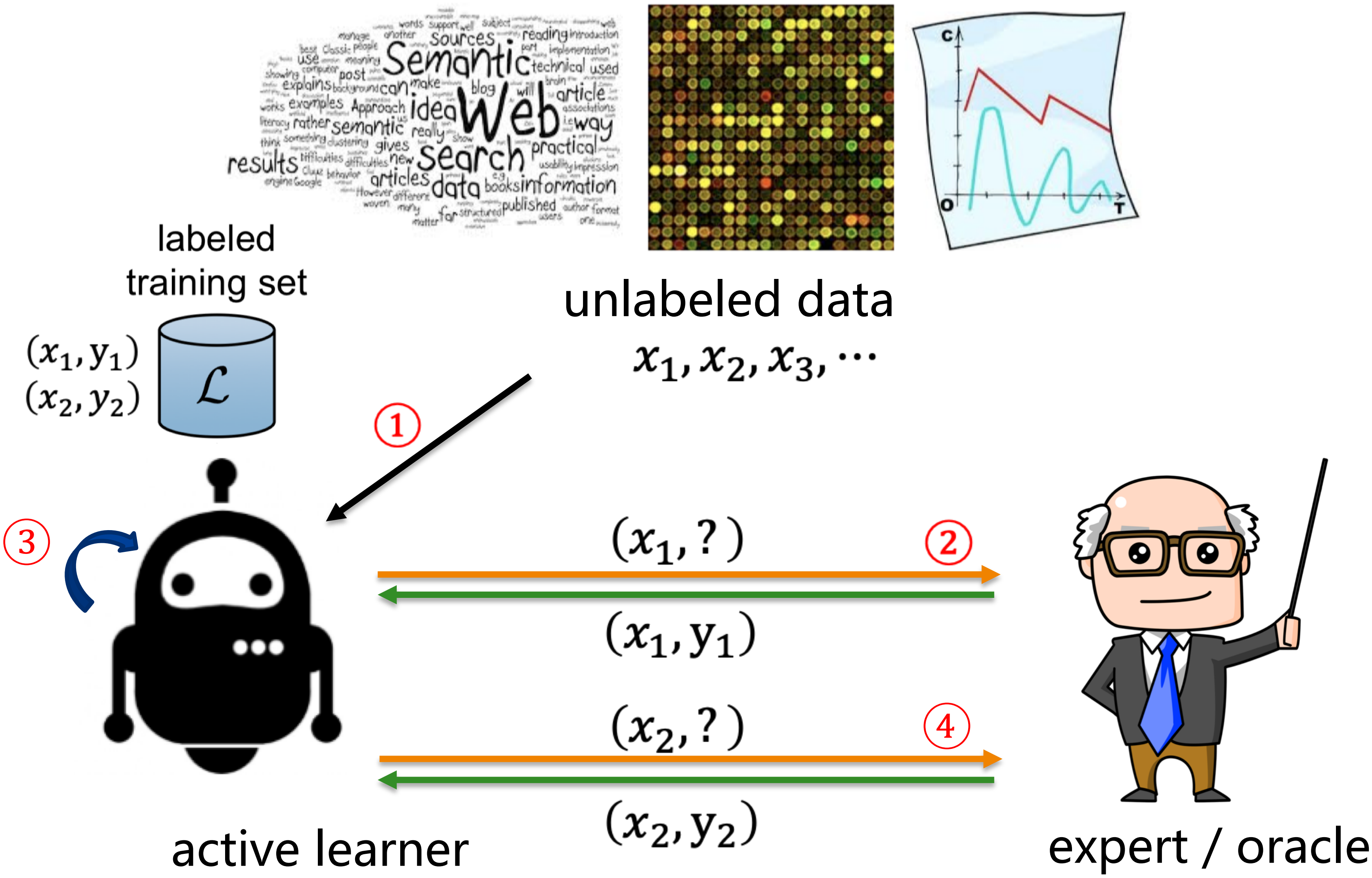
Efficient Nonmyopic Active Search

Jiang, Malkomes, Converse, Shofner, Moseley, Garnett. ICML 2017

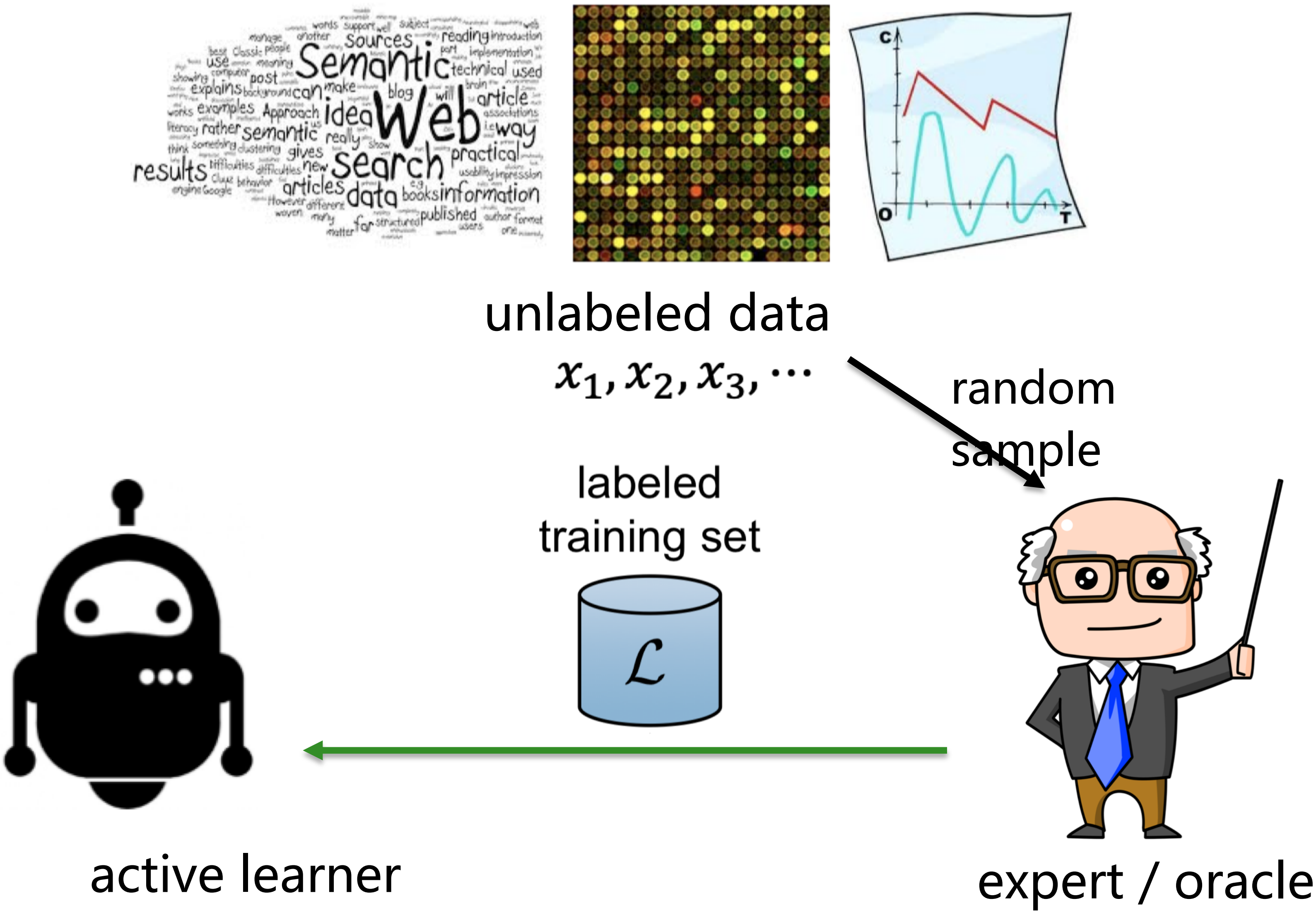
Presented by Arghavan Modiri, Jingkang Wang, Jinman Zhao

CSC 2547, Oct 18th

Active Learning



Supervised Learning



Why active learning matters?

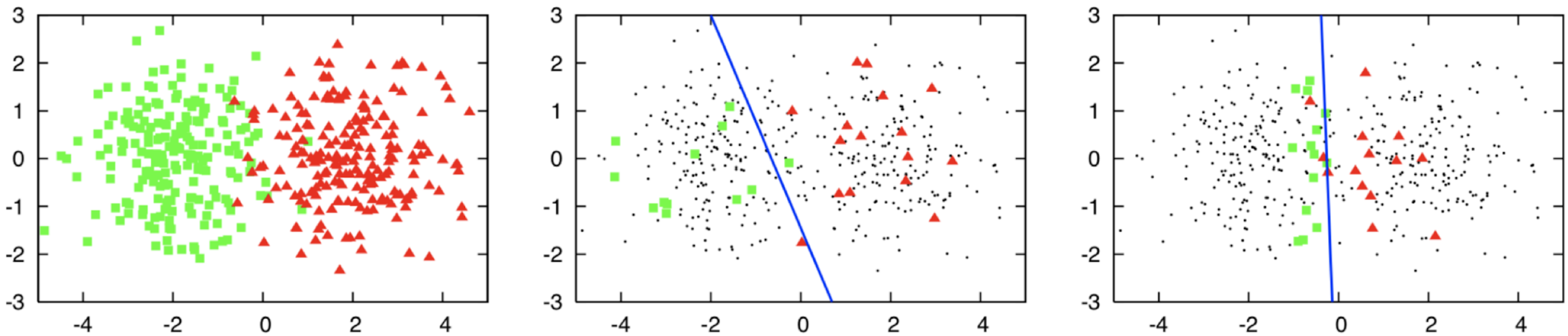
- Collecting data is much cheaper than annotating them
  - we have large-scale unlabeled data
- Labeling data is very difficult, time-consuming, or expensive

