Toward learning extended shuffle ideals

G. Ma*

April 2013

Abstract

We present a machine learning algorithm for determining the underlying shuffle string for an extended shuffle ideal, as well as useful results for proving that the algorithm functions as required in the statistical query model.

1 Introduction

Using a statistical query (SQ) model of machine learning, we attempt to find an algorithm that, given an SQ oracle that provides example strings with each character chosen independently and uniformly at random from a given alphabet and a labeling of “positive” or “negative” for each, can be used to determine the underlying rule for a particular type of string pattern that we call the extended shuffle ideals.

This class is a generalization of the shuffle ideals. A shuffle ideal is the set of all strings that contain a designated string, known as the shuffle string, as a subsequence. This means that for each string in the shuffle ideal, the characters of the shuffle string appear, in order, with interpolation of other characters allowed. For example, if our shuffle string were \textit{acc}, then \textit{accb}, \textit{aacc}, and \textit{bacbc} would all be in the associated shuffle ideal, while \textit{cac} and \textit{dcc} would not. A particular example string is designated to be a positive example if it is part of the shuffle ideal, and a negative example otherwise. The extended shuffle ideals are much the same, except that each character in the shuffle string need not be one specific character, but can be any non-empty strict subset of the alphabet. For example, an extended shuffle string could be defined \((a \lor c \lor d)(b)(b \lor c)\) (here denoted as a concatenation of or-clauses). Then the corresponding extended shuffle ideal would accept all of the following as positive examples: \textit{abb}, \textit{cbb}, \textit{dbb}, and \textit{dbc}.

If we are further given the length of the shuffle string and the constraint that the example strings are all of a specified length, we can use the SQ model algorithm presented by Angluin et al. [1] to determine the shuffle string of a shuffle ideal. There is a natural extension of this algorithm to the extended shuffle ideals, and in this paper we present this generalized algorithm and then focus primarily on analyzing its performance to prove that it functions in the SQ model. Our work thus far does not entirely complete this proof, but provides useful intermediate results.

*505 College St, New Haven 06511. Yale University.
2 Preliminaries

Let the alphabet of allowable characters be $\Sigma$, with $|\Sigma| = s$, and the extended shuffle string $u$ be of length $L$, with the $i^{th}$ character allowed to be any character in non-empty set $u_i \subset \Sigma$. (Note that these sets are strict subsets of the alphabet $\Sigma$.) Define $\text{III}(u)$ to be the extended shuffle ideal associated with $u$. For all $i$ such that $1 \leq i \leq L$, let $a_i$ be the size of set $u_i$. Given a positive example string, consider the leftmost embedding of the first $i - 1$ characters of $u$ (i.e. an embedding of some $c_1 c_2 \ldots c_{i-1}$, where $c_j \in u_j$ for all $j < i$), found by searching greedily from the left of the example string for the first match of any character in $u_1$, then $u_2$, and so on through $u_{i-1}$. Define $\sigma_i$ to be the first character in a positive example string after the leftmost embedding of the first $i - 1$ characters in $u$. Note that $\sigma_i$ is guaranteed to exist, for all $1 \leq i \leq L$, since there must be at least enough characters remaining after the leftmost embedding for the string to be a positive example, and furthermore is well-defined because for each $j < i$, we start searching after the character found for the $j - 1^{th}$ character of $u$ for the first character that is in $u_j$. The greedy leftmost embedding is thus a completely deterministic process.

We now define sets of positive example strings for a given extended shuffle string. $P_n$ denotes the set of positive example strings of length $n$. For any $1 \leq i \leq L$ and any $c \in u_i$, let $R_{n,i,c}$ be the set of positive example strings of length $n$ for which $\sigma_i = c$. Analogously, for any such $i$ and any $d \notin u_i$, let $W_{n,i,d}$ be the set of positive example strings for which $\sigma_i = d$. These sets have been so named because $R_{n,i,c}$ refers to the strings for which is $\sigma_i$ is "right" (i.e. matches some character in $u_i$) with some specific character $c \in u_i$, and $W_{n,i,d}$ similarly refers to the strings for which $\sigma_i$ is "wrong" with some specific character $d \notin u_i$.

