Bayesian Active Learning for Finding Maximally-valued Exemplars

Jialei Wang
School of Operations Research & Information Engineering, Cornell University, Ithaca, NY 14853, jw865@cornell.edu

Pu Yang
School of Operations Research & Information Engineering, Cornell University, Ithaca, NY 14853, py75@cornell.edu

Peter I. Frazier
School of Operations Research & Information Engineering, Cornell University, Ithaca, NY 14853, pf98@cornell.edu

We are motivated by a problem of searching short peptides with desired property in biochemistry, and generalize it as a Bayesian optimal search problem related to active learning. In active learning, we have training data with unknown binary labels. Obtaining labels is expensive, and we wish to obtain a small number of labels, so that the statistical classifier built from them is good. We consider a variant in which each datapoint has an associated known value, and our goal is to find a datapoint with a positive label and large value. We proposed two greedy algorithms to address this problem and proved lower bound of the algorithms that guarantees their performance. We also built a prediction model for our motivated problem, and use the proposed optimal search algorithm to find desired peptides. We showed our proposed method did significant better job than typical approach in biochemistry field.

1. Introduction (NOT READY YET)

In many optimal search problems, we need to effectively collect information so as to make the best decisions under uncertainty. In this setting, we need to trade off the reward by sampling (i.e. exploitation) and the cost by aquiring this information (i.e. exploration). For example, in drug discovery, we need to search for a chemical derivative of the base molecule that best treats disease. To achive the goal, we choose molecules to test to maximize the expected quality of the best compound discovered (Negoescu et al. 2010). Since the budget for testing is limited, we need to test the most informative and high quality molecules. To address this problem, Jones & Schonlau proposed Expected Improvement algorithm to sample points sequentially (Jones et al. 1998) . Ginsbourger used constant liar heuristic to extend Expected Improvement algorithm to parallel setting (Ginsbourger et al. 2008) . There are quite a few papers about parallel sampling in active learning research community (Chen et al. 2013, Hoi et al. 2006b,a) , but they only aim to maximize information gain (i.e. pure exploration). In this paper, we consider an optimal search problem in parallel setting, and propose search algorithm using greedy heuristic. We also prove that, our greedy algorithm has a guarantee of performace compared with the optimal solution.

2. Problem Formulation

We first describe the application that motivates our research, and then we provide mathematical formalism to address a more general problem. In the last sub-section we derive our method in solving this problem.

2.1. Motivating application.

Over the past decade, many chemistry-based tools have been developed for protein labeling, which have great use in biological studies and preparation of novel materials. In the recent years, focus has been moved to short peptide labeling techniques. Peptides are short chains of amino acid monomers, and they are much shorter than protein. Because of that, peptide labeling has minimal perturbation to native systems, which is a much desired feature. Based on prevailing short peptide labeling methods, a biochemistry research group in UC San Diego takes a step further and aims to develop a novel method of selectively adding and removing the desired label. To support this goal, they need to find such peptide from an enormous peptide library ($\sim 10^{27}$ peptides) that only reacts with some specific enzymes, while stays inert to other enzymes. This is a challenging task because this kind of peptide is extremly rare in nature (less than 1 in 10^4), and experiment cost is so expensive that the project can only afford to test hundreds of peptide candidates.

To design optimal experiments for this project is critical, because with so large search space, so rare the occurrence of target and so small amount of permitted evaluations, random sampling will almost surely lead to failure. The motivation for our work lies in developing robust prediction model given a small training dataset, and optimal search algorithm that finds target with least number of evaluations.

2.2. General problem statement.

We now formalize and generalize our problem as an active learning problem, which includes but is not limited to our motivating application.

Let E be a generic search space of exemplars. In our motivating application, E is the search space of peptide candidates. Each element $x \in E$ has an unknown binary label $y(x) = \{0, 1\}$. A known deterministic function $f(x)$ measures the cost or disutility associated with x, which is length of the peptide candidate in the motivating application. Our goal is to perform experiments so as to find x such that it has positive label and its cost function $f(x)$ is as small as possible.

To obtain labels of exemplars, we can do a batch of experiments, which evaluates the labels for a subset $S \subseteq E$. We let cardinality of S be constrained by some number K (i.e. $|S| \leq K$), because budget of doing experiments is usually limited. We measure the quality of S by

$$f^*(S) = \min_{x \in S: y(x)=1} f(x), \tag{1}$$

where we assume $\min \emptyset = \infty$. $f^*(S)$ measures the smallest cost function for positive labeled elements in the set S, and in our motivating application, that means the shortest length of desired peptides in the set of peptides to test.

