10 Computer Science: Imagination Sampling

Eric Holloway

Baylor University

Abstract

Machine Learning, despite its name, can incorporate an oracle. One common form of oracle interaction is known as *active learning.* Active learning samples $\{x, y\}$ from an oracle for f (the function to be learned). *Imagination sampling* is the converse of active learning. Imagination sampling asks an oracle for hypotheses h from $\mathcal{H}$ (hypothesis space). In this paper imagination sampling is compared with a purely algorithmic approach to determine if oracle interaction outperforms a purely algorithmic approach. The theoretical basis for imagination sampling is developed and illustrated by simulating an oracle.

1 Introduction

The effectiveness of a machine learning algorithm can be measured by the number of samples required to match f (problem) and $h \in \mathcal{H}$ (hypothesis space). One approach to improving machine learning is to incorporate oracle interaction. An oracle is an external source of information. Machine learning approaches that incorporate oracle interaction focus on how oracle interaction significantly improves the sampling of f. This is known as *active learning*, along with a variant known as *guided search* described below. However, there is no research into whether an oracle improves sampling of $\mathcal{H}$.

2 Background

A sub-field of active learning is guided search. In guided search an oracle not only labels data items, but also identifies data items for labeling. Attenberg et al. have shown that a guided search is superior to active learning in the area of website classification (Attenberg and Provost, 2010). It is useful for domains where the target classification has a very small number of instances compared to the general population. Additionally, guided search is good for classifications that are formed from disjunctive subclasses. These are common problems encountered when putting active learning to use (Attenberg and Provost, 2011). A similar approach is oracle guided feature selection for particular classes. This is known as *guided feature labeling* (Attenberg, Melville, and Provost, 2010). Most recently, Attenberg has proposed a gamified system called *beat the machine* (BTM). BTM uses oracles to identify the unknown unknowns of a machine learning model. (See Attenberg, Ipeirotis, and Provost, 2015 for more information.)

Classification model learning is divided into three different areas:

1. Query to classify data: $y = f(x)$
2. Select data to be classified: $x \in \mathcal{D}$
3. Select predictive model: $h \in \mathcal{H}$

There are four different approaches to optimizing these areas:

A. Random

B. Algorithmic

C. Oracle

D. Ground truth

3D is the goal of machine learning—an accurate prediction model. Traditional machine learning is a combination of 2A (random sampling), 3B, and 1D. Semi-supervised learning introduces 1B. The learned model is used to classify further data for learning.

3A is used in the *No Free Lunch Theorem* to characterize the expected performance of machine learning.

Active learning incorporates 1C (oracle labeler) and 2B into traditional machine learning. Guided search adds 2C. Finally, BTM and its predecessor *equivalence querying* (Angluin, 1988) are another form of 2C.

Table 10.1 shows the areas covered in the literature. Most of the combinations have been addressed.

Another approach is for an oracle to sample from $\mathcal{H}$, the hypothesis space. This approach is 3C. Sampling from $\mathcal{H}$ uses the oracle's imagination so this approach is called "imagination sampling."

Oracle sampling of $\mathcal{H}$ is unexplored in the literature. This project investigates whether an oracle improves the sampling of $\mathcal{H}$ by comparing the effectiveness of "imagination sampling" with algorithmic approaches.

Table 10.1: Research grid

	A	**B**	**C**	**D**
1	X	X	X	X
2	X	X	X	
3	X	X		X

3 Testing “Imagination Sampling”

The No Free Lunch Theorem (NFLT) states that all algorithms have exactly the same performance when averaged over all problem domains. While particular algorithms perform better on particular problem domains, it is extremely unlikely to pair the right algorithm with the right domain. Consequently, the NFLT is used in two different ways to test for non-algorithmic learning in the oracle: (1) in terms of the problem domain, and (2) of the learning algorithm.

As stated, it is unlikely a particular algorithm will do well on a randomly selected problem. If imagination sampling performs well on an arbitrary problem domain, then the oracle is highly likely to have a non-algorithmic learning ability. The oracle cannot have information about the dataset that the algorithms do not have.[1]

Similarly, for a particular problem, it is unlikely a randomly selected algorithm will do well. The learning problem is constructed so algorithmic learning cannot perform better than random sampling that is averaged over many problems. This is a No Free Lunch (NFL) construction.

In either case, if the oracle performs better than algorithmic approaches, this shows imagination sampling is generally better than purely algorithmic learning.

The NFL construction is a hypothesis space that shatters every dataset. A hypothesis space shatters a dataset when it can represent any possible labeling of items in the dataset. Algorithmic learning based on in-sample error (E_{in}) is not possible in this hypothesis space, since a hypothesis with zero in-sample error can always be found. However, learning based on compression is possible and will be covered.

The oracle is compared to two classes of algorithms. The first class samples hypotheses over the entire hypothesis space. The second class learns using a learnable subset. The subset is learnable because it does not cover all possibilities and in-sample error cannot be completely minimized. The oracle should be more effective than the first class, and may be more effective than the second class. Effectiveness is measured by the lowest out-of-sample error, E_{out}, obtained.

A second issue is identifying when the oracle has found a good hypothesis. Since

[1] This sounds like a contradiction since an oracle is defined as an external source of information. If the oracle does not have information, then it is not a source of information. However, some oracles can create information. Algorithms cannot create information. So, relying on an oracle that does not initially have information can still be useful if the oracle can create information.

the NFL construction shatters every dataset, overfitting is more likely to occur than generalization. This means the hypothesis cannot be rated in-sample using E_{in} and E_{test}, the test dataset. Instead, the hypothesis must be rated by its conciseness. Conciseness is measured by Kolmogorov complexity (KC).