Active learning helps model learn more efficiently (compared to random sampling)

Uncertainty Sampling

- Query examples that the learner are most uncertain about (i.e., instances near the decision boundary of the model)

Binary: query the instance whose posterior probability of being positive is nearest 0.5



400 instances sampled from 2 class Gaussians

random sampling
30 labeled instances
(accuracy=0.7)

uncertainty sampling
30 labeled instances
(accuracy=0.9)

$$x^* \triangleq \arg\min_x \left|\Pr(y=1 \mid x, \mathcal{D}) - {}^1\!/_2\right|$$

[Lewis & Gale, SIGIR'94]

Uncertainty Sampling

- For multiclass problems
  - *least confidence*

$$x^*_{LC} = \underset{x}{\operatorname{argmax}} \; 1 - P_\theta(\hat{y}|x)$$

$$\hat{y} = \operatorname{argmax}_y P_\theta(y|x)$$

  - *margin sampling*

$$x^*_M = \underset{x}{\operatorname{argmin}} \, P_\theta(\hat{y}_1|x) - P_\theta(\hat{y}_2|x)$$

$\hat{y}_1$ and $\hat{y}_2$ are the first and second most probable class labels

  - *entropy*

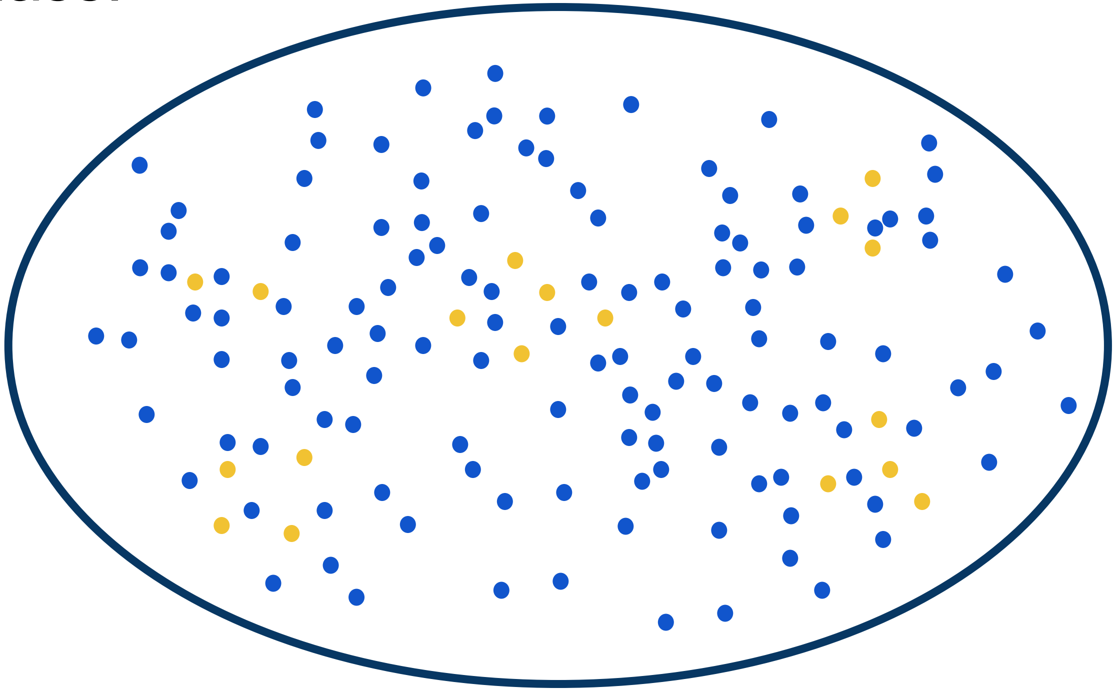
$$x^*_H = \underset{x}{\operatorname{argmax}} - \sum_i P_\theta(y_i|x) \log P_\theta(y_i|x)$$

Other Query Strategies

- Query-By-Committee (QBC)
  - maintain a committee for voting query candidates
- Expected Model Change
  - impart the greatest change to the current model
- Expected Error Reduction
  - how much its generalization error is likely to be reduced
- Variance Reduction
  - minimizing output variance
- Density-Weighted Methods
  - modifying the input distribution and pick informative instances (uncertain and representative)

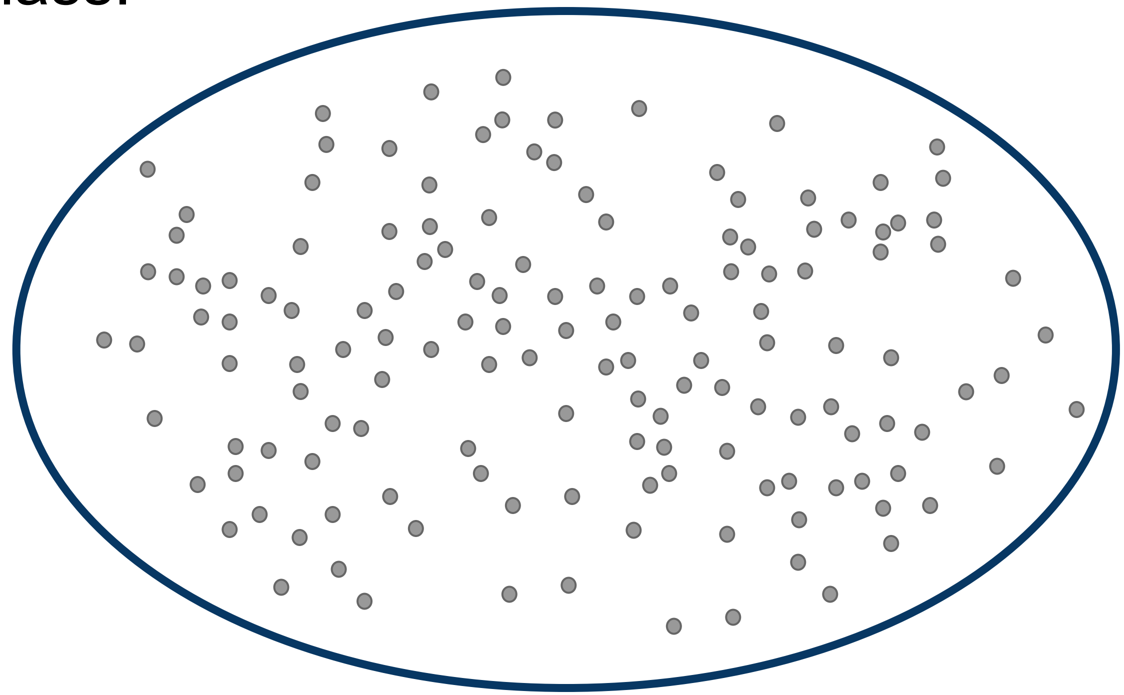
Active Search

sequentially inspecting data to discover members of a rare, desired class.



Active Search

sequentially inspecting data to discover members of a rare, desired class.



What is the best policy to select between data points such that we can find more of the *target* class in a given number of queries?

Active Search

- Given a finite domain of elements $\mathcal{X} \triangleq \{x_i\}$
- target set $\mathcal{R} \subset \mathcal{X}$
- budget t

Goal: Maximizing the utility function in budget t

$$u(\mathcal{D}) \triangleq \sum_{y_i \in \mathcal{D}} y_i$$

where

$$\mathcal{D} \triangleq \{(x_i, y_i)\} \quad y \triangleq \mathbb{1}\{x \in \mathcal{R}\}$$

Optimal Bayesian Policy

- Assume we have a probabilistic classification model that provides

$$\Pr(y = 1 \mid x, \mathcal{D})$$

- The optimal policy

$$x_i^* \triangleq \operatorname*{arg\,max}_{x_i \in \mathcal{X} \setminus \mathcal{D}_{i-1}} \mathbb{E}\big[u(\mathcal{D}_t) \mid x_i, \mathcal{D}_{i-1}\big]$$

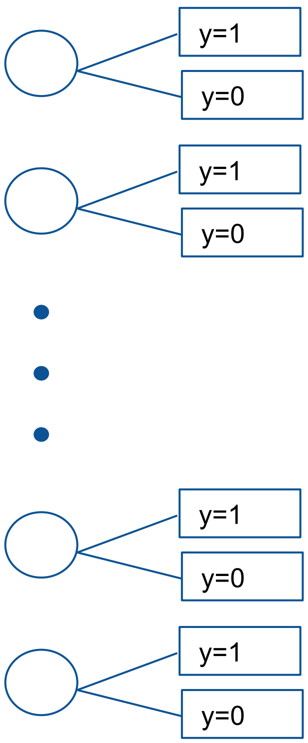
- How to solve above Equation?