Let b be a target value and we wish to find $S \subseteq E$ such that $f^*(S)$ is better than b. In our motivating application, we let b be the length of the shortest previously known desired peptide. Since $f^*(S)$ is an unknown random variable, we consider the following two probabilistic measures:

$$\begin{aligned} &\text{Probability of Improvement:} && \mathrm{P}^*(S) = \mathbb{P}(f^*(S) < b) \\ &\text{Expected Improvement:} && \mathrm{EI}(S) = \mathbb{E}[(b - f^*(S))^+] \end{aligned} \tag{2}$$

Given constraint on the cardinality of S, we wish to find S that maximizes one of these two measures. Let $g(S)$ be either $\mathrm{P}^*(S)$ or $\mathrm{EI}(S)$, and we can write our goal as

$$\max_{S \subseteq E : |S| \leq K} g(S). \tag{3}$$

3. Statistical model

We use Naive Bayes as the classification method, which, despite the name, has performed quite well in many cases. Let $X = (X_1, \ldots, X_n)$ be an instance with n features and Y be its label. By Bayes's Rule, we have:

$$\mathbb{P}(Y = y | X = x) = \frac{\mathbb{P}(X = x | Y = y)\mathbb{P}(Y = y)}{\mathbb{P}(X = x)} = \frac{\mathbb{P}(X = x | Y = y)\mathbb{P}(Y = y)}{\sum_{y'} \mathbb{P}(X = x | Y = y')\mathbb{P}(Y = y')}$$

The Naive Bayes classifier assumes that the presence or absence of a particular feature is unrelated to the presence or absence of any other feature, given the class variable, i.e.

$$\mathbb{P}(Y = y | X = x) = \frac{\prod_{j=1}^{n} \mathbb{P}(X_j = x_j | Y = y)\mathbb{P}(Y = y)}{\sum_{y'} \prod_{j=1}^{n} \mathbb{P}(X_j = x_j | Y = y)\mathbb{P}(Y = y)}$$

In our motivation application, we have a set of peptides, each with length less than or equal to L. Each peptide is a sequence of amino acids. We use a reduced alphabet for amino-acids, i.e., we group them into K groups. For each peptide, let A_i be the amino acid on position j, and let X_i be the class of this amino acid. For a specific enzyme, let $Y(x) = 1$ if peptide x is a substrate for that enzyme and 0 if not.

We let $\theta_{y,j}(k) = \mathbb{P}(X_i = k | Y(X) = y)$, for each $j = 1, \ldots, L$, $k = 1, \ldots, K$ and $y \in \{0, 1\}$. We further assume some known prior distribution $\mathbb{P}(Y(x) = y)$, $y \in \{0, 1\}$. Let θ be the full set of parameters $\theta_{y,j}(k)$, for $j = 1, \ldots, L$, $k = 1, \ldots, K$ and $y \in \{0, 1\}$. Then, given an unlabeled peptide, we can calculate its probability being a substrate as:

$$\mathbb{P}\left(Y(x)=1|\theta\right)=\frac{\mathbb{P}(Y(x)=1)\prod_j\theta_{1,j}(x_j)}{\left[\mathbb{P}(Y(x)=1)\prod_j\theta_{1,j}(x_j)\right]+\left[\mathbb{P}(Y(x)=0)\prod_j\theta_{0,j}(x_j)\right]} \tag{4}$$

We estimate the parameters $\theta_{y,j}(k)$ using Bayesian inference. We assume for each $j=1,\ldots,L$, $y\in\{0,1\}$, the vector $\theta_{y,j}\sim\text{Dirichlet}(\alpha_{y,j}(1),\ldots,\alpha_{y,j}(K))$. A good initial choice for the parameter vector $\alpha_{y,j}=(\alpha_{y,j}(1),\ldots,\alpha_{y,j}(6))$ can be choosing $\alpha_{y,j}(k)$ to be constant across k, and y, and to only depend upon j. Since amino acids further from the serine are less likely to have a strong influence on its activity, we choose this value to be 1 in the positions next to the serine and to increase as j moves further.

We further assume two hyper parameters γ_0 and γ_1 that characterize the distribution for $y=0$ and $y=1$ respectively. Then, with the prior distribution and hyper parameters, our posterior distribution is also Dirichlet. In particular, it is $\text{Dirichlet}(\alpha_{y,j}(1)+\gamma_y N_{y,j}(1),\ldots,\alpha_{y,j}(K)+\gamma_y N_{y,j}(K))$, where $N_{y,j}(k)$ counts how many peptides x in the training data with $Y(x)=y$ had $x_j=k$. That is, it counts how many peptides had amino acid j in class j.

Since our training data is expensive and highly skewed, we use the leave-one-out cross validation procedure to choose the optimal hyper parameters. For each setting of the hyper parameters, we obtain an receiver operating characteristic(ROC) curve using the result of the leave-one out procedure and choose the setting with highest AUC(area under curve).

