Supplementary Material for Efficient Nonmyopic Active Search

Shali Jiang [1] **Gustavo Malkomes** [1] **Geoff Converse** [2] **Alyssa Shofner** [3] **Benjamin Moseley** [1] **Roman Garnett** [1]

1. Hardness of Active Search

In this section, we present the proof of Theorem 1. We assume that active search policies have access to the correct marginal probabilities $f(x; \mathcal{D}) = \Pr(y = 1 \mid x, \mathcal{D})$, for any given point x and labeled data $\mathcal{D}$, which may include "ficticious" observations. Further, the computational cost will be analyzed as the number of calls to f, i.e., $f(x; D)$ has unit cost. Note that the optimal policy operates in such a computational model, with exponentially many calls (in terms of $|\mathcal{X}|$) to the marginal probability function f.

Our proof technique consists of constructing an explicit active search instance where a small "secret" set of points S encodes the location of a larger "hidden" group of positive points. A particular feasibly exponential-cost policy identifies this small set first, and then obtains a large reward by collecting the revealed targets. We will show that an algorithm with limited computational power (i.e., polynomial in $n = |\mathcal{X}|$) will not be able to identify the set S. As a consequence, its performance will be arbitrarily worse than an optimal solution as the size of the instance increases.

The crux of the proof is to construct a class of instances $\mathcal{H}$ that we present next. Figure 1 shows a schematic representation of an example instance $\mathcal{I} \in \mathcal{H}$. The instances in $\mathcal{H}$ differ from each other by a permutation of the labels. An instance has $n = |\mathcal{X}| = 2^{2m}$ points, where m is a parameter of the instance. The search budget is defined to be $B = m^2$. The points in each instance can be categorized as follows.

"Clumps." These points are partitioned into 2^m groups, which we will call "clumps," each of size B. All points in a clump share the same label. Additionally, exactly one of the clumps comprises all positive points; the remaining points are all negative. The clump containing the positive points is chosen uniformly at random; therefore, the prior marginal probability for any point x_c in this category is $f(x_c; \emptyset) = p_c = 2^{-m}$. We denote the clump of all positive points C_*, where $*$ can be regarded as a m-bit integral index $1 \leq * \leq 2^m$. Figure 1(b) illustrates these points.

"Isolated points." The remaining points share the property that observing any *single* one of them does not change the marginal probabilities of any other point. These points are illustrated in Figures 1(a) (black dots) and 1(c). The marginal probabilities for any point x_b in this category is defined to be $f(x_b; \emptyset) = p_b = 1 - \sqrt[c]{1/2}$, where we define $c = \sqrt{m}/2$ for convenience. These points can be further classified into two categories:

- A "secret set," denoted by S; see Figure 1(a) (black dots). These points encode which of the clumps C_* contains the positive points, using a scheme we describe below. For ease of exposition, we partition the set S into m subsets $S_1, \ldots, S_m$, each of size dc, where we define $d = \sqrt{m}$. Thus $|S| = mdc = {}^{m^2}/_2 = {}^{B}/_2$; the size of this secret set is exactly half the budget.

  The key of this construction is that each subset S_i encodes one bit b_i of information about which clump C_* contains the positive points, using a simple encoding scheme: the binary representation of the positive clump C_*'s index is $* = b_1 b_2 \ldots b_m$. Each bit is encoded with a two-step mechanism. First, each S_i is partitioned into d groups of c points, each group encoding a "meta" bit of information $b_{ij}, 1 \leq i \leq m, 1 \leq j \leq d$, by a logical OR. These meta-bits, not in the problem instance, are illustrated by the white dots in 1(a). Finally, the meta-bits associated with S_i encode the bit b_i via a logical XOR,[1] $b_i = b_{i1} \oplus \cdots \oplus b_{id}$.

- Independent points. The remaining $2^{2m} - 2^m B - mdc$ points are totally independent from the others; revealing them conveys no information for any other point. We denote this set of points by R.