Here, we will briefly note that given a choice of $i$, $R_{n,i,c}$ is of equal size for any $c \in u_i$. This is readily proven by noting that given any two characters $c, d \in u_i$, we can create a one-to-one mapping from the strings in $R_{n,i,c}$ to those in $R_{n,i,d}$ by changing $\sigma_i$ from $c$ to $d$, or vice versa in the reverse direction. An analogous mapping proves that $W_{n,i,c}$ is of equal size for any $c \notin u_i$. We note this now because it is a relevant fact for the algorithm outlined later in this section.

For all $1 \leq i \leq L$, let $X_i$ be the geometric random variable that denotes how many characters are needed to match the $i^{th}$ character in $u$, where the characters are chosen independently and uniformly at random from $\Sigma$. In particular, $X_i$ counts the number of characters after the leftmost embedding of the first $i - 1$ characters of $u$, up to and including the first character $c$ in the example string such that $c \in u_i$. Then let $Y$ be the sum of all such $X_i$, or the total number of characters needed to match the entire extended shuffle string. Thus, given any independently uniformly randomly chosen infinite string of characters, the first $n$ characters constitute a positive example if and only if $Y \leq n$.

Section 3.3 will deal with distributions of geometric random variables through an argument regarding unimodality. A discrete distribution $\{d_n\}$ is unimodal if there is only one local maximum,
i.e. if there exists some $M$ such that

for all $n \leq M, d_n - d_{n-1} \geq 0,$
for all $n \geq M, d_n - d_{n-1} \leq 0.$

A discrete distribution $\{d_n\}$ is strongly unimodal if for any unimodal discrete distribution $\{e_n\}$, the convolution of $\{d_n\}$ and $\{e_n\}$ is unimodal.

Finally, we define the relevant variables related to the SQ model. Let $\epsilon > 0$ be the allowable error of our algorithm, and $\tau > 0$ be the tolerance of the SQ oracle. Let predicate $f(x, y)$ be a function on example string $x$ with labeling $y$ (equal to 1 if the string is a positive example and 0 otherwise) that simply returns $y$.

Our proposed algorithm, inspired by Angluin et al.’s algorithm for shuffle ideals [1], is given example string length $n$, extended shuffle string length $L$, allowable error $\epsilon > 0$, and runs as follows:

1. Set tolerance $\tau = \frac{\epsilon}{3}$. Query the SQ oracle for $E(f)$, the expected value of labelings of example strings, recording the response as $\alpha$. (By definition, $\alpha$ will be within $\tau$ of the actual value of $E(f)$.) If $\alpha < \frac{2\epsilon}{3}$, return constant 0. If $\alpha > 1 - \frac{2\epsilon}{3}$, return constant 1. In either case, exit the algorithm.

2. For all other expected values, do the following: For each $i = 1$ to $L$, look at the $\sigma_i$ for all the positive example strings drawn, and count the occurrences of each $c \in \Sigma$. Select all the characters that have frequencies “close to” the higher of the two frequencies found, and determine their union to be the set $u_i$.

To clarify the second step, for each $i$, we expect to find roughly two frequencies of characters: a higher one for each character $c \in u_i$, and a lower one for each character $d \notin u_i$. (This follows from the earlier observation that $R_{n,i,c}$ is of equal size for any $c \in u_i$, and $W_{n,i,d}$ is of equal size for any $d \notin u_i$.) To prove that this algorithm fits the requirements of an SQ model algorithm, we must do two things. First, we must more rigorously define “close to” by showing that there is a “sufficiently large” difference between the expected frequencies of allowable characters $c$ and non-allowable characters $d$ that is inverse polynomial in $s, n, L$, and $\frac{1}{\epsilon}$. (We will henceforth refer to this property simply as being “sufficiently large.”) This allows us to distinguish between the two frequencies such that we can bound the running time when this SQ model is converted into its corresponding probably approximately correct (PAC) model. Second, we must also show that we can always bound the algorithm’s answer $u'$ to within $\epsilon$ of the true extended shuffle string $u$, i.e. ensure that $Pr(x \in \Pi(u) \leftrightarrow x \in \Pi(u')) \geq 1 - \epsilon$.