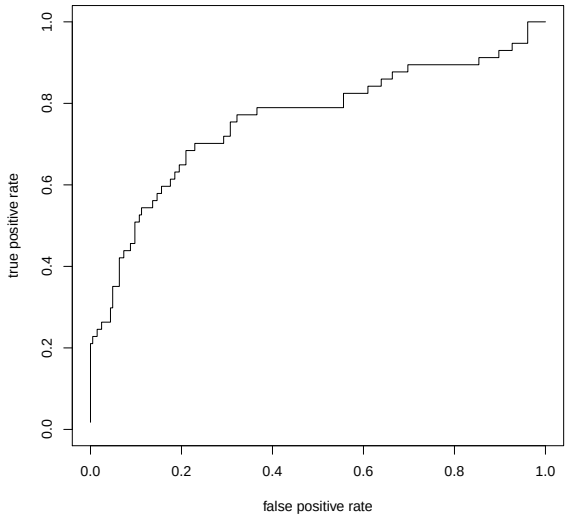

Figure 0 using data set #0 & #1

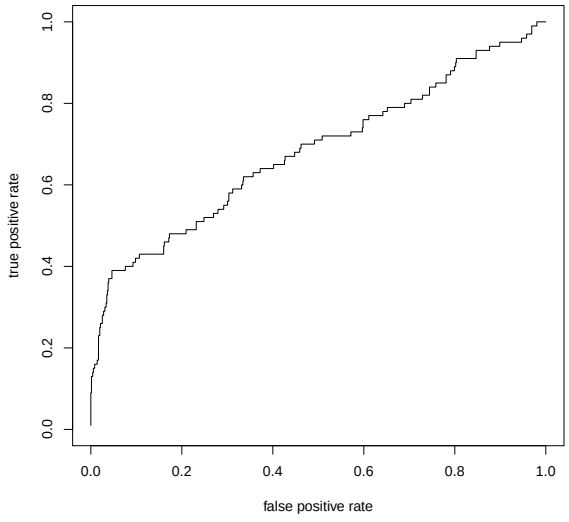

Figure 0 using data set #0,#1 & #2

Figure 1 ROC curve using leave-one-out cross validation

In Figure 1, note ROC curve to the left is better than the one to the right. This is because data set #2 was generated by our algorithm based on the previous two data sets, and due to the

exploration manner of our algorithm, data set #2 should lie in the region that is more challenging for the classifier. Thus it is reasonable that our classifer performs worse using all available data sets.

4. Optimization Algorithm

(3) is a combinatorial optimization problem with huge feasible region. This problem is considered NP-hard and yet there is no known algorithm that solves it quickly. We propose an approximation algorithm, which solves (3) using greedy heuristic, that is, starting with the empty set $S = \emptyset$, iteratively find element e such that

$$\arg\max_{e \in E \setminus S} g(S \cup \{e\}), \tag{5}$$

and incorporate it into S until $|S| = K$ for some chosen K. This approach reduces the size of search space dramatically from $|E|^{|S|}$ down to $|E| \times |S|$, and we prove that if objective function g is either P^* or EI, the proposed greedy algorithm is guaranteed to be near-optimal with a lower bound. Additionally, we show that under the statistical model described in section 3, we can formulate (5) as Mixed-Integer Nonlinear Programming (MINLP) and solve it efficiently using a off-the-shelf MINLP solver.

4.1. Lower bound of greedy algorithm

In this subsection, we show that when the objective function g in (5) is either P^* or EI, the proposed greedy algorithm has performance guarantee. The result is stated in the following:

THEOREM 1. *If objective function is probability of improvement (i.e $P^*(S)$) or expected improvement (i.e* $\mathrm{EI}(S)$*), the greedy algorithm is guaranteed to achieve a factor* $(1 - 1/e)(\approx 63\%)$ *of the optimal value.*

We prove the theorem using the following three lemmas. Lemma 1 is a result from the analysis of greedy heuristic in combinatorial optimization by Nemhauser, which provides lower bound of greedy heuristic given that objective function satisfies certain conditions. Lemma 2 and 3 show P^* and EI are the objective functions that satisfy conditions stated in Lemma 1, and thus greedy algorithm has a lower bound.

LEMMA 1. *NemHauser et al. (1978) If $F(S)$ is submodular, nondecreasing and $F(\emptyset) = 0$, the greedy heuristic always produces a solution whose value is at least $1 - [(K-1)/K]^K$ times the optimal value, where $|S| \leq K$. This bound can be achieved for each K and has a limiting value of $1 - 1/e$, where e is the base of the natural logarithm.*

LEMMA 2. *Probability of improvement $P^*(S)$ is submodular, nondecreasing and $P^*(\emptyset) = 0$.*

LEMMA 3. *Expected improvement* $\mathrm{EI}(S)$ *is submodular, nondecreasing and* $\mathrm{EI}(\emptyset)=0$.