This paper uses KC to identify when the oracle has found a good hypothesis. Only an upper bound for KC can be calculated. Consequently, a gradient-based approach is used with the upper bound to identify a good hypothesis.

4 Representation

As discussed previously, an NFL construction for the learning problem is a hypothesis space that shatters every dataset. To shatter a dataset, we need a hypothesis space that can represent all possible classifications for a dataset. The multibox is a hypothesis space that represents all possible classifications for any dataset, and can shatter any dataset. This makes the multibox an NFL construction. The multibox is defined after the problem domain is described.

4.1 Problem Domain

The problem domain for this project is a 2D discrete grid. The target function f is a particular classification of the cells, such as in Figure 10.1. The classification in the image is represented by white and black cells. A hypothesis h from the hypothesis space $\mathcal{H}$ is a classification region represented by light and dark gray cells. The classification region does not have to classify the entire problem domain.

4.2 Multibox Definition

A multibox is a set of n coordinate 5-tuples. The fifth element is the box's binary classification. Each tuple defines an axis-aligned, rectangular box. The coordinates are integers. The set of all boxes is MB. The set of boxes in the hypothesis is MB_g.

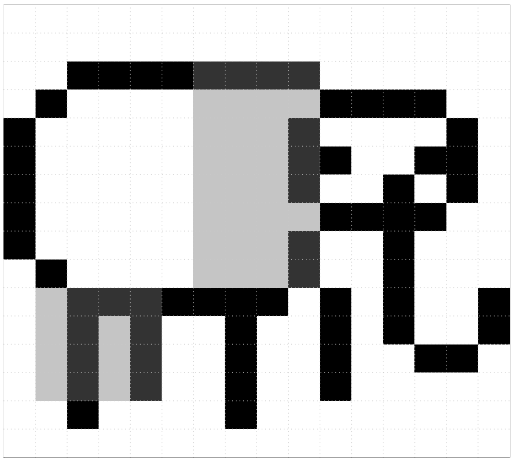
$$MG_g = \{Z, Z, Z, Z, 0 \text{ or } 1\}^M$$

The parameters of each box are referenced by a subscript, box_{0-4} with box_4 as the 0 or 1 classification.

The oracle places multiple axis-aligned, rectangular boxes on the data to form a hypothesis. Each box classifies samples according to a single classification. For example, a box that classifies samples as "ones" will classify all contained samples as ones. If boxes overlap, samples are classified according to the last placed box. The classification algorithm is shown in Algorithm 1.

We can see the multibox represents any classification if each box is the size of an individual cell in Figure 10.1. This means the multibox can shatter all datasets, and

Figure 10.1: Problem domain with an example target function (white and pure black) and classification region (light gray and dark gray).



algorithmic learning is not possible. Algorithmic learning based on error minimization cannot happen because an error can always be taken to zero. Learning through compression cannot happen as compression is not computable in general.

Due to the NFLT the odds of a learning approach performing well on a particular problem are too small to happen by chance. This is still true even if most or all real world problems are learnable because the NFLT applies to learning approaches as well as problems. If an approach does well on a decent number of different datasets, and it is selected independently from the problem domain, then it is superior to algorithmic learning.

Consequently, if experiments show imagination sampling is superior to algorithmic approaches for multibox classification and other classification tasks then it is superior to algorithmic learning in general.

Algorithm 1 Multibox classifier

```
procedure CLASSIFY($x$)
    $result \leftarrow$ NULL
    for all $box \in MB_g$ do
        if $x \in box$ then
            $result \leftarrow box_4$
    return $result$
```

5 Methodology

To test whether imagination sampling is superior to algorithmic techniques, the oracle is compared to two classes of algorithms. The first class learns over the entire hypothesis space. The second class learns a learnable subset.

5.1 First Class of Algorithm

The algorithms in the first class are random box placement and guided box placement. The random approach places boxes randomly until it has placed M boxes (see Algorithm 2). Each box's classification is assigned randomly.

Algorithm 2 Random placement

```
procedure RandomPlacement($X, M$)
    $MB_g \leftarrow$ list
    for all $iteration \in 1$ to $M$ do
        $x_{00} \leftarrow \text{Uniform}(\min_i X_{0i}, \max_i X_{0i})$
        $x_{01} \leftarrow \text{Uniform}(\min_i X_{0i}, \max_i X_{0i})$
        $x_{10} \leftarrow \text{Uniform}(\min_i X_{1i}, \max_i X_{1i})$
        $x_{11} \leftarrow \text{Uniform}(\min_i X_{1i}, \max_i X_{1i})$
        $class \leftarrow \text{Choose}(0, 1)$
        $MB_g.\text{Append}(\{x_{00}, x_{01}, x_{10}, x_{11}, class\})$
    return $MB_g$
```

The guided box placement algorithm is initiated with the random algorithm by placing N boxes. Then, it selects a subset M that maximizes the significance score in Algorithm 3.

The first term, *accuracy*, prioritizes boxes that contain an unlikely number of samples. The second term, *correctness*, prioritizes boxes that are likely to correctly classify a sample.

