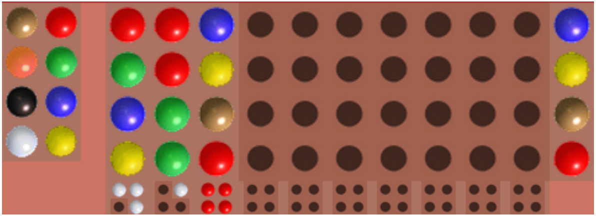
Query Complexity of Mastermind Variants

Aaron Berger, Christopher Chute, Matthew Stone

August 4, 2015

Mastermind

i. Codemaker vs. Codebreaker
ii. Queries: Guess a vector from $\{1, 2, \ldots, 6\}^4$
iii. Response
    i. Black (Red) hits
    ii. White hits



Knuth Paper – 1976

At most five turns needed to guarantee a victory

Minimax

For each possible guess

  For each possible response to that guess

    Check how many possible solutions remain

  Let *score* be min. number solutions eliminated

Make guess with maximum score

Optimality of the Minimax Algorithm

Lemma (Pigeonhole Principle)

For a query with r possible responses, there exists a response that would leave at leasr $1/r$ of possible solutions remaining.

Analyze the worst-case performance of any algorithm.

i. Worst case response to first guess $\Rightarrow$ At least 256 solutions remain.
ii. Second guess: At least $\lceil 256/14 \rceil = 19$ solutions remain.
iii. Third guess: At least $\lceil 19/14 \rceil = 2$ solutions remain
iv. Fourth guess: At least two possible solutions left $\Rightarrow$ cannot guarantee to guess the solution on the fourth turn.

Extensions

i. Basic Extension: n spots, k colors
ii. Repeats vs. no repeats
iii. Non-adaptive vs. adaptive strategies

Trivial Lower Bound

Trivial Lower Bound

Lemma (Pigeonhole Principle)

For a query with r possible responses, there exists a response that would leave at leasr $1/r$ of possible solutions remaining.

Trivial Lower Bound

Lemma (Pigeonhole Principle)

For a query with r possible responses, there exists a response that would leave at leasr $1/r$ of possible solutions remaining.

- $\#$ of solutions $= k^n$

Trivial Lower Bound

Lemma (Pigeonhole Principle)

For a query with r possible responses, there exists a response that would leave at leasr $1/r$ of possible solutions remaining.

- $\#$ of solutions $= k^n$
- $\binom{n+2}{2} \approx n^2$ possible responses per turn

Trivial Lower Bound

Lemma (Pigeonhole Principle)

For a query with r possible responses, there exists a response that would leave at leasr $1/r$ of possible solutions remaining.

- \# of solutions $= k^n$
- $\binom{n+2}{2} \approx n^2$ possible responses per turn

Theorem

For any strategy that guarantees a win in s turns,

$$\frac{k^n}{\binom{n+2}{2}^s} \leq 1$$

(Somewhat) Trivial Upper Bound

1. Represent guesses and solutions as matrices ($Q_{ij} = 1$ iff the i-th spot is the j-th color)

Example

The guess $(1, 1, 3)$ would become:

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

(Somewhat) Trivial Upper Bound

1. Represent guesses and solutions as matrices ($Q_{ij} = 1$ iff the i-th spot is the j-th color)

Example

The guess $(1, 1, 3)$ would become:

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

2. $Q_{ij} \in \mathbb{R}^{nk}$

(Somewhat) Trivial Upper Bound

1. Represent guesses and solutions as matrices ($Q_{ij} = 1$ iff the i-th spot is the j-th color)

   Example

   The guess $(1, 1, 3)$ would become:

   $$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

2. $Q_{ij} \in \mathbb{R}^{nk}$
3. # of black hits of Q with hidden matrix X is $Q \cdot X$.

(Somewhat) Trivial Upper Bound

1. Represent guesses and solutions as matrices ($Q_{ij} = 1$ iff the i-th spot is the j-th color)

Example

The guess $(1, 1, 3)$ would become:

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

2. $Q_{ij} \in \mathbb{R}^{nk}$
3. # of black hits of Q with hidden matrix X is $Q \cdot X$.
4. Guess a basis of span of all such matrices ($\leq nk$ guesses)

(Somewhat) Trivial Upper Bound

1. Represent guesses and solutions as matrices ($Q_{ij} = 1$ iff the i-th spot is the j-th color)

Example

The guess $(1, 1, 3)$ would become:

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

2. $Q_{ij} \in \mathbb{R}^{nk}$
3. # of black hits of Q with hidden matrix X is $Q \cdot X$.
4. Guess a basis of span of all such matrices ($\leq nk$ guesses)
5. Dot products with basis $\Rightarrow$ Uniquely determine X

(Somewhat) Trivial Upper Bound

1. Represent guesses and solutions as matrices ($Q_{ij} = 1$ iff the i-th spot is the j-th color)

Example

The guess $(1, 1, 3)$ would become:

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Theorem

There exists a set of at most nk guesses such that any hidden vector is uniquely determined by the responses to those guesses.

