Query Complexity of Mastermind Variants

Aaron Berger Christopher Chute Matthew Stone

December 26, 2015

Abstract

We study variants of Mastermind, a popular board game in which the objective is sequence reconstruction. In this two-player game, the codemaker constructs a hidden sequence $H = (h_1, h_2, \ldots, h_n)$ of colors selected from an alphabet $\mathcal{A} = \{1, 2, \ldots, k\}$ (*i.e.,* $h_i \in \mathcal{A}$ for all $i \in \{1, 2, \ldots, n\}$). The game then proceeds in turns, each of which consists of two parts: in turn t, the codebreaker first submits a query sequence $Q_t = (q_1, q_2, \ldots, q_n)$ with $q_i \in \mathcal{A}$ for all i, and second receives feedback $\Delta(Q_t, H)$, where Δ is a function of distance between two n-sequences. The game terminates when $Q_t = H$, and the codebreaker seeks to end the game in as few turns as possible. Throughout we let $f(n, k)$ denote the smallest integer such that the codebreaker can determine any H in $f(n, k)$ turns.[1] We prove three main results: First, we show that given a set S_t of sequences which is known to contain H, there always exists a query whose feedback implicitly produces a smaller set S_{t+1} such that $H \in S_{t+1}$ and $|S_{t+1}| \leq (1 - 1/(nk))|S|$. Second, when $k = n$ and $h_i \neq h_j$ for all $i \neq j$, we prove that $f(n, k) \geq n - \log\log(n)$ for all sufficiently large n. Third, when feedback is not present in the game, we show that there exists a constant $c > 0$ such that $f(n, k) \geq c \cdot n \log(k)$.

1 Introduction

Variants of Mastermind are a family of two-player games centered around reconstruction of a hidden sequence. In all variants of the game, one player is given the role of "codemaker," and the other is denoted the "codebreaker." The codemaker begins the game by constructing a hidden sequence $H = (h_1, h_2, \ldots, h_n)$ where each component is selected from an alphabet $\mathcal{A} = \{1, 2, \ldots, k\}$ of k colors (that is, $h_i \in \mathcal{A}$ for all $i \in \{1, 2, \ldots, n\}$). The goal of the codebreaker is to uniquely determine the hidden sequence H through a series of queries, which are submissions of vectors of the form $Q_t = (q_1, q_2, \ldots, q_n)$. The codebreaker always seeks to determine H with as few queries as possible, however the nature of these queries, the feedback received after a query, and the restrictions on H differ between variants.

The variants which we will study are differentiated by settings of the tuple (n, k, Δ, R, A). These parameters are defined as follows:

(i) (n) *Length of Sequence.* The parameter n denotes the length of the hidden sequence H created by the codemaker, hence $H = (h_1, h_2, \ldots, h_n)$. The codebreaker is also required to submit query vectors of length n, so the t^{th} query vector takes the form $Q_t = (q_1, q_2, \ldots, q_n)$.

(ii) (k) *Size of Alphabet.* This parameter determines the number of possible values for components of H and Q_t. Each game is accompanied by an alphabet $\mathcal{A} = \{1, 2, \ldots, k\}$ from which the components of H and Q_t are selected. That is, $h_i \in \{1, 2, \ldots, k\}$ and $q_i \in \{1, 2, \ldots, k\}$.

(iii) (Δ) *Distance Function.* On the t^{th} turn of a game, the codebreaker submits a query sequence $Q_t = (q_1, q_2, \ldots, q_n)$. The codemaker then gives feedback $\Delta(Q_t, H)$, which is roughly a measure of distance between Q_t and H. The information yielded by $\Delta(Q_t, H)$ may be used to guide the choice of Q_{t+1}, the next sequence to be guessed. Hence the choice of distance function affects the codebreaker's ability to make an informed query on the following turn. We study the following distance functions:

a. *"Black hits and white hits."* Let Q_t and H be as above. The black hits and white hits distance function is defined by $\Delta(Q_t, H) = (b(Q_t, H), w(Q_t, H))$ where

$$b(Q_t, H) = |\{i \in \mathbb{Z} \mid q_i = h_i,\ 1 \leq i \leq n\}|, \tag{1}$$

and

$$w(Q_t, H) = \max_{\sigma \in S_{Q_t}} \ b(\sigma(Q_t), H) - b(Q_t, H),$$

We note that this is the distance function used in the original game of Mastermind.

b. *"Black hits only."* When Δ is the black hits only distance function, it is defined by $\Delta(Q_t, H) = b(Q_t, H)$, where b is defined as in equation (1).

(iv) (R) *Repetition.* The parameter R is a Boolean restriction on the components of H. If R is true, we say that the variant game is *with repeats* or that repeats are allowed. In this case, the hidden vector H may have repeated colors, that is, we allow $h_i = h_j$ for any $i, j \in \{1, 2, \ldots, n\}$. When R is false, we say that the variant is *no repeats*, and we require $h_i \neq h_j$ when $i \neq j$. In particular, when R is false and $k = n$, we have that H must be a permutation of $(1, 2, \ldots, n)$, and we refer to this variant as the *Permutation Game.*

(v) (A) *Adaptiveness.* The Boolean parameter A determines whether the codebreaker receives feedback after each query. If A is true, we say that the game is *adaptive.* In this case the game consists of two-part turns: on the t^{th} turn, the codebreaker first submits a query sequence Q_t, and then receives feedback $\Delta(Q_t, H)$. The codebreaker may use the feedback to inform the query Q_{t+1} made in turn $t + 1$, and the game ends in turn s if and only if $Q_s = H$.