α and β are tunable parameters between zero and one. When both parameters are zero, guided placement reduces to random placement. Two parameters are used instead of just setting $\beta = 1 - \alpha$, otherwise it is not possible to set both to zero and achieve a random box placement. Consequently, both the random and guided algorithms can be described by Algorithm 4.

5.2 Second Class of Algorithm

The second class of algorithms uses a learnable subset of $\mathcal{H}$. The algorithm in the second class of algorithms is the Set Cover Machine (SCM) algorithm (Marchand and Taylor, 2003). The SCM algorithm finds a minimal multibox of square boxes that cover all of one class of samples. A regularization parameter p penalizes classification

Algorithm 3 Significance scoring

```
procedure $\textsc{Significance}(X, Y, box, \alpha, \beta)$
    $X_{width} \leftarrow \max_i X_{0i} - \min_i X_{0i}$
    $X_{height} \leftarrow \max_i X_{1i} - \min_i X_{1i}$
    $X_{area} \leftarrow X_{width} X_{height}$
    $\lambda \leftarrow \frac{|X|}{X_{area}}$
    $box_{width} \leftarrow box_1 - box_0$
    $box_{height} \leftarrow box_3 - box_2$
    $box_{area} \leftarrow \textsc{abs}(box_{width} box_{height})$
    $k \leftarrow \frac{|x \in box|}{box_{area}}$
    $probability \leftarrow \textsc{poisson}(k, \lambda)$
    $accuracy \leftarrow -\alpha \textsc{log}(probability)$
    $probability \leftarrow \textsc{abs}(1 - box_4 - \frac{\sum_{y \in box} y}{|y \in box|})$
    $correctness \leftarrow -\beta \textsc{log}(probability)$
    return $accuracy + correctness$
```

Algorithm 4 Generalized box placement

```
procedure $\textsc{BoxPlacement}(X, Y, N, M, \alpha, \beta)$
    $MB_g \leftarrow \textsc{RandomPlacement}(X, N)$
    $scores \leftarrow \textsc{LookupTable}$
    for all $box \in MB_g$ do
        $scores[\textsc{Significance}(X, Y, box, \alpha, \beta)] \leftarrow box$
    $\textsc{SortDescending}(scores)$
    $MB_g \leftarrow scores.\textsc{values}[1 \dots M]$
    return $MB_g$
```

error in the SCM. p can range from zero to ∞. In this experiment, $p = \infty$ so there is no misclassification error. Other values were tried but did not noticeably improve accuracy, and they greatly increased computation time.

The standard SCM algorithm classifies samples outside of the set cover as the alternative class. The SCM algorithm is modified for this experiment to only classify the samples covered by the multibox and ignore samples outside the multibox. In this way, it is similar to a deterministic version of generalized box placement in Algorithm 3.

The standard SCM algorithm has an $O(N^4)$ complexity. To improve runtime, the sample count for the SCM is reduced to $\sqrt{N}$, while the other approaches still use N samples.

6 Experiment

The dataset comes from a BNP insurance competition on Kaggle.com. and has 130 anonymized features and 130k samples. Anonymization keeps the oracle from having access to domain knowledge.

The four approaches are trained and validated across a range of sample sizes for E_{in} and E_{test}, from 100 to 500 samples using 50 sample increments. A separate set of 1000 samples is used for calculating E_{out}. The oracle's multibox placement is compared to the algorithmic approaches using E_{out} in a batch after all experiments are completed. E_{out} is calculated using Algorithm 1. $s_{\mathcal{C}}$ represents the E_{out} samples covered by the hypothesis.

$$E_{out} = \sum_{x,y \in s_{\mathcal{C}}} \text{abs}\left(\frac{y - \text{classify}(x)}{|s_{\mathcal{C}}|}\right)$$

1. The samples are drawn randomly without replacement from the dataset to create the training set. The samples are reduced to remove samples that are missing data and are further reduced so there are an equal number of both classes. This can result in a sample set containing much less than the initial amount. For instance, if 100 samples are initially drawn, cleaning and equalizing can leave only 30 samples.

2. The samples are preprocessed to be centered, normalized, and whitened. The parameters for the preprocessing are recorded for use on the E_{test} and E_{out} samples.

3. The two algorithmic multibox hypotheses are generated with $M = 5$, meaning each hypothesis consists of 5 boxes. This number is selected to be small so the resulting hypothesis has a low complexity. For the guided multibox selection, $N = 100$. This means 100 boxes are generated randomly. Then the 5 best boxes

are selected for the hypothesis, as shown in Algorithm 4. The guided multibox algorithm has parameters $\alpha = 0.5$ and $\beta = 0.5$ in Algorithm 3.

4. The oracle (human user) visually selects a set of boxes it thinks will create a good classification hypothesis. Figure 10.2 is an example MB_g created by an oracle.

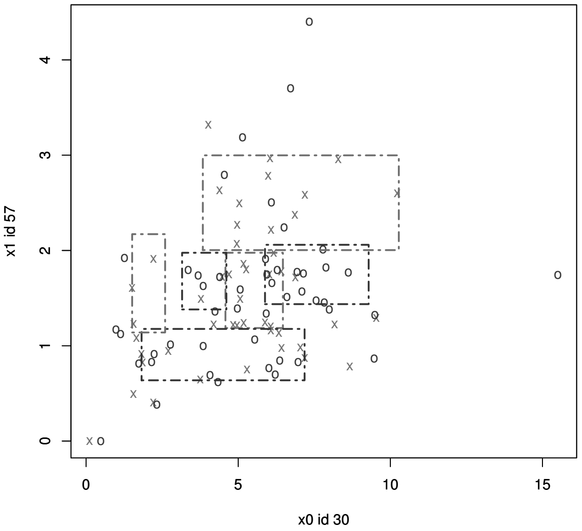


Figure 10.2: Example of MB_g generated by oracle.