3 Main Results

3.1 Justification for discarded tail ends

We first justify Step 1 of the algorithm. If $\alpha < \frac{2\epsilon}{3}$ then we know that the true value of $E(f)$ is less than $\epsilon$ because $\tau = \frac{\epsilon}{3}$. Thus we can act on the assumption that the labelings of all example strings
of length $n$ are negative, i.e. that no strings of length $n$ fulfill the given extended shuffle string. We can then simply return constant 0, and be guaranteed to be within $\epsilon$ of the true extended shuffle ideal because this will be correct with a probability of at least $1 - \epsilon$. Similarly, if $\alpha > 1 - \frac{2\epsilon}{3}$, we know the true value of $E(f)$ is greater than $1 - \epsilon$, and we can assume all example strings are positive and return constant 1.

Having entered Step 2, we are now guaranteed that the algorithm’s answer $u'$ will be precisely equal to the actual extended shuffle string $u$, due to the nature of the SQ oracle. We have thus shown the second of the required properties for the algorithm to be a SQ model. We now spend the remainder of the paper deriving results that aid a proof of the first.

3.2 Derivation of expression for difference between correct and incorrect character frequencies

We now consider the cases where $\frac{2\epsilon}{3} \leq \alpha \leq 1 - \frac{2\epsilon}{3}$, for which we enter Step 2. (Again, because $\tau = \frac{\epsilon}{3}$, we note that we can conclude that $\frac{\epsilon}{3} \leq E(f) \leq 1 - \frac{\epsilon}{3}$. This fact will come in useful in Section 3.3.) We must prove that for all $1 \leq i \leq L$ and for any $c \in u_i$ and $d \notin u_i$,

$$\left| R_{n,i,c} \right| - \left| W_{n,i,d} \right| = \frac{|P_n| - (s - a_i)|W_{n,i,d}|}{a_i}.$$

Solving for $|R_{n,i,c}|$, we find that

$$|R_{n,i,c}| = \frac{|P_n| - (s - a_i)|W_{n,i,d}|}{a_i}.$$

We substitute in this value for $|R_{n,i,c}|$ to remove that term from our initial expression:

$$\frac{|R_{n,i,c}| - |W_{n,i,d}|}{|P_n|} = \frac{|P_n| - (s - a_i)|W_{n,i,d}|}{a_i} - \frac{|W_{n,i,d}|}{sn} = \frac{|P_n| - s|W_{n,i,d}|}{a_i \cdot sn}.$$

We now prove a theorem that will allow us to remove $|W_{n,i,d}|$ from this expression.

**Theorem 1.** For any $i$, $1 \leq i \leq L$, $c \notin u_i$, and $n > 0$, $|W_{n,i,c}| = |P_{n-1}|$.

**Proof.** Let $d \notin u_1$. We will prove that there is a one-to-one-mapping from $W_{n,1,d}$ to $P_{n-1}$, and then we will show that there is a one-to-one mapping from $P_{n-1}$ to $W_{n,i,c}$ for any $i$ and $c$ as defined in the theorem statement.

For the first claim, note that since the position of $\sigma_1$ is always simply the first character of a string, it follows that for each string in $W_{n,1,d}$, the entire shuffle string must be embedded in
the latter \(n - 1\) characters of the string. Consider the set of strings of length \(n - 1\) created by simply removing the first character from each of the strings in \(W_{n,1,d}\), and call this set \(U\). Clearly, \(U \subseteq P_{n-1}\). Furthermore, \(P_{n-1} \subseteq U\) because for any string in \(P_{n-1}\), if we append \(c\) as its first character, we will get a positive example of length \(n\) whose first character \(\sigma_1 = c \notin u_1\), i.e. a string in \(U\). Therefore, \(U = P_{n-1}\).