Optimal Bayesian Policy

Optimal Policy for the last query ($i = t$) :

- Intuition
  - There is no need to explore
  - The optimal decision should be greedy

Time step i = t
[n - (t-1)] nodes are unlabeled

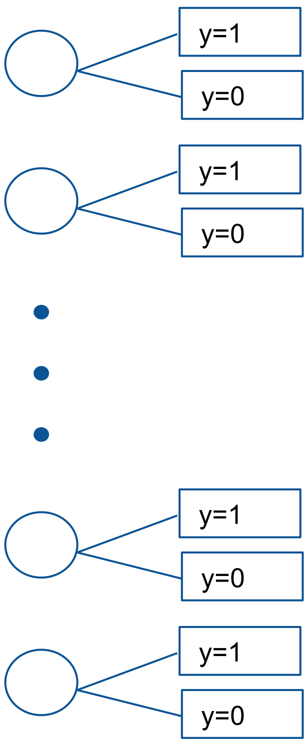


Optimal Bayesian Policy

Optimal Policy for the last query ($i = t$) :

- Intuition
  - There is no need to explore
  - The optimal decision should be greedy
- Solving Bayesian Policy equation confirms

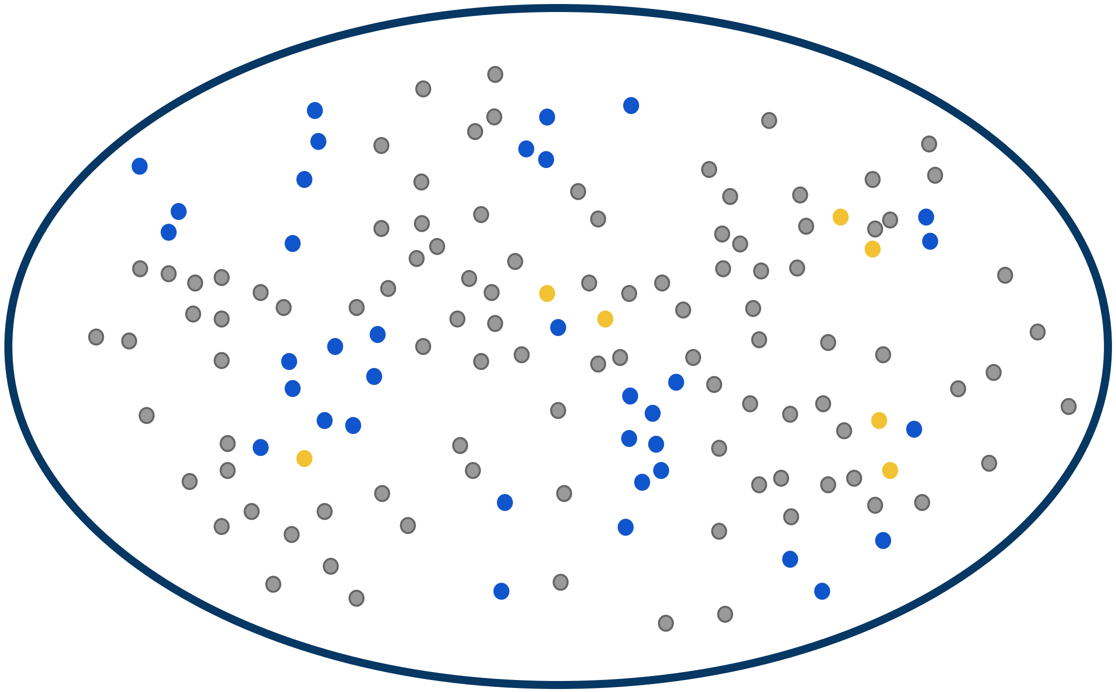
Time step i = t
[n - (t-1)] nodes are unlabeled



$$\mathbb{E}\left[u(\mathcal{D}_t) \mid x_t, \mathcal{D}_{t-1}\right] = u(\mathcal{D}_{t-1}) + \Pr(y_t = 1 \mid x_t, \mathcal{D}_{t-1})$$

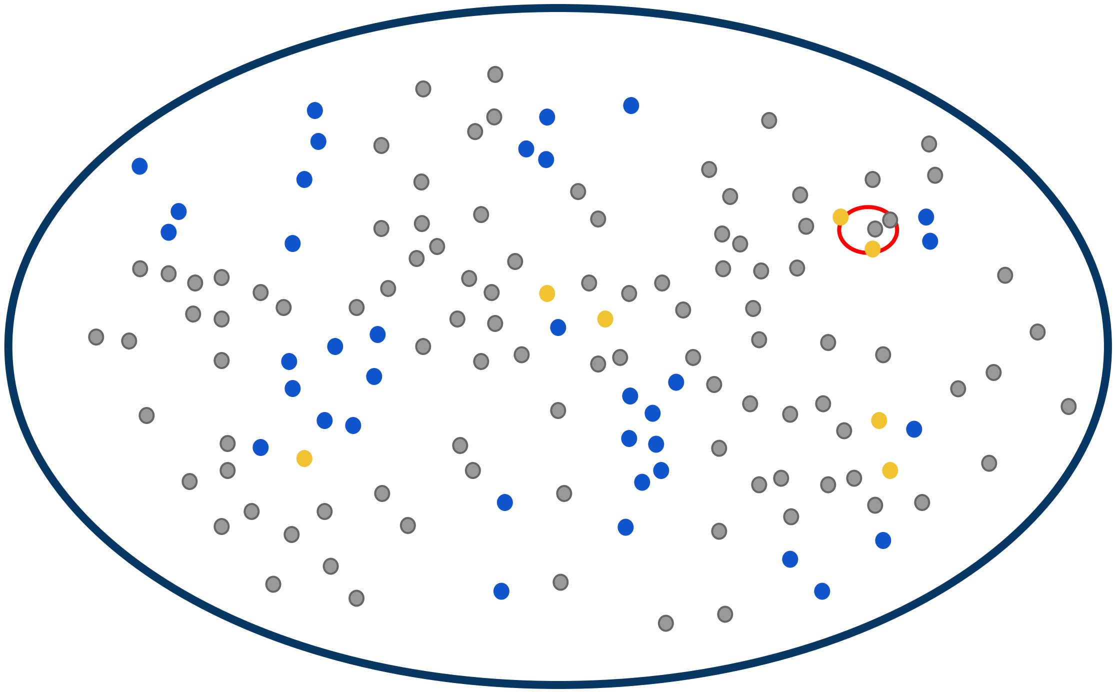
Optimal Bayesian Policy (Example)

last query for our example:



Optimal Bayesian Policy (Example)

last query for our example:

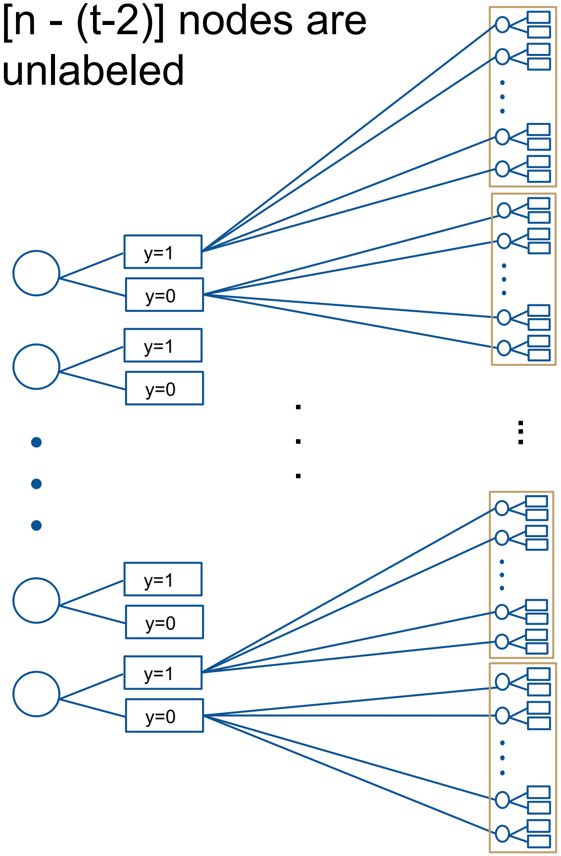


Optimal Bayesian Policy

Optimal Policy when two queries are left
$(i = t - 1)$

- policy is not as trivial
- the probability model changes after the first choice

Time step i = t-1
[n - (t-2)] nodes are unlabeled



Optimal Bayesian Policy

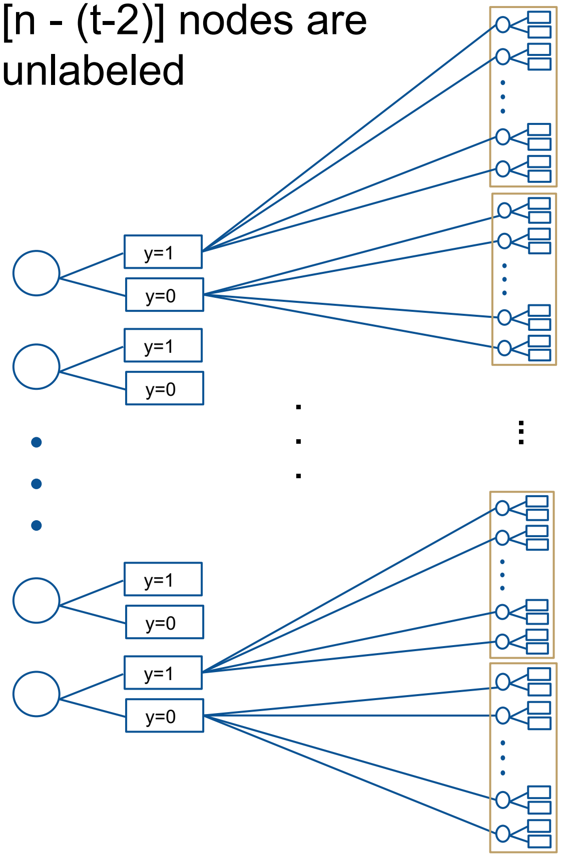
Optimal Policy when two queries are left
$(i = t - 1)$

- policy is not as trivial
- the probability model changes after the first choice

Solving Bayesian Policy equation

$$\mathbb{E}\left[u(\mathcal{D}_t) \mid x_{t-1}, \mathcal{D}_{t-2}\right] = u(\mathcal{D}_{t-2}) + \\ \Pr(y_{t-1} = 1 \mid x_{t-1}, \mathcal{D}_{t-2}) + \\ \mathbb{E}_{y_{t-1}}\left[\max_{x_t} \Pr(y_t = 1 \mid x_t, \mathcal{D}_{t-1})\right]$$