Proof of Theorem 1. From Lemma 2 and 3, we know that P^* and EI are submodular, nondecreasing and their measure of the empty set is 0. From Lemma 1, we conclude that if the objective function g in (5) is P^* or EI, the greedy algorithm is guaranteed to achieve a factor of $(1-1/e)$ of the optimal value. □

Proof of Lemma 2. First we show $\mathrm{P}^*(\emptyset)=0$.

$$\mathrm{P}^*(\emptyset)=\mathbb{P}(f^*(\emptyset)<b)=\mathbb{P}(\infty<b)=0.$$

To show $\mathrm{P}^*(S)$ is nondecreasing, let $A\subseteq B\subseteq E$ where E is a finite set, then

$$\begin{aligned}
\mathrm{P}^*(B)&=\mathbb{P}(f^*(B)<b)\\
&=\mathbb{P}(f^*(B)<b|f^*(A)\geq b)\mathbb{P}(f^*(A)\geq b)+\mathbb{P}(f^*(B)<b|f^*(A)<b)\mathbb{P}(f^*(A)<b)\\
&=\mathbb{P}(f^*(B)<b|f^*(A)\geq b)\mathbb{P}(f^*(A)\geq b)+\mathbb{P}(f^*(A)<b)\\
&\geq\mathbb{P}(f^*(A)<b)\\
&=\mathrm{P}^*(A)
\end{aligned}$$

Lastly, we want to show $\mathrm{P}^*(S)$ is submodular. For $e\in E\backslash B$,

$$\begin{aligned}
&\mathrm{P}^*(A\cup\{e\})-\mathrm{P}^*(A)\\
&=\mathbb{P}(f^*(A\cup\{e\})<b)-\mathbb{P}(f^*(A)<b)\\
&=\mathbb{P}(f^*(A\cup\{e\})<b|f^*(A)<b)\mathbb{P}(f^*(A)<b)+\mathbb{P}(f^*(A\cup\{e\})<b|f^*(A)\geq b)\mathbb{P}(f^*(A)\geq b)-\mathbb{P}(f^*(A)<b)\\
&=\mathbb{P}(f^*(A)<b)+\mathbb{P}(f^*(A\cup\{e\})<b|f^*(A)\geq b)\mathbb{P}(f^*(A)\geq b)-\mathbb{P}(f^*(A)<b)\\
&=\mathbb{P}(f^*(A\cup\{e\})<b|f^*(A)\geq b)\mathbb{P}(f^*(A)\geq b)\\
&=\mathbb{P}(f(e)<b,y(e)=1|f^*(A)\geq b)\mathbb{P}(f^*(A)\geq b)\\
&=\mathbb{P}(f(e)<b,y(e)=1,f^*(A)\geq b)
\end{aligned}$$

Using similar argument,

$$\begin{aligned}
&\mathrm{P}^*(B\cup\{e\})-\mathrm{P}^*(B)\\
&=\mathbb{P}(f(e)<b,y(e)=1,f^*(B)\geq b)\\
&=\mathbb{P}(f(e)<b,y(e)=1,f^*(A)\geq b,f^*(B\backslash A)\geq b)
\end{aligned}$$

Therefore, $\mathrm{P}^*(A\cup\{e\})-\mathrm{P}^*(A)\geq\mathrm{P}^*(B\cup\{e\})-\mathrm{P}^*(B)$, thus $\mathrm{P}^*(S)$ is submodular. □

Proof of Lemma 3. First we show $\mathrm{EI}(\emptyset)=0$.

$$\mathrm{EI}(\emptyset)=\mathbb{E}[(b-f^*(\emptyset))^+]=\mathbb{E}[0]=0.$$

To show $\mathrm{EI}(S)$ is nondecreasing, let $A\subseteq B\subseteq E$ where E is a finite set. Since $f^*(B)\leq f^*(A)$, $b-f^*(B)\geq b-f^*(A)$, and $(b-f^*(B))^+\geq(b-f^*(A))^+$, therefore, $\mathbb{E}[(b-f^*(B))^+]\geq\mathbb{E}[(b-f^*(A))^+]$.

Lastly, we want to show $\mathrm{P}^*(S)$ is submodular. For $e \in E \backslash B$, consider $\mathbb{E}[(b - f^*(A \cup \{e\}))^+] - \mathbb{E}[(b - f^*(A))^+]$. We can write

$$(b - f^*(A \cup \{e\}))^+ = \begin{cases} (b - f^*(A))^+ & \text{if } y(e) = 0 \\ (b - \min\{f(e), f^*(A)\})^+ & \text{if } y(e) = 1 \end{cases}$$

Then

$$\begin{aligned}
&\mathbb{E}[(b - f^*(A \cup \{e\}))^+] - \mathbb{E}[(b - f^*(A))^+] \\
&= \mathbb{P}(y(e) = 1)\mathbb{E}[(b - \min\{f(e), f^*(A)\})^+ - (b - f^*(A))^+ | y(e) = 1] \\
&= \mathbb{P}(y(e) = 1)\mathbb{P}(f(e) < f^*(A) | y(e) = 1)\mathbb{E}[(b - e)^+ - (b - f^*(A))^+ | y(e) = 1, f(e) < f^*(A)] \\
&= \mathbb{E}[\mathbb{1}_{y(e)=1, f(e)<f^*(A)}((b - e)^+ - (b - f^*(A))^+)]
\end{aligned}$$