For the second claim, we show two things: that every string in \(P_{n-1}\) maps to a distinct string in \(W_{n,i,c}\), and the same is true in the reverse direction. We repeat the prior observation that for any string in \(P_{n-1}\), and any \(1 < i \leq L\), the position of \(\sigma_i\) in \(P_{n-1}\) is determined: namely, it is the position directly after the leftmost embedding of \(c_1c_2\ldots c_{i-1}\), where \(c_j \in u_j\) for all \(j < i\). Thus, if we simply insert a \(c \notin u_i\) at that position, we derive a string of length \(n\) that embeds all of \(u\) (i.e. is a positive example) with \(\sigma_i = c \notin u_i\), i.e. a string in \(W_{n,i,c}\). Therefore, every string in \(P_{n-1}\) maps to a string in \(W_{n,i,c}\). For the reverse direction, consider any string in \(W_{n,i,c}\). Since \(\sigma_i = c \notin u_i\), we can remove that character and be left with a string of length \(n - 1\) that embeds all of \(u\), i.e. a string in \(P_{n-1}\).

By substituting in \(P_{n-1}\) for \(W_{n,1,d}\), the expression we must prove is “sufficiently large” now becomes

\[
\frac{|P_n| - s|P_{n-1}|}{(s - 1)s^n}. 
\]

Note that we have replaced the term \(a_i\) in the denominator with \(s - 1\) because \(a_i \leq s - 1\), and thus the original expression is at least as large as the expression written above.

This remaining expression is simply equal to the probability that \(Y = n\). This is intuitively explained by noting that \(P_n\) is the set of all positive example strings of length \(n\). Take any infinitely long string of characters \(w\), and consider its corresponding \(Y\). The first \(n\) characters of \(w\) constitute a member of \(P_n\) if and only if \(Y \leq n\). In other words, the example string must satisfy the extended shuffle string in \(n\) characters or fewer to be a positive example. \(s|P_{n-1}|\) is then equivalent to the number of strings of length \(n\) that satisfy the extended shuffle string in \(n - 1\) characters or fewer, so their difference is precisely the number of strings that take exactly \(n\) characters to satisfy the extended shuffle string.

### 3.3 Results on lower bound of difference

We now attempt to find a lower bound on \(Pr(Y = n)\) for all relevant \(n\). We first prove a useful lemma.

**Lemma 2.** Distributions of geometric random variables are strongly unimodal.

We provide two different proofs for this lemma, both based closely on the proofs by Keilson and Gerber [2]:

**Proof 1.** We use the following lemma from Keilson and Gerber [2]:

**Lemma 3.** A necessary and sufficient condition that a discrete distribution \(\{d_n\}\) be strongly unimodal is that \(\{d_n\}\) be log-concave, i.e. that \(d_n^2 \geq d_{n+1}d_{n-1}\).
Recall that a geometric distribution \( g_n \) with probability \( p \) of failure, \( 0 \leq p \leq 1 \), is equal to \((1 - p)^n p\) for all \( n \geq 0 \), and equal to 0 for all \( n < 0 \). For any \( n > 0 \),
\[
  g_n^2 = ((1 - p)^n p)^2 = (1 - p)^{2n} p^2 = (1 - p)^{n+1} (1 - p)^{n-1} p^2 = g_{n+1} g_{n-1},
\]
and for any \( n \leq 0 \),
\[
  g_n^2 = 0 = g_{n-1} = g_{n+1} g_{n-1}.
\]
Thus, we have shown that for all \( n \),
\[
  g_n^2 \geq g_{n+1} g_{n-1}.
\]

**Proof 2.** Given geometric distribution \( \{g_n\} \) with probability \( p \) of failure, as above, and another unimodal distribution \( \{d_n\} \), we need to prove that \( \{q_n\} = \{g_n\} \ast \{d_n\} \) is unimodal.