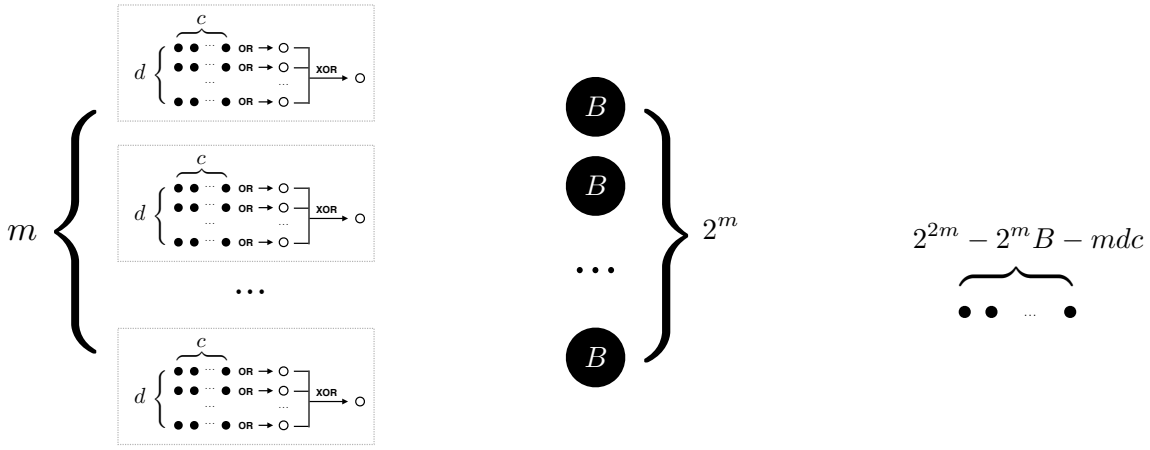
Observation 1. *At least d points from S_i need to be observed in order to infer one bit b_i of information about the positive clump.*

A virtual bit b_i is computed by the XOR operation of the d associated meta-bits $b_i = b_{i1} \oplus \cdots \oplus b_{id}$. Notice that each b_{ij} has same marginal probability of being positive,

[1]Washington University in St. Louis, St. Louis, MO, USA [2]Simpson College, Indianola, IA, USA [3]University of South Carolina, Columbia, SC, USA. Correspondence to: Shali Jiang <jiang.s@wustl.edu>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017. Copyright 2017 by the author(s).

[1]Sum of the bits modulo 2.



(a) Secret set S of mdc isolated points. (b) $2^m B$ points from clumps C_j. (c) Isolated and independent points R.

Figure 1: An instance of active search where any efficient algorithm can be arbitrarily worse than an optimal policy.

i.e., $\forall i, j, \Pr(b_{ij} = 1) = (1 - p_b)^c = 1/2$. We also have $\forall i, \Pr(b_i = 1) = 1/2$. It is necessary to observe all d meta-bits b_{ij} from the same group S_i to infer the bit b_i, since observing a fraction of the inputs of an XOR does not change the marginal belief about its outcome. So observing $d - 1$ or fewer points from S conveys no information about the positive clump. Equivalently, $\forall x, \Pr(y = 1 \mid x, \mathcal{D}) = \Pr(y = 1 \mid x)$ if $|\mathcal{D} \cap S| \leq d - 1$.

Observation 2. *Observing any number of clump points does not change the marginal probability of any point in the secret set S.*

We need to make sure that no external information can help to identify the secret set S. Notice that the knowledge of b_i does not change the marginal probability of any b_{ij}; hence, no point x in S will have a different probability after observing b_i. This means that observing points outside S does not help distinguish S from the remaining isolated points R.

Now we formally restate Theorem 1 and provide its proof. The theorem implies that a polynomial time algorithm cannot achieve a constant approximation ratio.

Theorem 1. *Any (possibly randomized) policy for active search that performs $o\left(n^{\sqrt{\frac{1}{2}\log n}}\right)$ inference calls $f(x, \mathcal{D})$, with $|\mathcal{D}| \leq B$, has approximation ratio with respect to the expected utility of the optimal policy of $\mathcal{O}\left(\frac{1}{\sqrt{\log n}}\right)$ where $|\mathcal{X}| = n$ is the number of points, and B is the budget.*[2]

Proof. Consider a random instance $\mathcal{I} \in \mathcal{H}$ and fix a policy $\mathcal{A}$. Let α be the total number of inference calls performed by $\mathcal{A}$ throughout its execution. At the ith inference call $\Pr(y = 1 \mid x, \mathcal{D}_i)$, $\mathcal{A}$ might use an arbitrary training set $\mathcal{D}_i$ of size at most B. We will show that $\mathcal{A}$ has a very small probability of collecting a large reward on $\mathcal{I}$.

Before analyzing the algorithm $\mathcal{A}$, we present a lower bound on the performance of an optimal policy. Consider the following policy with unlimited computational power: In the first iteration, compute the marginal probability of an arbitrary fixed clump point, conditioning on observing every possible subset of the isolated points of size d with labels all equal to 1. This set of $\mathcal{O}(n^d) = \mathcal{O}(n^{\sqrt{\frac{1}{2}\log n}})$ inference calls will reveal the location of the secret set S: exactly those points will modify the probabilities of the fixed clump point. Now the policy spends the first half of its budget querying the points in S (recall $|S| = B/2$). These outcomes reveal the hidden clump of positives C_*. The policy now spends the second half of the budget querying (collecting) these positive points. The expected performance of this strategy is $B/2 + p_b \, B/2 > B/2$. Certainly this is a lower bound on the optimal performance; hence