Time step i = t-1
[n - (t-2)] nodes are unlabeled



(n-(t-2)) * 2 * (n-(t-1) * 2) computation

Optimal Bayesian Policy

Optimal Policy when two queries are left $(i = t - 1)$

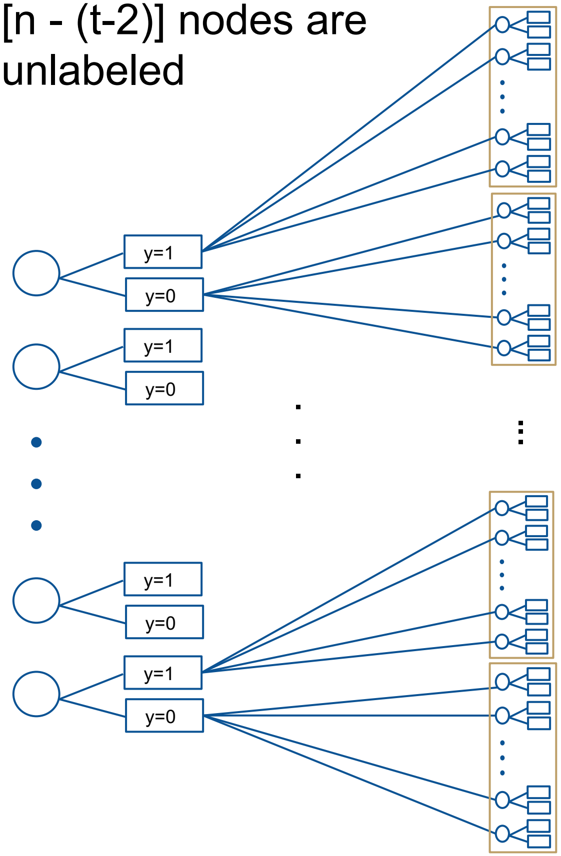
- policy is not as trivial
- the probability model changes after the first choice

Solving Bayesian Policy equation

$$\mathbb{E}\left[u(\mathcal{D}_t) \mid x_{t-1}, \mathcal{D}_{t-2}\right] = u(\mathcal{D}_{t-2}) + \Pr(y_{t-1} = 1 \mid x_{t-1}, \mathcal{D}_{t-2}) + \mathbb{E}_{y_{t-1}}\left[\max_{x_t} \Pr(y_t = 1 \mid x_t, \mathcal{D}_{t-1})\right]$$

Exploitation

Time step i = t-1
[n - (t-2)] nodes are unlabeled



(n-(t-2)) * 2 * (n-(t-1) * 2) computation

Optimal Bayesian Policy

Optimal Policy when two queries are left
$(i = t - 1)$

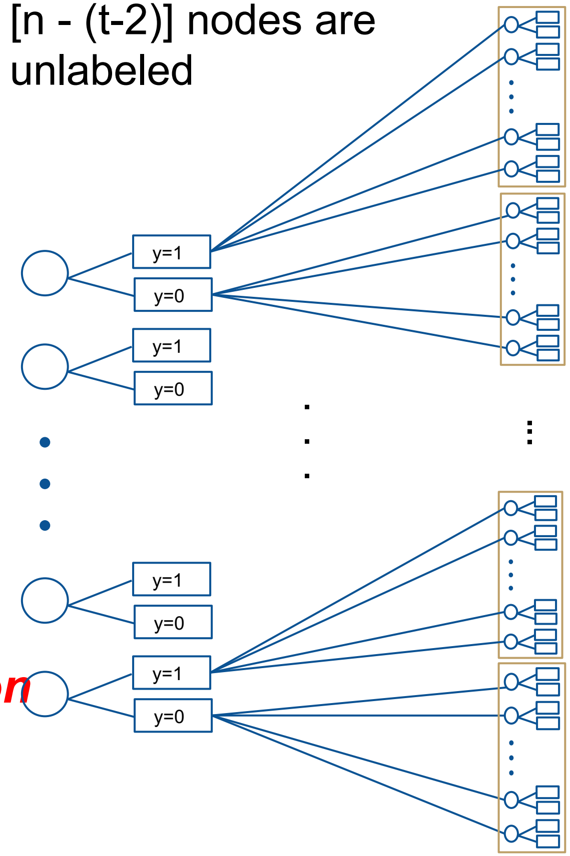
- policy is not as trivial
- the probability model changes after the first choice

Solving Bayesian Policy equation

$$\mathbb{E}\big[u(\mathcal{D}_t) \mid x_{t-1}, \mathcal{D}_{t-2}\big] = u(\mathcal{D}_{t-2}) + \Pr(y_{t-1} = 1 \mid x_{t-1}, \mathcal{D}_{t-2}) + \mathbb{E}_{y_{t-1}}\Big[\max_{x_t} \Pr(y_t = 1 \mid x_t, \mathcal{D}_{t-1})\Big]$$

Exploration

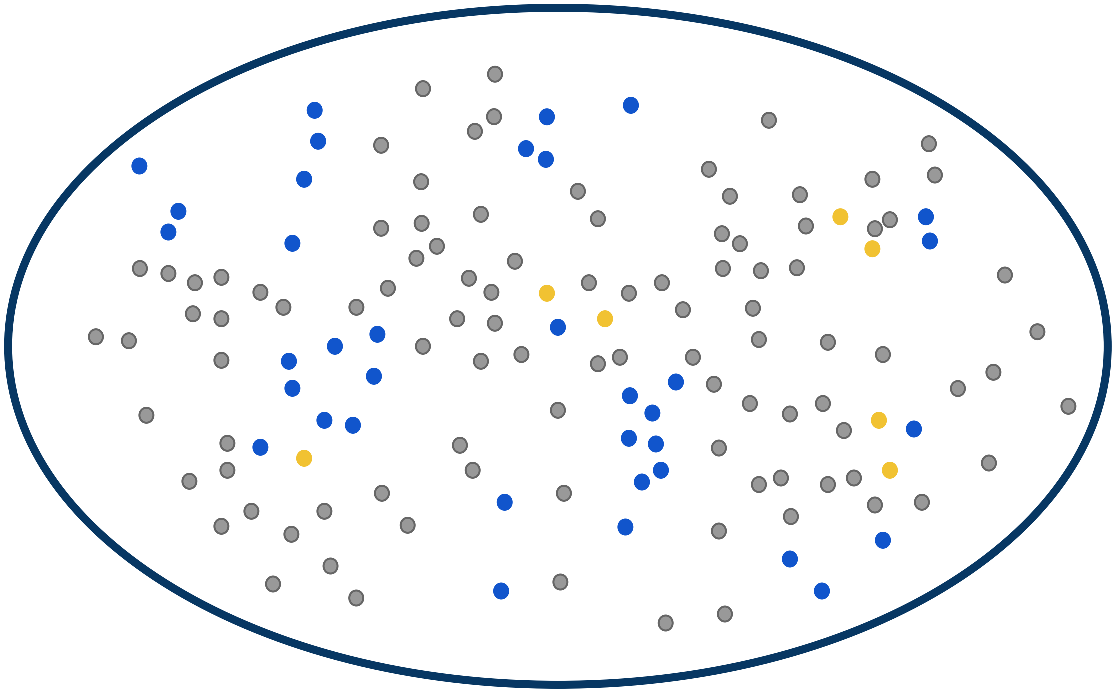
Time step i = t-1
[n - (t-2)] nodes are unlabeled



(n-(t-2)) * 2 * (n-(t-1) * 2)
computation

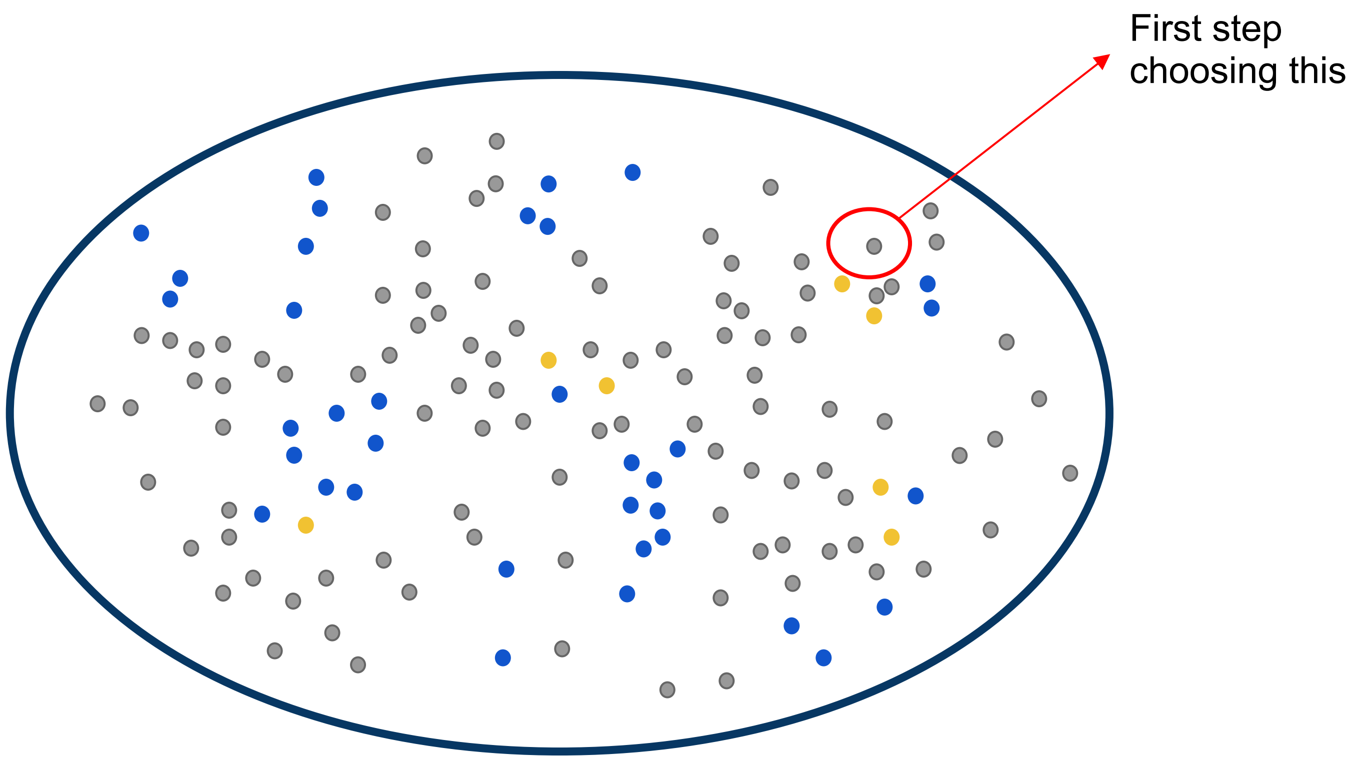
Optimal Bayesian Policy (Example)