Entropy Method

Surprise Function: For an event x, we want

Definition (Surprise Function)

Entropy Method

Surprise Function: For an event x, we want

Definition (Surprise Function)

1. $S(x) = 0$ when $\mathbb{P}[x] = 1$

Entropy Method

Surprise Function: For an event x, we want

Definition (Surprise Function)

1. $S(x) = 0$ when $\mathbb{P}[x] = 1$
2. $S(x) = 1$ when $\mathbb{P}[x] = 1/2$

Entropy Method

Surprise Function: For an event x, we want

Definition (Surprise Function)

1. $S(x) = 0$ when $\mathbb{P}[x] = 1$
2. $S(x) = 1$ when $\mathbb{P}[x] = 1/2$
3. Decreasing function of $\mathbb{P}[x]$

Entropy Method

Surprise Function: For an event x, we want

Definition (Surprise Function)

1. $S(x) = 0$ when $\mathbb{P}[x] = 1$
2. $S(x) = 1$ when $\mathbb{P}[x] = 1/2$
3. Decreasing function of $\mathbb{P}[x]$
4. $S(x \wedge y) = S(x) + S(y|x) \quad (= S(x) + S(y)$ if independent$)$

Entropy Method

Surprise Function: For an event x, we want

Definition (Surprise Function)

1. $S(x) = 0$ when $\mathbb{P}[x] = 1$
2. $S(x) = 1$ when $\mathbb{P}[x] = 1/2$
3. Decreasing function of $\mathbb{P}[x]$
4. $S(x \wedge y) = S(x) + S(y|x) \quad (= S(x) + S(y)$ if independent$)$

Unique Surprise Function

$S(x) = -\log_2(\mathbb{P}[x])$.

Entropy Method

Entropy is the expected surprise of a random variable.

Entropy Method

Entropy is the expected surprise of a random variable.

Definition (Entropy)

Let X be a random variable with domain D.

$$H(X) = \sum_{x \in D} \mathbb{P}[X = x] \cdot (-\log_2 (\mathbb{P}[X = x]))$$

Entropy Method

Lemma

A single turn in a non-adaptive strategy has < 4 bits of entropy.

Entropy Method

Lemma

A single turn in a non-adaptive strategy has < 4 bits of entropy.

Lemma

The number of bits of entropy in any winning strategy is exactly

$$\log_2\left(\frac{k!}{(k-n)!}\right).$$

Entropy Method

Lemma

A single turn in a non-adaptive strategy has < 4 bits of entropy.

Lemma

The number of bits of entropy in any winning strategy is exactly

$$\log_2\left(\frac{k!}{(k-n)!}\right).$$

Theorem

The number of queries submitted by any winning non-adaptive strategy is at least

$$\frac{1}{4}\log_2\left(\frac{k!}{(k-n)!}\right).$$

Probabilistic Method

Choose a random set of queries $Q = \{q_1, q_2, \ldots, q_s\}$.

Calculate $\mathbb{P}[Q \text{ is a winning set of guesses}]$

Probabilistic Method

Choose a random set of queries $Q = \{q_1, q_2, \ldots, q_s\}$.

Calculate $\mathbb{P}[Q \text{ is a winning set of guesses}]$

Theorem

When the random set of guesses is of magnitude $s = 4k \log k$

$$\mathbb{P}[Q \text{ is a winning set of guesses }] > 0$$

Probabilistic Method

Choose a random set of queries $Q = \{q_1, q_2, \ldots, q_s\}$.

Calculate $\mathbb{P}[Q \text{ is a winning set of guesses}]$

Theorem

When the random set of guesses is of magnitude $s = 4k \log k$

$$\mathbb{P}[Q \text{ is a winning set of guesses }] > 0$$
$$\Downarrow$$
$$\exists \text{ a winning set of } 4k \log k \text{ guesses}$$

Thanks!

1. Special thanks to Danny Montealegre, Nathan Kaplan.
2. Thanks to SUMRY for this research opportunity.
3. Thanks to MathFest for the opportunity to present.