$$\text{OPT} > \frac{B}{2}. \tag{1}$$

Now consider the algorithm $\mathcal{A}$ at the ith inference. By Observations 1 and 2, $\mathcal{A}$ cannot differentiate between the points in S and those in R unless $|\mathcal{D}_i \cap S| \geq d$. Suppose that before the ith inference, the algorithm has no information about S. Then the chance of $\mathcal{A}$ choosing a $\mathcal{D}_i$ such that $|\mathcal{D}_i \cap S| \geq d$ is no better than that of a random selection from n' points, where $n' = n - 2^m B$ is the number of isolated points. We can upper bound the probability of $\mathcal{A}$ choosing a dataset $\mathcal{D}_i$ such that $|\mathcal{D}_i \cap S| \geq d$, by counting how many subsets would contain at least d points from S,

[2] Note we used the little-o notation for the number of inference calls, and the big-O notation of the approximation ratio.

among all subsets of the n' points of size at most B:

$$\Pr\big(|\mathcal{D}_i \cap S| \geq d\big) \leq \frac{\binom{B/2}{d}\binom{n'-d}{B-d}}{\binom{n'}{B}}. \tag{2}$$

We only consider the isolated points because an algorithm $\mathcal{A}$ that only queries the isolated points has a higher probability of hitting the secret set. Also note that technically the denomenator in (2) should be $\binom{n'}{B} - i + 1$ since one would not choose the same subsets $\mathcal{D}_j, j < i$ as those before the ith inference. But asymptotically $i \leq \alpha$ (assuming $\alpha = \mathcal{O}(2^n)$ for now) is of much lower order than $\binom{n'}{B}$, therefore $\binom{n'}{B} - \alpha + 1 = \Theta\Big(\binom{n'}{B}\Big)$. Denote p_h as the probability of algorithm $\mathcal{A}$ ever "hiting" the secret set after α inferences; then p_h can be union-bounded:

$$\begin{aligned} p_h &\leq \frac{\alpha\binom{B/2}{d}\binom{n'-d}{B-d}}{\binom{n'}{B}} \\ &< \frac{\alpha\left(\frac{B}{2}\right)^d B^d}{(n')^d} \\ &= \frac{\alpha}{\left(\frac{2n'}{B^2}\right)^d} \end{aligned}$$

Note $B = m^2 = \frac{1}{4}\log^2 n$ and $n' = n - 2^m B = n - \sqrt{n}\frac{1}{4}\log^2 n = \Theta(n)$, so

$$\begin{aligned} \left(\frac{2n'}{B^2}\right)^d &= \Theta\left(\left(\frac{2n}{B^2}\right)^d\right) \\ &= \Theta\left(\left(\frac{n}{\frac{1}{32}\log^4 n}\right)^{\sqrt{\frac{1}{2}\log n}}\right) \\ &= \Theta\left(n^{\sqrt{\frac{1}{2}\log n}}\right). \end{aligned}$$

We have now derived

$$p_h < \frac{\alpha}{\Theta(n^{\sqrt{\frac{1}{2}\log n}})}.$$

So for any $\alpha = \mathcal{O}\big(n^{\sqrt{\frac{1}{2}\log n}-\varepsilon}\big)$, where ε is a positive constant, we have

$$p_h < \mathcal{O}\left(\frac{1}{n^\varepsilon}\right). \tag{3}$$

If $\mathcal{A}$ ever hits the secret set S, for simplicity, we will assume that it achieves maximal performance B. If $\mathcal{A}$ never finds the secret set, we can further consider the following two cases. If the algorithm queries points $x \in R$, no marginal probabilities are changed; if a point $x \in C_j$ is queried, for any clump j, only the marginal probabilities of the clumps are changed. The expected performance in these two cases can be upper bounded by pretending that the algorithm had a larger budget of size $2B$; in which half the budget (i.e., B) is spent on querying points in R, and the other half on querying the clump points. The expected number of targets found after B queries on R is

$$Bp_b = B(1 - \sqrt[c]{1/2}) = B\left(1 - 2^{-\frac{2}{\sqrt{m}}}\right). \tag{4}$$

The expected number of targets found after B queries on the clump points is

$$\begin{aligned} &\frac{B}{2^m} + \left(1 - \frac{1}{2^m}\right)\left(\frac{B-1}{2^m - 1} + \left(1 - \frac{1}{2^m - 1}\right)(\cdots)\right) \\ &\quad = \frac{B(B+1)}{2^{m+1}}. \end{aligned} \tag{5}$$