For all \( n \),
\[
  q_n = \sum_{k=-\infty}^{\infty} g_{n-k} d_k = \sum_{k=-\infty}^{\infty} g_k d_{n-k} = \sum_{k=0}^{\infty} g_k d_{n-k},
\]
with the second equality due to the fact that \( g_k = 0 \) for all \( k < 0 \). Then,
\[
  q_n = p \cdot d_n + (1 - p) p \cdot d_{n-1} + (1 - p)^2 p \cdot d_{n-2} + \ldots
\]
\[
  = p \cdot [d_n + (1 - p) d_{n-1} + (1 - p)^2 d_{n-2} + \ldots]
\]
\[
  = p \cdot d_n + p(1 - p)[d_{n-1} + (1 - p) d_{n-2} + \ldots]
\]
\[
  = p \cdot d_n + (1 - p) q_{n-1}.
\]
Therefore,
\[
  \Delta q_n = p \Delta d_n + (1 - p) \Delta q_{n-1}.
\]

From the first equality in Equation 1, we note that
\[
  \Delta q_n = \sum_{k=-\infty}^{\infty} g_{n-k} \Delta d_k.
\]
Because \( \{d_n\} \) is unimodal, by definition there exists some \( M \) such that for all \( n \leq M \), \( \Delta d_n \geq 0 \), and for all \( n > M \), \( \Delta d_n \leq 0 \). Thus, for all \( n \leq M \), \( \Delta q_n \geq 0 \). However, because
\[
  \sum_{n=-\infty}^{\infty} q_n = 1,
\]
\( \Delta q_n \) cannot be positive for all \( n \), so there must be some first value \( M + K \), \( K > 0 \), such that \( \Delta q_{M+K} < 0 \). For each subsequent \( n > M + K \), we can conclude by induction that \( \Delta q_n \leq 0 \) by using Equation 2 and noting that both \( \Delta d_n \) and \( \Delta q_{n-1} \) are negative. Thus, \( \{q_n\} \) is unimodal. \( \square \)

With this lemma, we can now prove a theorem that will lead to a useful conclusion on the lower bound of \( Pr(Y = n) \).

**Theorem 4.** The distribution of a sum of geometric random variables is unimodal.
Proof. We will name the $L$ geometric random variables $X_1, \ldots, X_L$, and let their sum be $Y$. (These are the same names as the random variables defined in the Preliminaries, Section 2, which is convenient for our purposes since we prove this theorem to prove that the distribution of $Y$ is unimodal.)

For any $L \geq 2$, first consider the first two terms of $Y$, i.e. $X_1 + X_2$. Then,

$$Pr(X_1 + X_2 = n) = \sum_{k=-\infty}^{\infty} Pr(X_1 = k \cap X_2 = n - k)$$

$$= \sum_{k=-\infty}^{\infty} Pr(X_1 = k) \ast Pr(X_2 = n - k),$$

since $X_1$ and $X_2$ are independent. This is the convolution of two strongly unimodal geometric distributions, so its distribution is also unimodal. For each subsequent $X_i$, $3 \leq i \leq L$, we simply convolve this resulting unimodal distribution with the geometric distribution for $X_i$ to get the distribution of $X_1 + X_2 + \ldots + X_i$. Because the distribution for $X_i$ is strongly unimodal, the distribution of the resulting convolution remains unimodal at each step. Thus, after convolving with the final $X_L$, we find that the distribution of $Pr(Y = n)$ is unimodal.

We have thus shown that the distribution of $Y$ is unimodal. Let $k_1$ be the largest integer such that $Pr(Y \leq k_1) \leq \frac{\epsilon}{3}$, and let $k_2$ be the smallest integer such that $Pr(Y \geq k_2) \leq \frac{\epsilon}{3}$. The set of all $n < k_1$ are the values of $n$ that give a low enough $\alpha$ for the algorithm to return constant $0$ in Step 1. Similarly, the set of all $n > k_2$ are the values of $n$ that give a high enough $\alpha$ for the algorithm to return constant $1$. We can then use the unimodality of $Pr(Y = n)$ to assert that for all $n$ such that $k_1 \leq n \leq k_2$, a lower bound for $Pr(Y = n)$ is $\min(Pr(Y = k_1), Pr(Y = k_2))$.

It then remains to prove that this final expression is “sufficiently large.”

4 Acknowledgements

Many thanks to Professor Dana Angluin for her mentorship and patience in directing this research, and thanks also to Professor Joseph Chang for pointing us in the direction of Keilson and Gerber’s article. It was a privilege and a pleasure to work on this paper.

References