Two queries are left:



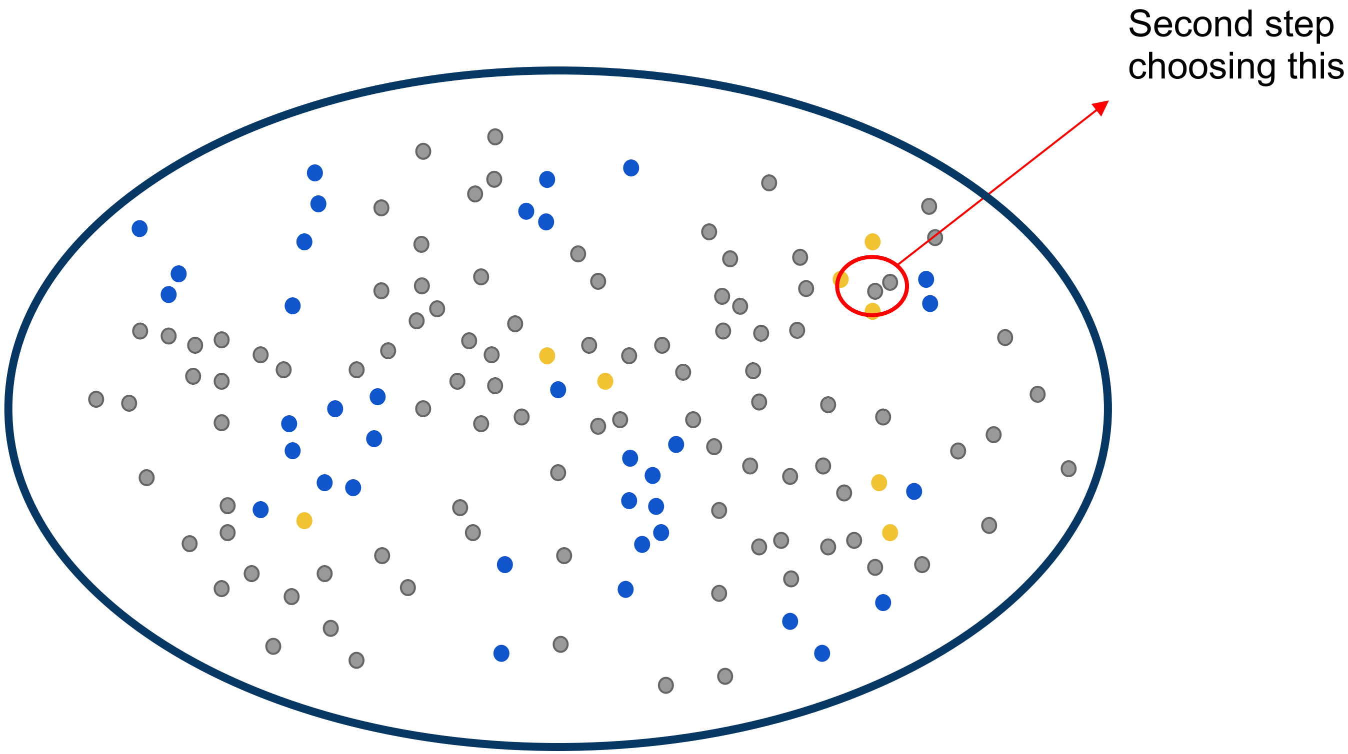
Optimal Bayesian Policy (Example)

Two queries are left:



Optimal Bayesian Policy (Example)

Two queries are left:



Optimal Bayesian Policy

Bayesian Policy equation (General Form)

$$\mathbb{E}\big[u(\mathcal{D}_t) \mid x_i, \mathcal{D}_{i-1}\big] = u(\mathcal{D}_{i-1}) + \underbrace{\Pr(y_i = 1 \mid x_i, \mathcal{D}_{i-1})}_{\text{exploitation, } < 1} + \underbrace{\mathbb{E}_{y_i}\Big[\max_{x'} \mathbb{E}\big[u(\mathcal{D}_t \setminus \mathcal{D}_i) \mid x', \mathcal{D}_i\big]\Big]}_{\text{exploration, } < t-i}$$

Time complexity: $\mathcal{O}\big((2n)^{\ell}\big)$

- where ℓ is the lookahead $\ell = t - i + 1$
- n is the total number of unlabeled point

Optimal Bayesian Policy

Bayesian Policy equation (General Form)

$$\mathbb{E}\big[u(\mathcal{D}_t) \mid x_i, \mathcal{D}_{i-1}\big] = u(\mathcal{D}_{i-1}) + \underbrace{\Pr(y_i = 1 \mid x_i, \mathcal{D}_{i-1}) +}_{\text{exploitation, } \leq 1} \underbrace{\mathbb{E}_{y_i}\Big[\max_{x'} \mathbb{E}\big[u(\mathcal{D} \setminus \mathcal{D}_i) \mid x', \mathcal{D}_i\big]\Big]}_{\text{exploration, } < t-i}$$

Time complexity: $O((2n)^{\ell})$

- where ℓ is the lookahead $\ell = t - i + 1$
- n is the total number of unlabeled point

infeasible

Hardness of Approximation

There is no *polynomial-time* active search policy with a constant factor approximation ratio for optimizing the expected utility.

Myopic Approach

- **1-step ahead myopic**

$$x_i^* = \arg\max_{x_i} \mathbb{E}[u(\mathcal{D}_i)|x_i, \mathcal{D}_{i-1}]$$

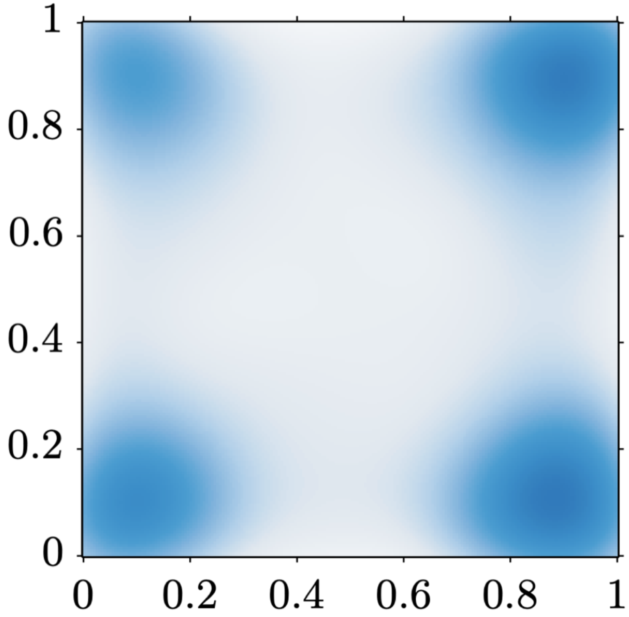
- **2-step ahead myopic**

$$x_i^* = \arg\max_{x_i} \mathbb{E}[u(\mathcal{D}_{i+1})|x_i, \mathcal{D}_{i-1}]$$

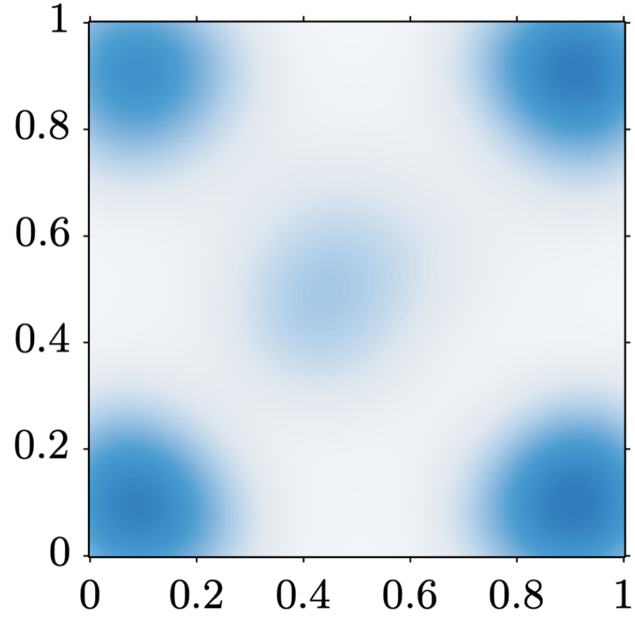
Toy Example

$$I \triangleq [0,1]^2$$

- Target: all points within Euclidean distance $1/4$ from either the center or any corner of I



uncertainty sampling



1-step optimal

Experiments (Active Search)

- Dataset: CiteSeer citation network (38079 nodes)
- Target: Papers appearing in NeurIPS (2198 in total, 5.2%)
- Features: extracted by PCA

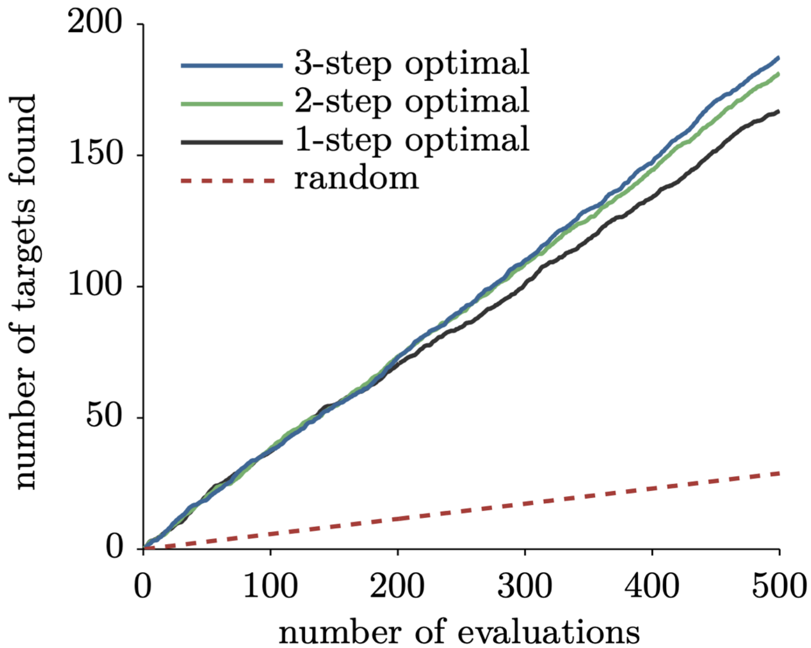


Figure 3: Cumulative number of targets found during 1 000 steps of several active querying schemes on the CiteSeer^x data. The dashed red line shows the expected performance of random sampling.