Combining (4) and (5), we get that the expected performance in the case when $\mathcal{A}$ never hits S can be upper bounded by

$$\frac{B(B+1)}{2^{m+1}} + B\big(1 - 2^{-\frac{2}{\sqrt{m}}}\big).$$

The overall expected performance of $\mathcal{A}$ can be upper bounded by

$$E_{\mathcal{A}} < Bp_h + \frac{B(B+1)}{2^{m+1}} + B\big(1 - 2^{-\frac{2}{\sqrt{m}}}\big),$$

where we have used the trivial upper bound $1 > (1 - p_h)$. Finally, combining with the lower bound of OPT in (1), the approximation ratio can be upper bounded by

$$\begin{aligned} \frac{E_{\mathcal{A}}}{\text{OPT}} &< \frac{Bp_h + \frac{B(B+1)}{2^{m+1}} + B\big(1 - 2^{-\frac{2}{\sqrt{m}}}\big)}{B/2} \\ &= 2p_h + \frac{B+1}{2^m} + 2\big(1 - 2^{-\frac{2}{\sqrt{m}}}\big) \\ &= \mathcal{O}\left(\frac{1}{n^\varepsilon} + \frac{\log^2 n}{4\sqrt{n}} + 2\Big(1 - 2^{-\frac{2\sqrt{2}}{\sqrt{\log n}}}\Big)\right) \\ &= \mathcal{O}\left(\frac{1}{\sqrt{\log n}}\right). \end{aligned}$$

for any $\alpha = \mathcal{O}\big(n^{\sqrt{\frac{1}{2}\log n}-\varepsilon}\big) = o\big(n^{\sqrt{\frac{1}{2}\log n}}\big)$. Note that it is easy to verify that $2\big(1 - 2^{-\frac{2\sqrt{2}}{\sqrt{\log n}}}\big) = \Theta\left(\frac{1}{\sqrt{\log n}}\right)$ with L'Hôpital's rule. □

2. Results on Individual Activity Classes

Figure 2 shows the learning curves of two-step lookahead and our method, on the first six activity classes (of the total 120) for the ECFP4 fingerprint. We can see clear patterns indicating a transition from exploration to exploitation, especially for activity classes 1 and 2. The number of targets found by ENS first grows slowly, but after a certain point,

grows increasingly faster through the end of the budget. For the two-step-lookahead myopic policy, most of the time it behaves the opposite: it first greedily discovers targets faster than ENS, but in later stages, it usually levels off, not knowing where else to go. Note the point when ENS transitions from exploration to exploitation can be different for different problems, but we see a very clear pattern when these results are averaged over all 120 activity classes, as shown by the difference of the two average learning curves in Figure 3 of the main text. We also show the difference curve for the other fingerprint in Figure 3 here.

3. UCB-Style Score

We mentioned in Section 5 of the main text that we have investigated the UCB-style (Auer, 2002) score function

$$\alpha(x, \mathcal{D}) = \pi + \gamma\sqrt{\pi(1-\pi)}, \tag{6}$$

where $\pi = \Pr(y = 1 \mid x, \mathcal{D})$ and γ is a tradeoff parameter between exploitation (first term) and exploration (second term). The results of this score varying the hyperparameter γ are shown in Figure 4, for the CiteSeer^x dataset and the drug discovery dataset with fingerprint ECFP4, using exactly the same experimental setting as other methods (20 experiments, same random seed for each, sample model, etc.). Note with a fixed γ, the score α is maximized at some probability $\pi = p^*$; it is easy to derive that

$$p^* = \frac{1}{2} + \frac{1}{2\sqrt{\gamma^2 + 1}} \tag{7}$$

by setting the derivative to zero. To better present the results, we use $p^* \in [0.5, 1]$ as the hyper-parameterization of the score in Figure 4. In summary, on the CiteSeer^x dataset, as shown in Figure 4(a), the performance is maximized at some p^* near 0.6, but there does not seem to be a clear pattern. But when we average the performance on 2 400 experiments on the ECFP4 data, as shown in Figure 4(c), we see that the α score is monotonically performing better with larger p^* (or smaller γ), and converges to the greedy policy ($p^* = 1$).

4. Effect of Pruning: Detailed Results

Table 1 shows the detailed results of our pruning study described in Section 5.4 of the main text.

Table 1: Average number of pruned points in each iteration for the two chemical datasets.

fingerprint	# pruned	# total	pruned %
ECFP4	94 995	100 518	94.5%
GpiDAPH3	93 173	100 518	92.7%

5. Mean Difference Curves for CiteSeer^x and BMG Data

Figures 5 and 6 show the the mean difference curve between ENS and two-step, for various budgets, respectively for the CiteSeer^x and BMG data.