Since $f^*(A) \geq f^*(B)$, $\mathbb{1}_{y(e)=1,f(e)<f^*(A)}((b-e)^+ - (b - f^*(A))^+)) \geq \mathbb{1}_{y(e)=1,f(e)<f^*(B)}((b-e)^+ - (b - f^*(B))^+))$, thus

$$\mathrm{EI}(A \cup \{e\}) - \mathrm{EI}(A) \geq \mathrm{EI}(B \cup \{e\}) - \mathrm{EI}(B)$$

$\mathrm{EI}(S)$ is submodular. □

4.2. Probability of Improvement

We first show in Proposition 1 that (5) can be written into the form in which the objective function can be easily calculated using any statitical classifier. This is the general form of greedy algorithm for optimization over probability of improvement and can be used with any underlying statistical model. In addition, we show that, using the Naive Bayes classifier proposed in section 3, we can further formulate (5) as a MINLP, which can be solved efficiently using an off-the-shelf MINLP solver.

PROPOSITION 1. *If the objective function g is P^*, we can write* (5) *as*

$$\arg\max_{e \in E \backslash S, f(e) < b} \mathbb{P}(y(e) = 1 | y(x) = 0, \forall x \in S). \tag{6}$$

Proof of proposition 1.

$$\begin{aligned}
&\mathrm{P}^*(S \cup \{e\}) = \mathbb{P}(f^*(S \cup \{e\}) < b) \\
&= \mathbb{P}(f^*(S) < b) + \mathbb{P}(f^*(S) \geq b)\mathbb{P}(f(e) < b, y(e) = 1 | f^*(S) \geq b),
\end{aligned}$$

so (5) becomes

$$\max_{e \in E \backslash S} \mathrm{P}^*(S \cup \{e\}) = \max_{e \in E \backslash S} \mathbb{P}(f(e) < b, y(e) = 1 | f^*(S) \geq b). \tag{7}$$

Note that when $f(e) \geq b$, $\mathbb{P}(f(e) < b, y(e) = 1 | f^*(S) \geq b) = 0$, thus our algorithm will always propose e such that $f(e) < b$. Therefore, it is reasonable to assume that $f(x) < b$ for $\forall x \in S$, and $f^*(S) \geq b$ is equivalent to $y(x) = 0$ for $\forall x \in S$. Now we can write (7) as

$$\max_{e \in E \backslash S, f(e) < b} \mathbb{P}(y(e) = 1 \mid y(x) = 0, \forall x \in S). \quad \square$$

In our motivating application, we use the Naive Bayes model described in section 3, and we want to formulate (6) as a MINLP. First write equation (4) as

$$\mathbb{P}\left(Y(x)=1|\theta\right)=\frac{\prod_j \eta_j(x_j)}{\prod_j \eta_j(x_j)+\frac{\mathbb{P}(Y(x)=0)}{\mathbb{P}(Y(x)=1)}}, \tag{8}$$

where

$$\eta_j(x_j)=\frac{\theta_{1,j}(x_j)}{\theta_{0,j}(x_j)} \text{ for } \forall j\in\{1,\ldots,L\}.$$

Then we can write equation (6) as

$$\underset{e\in E\setminus S, f(e)<b}{\arg\max}\ \frac{\prod_j \eta_j(e_j)}{\prod_j \eta_j(e_j)+\frac{\mathbb{P}(Y(e)=0)}{\mathbb{P}(Y(e)=1)}}, \tag{9}$$

where

$$\eta_j(e_j)=\frac{\mathbb{P}(e_j|Y(e)=1,Y(x)=0,\forall x\in S)}{\mathbb{P}(e_j|Y(e)=0,Y(x)=0,\forall x\in S)}.$$

Now we can formulate equation (9) as a MINLP,

$$\begin{aligned}\max\quad & \frac{\prod_j \Sigma_k x_j(k)\eta_j(k)}{\prod_j \Sigma_k x_j(k)\eta_j(k)+\frac{\mathbb{P}(Y(x)=0)}{\mathbb{P}(Y(x)=1)}}\\ \text{s.t}\quad & k\in\{1,\ldots,K\}\\ & x_j(k)\in\{0,1\}\\ & \Sigma_k x_j(k)=1,\end{aligned} \tag{10}$$

where

$$x_j(k)=\begin{cases}1 & \text{if } e_j=k\\ 0 & \text{else.}\end{cases}$$

We summarize the algorithm in Algorithm 1.