5. Each hypothesis is then evaluated for classification error on a validation and test dataset. Both datasets use different samples, which are also cleaned, equalized, and preprocessed. The training dataset values are used for preprocessing.

7 Imagination Sampling Results

The experiment is repeated 351 times and is carried out across sample sizes from 100 to 500, with 50 sample increments. In each experiment, the approaches are compared based on accuracy, $1-E_{out}$. The outcome of an experiment identifies which hypothesis of the four approaches has the best accuracy. The approach with the highest accuracy wins the experiment.

The overall results as well as wins, min, mean, and max accuracy for each sample size are shown in Table 10.2. Some numbers are truncated to fit the table. As the results show, the oracle performs the best.

Table 10.2: Results from 351 experiments.

SCM	**All**	100	150	200	250	300	350	400	450	500
Wins	91	**12**	10	**12**	10	11	**12**	6	9	9
Max	1	.82	.64	.75	.83	.66	1	.70	.59	.60
Mean	.50	.51	.51	.50	.51	.48	.50	.48	.48	.50
Min	0	.33	.40	.14	.33	.22	.33	0	0	.23
Oracle	**All**	100	150	200	250	300	350	400	450	500
Wins	**114**	8	10	9	**11**	**16**	**12**	**18**	**15**	**15**
Max	.75	.60	.58	.63	.75	.70	.61	.62	.62	.60
Mean	.50	.49	.49	.49	.51	.50	.50	.51	.50	.51
Min	.33	.36	.33	.37	.42	.40	.39	.43	.41	.38
Rand	**All**	100	150	200	250	300	350	400	450	500
Wins	71	10	7	8	7	8	8	4	9	10
Max	1	.75	.65	1	1	.75	.64	.69	.66	.66
Mean	.49	.50	.50	.51	.49	.50	.47	.48	.50	.50
Min	0	.32	.33	.33	0	0	0	.34	.33	.39
Guided	**All**	100	150	200	250	300	350	400	450	500
Wins	75	9	**12**	10	**11**	4	7	11	6	5
Max	.74	.63	.67	.74	.55	.58	.52	.55	.54	.56
Mean	.49	.49	.50	.51	.49	.48	.49	.49	.49	.49
Min	.31	.33	.40	.36	.34	.32	.41	.31	.40	.45

The interesting trend in the results is that as the number of samples increases, the oracle's performance improves relative to the other algorithms. It is also evident that, in general, none of the approaches did very well.

8 Identifying Good Oracle Hypotheses

Using the multibox model means E_{in} and E_{test} cannot be used to evaluate hypothesis accuracy. Since the model shatters every dataset, a hypothesis can always be found that makes $E_{in} = 0$ and $E_{test} = 0$. We cannot trust these error metrics to identify a good hypothesis.

Instead, we use algorithmic complexity theory to identify a good hypothesis. Identifying a good hypothesis is a compression problem (Kearns and Vazirani, 1994). But, calculating an arbitrary bitstring's optimum compression is undecidable (Kolmogorov, 1998). However, an upper bound on compression can be calculated, and consequently, an oracle's good hypotheses can be identified by using the upper bound with a gradient approach.

To derive the upper bound, we must first define the problem domain. The problem domain is finite and discrete. There are two classes and an equal count of both classes.

- $\mathcal{A}$ is the complete sample space. For example, the problem domain is a 2D grid. Each cell in the grid is a sample. In this case, $\mathcal{A}$ is all of the grid cells.
- h is the hypothesis being examined. $\mathcal{H}$ is the hypothesis space. $\mathcal{H}$ represents all classifications on the dataset. *diversity*$(\mathcal{H})$ counts the unique classifications represented by $\mathcal{H}$. Thus, *diversity*$(\mathcal{H}) = 2^{|\mathcal{A}|}$ and $|\mathcal{H}| \geq 2^{|\mathcal{A}|}$.
- h does not necessarily classify every cell in $\mathcal{A}$. The set of cells that are classified by h is $\mathcal{C}$.
- The set of samples is $\mathcal{S}$. The subset of $\mathcal{S}$ in $\mathcal{C}$ is $s_{\mathcal{C}} \in \mathcal{S}$.
- The Kolmogorov complexity of h is $K(h)$. $\mathcal{H}_k$ is the set of hypotheses where $K(h) = k$. As such, *diversity*$(\mathcal{H}_k) \leq 2^k$ and $|\mathcal{H}_k| = 2^k$. For a classification region $\mathcal{C}$ there are $2^{|\mathcal{C}|}$ different classifications. $\mathcal{H}_k$ can only describe, at most, 2^k different classifications. So $\mathcal{H}_k$ covers, at most, $2^{k-|\mathcal{C}|}$ possible classifications.
- If the hypothesis correctly classifies all samples, then the conditional algorithmic complexity of the samples given the hypothesis is zero, $K(s_{\mathcal{C}}|h) = 0$. However, if there are misclassifications, then the conditional complexity is non-zero because extra information is needed to describe the misclassified samples. The combination of both hypothesis complexity and sample conditional complexity is *classification complexity.* The expression for classification complexity is $CC(s_{\mathcal{C}}, h) = K(h) + K(s_{\mathcal{C}}|h)$.
- The calculable upper limit for classification complexity with no misclassifications is $D(h) \geq CC(s_{\mathcal{C}}, h)$.