- 1-step: 167 targets
- 2-step: 180 targets
- 3-step: 187 targets
- 6.5 times better than random search

Search-space pruning

- Pruning improves the search efficiency
- Still exponential

Table 1: The average time (in seconds) taken for one iteration of the ℓ-step lookahead optimal search policy on the CiteSeer^{x} data, for $1 \leq \ell \leq 4$. Some times are approximate. For reference, the one-step policy took an average of 2.24×10^{-3} s per iteration.

	$\ell = 2$	$\ell = 3$	$\ell = 4$
pruning	0.228 s	15.0 s	745 s
no pruning	166 s	≈146 days	≈30 500 years
speedup	731	8.42×10^5	1.29×10^9

Approximating Bayesian Optimal Policy

Reminder: Bayesian Optimal Policy

$$
\begin{aligned}
\mathbb{E}\big[u(\mathcal{D}_t) \mid x_i, \mathcal{D}_{i-1}\big] = u(\mathcal{D}_{i-1}) + \\
\Pr(y_i = 1 \mid x_i, \mathcal{D}_{i-1}) + \\
\underbrace{\mathbb{E}_{y_i}\Big[\max\nolimits_{x'} \mathbb{E}\big[u(\mathcal{D}_t \setminus \mathcal{D}_i) \mid x', \mathcal{D}_i\big]\Big]}_{\text{exploration, } < t-i}
\end{aligned}
$$

Approximating Bayesian Optimal Policy

$$\begin{aligned}\mathbb{E}\big[u(\mathcal{D}_t) \mid x_i, \mathcal{D}_{i-1}\big] \approx u(\mathcal{D}_{i-1}) &+ \\ \Pr(y_i = 1 \mid x_i, \mathcal{D}_{i-1}) &+ \\ &\underbrace{\mathbb{E}_{y_i}\Big[\textstyle\sum'_{t-i} \Pr\big(y = 1 \mid x, \mathcal{D}_i\big)\Big]}_{\text{exploration, } < t-i}\end{aligned}$$

assume that any remaining points, in our budget will be selected simultaneously in one big batch

Approximating Bayesian Optimal Policy

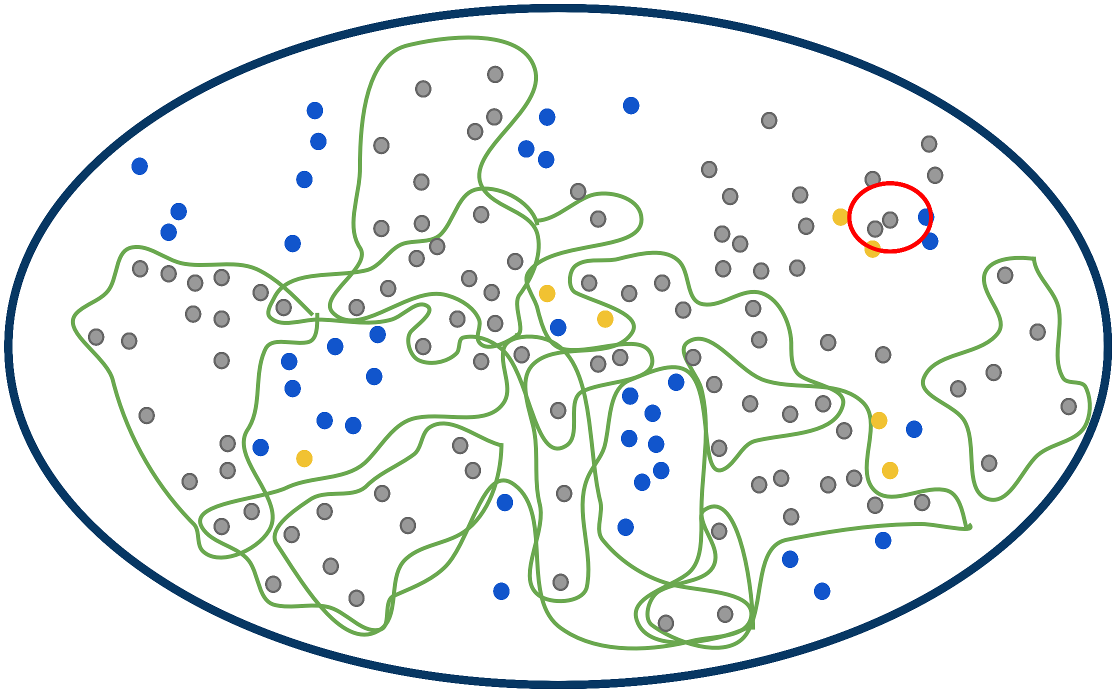
We will call this policy **efficient nonmyopic search (ENS)**.

$$
\begin{aligned}
\mathbb{E}\big[u(\mathcal{D}_t) \mid x_i, \mathcal{D}_{i-1}\big] \approx\; & u(\mathcal{D}_{i-1}) + \\
& \Pr(y_i = 1 \mid x_i, \mathcal{D}_{i-1}) + \\
& \underbrace{\mathbb{E}_{y_i}\Big[\textstyle\sum'_{t-i} \Pr\big(y = 1 \mid x, \mathcal{D}_i\big)\Big]}_{\text{exploration, } < t-i}
\end{aligned}
$$

Time complexity: $\mathcal{O}(n^2 \log n)$

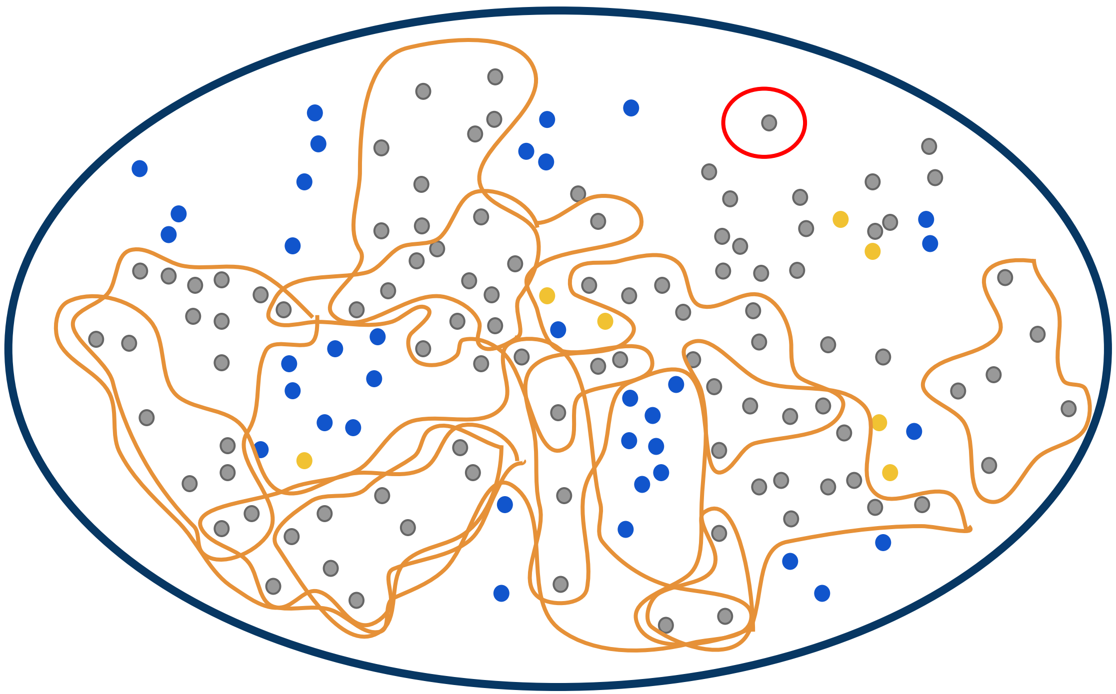
ENS (Example)

at ith query ($t-i$ nodes are left to be labelled)



ENS (Example)

at ith query ($t-i$ nodes are left to be labelled)



Until we find the x with maximum utility...

Efficient nonmyopic search (ENS)

When does ENS become the exact Bayesian optimal policy?

Efficient nonmyopic search (ENS)

When does ENS become the exact Bayesian optimal policy?