Acknowledgments

We would like to thank Brendan Juba for insightful discussion. SJ, GM, and RG were supported by the National Science Foundation (NSF) under award number IIA–1355406. GM was also supported by the Brazilian Federal Agency for Support and Evaluation of Graduate Education (CAPES). GC and AS were supported by NSF under award number CNS–1560191. BM was supported by a Google Research Award, a Yahoo Research Award, and by NSF under award number CCF–1617724.

References

ASM Alloy Center Database. URL http://mio.asminternational.org/ac/.

Auer, Peter. Using Confidence Bounds for Exploitation–Exploration Trade-offs. *Journal of Machine Learning Research*, 3:397–422, 2002.

Kawazoe, Yoshiyuki, Yu, Jing-Zhi, Tsai, An-Pang, and Masumoto, Tsuyoshi (eds.). *Nonequilibrium Phase Diagrams of Ternary Amorphous Alloys*, volume 37A of *Condensed Matter*. Springer–Verlag, 1997.

Wang, Xuezhi, Garnett, Roman, and Schneider, Jeff. Active Search on Graphs. In *Proceedings of the 19th* ACM SIGKDD *International Conference on Knowledge Discovery and Data Mining*, pp. 731–738, 2013.

Ward, Logan, Agrawal, Ankit, Choudhary, Alok, and Wolverton, Christopher. A General-Purpose Machine Learning Framework for Predicting Properties of Inorganic Materials. 2016. arXiv preprint arXiv:1606.09551 [cond-mat.mtrl-sci].