In this analysis, $\mathcal{H}$ is the multibox classifier. (See Algorithm 1.) For an $h \in \mathcal{H}$, $|h|$ is the number of boxes in h. Since each box in h can only have a single classification, then *diversity*$(h) \leq 2^{K(h)} \leq 2^{|h|}$.

Note, $K(h) \leq |h|$ because there can be a shorter description of h than to enumerate all the boxes. As an example, the boxes form an infinitely long diagonal line. The equation for the line has finite Kolmogorov complexity. The enumeration of the boxes is an infinitely long bitstring. Thus, trivially $K(h) < |h| = \infty$.

As an upper bound on $CC(s_{\mathcal{C}}, h)$, we have $D_1(h) = |h| - \sum_{box \in h} log_2(1 - E_{in}^{box}) * |box|$. The notation $|box|$ counts how many samples from $\mathcal{S}$ are in the box. If all boxes have $E_{in}^{box} = 0$, then there is no need for the second term, and $CC(s_{\mathcal{C}}, h) = K(h)$. However, if the boxes do have an $E_{in}^{box} > 0$, there are a couple of key cases to consider.

1. If $E_{in}^{box} = \frac{1}{2}$, then the box has a 50/50 chance of correct classification. If we go back to our definition of $\mathcal{H}_k$, we see it can describe, at most, $2^{k-|\mathcal{C}|}$ of the

classifications in $\mathcal{C}$. For our subset of samples $s_{\mathcal{C}}$, $\mathcal{H}_k$ can describe $2^{k-|s_{\mathcal{C}}|}$ classifications. If $k = |s_{\mathcal{C}}|$, then $2^{k-|s_{\mathcal{C}}|} = 1$. This means $\mathcal{H}_{k=|s_{\mathcal{C}}|}$ can correctly classify any set of samples of that size and cannot generalize. A hypothesis that cannot generalize has a 50/50 chance of correct classification. Consequently, if $E_{in}^{box} = \frac{1}{2}$, then this is equivalent to $\mathcal{H}_{k=|s_{\mathcal{C}}|}$ for the samples in the box.

2. If $E_{in}^{box} = 1$, then this box is not an acceptable classifier. However, the classification of the box cannot always be changed to turn it into a good classifier. In this case, there are only two classifications, so the box can be fixed. But in general, there are an unlimited number of classifications. In the unlimited case, a box that misclassifies everything cannot be fixed to provide a good classifier.

All these criteria are met by using $log_2(1 - E_{in}^{box}) * |box|$ for the second term.
If $E_{in}^{box} = 0$, $D_1(box) = 1 - log_2(1) * |box| = 1$.
If $E_{in}^{box} = \frac{1}{2}$, $D_1(box) = 1 - log_2(\frac{1}{2}) * |box| = 1 + |box|$.
If $E_{in}^{box} = 1$, $D_1(box) = 1 - log_2(0) * |box| = \infty$.

With a definition of the upper bound on imagination sampling complexity, $D_1(h)$, we need a metric to measure how well a particular classification will perform. In the following discussion, we assume h correctly classifies $s_{\mathcal{C}}$ for simplicity of notation. If h correctly classifies, then $CC(s_{\mathcal{C}}, h) = K(h)$. Additionally, k is used interchangeably with $K(h)$.

For this metric, we need a measure that

1. becomes 0 as $K(h) \to \infty$
2. becomes $\frac{1}{2}$ as $K(h) \to |s_{\mathcal{C}}|$
3. becomes 1 as $K(h) \to 0$

all as $|s_{\mathcal{C}}| \to |\mathcal{C}|$.

We measure the proportion of classifications by $\mathcal{H}_k$ on $s_{\mathcal{C}}$ by $2^{k-|s_{\mathcal{C}}|}$. Therefore, we can define an accuracy metric that follows these criteria.

$$\text{acc}(s_{\mathcal{C}}, \mathcal{C}, K(h)) \leq 2^{\frac{|s_{\mathcal{C}}|-K(h)}{|\mathcal{C}|}-1}$$

If $K(h) \to 0$, then $\text{acc}(s_{\mathcal{C}}, \mathcal{C}, K(h)) \leq 2^{\frac{|s_{\mathcal{C}}|}{|\mathcal{C}|}-1} \to 1$.
If $K(h) = s_{\mathcal{C}}$, then $\text{acc}(s_{\mathcal{C}}, \mathcal{C}, K(h)) \leq 2^{-1} = \frac{1}{2}$.
If $K(h) \to \infty$, then $\text{acc}(s_{\mathcal{C}}, \mathcal{C}, K(h)) \leq 2^{-\infty} \to 0$.
These trends are illustrated in Figure 10.3.