Algorithm 1 *(Probability of Improvement)*

Require: *Inputs M, J, K, data set D and prior distribution of* $\theta_y\sim$ *Dirichlet*$(\boldsymbol{\alpha}_y), y\in\{1,0\}$

1: $S\leftarrow\emptyset$

2: Calculate posterior distribution of $\theta_1\sim$ *Dirichlet*$(\boldsymbol{\alpha}_1|\{x|x\in D, y(x)=1\})$.

3: ***for*** $m=1$ *to* M ***do***

4: $COUNT\leftarrow 0$

5: Calculate posterior distribution of $\theta_0\sim$ *Dirichlet*$(\boldsymbol{\alpha}_0|\{x|x\in D, y(x)=0\}\cup S)$.

6: ***loop***

7: Sample θ_1 *from Dirichlet*$(\boldsymbol{\alpha}_1|\{x|x\in D, y(x)=1\})$ *and* θ_0 *from Dirichlet*$(\boldsymbol{\alpha}_0|\{x|x\in D, y(x)=0\}\cup S)$.

8: $\eta\leftarrow\frac{\theta_1}{\theta_0}$

9: *Solve MINLP in equation* (10) *to find* x.

10: $COUNT \leftarrow COUNT + x$.

11: ***end loop***

12: ***for*** $j = 1$ *to* J ***do***

13: $e_j \leftarrow \arg\max_{k \in \{1,\ldots,K\}} COUNT_{kj}$

14: ***end for***

15: $S \leftarrow (S, e)$

16: ***end for***

4.3. Expected Improvement

The formulation for the case of expected improvement is similar to that of probability of improvement, and for our motivating application, we can also formulate (5) as a MINLP.

PROPOSITION 2. *If the objective function* g *is* EI, *we can write* (5) *as*

$$\arg\max_{e \in E \setminus S} c_0 \mathbb{P}_0(e)(b - f(e))^+ + \sum_{i=1}^{|S|} c_i \mathbb{P}_i(e)(f(x_i) - f(e))^+, \tag{11}$$

where

$$\mathbb{P}_0(e) = \mathbb{P}(y(e) = 1 | y(x) = 0, \forall x \in S),$$
$$\mathbb{P}_i(e) = \mathbb{P}(y(e) = 1 | y(x_i) = 1, y(x_j) = 0, \forall j < i, x_i, x_j \in S),$$

and $c_i (i = 0, \ldots, |S|)$ *are known coefficients.*

Proof of proposition 2. Since choosing e such that $f(e) \geq b$ has no contribution to the objective function, by using similar argument as dealing with probability of improvement, we argue that $f(x) < b$ for $\forall x \in S$. Thus

$$f^*(S) \begin{cases} = \infty & \text{if } y(x) = 0 \text{ for } \forall x \in S, \\ < b & \text{else.} \end{cases}$$

Now objective function we want to maximize becomes

$$\begin{aligned}
&\mathbb{E}\left[(b - f^*(S \cup \{e\}))^+\right] \\
&= \mathbb{E}[(b - f(e))^+ \mathbb{1}_{f^*(S)=\infty, y(e)=1}] + \mathbb{E}[(b - f^*(S \cup \{e\}))^+ \mathbb{1}_{f^*(S)<b}] \\
&= \mathbb{E}[(b - f(e))^+ \mathbb{1}_{f^*(S)=\infty, y(e)=1}] + \mathbb{E}[(b - f^*(S)) \mathbb{1}_{f^*(S)<b}] + \mathbb{E}[(f^*(S) - f(e)) \mathbb{1}_{y(e)=1, f(e)<f^*(S)<b}].
\end{aligned}$$

so

$$\begin{aligned}
&\max_{e \in E \setminus S} \mathbb{E}\left[(b - f^*(S \cup \{e\}))^+\right], \\
&= \max_{e \in E \setminus S, f(e) < b} \mathbb{E}[(b - f(e)) \mathbb{1}_{f^*(S)=\infty, y(e)=1}] + \mathbb{E}[(f^*(S) - f(e)) \mathbb{1}_{y(e)=1, f(e)<f^*(S)<b}].
\end{aligned} \tag{12}$$

For $e \in E \backslash S, f(e) < b$,

$$\begin{aligned}
&\mathbb{E}[(b-f(e))\mathbb{1}_{f^*(S)=\infty, y(e)=1}] = \mathbb{P}(y(e)=1, y(x)=0, \forall x \in S)(b-f(e)), \\
&\quad \mathbb{E}[(f^*(S)-f(e))\mathbb{1}_{y(e)=1, f(e)<f^*(S)<b}], \\
&\quad = \mathbb{E}[\mathbb{E}[(f^*(S)-f(e))\mathbb{1}_{y(e)=1, f(e)<f^*(S)<b}]|f^*(S)=l], \\
&\quad = \sum_{l \in L, f(e)<l} \mathbb{P}(y(e)=1|f^*(S)=l)(l-f(e))\mathbb{P}(f^*(S)=l),
\end{aligned} \tag{13}$$

where $L = \{f(x) : x \in S\}$. If we rank elements in S such that $f(x_i) \leq f(x_j), \forall i < j, x_i, x_j \in S$, we can write equation above as

$$\sum_{i=1}^{|S|} \mathbb{P}(y(e)=1, y(x_i)=1, y(x_j)=0, \forall j < i, x_i, x_j \in S)(f(x_i)-f(e))^+. \tag{14}$$