- if after observing $\mathcal{D}_i$, the labels of all remaining unlabeled points are conditionally independent

Nonmyopic Behavior

$$I \triangleq [0,1]^2$$

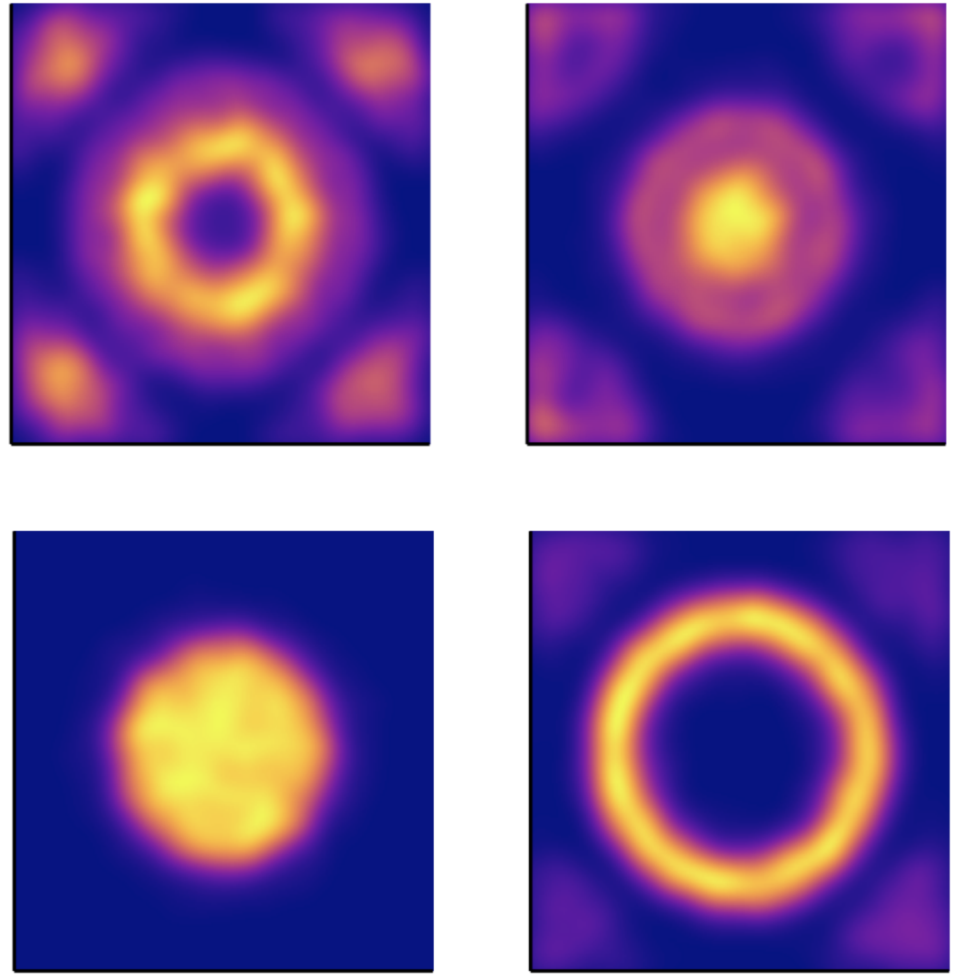
- Target:

  all points within Euclidean distance $1/4$ from either the center or any corner of I

- Budget: 200

ENS:

2-step lookahead:



first 100 points **last 100 points**

Experiment

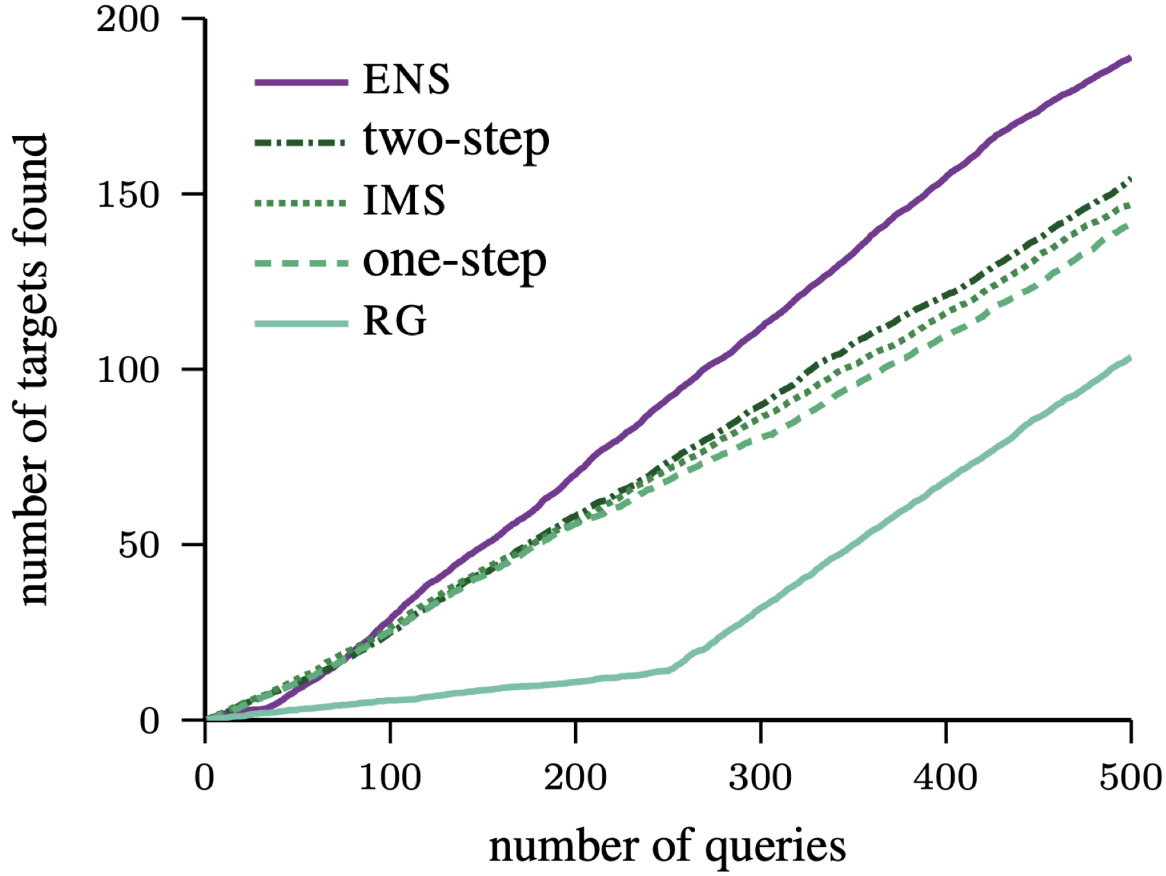
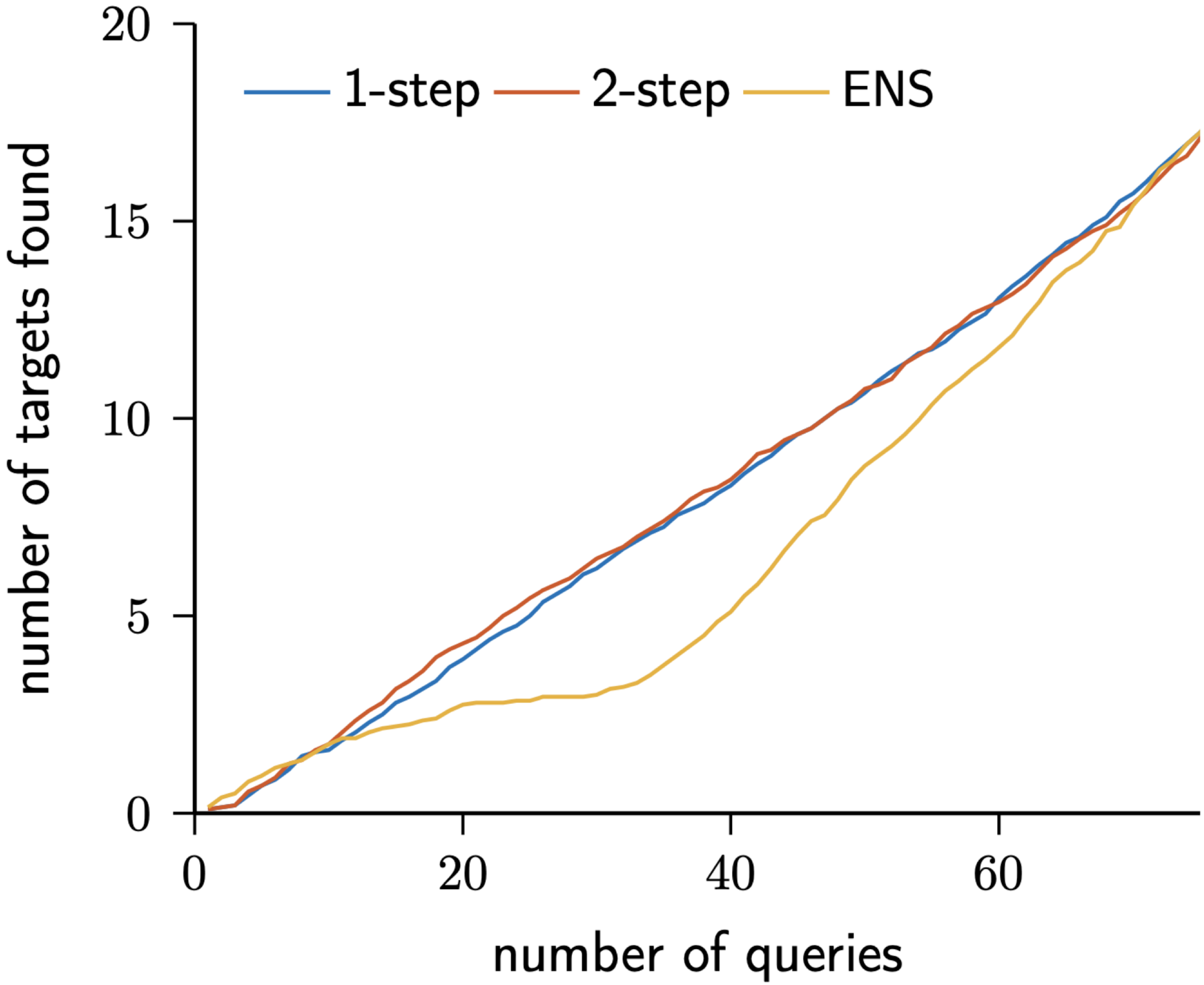


Figure 2: The learning curve of our policy and other baselines on the CiteSeer^x dataset.

