1 Introduction

2 Problem Statement and Application

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

We have two enzymes (Sfp from *Bacillus subtilis*, and PaAcpH from *Pseudomonas aeruginosa*), and a collection of peptides that can potentially act as a substrate for one or both of these enzymes. Our goal is to find a peptide that acts as a substrate for both of these enzymes, and is as short as possible.

To support this goal, we can do lab experiments, in which we synthesize a peptide and test, for each enzyme, whether it is a substrate or not. We need to find a policy that suggests which peptide to synthesize and test next, so as to reach our goal with as few experiments as possible.

Experiments have parallel setup, thus can be done with a batch of peptides at a time, and so the algorithm suggests a batch of peptides at a time, waiting for the results from the experiment before suggesting the next batch of peptides. A large collection of peptides would be considered by the algorithm for potential synthesis and testing, e.g., all peptides with length less than a given threshold. That is, we would consider more peptides than just those that are sub-peptides of peptides from the literature known to be substrates for one enzyme.

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 space of peptides. 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. Our goal is to perform experiments so as to find x such that it has positive label and its cost function $f(x)$ is minimum.

To obtain labels of exemplars, we can do a batch of experiments, which evaluate a subset $S \subseteq E$ and obtain labels at each time. We measure quality

of S by

$$f^*(S) = \begin{cases} \min\limits_{x \in S : y(x)=1} f(x), & \text{if } \{x \in S : y(x) = 1\} \neq \emptyset, \\ \infty, & \text{if } \{x \in S : y(x) = 1\} = \emptyset. \end{cases} \tag{1}$$

Let b be a target value and we wish to find $S \subseteq E$ such that $f^*(S)$ is, in some sense, better than b. Specifically, we consider the following two measures:

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

We wish to find S that maximize one of these two measures. Let $g(S)$ be either $\mathrm{P}^*(S)$ or $\mathrm{EI}(S)$ and let the cardinality of S be the only constraint on S. Our goal is then:

$$\max_{S \subseteq E : |S| < k} g(S) \tag{3}$$

3 Solution Method

We solve (3) by greedy algorithm, that is, starting with empty set $S = \emptyset$, find element $e = \arg\max_e g(S \cup \{e\}) - g(S)$ to include in S iteratively until $|S| = k$ for some chosen k. We can show the greedy solution has lower bound, which is a factor $(1 - 1/e)$ of the optimal objective value.

3.1 Lower bound of greedy algorithm

3.2 Algorithm

Suppose we have chosen $S = \{x_1, x_2, \ldots, x_n\}$ as a batch of points we are going to evaluate next, and if we want to incorporate one more point e, which is distinct from $x_1, x_2, \ldots, x_n$, such that the expected improvement increases most, we use the following criterion to find e:

$$\arg\max_{e \in E \setminus S} \mathbb{E}\left[(b - f^*(S \cup \{e\}))^+\right] \tag{4}$$

From (1) we write expected improvement part in (3) as

$$\begin{aligned}\mathbb{E}\left[(b - f^*(S))^+\right] &= \mathbb{E}[b - \min_{x \in S \cup \{e\}: y(x)=1} f(x)] \\ &= \begin{cases} \mathbb{E}[b - f^*(S)] & \text{if } y(e) = 0 \ , \\ \mathbb{E}[b - \min\{f(e), f^*(S)\}] & \text{if } y(e) = 1 \ , \end{cases} \\ &= \mathbb{E}[b - f^*(S) + \mathbb{1}_{\{y(e)=1, f(e)<f^*(S)\}}[f^*(S) - f(e)]]\end{aligned}$$

After some algebra we can write (4) as

$$\begin{aligned}&\arg\max_{e \in E \setminus S} \mathbb{E}[\mathbb{1}_{\{y(e)=1, f(e)<f^*(S)\}}[f^*(S) - f(e)]] \\ &= \arg\max_{e \in E \setminus S} \sum_{i=1}^{|S|} \mathbb{P}(y(e) = 1, y(x_i) = 1, y(x_j) = 0, \forall j < i)[f(x_i) - f(e)]^+ \\ &+ \mathbb{P}(y(e) = 1, y(x_j) = 0, \forall j)[b - f(e)]^+\end{aligned}$$

where $f(x_i) <= f(x_j)$ for $\forall i < j, x_i, x_j \in S$. Since

$$\begin{aligned}&\mathbb{P}(y(e) = 1, y(x_i) = 1, y(x_j) = 0, \forall j < i) \\ &= \mathbb{P}(y(x_1) = 0)\mathbb{P}(y(x_2) = 0 | y(x_1) = 0) \ldots \mathbb{P}(y(e) = 1 | \mathcal{F}(x_1, x_2, \ldots, x_i) \\ &\propto \mathbb{P}(y(e) = 1 | \mathcal{F}(x_1, x_2, \ldots, x_i)\end{aligned}$$

4 Application

4.1 Statistical Method

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.

$$\mathbb{P}(Y = y|X = x) = \frac{\prod_{i=1}^{n} \mathbb{P}(X_i = x_i|Y = y)\mathbb{P}(Y = y)}{\sum_{y'} \prod_{i=1}^{n} \mathbb{P}(X_i = x_i|Y = y)\mathbb{P}(Y = y)}$$

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 i, 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 $\theta_{y,i}(j) = \mathbb{P}(X_i = j|Y(X) = y)$, for each $i = 1, \ldots, L$, $j = 1, \ldots, K$ 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 $\theta_{y,i}(j)$, for $i = 1, \ldots, L$, $j = 1, \ldots, K$ and $y \in \{0, 1\}$. Then, given an unlabeled peptide, we can calculate its probability being a substrate as:

$$\mathbb{P}\left(Y(x) = 1|\theta\right) = \frac{\mathbb{P}(Y(x) = 1) \prod_i \theta_{1,i}(x_i)}{[\mathbb{P}(Y(x) = 1) \prod_i \theta_{1,i}(x_i)] + [\mathbb{P}(Y(x) = 0) \prod_i \theta_{0,i}(x_i)]}$$

We estimate the parameters $\theta_{y,i}(j)$ using Bayesian inference. We assume for each $i = 1, \ldots, L$, $y \in \{0, 1\}$, the vector $\theta_{y,i} \sim \text{Dirichlet}(\alpha_{y,i}(1), \ldots, \alpha_{y,i}(K))$. A good initial choice for the parameter vector $\alpha_{y,i} = (\alpha_{y,i}(1), \ldots, \alpha_{y,i}(6))$ can be choosing $\alpha_{y,i}(j)$ to be constant across j, and y, and to only depend upon i. 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 i 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 $\text{Dirichlet}(\alpha_{y,i}(1) + \gamma_y N_{y,i}(1), \ldots, \alpha_{y,i}(K) + \gamma_y N_{y,i}(K))$, where $N_{y,i}(j)$ counts how many peptides x in the training data with $Y(x) = y$ had $x_i = j$. That is, it counts how many peptides had amino acid i 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).

[Put two ROC curves here, one for leave-one-out and one for using the 1st data set as training and 2nd data set as test, to be continued ...]