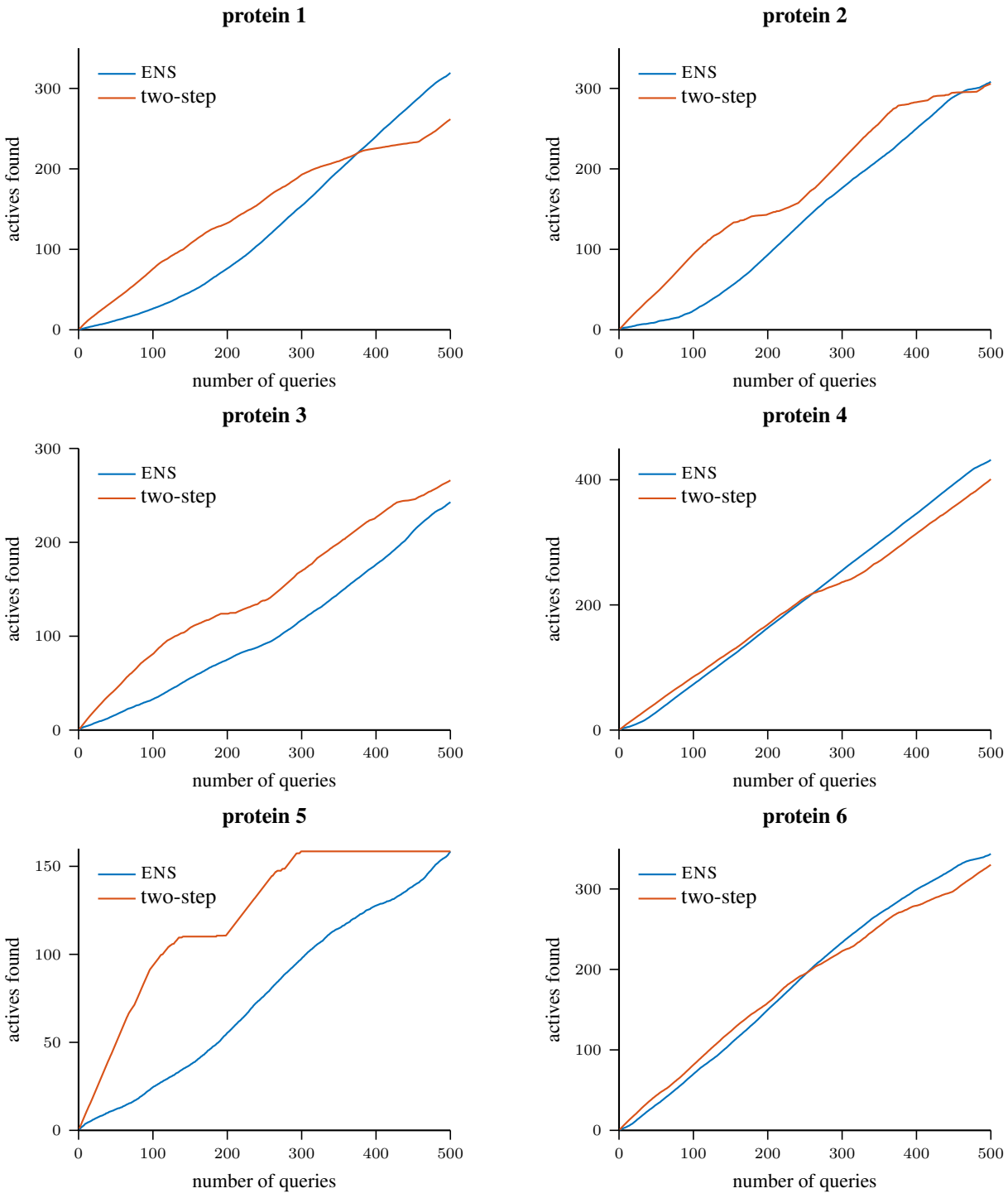


Figure 2: Number of active compounds found as a function of the number of queries, for protein (activity class) 1 to 6, fingerprint ECFP4, averaged over 20 experiments for each protein.

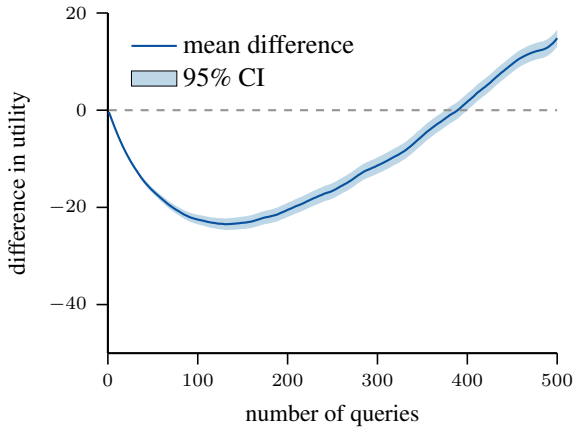
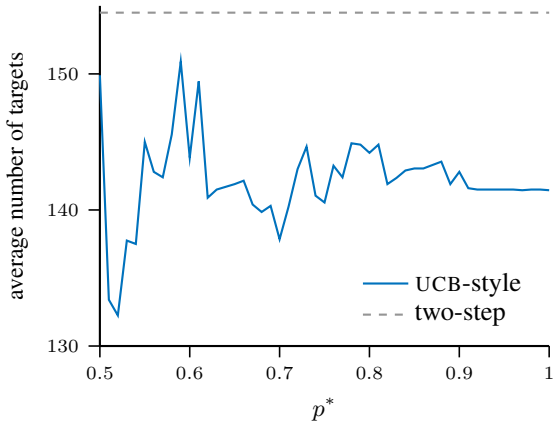
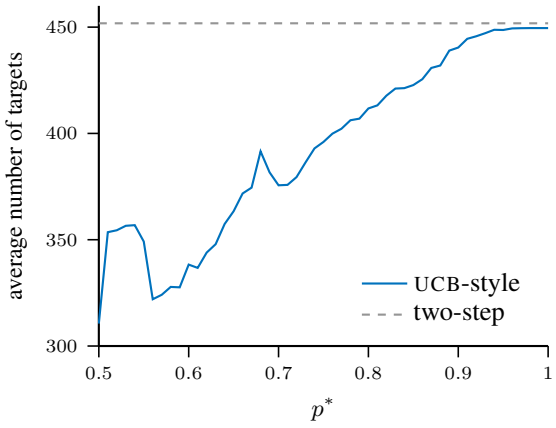


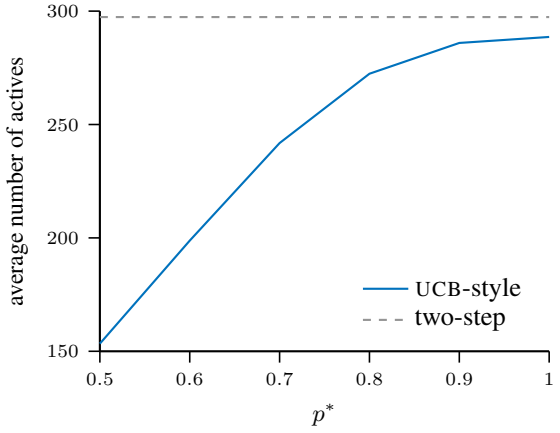
Figure 3: The average difference in cumulative targets found between our method and the two-step policy, averaged over all 120 activity classes and 20 experiments, on the fingerprints GPIDAPH3.