Substitute (13) (14) into (12), and note that $\mathbb{P}(y(e)=1, \mathcal{F}(x_1, \ldots, x_{|S|}) \propto \mathbb{P}(y(e)=1|\mathcal{F}(x_1, \ldots, x_{|S|})$ with known coefficient given S, we get (11) in proposition 2. $\square$

Note that each term in the summation of (11) has a similar structure as (6). Let $S = \{p^1, \ldots, p^{|S|}\}$, we can write (11) as a MINLP:

$$\begin{aligned}
\max \quad & \Sigma_{i=0}^{|S|} c_i \frac{\prod_j \Sigma_k x_j(k)\eta_j^i(k)}{\prod_j \Sigma_k x_j(k)\eta_j^i(k) + \frac{\mathbb{P}(Y(x)=0)}{\mathbb{P}(Y(x)=1)}} (f_i - f(e))^+ \\
\text{s.t} \quad & k \in \{1, \ldots, K\} \\
& x_j(k) \in \{0,1\} \\
& \Sigma_k x_j(k) = 1,
\end{aligned} \tag{15}$$

where

$$x_j(k) = \begin{cases} 1 & \text{if } e_j = k \\ 0 & \text{else,} \end{cases}$$

$$f_i = \begin{cases} b & \text{if } i = 0 \\ f(p^i) & \text{else,} \end{cases}$$

and c_i's are known coefficients. We summarize it in Algorithm 2.

Algorithm 2 *(Expected Improvement)*

Require: *Inputs M, J, K, data set D and prior distribution of* $\theta_y \sim Dirichlet(\boldsymbol{\alpha}_y), y \in \{1,0\}$

1: $S \leftarrow \emptyset$

2: ***for*** $m = 1$ *to* M ***do***

3: $COUNT \leftarrow 0$

4: ***if*** S *is not empty* ***then***

5: *Sort elements in* S *as* $\{p^1, \ldots, p^{|S|}\}$ *such that* $f(p^i) \leq f(p^j), \forall i < j$.

end if

Calculate posterior distribution of $\theta_1^0 \sim Dirichlet(\boldsymbol{\alpha}_1|\{x|x \in D, y(x) = 1\})$ *and* $\theta_0^0 \sim Dirichlet(\boldsymbol{\alpha}_0|\{x|x \in D, y(x) = 0\} \cup S)$.

for $i = 1$ *to* $|S|$ ***do***

Calculate posterior distribution of $\theta_1^i \sim Dirichlet(\boldsymbol{\alpha}_1|\{x|x \in D, y(x) = 1\} \cup \{p^i\})$ *and* $\theta_0^i \sim Dirichlet(\boldsymbol{\alpha}_0|\{x|x \in D, y(x) = 0\} \cup \{p^j|j < i\})$.

end for

loop

Sample $\theta_1^{i=0:|S|}$ *and* $\theta_0^{i=0:|S|}$ *from posterior distribution.*

$\eta^{i=0:|S|} \leftarrow \frac{\theta_1^{i=0:|S|}}{\theta_0^{i=0:|S|}}$

Solve MINLP in equation (15) *to find* x.

$COUNT \leftarrow COUNT + x$.

end loop

for $j = 1$ *to* J ***do***

$e_j \leftarrow \underset{k \in \{1,\ldots,K\}}{\arg\max}\ COUNT_{kj}$

end for

$S \leftarrow (S, e)$

end for

5. Performance and Validation (NOT READY YET)

5.1. Greedy Algorithm

5.1.1. Probability of Improvement We compare the performance of probability of improvement algorithm with two other methods: one method is to pick the most probable peptides based on posterior distribution, and the other is to randomly mutate known peptides x such that $y(x) = 1$ as our new recommendations. The benchmark was performed on a simulated dataset using the statistical model described in 3, and we show the result in 2

5.1.2. Expected Improvement Benchmark to be added here.

5.2. Performance on real experiment data.

Given the initial training data with size ?, we have performed two rounds of experiments using recommendations generated by Probability of Improvement algorithm. In the first round we performed experiments on