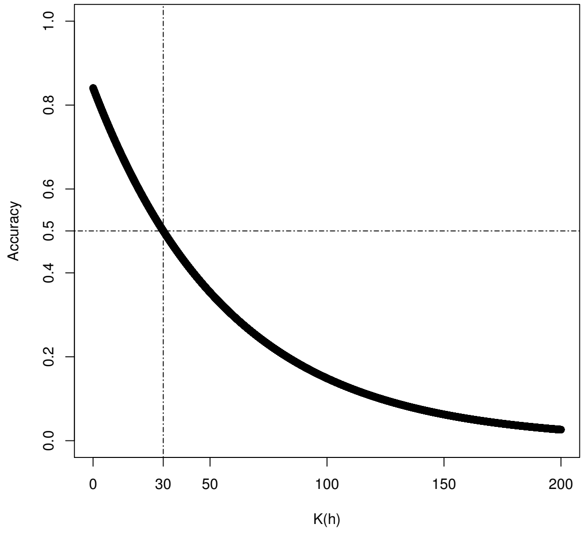


Figure 10.3: Graph of acc function for $\frac{s_{\mathcal{C}}}{C} = \frac{30}{40}$.

The next item is to invert acc. In other words, given E_{out} and $s_{\mathcal{C}}$, can we find $K(h)$? The answer is: sort of.

We set $acc(s_{\mathcal{C}}, \mathcal{C}, K(h)) = 1 - E_{out}$ and solve for $K(h)$ to get our $\mathrm{acc}^{-1}(s_{\mathcal{C}}, \mathcal{C}, E_{out})$ function:

$$\begin{aligned}
2^{\frac{|s_{\mathcal{C}}|-K(h)}{|\mathcal{C}|}-1} &\geq 1 - E_{out} \\
\frac{|s_{\mathcal{C}}| - K(h)}{|\mathcal{C}|} - 1 &\geq \log_2(1 - E_{out}) \\
\frac{|s_{\mathcal{C}}| - K(h)}{|\mathcal{C}|} &\geq \log_2(1 - E_{out}) + 1 \\
|s_{\mathcal{C}}| - K(h) &\geq |\mathcal{C}|(\log_2(1 - E_{out}) + 1) \\
K(h) &\leq |s_{\mathcal{C}}| - |\mathcal{C}|(\log_2(1 - E_{out}) + 1) = D_2(h)
\end{aligned}$$

This metric falls apart if $E_{out} = 0$ or $\frac{|s_{\mathcal{C}}|}{|\mathcal{C}|}$ is too small because $K(h)$ becomes negative. A negative Kolmogorov complexity does not make sense. This discrepancy is probably due to $\mathrm{acc}(s_{\mathcal{C}}, \mathcal{C}, K(h))$ having an asymptote of 1, but never reaching 1. Setting $E_{out} = 1$ assumes the asymptote is reached. The other issue is the hidden constant in Kolmogorov complexity, which is not addressed in these equations.

The acc metric does compare favorably with the equation boundaries for Occam Learning from Blumer, Ehrenfeucht, Haussler, and Warmuth (1987). The following equations show the equivalencies with acc.

- n is $K(h)$

- m is $|s_{\mathcal{C}}|$
- ϵ is E_{out}

The parameters $0 \leq \alpha < 1$ and $c \geq 1$ are parameters that specify a particular Occam algorithm.

$$n^c m^\alpha \ln(2) \leq -\frac{1}{2} m \ln(1-\epsilon)$$

With $\alpha = 0$ and $c = 1$, we can see the boundary conditions are similar to acc. The exception is for $\epsilon = \frac{1}{2}$, which is more stringent than acc.

$$\begin{aligned}
\epsilon = 0 : n &\leq \frac{-\frac{1}{2} m \ln(1)}{\ln(2)} = 0 \\
\epsilon = \frac{1}{2} : n &\leq \frac{-\frac{1}{2} m \ln(\frac{1}{2})}{\ln(2)} = \frac{1}{2} m \\
\epsilon = 1 : n &\leq \frac{-\frac{1}{2} m \ln(0)}{\ln(2)} = \infty
\end{aligned}$$

The final step in defining the theory of imagination sampling is to find out when we've converged on a good hypothesis. While it is not possible to know if we've found the optimum compression, we can at least measure our progress toward local optima.

If we have found the optimum compression, then $\frac{\Delta K(h)}{\Delta |s_{\mathcal{C}}|} = 0$. Since there is no closed form, or any form, of formula for $K(h)$ the best way we can find the derivative is empirical observation. We want to observe that as $|s_{\mathcal{C}}|$ increases, $K(h)$ remains constant. To estimate $K(h)$ we use validation to get an E_{out} score, and $\text{acc}^{-1}(s_{\mathcal{C}}, E_{out})$ to estimate $K(h)$.

Alternatively, we know that since $K(h) \leq D(h)$, then $\frac{\Delta K(h)}{\Delta |s_{\mathcal{C}}|} \leq \frac{\Delta D(h)}{\Delta |s_{\mathcal{C}}|}$ for a large enough Δ. Therefore, if $\frac{\Delta D(h)}{\Delta |s_{\mathcal{C}}|} = 0$, then $\frac{\Delta K(h)}{\Delta |s_{\mathcal{C}}|} = 0$. The intuitive reason for this is if $D(h)$ is constant, then eventually $K(h)$ must become constant. $K(h)$ will not decrease as the number of samples increases.