(a) CiteSeer^x data

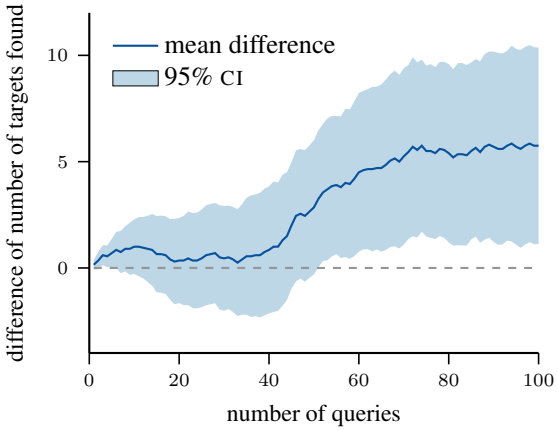


(b) BMG data

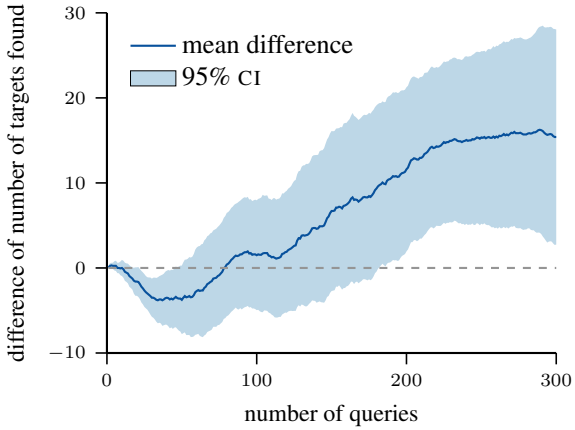


(c) ECFP4 data

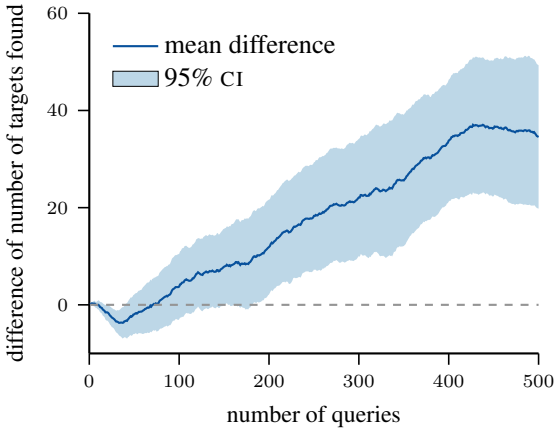
Figure 4: Number of targets found by the UCB-style policy (6), as a function of the hyperparameter p^* as derived in (7), averaged over 20 experiments. Note for CiteSeer^x and BMG datasets, the grid size of p^* is 0.01, but for ECFP4, we can only afford grid size of 0.1. To put these results into perspective, we also show the two-step performances by the red horizontal line, indicating two-step performs better than the UCB-style policy on all three domains. All these results are with budget 500.