Zoom



1-step
2-step
ENS
number of targets found
number of queries
0
5
10
15
20
0
20
40
60

Experiment

CiteSeer^x data

	query number				
policy	100	300	500	700	900
RG	19.7	60.0	104	140	176
IMS	26.3	86.3	147	214	281
one-step	25.5	80.5	141	209	273
two-step	24.9	89.8	155	220	287
ENS–900	25.9	94.3	163	239	**308**
ENS–700	28.0	105	188	**259**	
ENS–500	28.7	**112**	**189**		
ENS–300	26.4	105			
ENS–100	**30.7**				

Limitations

- Bayesian optimal policy and myopic methods (when lookahead step is large) are sample inefficient
- Assume the conditional independence of unlabelled data (ENS)
    - limited performance when budget is very small
- Can not deal with the continuous search space
- Difficult to generalize other more general setting
    - Bayesian Optimization, Multi-bandits, Reinforcement Learning

Takeaways

- Optimal Bayesian Policy (intractable)
- Myopic approach for approximating the optimal policy
  - Less-myopic approximations perform better
- Efficient nonmyopic search (ENS) improves the search efficiency but rely on strong assumptions

Related Work

1) ENS in batch mode (query a batch of points at a time)
    a) efficiency improvement
    b) theoretical guarantee of performance - not that worse compared to query one at a time (Jiang et al., 2018)

1) Bayesian Optimization (BO)
    a) AS can be seen as a special case of BO - **with binary observations and cumulative reward**
    b) Non-myopic policies for BO in the regression setting (Ling et al., 2016)
    c) ENS is similar to GLASS algorithm (González et al., 2016)

1) Multi-armed bandit
    a) electing an item can understood as “pulling an arm”
    b) items are correlated and cannot be played twice
    c) ENS is similar to *knowledge gradient* policy (Frazier et al., 2008)

References

[1] Settles, B. (2009). **Active learning literature survey**. University of Wisconsin- Madison Department of Computer Sciences.

[2] Garnett, R., Krishnamurthy, Y., Xiong, X., Schneider, J., & Mann, R. (2012). **Bayesian optimal active search and surveying**. arXiv preprint arXiv:1206.6406.

[3] Jiang, S., Malkomes, G., Converse, G., Shofner, A., Moseley, B., & Garnett, R. (2017, August). **Efficient nonmyopic active search**. In ICML 2017

[4] Jiang, S., Malkomes, G., Converse, G., Shofner, A., Moseley, B., & Garnett, R. **Efficient nonmyopic batch active search.** In NeurIPS 2018.

[5] González, J., Osborne, M. & Lawrence, N., 2016, May. **GLASSES: Relieving the myopia of Bayesian optimisation.** In Artificial Intelligence and Statistics

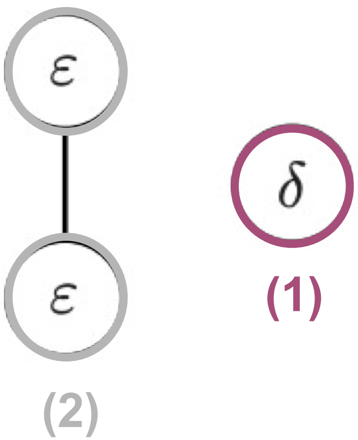
[6] Hasan Z. & Hidru D. Slide for Efficient nonmyopic batch active search. https://bayesopt .github.io/slides/2016/ContributedGarnett.pdf

[7] Jiang, S. Slide for Efficient nonmyopic batch active search. https://bayesopt.github.io/ slides/2016/ContributedGarnett.pdf

Q & A

Appendix: Myopic Approach

simple greedy one-step policy vs **two-step look ahead**:



one-step:

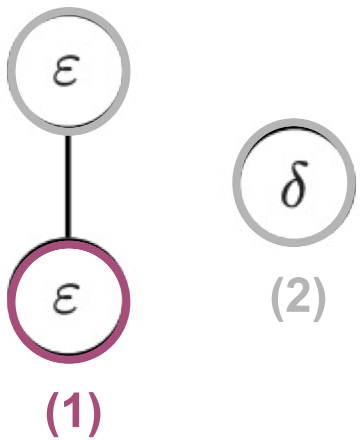
$$x_1* = \text{argmax}_{x_1}\,\mathbb{E}[u(\mathcal{D}_1)|x_1,\mathcal{D}_0] = \text{right point}$$

$$x_2* = \text{left point}$$

$$\mathbb{E}(u(\mathcal{D}_2)) = \delta + \epsilon$$

Appendix: Myopic Approach

simple greedy one-step policy vs **two-step look ahead**:



one-step:

$$\mathbb{E}(u(\mathcal{D}_2)) = \delta + \epsilon$$

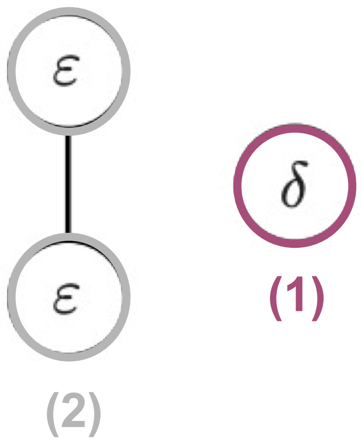
two-step(left):

$$\begin{aligned}\mathbb{E}\big[u(\mathcal{D}_t) \mid x_{t-1}, \mathcal{D}_{t-2}\big] = u(\mathcal{D}_{t-2}) &+ \\ \Pr(y_{t-1} = 1 \mid x_{t-1}, \mathcal{D}_{t-2}) &+ \\ \mathbb{E}_{y_{t-1}}\Big[\max_{x_t} \Pr(y_t = 1 \mid x_t, \mathcal{D}_{t-1})\Big]\end{aligned}$$

$$\begin{aligned}\mathbb{E}[u(\mathcal{D}_2)|x_1, \mathcal{D}_0] &= u(\mathcal{D}_0) + Pr(y_1 = 1|x_1, \mathcal{D}_0) + \\ &\quad \mathbb{E}_{y_1}[\max_{x_2} Pr(y_2 = 1|x_2, \mathcal{D}_1)] \\ &= 0 + \epsilon + Pr(y1 = 0) * [\max_{x_2} Pr(y_2 = 1|x_2, \mathcal{D}_1)] + \\ &\quad Pr(y1 = 1) * [\max_{x_2} Pr(y_2 = 1|x_2, \mathcal{D}_1)] \\ &= \epsilon + (1 - \epsilon) * \delta + \epsilon \times 1 \\ &= 2\epsilon + (1 - \epsilon) \times \delta\end{aligned}$$

Appendix: Myopic Approach

simple greedy one-step policy vs **two-step look ahead**:



one-step:

$$\mathbb{E}(u(\mathcal{D}_2)) = \delta + \epsilon$$

two-step (left):

$$\begin{aligned}\mathbb{E}[u(\mathcal{D}_2)|x_1, \mathcal{D}_0] &= 2\epsilon + (1-\epsilon) \times \delta \\ &= \epsilon + \delta + \epsilon(1-\delta)\end{aligned}$$

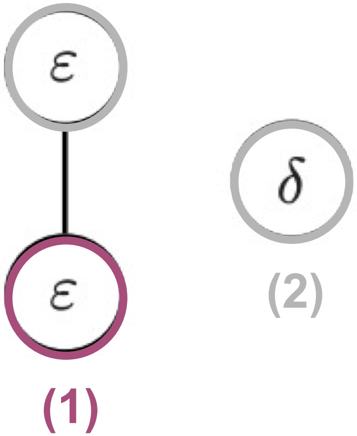
two-step (right):

$$\begin{aligned}\mathbb{E}\big[u(\mathcal{D}_t) \mid x_{t-1}, \mathcal{D}_{t-2}\big] = u(\mathcal{D}_{t-2}) +& \\ \Pr(y_{t-1} = 1 \mid x_{t-1}, \mathcal{D}_{t-2}) +& \\ \mathbb{E}_{y_{t-1}}\big[\max_{x_t} \Pr(y_t = 1 \mid x_t, \mathcal{D}_{t-1})\big]&\end{aligned}$$

$$\begin{aligned}\mathbb{E}[u(\mathcal{D}_2)|x_1, \mathcal{D}_0] &= u(\mathcal{D}_0) + Pr(y_1 = 1|x_1, \mathcal{D}_0) + \\ &\quad \mathbb{E}_{y_1}[\max_{x_2} Pr(y_2 = 1|x_2, \mathcal{D}_1)] \\ &= 0 + \delta + Pr(y1 = 0) * [\max_{x_2} Pr(y_2 = 1|x_2, \mathcal{D}_1)] + \\ &\quad Pr(y1 = 1) * [\max_{x_2} Pr(y_2 = 1|x_2, \mathcal{D}_1)] \\ &= \delta + (1-\delta) * \epsilon + \delta \times \epsilon \\ &= \epsilon + \delta\end{aligned}$$

Appendix: Myopic Approach

simple greedy one-step policy vs **two-step look ahead**:



one-step:

$$\mathbb{E}(u(\mathcal{D}_2)) = \delta + \epsilon$$

two-step(left):

$$\begin{aligned}\mathbb{E}[u(\mathcal{D}_2)|x_1, \mathcal{D}_0] &= 2\epsilon + (1-\epsilon) \times \delta \\ &= \epsilon + \delta + \epsilon(1-\delta)\end{aligned}$$

two-step(right):

$$\mathbb{E}[u(\mathcal{D}_2)|x_1\, \mathcal{D}_0] = \epsilon + \delta$$