Once we've empirically solved for $\frac{\Delta K(h)}{\Delta |s_{\mathcal{C}}|} = 0$, and found the optimum h, we will see $E_{out} \to 0$ as both $|\mathcal{C}| \to \infty$ and $|s_{\mathcal{C}}| \to \infty$. This is because in $\text{acc}(s_{\mathcal{C}}, \mathcal{C}, k)$, $|s_{\mathcal{C}}|$ and $|\mathcal{C}|$ will grow indefinitely as $K(h)$ remains constant. Thus, $\lim_{|s_{\mathcal{C}}| \to \infty, |\mathcal{C}| \to \infty} \frac{|s_{\mathcal{C}}| - k}{|\mathcal{C}|} = 1$.

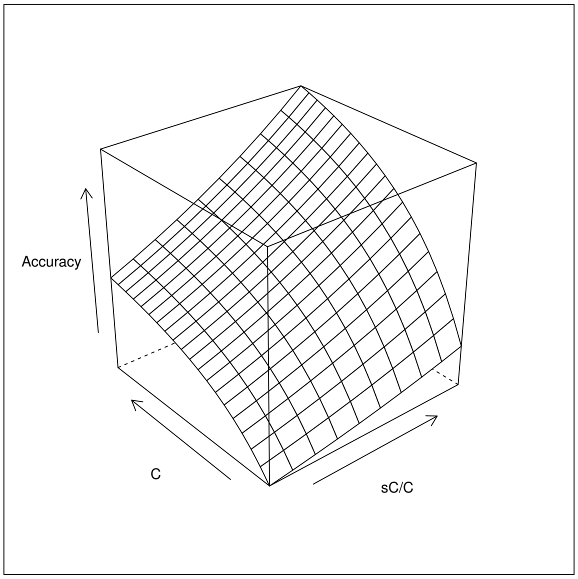


Figure 10.4: Keeping K(h) constant causes accuracy to increase as number of samples increase.

Of course, in a finite, discrete realm both $s_{\mathcal{C}}$ and $\mathcal{C}$ are bounded by $|\mathcal{A}|$. But as $|\mathcal{A}|$ is enlarged we will see $E_{out} \to 0$.

9 Simulated Oracle Experiment

To test this gradient-based approach with imagination sampling, a simulation of an oracle is used. In the problem, the target function is a tilted rectangle, as exemplified in Figure 10.5. The domain is a 100x100 grid making the rectangle pixelated.

The question is how to simulate a non-algorithmic oracle with an algorithm. Such a simulation seems to be a contradiction in terms. However, the benefit the oracle provides is the ability to infer the target function. To simulate the oracle, we use a learning algorithm that already has the correct class of target function, in this case a tilted rectangle. This learning algorithm is named an Occam learner. Figure 10.5 is the rectangle that the algorithm learned.

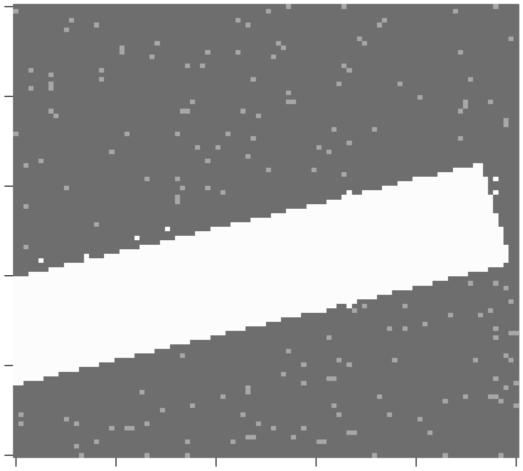


Figure 10.5: Function learned from training samples by Occam learner.

The learned rectangle is not used to calculate the value for $D(h)$. If we used the rectangle, $D(h)$ would give a complexity very close to $K(h)$. But without prior knowledge, we will not be able to calculate $K(h)$ for the oracle's hypothesis. The multibox classifier and E_{test} are used to calculate $D(h)$ instead. The goal is to see if the $D(h)$ metrics are reliable guides for identifying good oracle hypotheses.

The rectangular region is turned into a multibox, with one box for every cell in the rectangular region. $D_1(h)$ is calculated from this multibox hypothesis. $D_1(h)$ is not monotonic as its size will vary based on the samples used to construct the rectangular region. $K(h)$ is also estimated using E_{test} with $D_2(h)$.

The $D(h)$ metrics are calculated for hypotheses learned on different sample sizes. Once a large enough size range has been covered, the gradients $\frac{\Delta D(h)}{\Delta s_C}$ are calculated. The gradient technique is successful if it reliably identifies hypotheses that are highly accurate. To find a good region, we look for areas where at least one of the two gradients show a valley while the other gradient is negative. On the other hand, if the gradient cannot dependably identify accurate hypotheses, then it is not a useful technique. The graph in Figure 10.6 shows that the gradient technique can reliably identify high accuracy hypotheses with one anomaly.

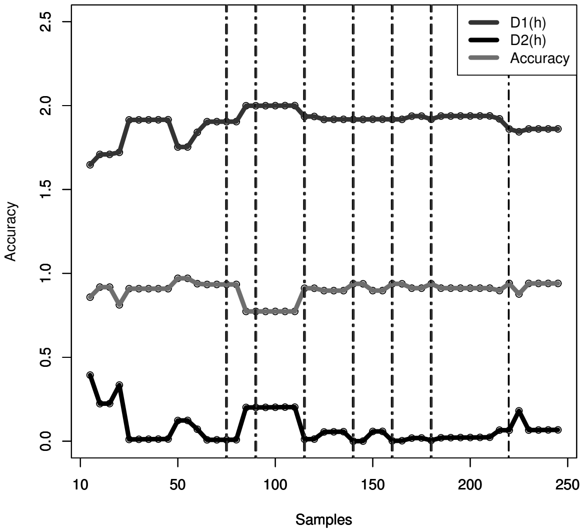


Figure 10.6: Example of the gradient-based approach working effectively. The vertical lines mark points where the gradient is zero and the second derivative is positive. The shade denotes which estimate of $K(h)$ is used to calculate the gradient. $D_1(h)$ is the top line, $D_2(h)$ is the bottom line, and the accuracy is the middle line. Scales are not given for the estimates as the important aspect of the estimates is the gradient.