When A is false, we say that the game is *non-adaptive.* In this case the codebreaker submits m queries $Q_1, Q_2, \ldots, Q_m$ all at once (where the codebreaker chooses m). The codemaker then reports a feedback vector of the form $(\Delta(Q_1, H), \Delta(Q_2, H), \ldots, \Delta(Q_m, H))$, after which the codebreaker must submit the final query $\overline{Q}$. The codebreaker wins if and only if $\overline{Q} = H$.

Throughout we define $f(n, k, \Delta, R, A)$ to be the smallest integer such that the codebreaker can determine any hidden sequence H in $f(n, k, \Delta, R, A)$ queries during a game with the corresponding assignment of n, k, Δ, R, and A. We will denote true and false by T and F, respectively, for assignment of Boolean variables. For example, Donald Knuth's result that the original game of Mastermind (four positions, six colors, black and white hits, with repeats, and adaptive) can always be determined after four turns is equivalently stated as $f(4, 6, \Delta = (b, w), R = T, A = T) \leq 4$ (in fact, this bound holds with equality). We will write simply $f(n, k)$ when the context is clear.

We prove three main results, each of which builds on the previous work related to a given Mastermind variant.

Theorem 1. Let $A = T$, let Δ be the black hits only distance function, and let n, k, and R be fixed. Given a set S_t of sequences which is known to contain H, there exists a query whose feedback produces a smaller set S_{t+1} such that $H \in S_{t+1}$ and $|S_{t+1}| \leq (1 - 1/(nk))|S|$.

This theorem facilitates analysis of a common variant of Mastermind, in which adaptive queries are made with feedback given by the black hits only distance function. To demonstrate the theorem's usefulness, suppose that the codebreaker is in turn t, hence has made queries $Q_1, Q_2 \ldots, Q_{t-1}$ and received feedback instances $\Delta(Q_1, H), \Delta(Q_2, H) \ldots, \Delta(Q_{t-1}, H)$. Given this set of feedback, the codebreaker can eliminate a number of possibilities for H. For example, suppose that a sequence $\tilde{H} = (\tilde{h}_1, \tilde{h}_2 \ldots, \tilde{h}_n)$ satisfies $\Delta(\tilde{H}, H) \neq \Delta(Q_s, H)$ for some $s \in \{1, 2, \ldots, t-1\}$. Then clearly $\tilde{H} \neq H$.

Using this process, the codebreaker may construct a set S_t of possibilities for H in turn t after $t-1$ query-feedback interactions have occurred. That is, S_t contains sequences of length n with components from the alphabet $\mathcal{A}$ such that no member of S_t has been eliminated by the responses $\Delta(Q_1, H), \Delta(Q_2, H), \ldots, \Delta(Q_{t-1}, H)$. Theorem 1 then states that given S_t, there always exists a query Q_t such that the response $\Delta(Q_t, H)$ will eliminate a fraction $1/(nk)$ of the members of S_t as possibilities for H. By Theorem 1, $|S_{t+1}| \leq (1 - 1/(nk))|S_t|$, which guarantees a quantifiable amount of progress at each turn of an adaptive, black hits only variant of mastermind.

Our second main result concerns the Permutation Game, in which $n = k$ and $R = F$, thus restricting the hidden sequence H to be a permutation of the alphabet $\mathcal{A}$.

> **Theorem 2.** Consider the Permutation Game defined by $n = k$ and $R = F$. Let $\Delta = b$ (black hits only distance function) and let A be fixed. Then for all sufficiently large n, we have
>
> $$f(n, k = n, \Delta = b, R = F, A) \geq n - \log\log(n).$$

Explicit algorithms that take $O(n \log n)$ turns to solve this variant were developed by Ko and Teng[2], and Ouali and Sutherland[3]. Ko and Teng approach the problem with an algorithm akin to binary search. The algorithm queries a sequence Q_t, swaps two components, and queries again, doing so repeatedly until a component of H is determined by the values of the distance function across these repeated queries. Ouali and Sutherland improve this algorithm primarily by altering the search routine once at least one component of H has been identified. In this way, Ouali and Sutherland achieve an average factor of 2 reduction in the number of queries needed to identify H. In our notation, these results state that there exists a constant $c > 0$ such that $f(n, k = n, \Delta = b, R = F, A = T) \leq c \cdot n \log(n)$.

Via a basic information-theoretic argument, one can show that the Permutation Game also satisfies $f(n) \geq n - n/\log(n) + c$ for some constant $c > 0$. We improve this lower bound to $f(n) \geq n - \log(n)$ for small n, and $f(n) \geq n - \log\log(n)$ for sufficiently large n. To our knowledge, this constitutes the first improvement over the trivial information-theoretic lower bound for the Permutation Game variant of Mastermind.

> **Theorem 3.** Consider non-adaptive Mastermind, defined by $A = F$. Let Δ be the black hits only distance function, and let n, k, and R be fixed. Then there exists a constant $c > 0$ such that
>
> $$f(n, k, \Delta = b, R, A = F) \geq c \cdot n \log(k).$$

2 Previous Work

References

[1] Chvátal, V.

[2] Ko, K.-I., and Teng, S.-C. On the number of queries necessary to identify a permutation. *Journal of Algorithms 7* (7 1986), 449–462.

[3] Ouali, M. E., and Sauerland, V. Improved approximation algorithm for the number of queries necessary to identify a permutation. *arXiv 1* (3 2013), 1–415.