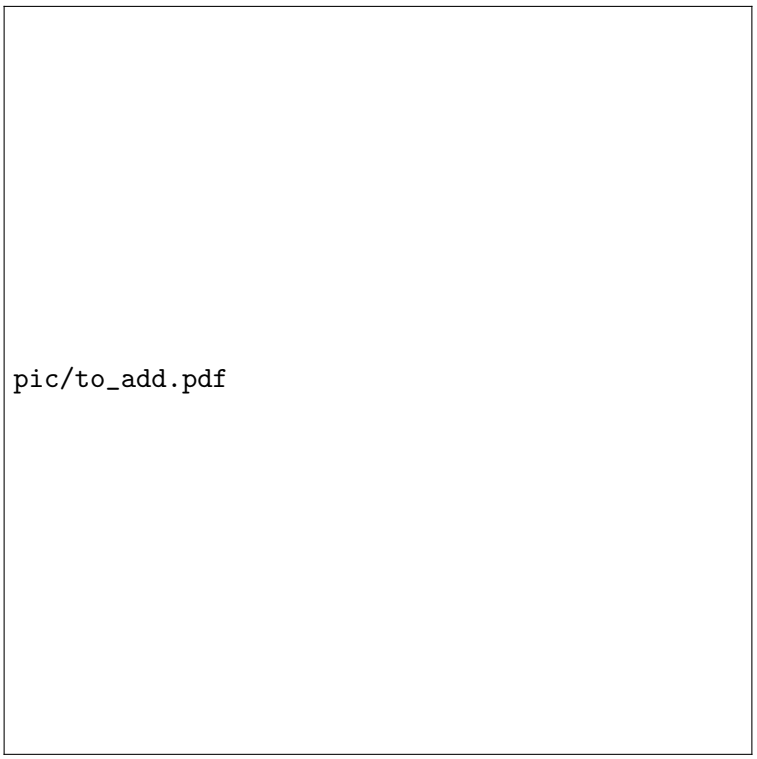

Figure 2 Benchmark of probability of improvement algorithm

6. Conclusion (NOT READY YET)

We presented two greedy heuristic algorithms solving active learning problem described in **??**, and proved that both of these two algorithms guarantee to achieve at least a factor (1-1/e) of the optimal value. From benchmark results, we further showed that these two algorithms outperformed another two heuristic search methods. In addition to theoretic results, We demonstrated effectiveness of our methods by applying them to optimal experimental design problem in material science, in which we are searching for shortest peptides that act as a substrate for some specific enzymes. We developed a Naive Bayes classifier to model the problem, and used our algorithm to propose candidates for testing by experiment. From the preliminary testing result made by our experimental collaborator, we have found a few short peptides that are very likely to be substrate of target enzymes.

References

Chen, Y., Krause, A., and Zurich, E. T. H. (2013). Near-optimal Batch Mode Active Learning and Adaptive Submodular Optimization. 28.

Ginsbourger, D., Riche, R. L., and Carraro, L. (2008). A multi-points criterion for deterministic parallel global optimization based on Gaussian processes. *In Intl. Conf. on Nonconvex Programming, NCP07, page ..., Rouen, France.*, pages 1–30.

Hoi, S. C. H., Jin, R., and Lyu, M. R. (2006a). Large-scale text categorization by batch mode active learning. *Proceedings of the 15th international conference on World Wide Web - WWW '06*, page 633.

Hoi, S. C. H., Jin, R., Zhu, J., and Lyu, M. R. (2006b). Batch mode active learning and its application to medical image classification. *Proceedings of the 23rd international conference on Machine learning - ICML '06*, pages 417–424.

Jones, D., Schonlau, M., and Welch, W. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, pages 455–492.

Negoescu, D. M., Frazier, P. I., and Powell, W. B. (2010). The Knowledge-Gradient Algorithm for Sequencing Experiments in Drug Discovery. *INFORMS Journal on Computing*, 23(3):346–363.

NemHauser, G., Fisher, M., and Wolsey, L. (1978). An Analysis of Approximations for Maximizing Submodular Set Functions. 14:265–294.

Feldman D, Schwan K (1990) *Who Put the Butter in Butterfly and Other Fearless Investigations into Our Illogical Language* (HarperPerennial, New York).

Fontecha J, Mayo I, Toledano G, Juarez M (2006) Triacylglycerol composition of projected designation of origin cheeses during ripening. *J. Dairy Sci.* 89:882–887.

Geisel TS (1984) *The Butter Battle Book* (Random House, New York).

Hodson L, Skeaff CM, Chisholm W-AH (2001) The effect of replacing dietary saturated fat with polyunsaturated or monounsaturated fat on plasma lipids in free-living young adults. *Eur. J. Clinical Nutrition* 55(10):908–915.

Tholstrup T, Raff M, Basu S, Nonboe P, Sejrsen K, Staarup EM (2006) Effects of butter and monounsaturated fatty acids into lipid classes, plasma C-reactive protein, oxidative stress, hemostatic variables, and insulin in healthy young men. *Amer. J. Clinical Nutrition* 83:237–243.

Torbica A, Jovanovic O, Pajin B (2006) The advantages of solid fat content determination in cocoa butter and cocoa butter equivalents by the Karlshamns method. *Eur. Food Res. Techn.* 222:385–391.