The gradient technique does not guarantee the global best accuracy. See the second gradient marker from the left in Figure 10.6. While it looks like the marking is in error since it appears that both $K(h)$ estimates are peaks, there are actually slight depressions in the peaks. This shows the gradient method can only guarantee local optima, which may be very local.

10 Empirical Gradient-Based Approach Results

The gradient-based approach is tested using the results from the first experiment.

There are nine different sample sizes ranging from 100 to 500 samples in increments of 50 samples. Each increment is the addition of new samples to the previous set. The actual sample sizes are smaller due to the cleaning and equalizing processes.

The range of samples is tested on a pair of variables, which form the x,y coordinates for the scatterplot. An example of the scatterplot is in Figure 10.2. There are 37 variable pairs in the results.

A particular experiment is identified by the sample size and variable pair. The gradient is calculated over a range of sample sizes for a variable pair. The gradient is then calculated with both the D_1 and D_2 complexity metrics.

The gradient-based approach looks for three consecutive experiments that meet these criteria:

1. s_C increases across all experiments
2. A $D(h)$ metric decreases and then increases
3. The other $D(h)$ metric is not increasing

If these conditions are met, then the gradient has hit a minimal point. When this happens, E_{out} is at a minimal, or has a negative gradient.

There are 259 experiments that can potentially meet the gradient criteria. We cannot know whether the first and last sample sizes are at minimal points so they are excluded. There are 92 (36%) experiments where the E_{out} is at a minimal point leaving only 10 experiments that meet the criteria. Five of the 10 (50%) have an E_{out} at a minimal. The gradient-based approach boosts the accuracy of identifying minimal E_{out} by 14%. The mean accuracy of the 10 experiments is 0.52 and the median is 0.54, both higher than the mean of all the oracle's hypotheses as well as the algorithmically generated hypotheses. However, the p-values for two sided t-tests on these results are 0.27 and 0.24, respectively. Thus, the results are not statistically significant.

11 Conclusion

The purpose of this project is to demonstrate that oracles can generate better hypotheses when compared to algorithmic approaches. To test the oracles' performance against algorithms, an algorithmically unlearnable classification model is used. The classification model, multiboxes, shatters all datasets. This means a good hypothesis must be selected based on compression, but finding a good compression is an undecidable problem. Consequently, due to the No Free Lunch Theorem, no multibox learning algorithm will do better than random sampling.

A dataset with anonymized features is chosen so the oracle does not have access to domain knowledge. The oracle outperforms the tested algorithmic approaches on the anonymized dataset by 114-to-91 successes. Due to the improbability of this result given the NFLT, it shows that the oracle has a non-algorithmic learning capability and can out-perform algorithmic learning in general.

Furthermore, a gradient-based approach for identifying good hypotheses is derived from the theory of imagination sampling. The theory defines how the accuracy of an oracle's hypothesis is based on hypothesis complexity. The complexity of the oracle's hypothesis is not directly calculable but can be estimated with an upper bound. The gradient-based approach is used on the upper bound to identify when a minimal complexity has been found. This minimal complexity identifies a good hypothesis.

The gradient-based approach works with a simulated oracle. The approach is also tested on the results from the initial experiment. It boosts identification of good hypotheses by 14% and improves the mean and median hypothesis accuracy to 0.52 and 0.54. However, the results are not statistically significant, with p-values of 0.27 and 0.24 using the t-test.

References

Angluin, D. 1988. Queries and concept learning. *Machine learning* 2(4):319–342.

Attenberg, J., Ipeirotis, P., and Provost, F. 2015. Beat the machine: Challenging humans to find a predictive model's "unknown unknowns". *Journal of Data and Information Quality (JDIQ)* 6(1).

Attenberg, J., Melville, P., and Provost, F. 2010. Guided feature labeling for budget-sensitive learning under extreme class imbalance. *ICML Workshop on Budgeted Learning* .

Attenberg, J. and Provost, F. 2010. Why label when you can search?: alternatives to active learning for applying human resources to build classification models under extreme class imbalance. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 423–432, ACM.

Attenberg, J. and Provost, F. 2011. Inactive learning?: difficulties employing active learning in practice. *ACM SIGKDD Explorations Newsletter* 12(2):36–41.

Blumer, A., Ehrenfeucht, A., Haussler, D., and Warmuth, M.K. 1987. Occam's razor. *Information Processing Letters* 24:377–380.

Kearns, M.J. and Vazirani, U.V. 1994. *An Introduction to Computational Learning Theory*. Massachusetts Institute of Technology.

Kolmogorov, A.N. 1998. On tables of random numbers. *Theoretical Computer Science* 207:387–395.

Marchand, M. and Taylor, J.S. 2003. The set covering machine. *The Journal of Machine Learning Research* 3:723–746.