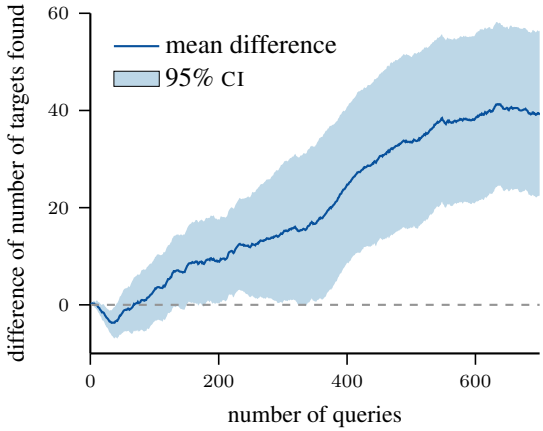
(a) Budget 100; $p = 0.017$



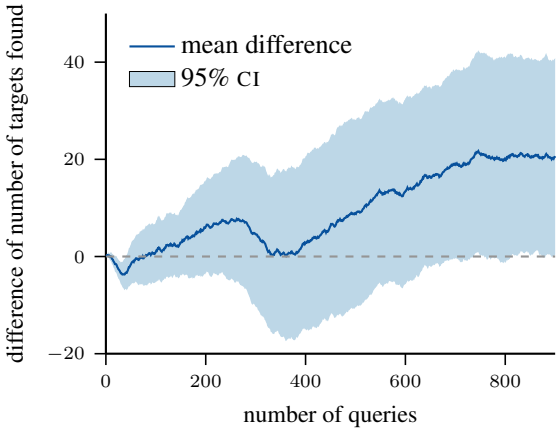
(b) Budget 300; $p = 0.020$



(c) Budget 500; $p = 9.9 \times 10^{-5}$

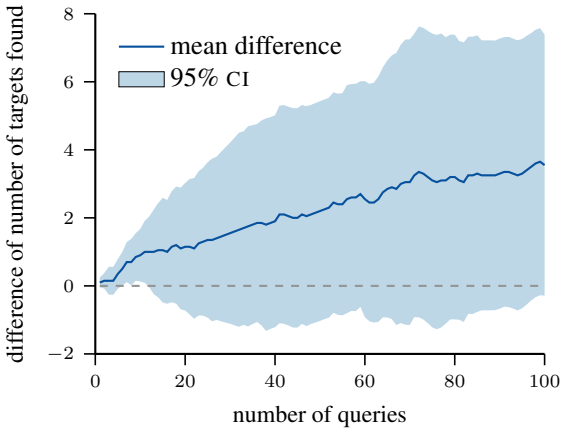


(d) Budget 700; $p = 1.0 \times 10^{-4}$

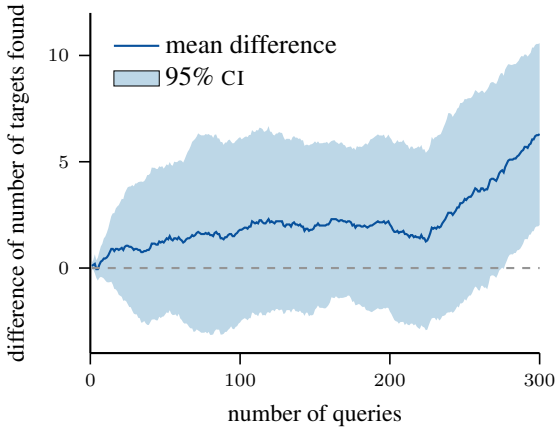


(e) Budget 900; $p = 0.046$

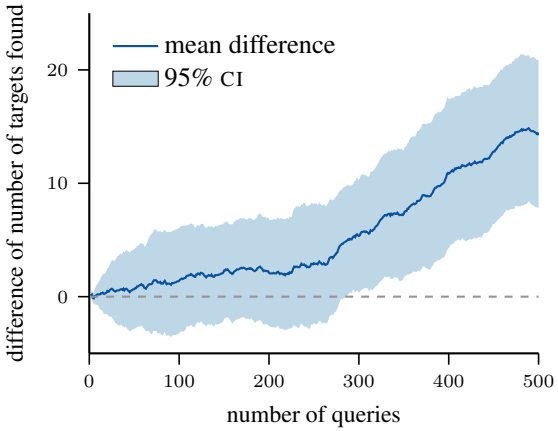
Figure 5: Mean difference curves between our policy and two-step lookahead in terms of number of actives found, along with the confidence interval, for CiteSeer^x data, with budget varying from 100 to 900. Also shown are the p-values of a two-sided paired t-test testing the null hypothesis that the performance of the policies is equal at termination.



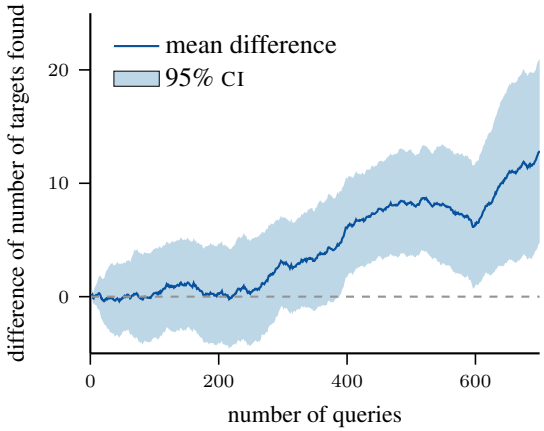
(a) Budget 100; $p = 0.068$



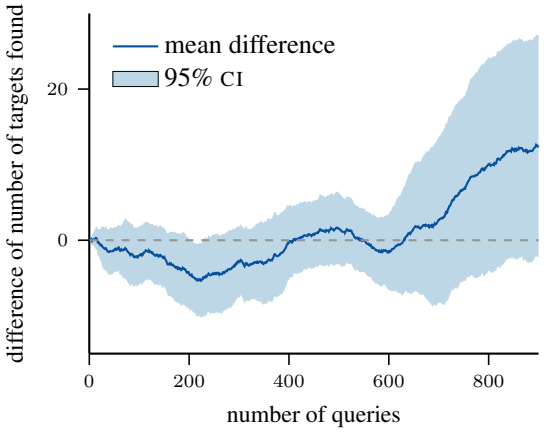
(b) Budget 300; $p = 6.1 \times 10^{-3}$



(c) Budget 500; $p = 2.1 \times 10^{-4}$



(d) Budget 700; $p = 3.7 \times 10^{-3}$



(e) Budget 900; $p = 0.091$

Figure 6: Mean difference curves between our policy and two-step lookahead in terms of number of actives found, along with the confidence interval, for BMG data, with budget varying from 100 to 900. Also shown are the p-values of a two-sided paired t-test testing the null hypothesis that the performance of the policies is equal at termination.