Mastermind by Importance Sampling and Metropolis-Hastings

Angela Snyder, Ph.D. Candidate
University of Chicago, Department of Statistics
5734 S. University Avenue
Chicago, IL 60637, USA
snyder@galton.uchicago.edu

1. Introduction

In the game of Mastermind, one player creates a sequence of colored pegs known as the "secret code," and their opponent attempts to deduce that code by making a series of guesses and receiving the following two integers as feedback: the number of pegs of the correct color in the correct position, and the number of additional pegs of the correct color but in the wrong position. In the standard game of Mastermind, codes of length four are selected with replacement from six colors, and the decoder has 8 turns to guess the secret code.

In this paper, Importance Sampling [1] and the Metropolis-Hastings algorithm [2] are utilized to construct stochastic strategies for a one-player version of the game where the secret code is generated uniformly at random. Koyama and Lai have shown that the optimal decision-tree strategy requires $\mu = 4.340$ turns on average, and always wins in at most 6 turns [3].

2. Mastermind by Importance Sampling

The first strategy is modeled as a series of stochastic decisions where the probability of guessing any code during the game is given by an exponential scoring model on a simple set of code features f_i that describe the number of colors in the code treated as a factor, and an indicator of whether the code is consistent with all prior information obtained. Thus, for a given code x_i in the code space $X = \{x_1, ..., x_n\}$, we have that

$$P_\theta(x_i) = \frac{S_\theta(x_i)}{\sum_{j=1}^{n} S_\theta(x_j)} \quad \text{, where} \quad S_\theta(x_i) = \exp\{\sum_{j=1}^{k} f_{ij}\theta_j\}.$$

In order to train the feature parameter vector θ, a gradient descent algorithm was performed on the Importance Sampling function for the mean number of turns required to complete the game. Beginning with an initial parameter vector θ_0, game data was simulated under the corresponding strategy and the parameter was subsequently updated in the direction of the importance sampling gradient until the process became stable. Resulting parameter values through the first 25,000 steps of the training process are shown in Figure 1. The final model for strategy prefers a first choice with three distinct colors, and subsequent choices that coincide with all information received. Simulation of n=10,000 games under this strategy resulted in a mean game length of $\mu = 4.606$ turns and standard deviation $\sigma = 0.877$ turns.

3. Mastermind by Metropolis-Hastings Algorithm

If we expand the game to allow for longer codes and additional color choices, it becomes impractical to evaluate all codes in the code space before making each guess. A more efficient means of searching through the code space is required. One strategy is to select each guess after making a series of Metropolis-Hastings steps through the space of possible codes. This strategy resembles simulated annealing [4].

Figure 1: Training the Parameter Vector by Importance Sampling

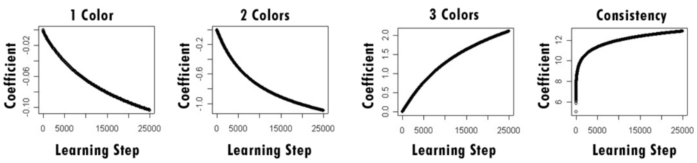

Table 1: Simulation Results for the Metropolis-Hastings Based Strategy

Colors	Pegs	μ (turns)	σ (turns)	Size of Code Space = c^p	Mean Prop'n Searched
5	3	3.918	0.979	125	2.306
6	4	4.812	1.041	1,296	0.4662
7	5	5.715	1.138	16,807	0.07342
8	6	6.618	1.174	262,144	0.009298
9	7	7.529	1.494	4,782,969	0.000966
10	8	8.477	1.585	100,000,000	0.000000859

Proposals are obtained by swapping the position of two pegs or by changing the color of a single peg. Each proposal is then evaluated by an exponential score based on that code's adherence to all feedback obtained thus far. Stopping time was determined through simulation such that the strategy results in no more than a 10% increase in game length over the mean length of a game played under the corresponding exact probability distribution on the entire code space (the theoretical equivalent of running the M-H algorithm to infinity). Simulations of $n = 10,000$ games were simulated under this strategy for six different versions of the game, and a summary of the result are presented in Table 1.

Lastly, it should be noted that the M-H based strategy is able to play efficiently even in the presence of false information. For a game where 20% of the feedback replies are selected uniformly at random from the set of all possible replies, simulation has shown that this strategy requires a mean of $\mu = 7.6388$ turns with a standard deviation of $\sigma = 4.351$ turns.

REFERENCES

[1] Hammersley, J. M., and Handscomb, D. *Monte Carlo Methods.* Wiley, New York, USA, 1964.

[2] Gill, J. *Bayesian Methods: A Social and Behavioral Sciences Approach.* CRC Press, New York, USA, 2002.

[3] Koyama, K., and Lai, T. An Optimal Mastermind Strategy. *Journal of Recreational Mathematics 25* (1993) 251-256.

[4] Van Laarhoven, P. J. M. and Aarts, E. H. L. *Simulated Annealing: Theory and Applications* D. Reidel Publishing Company, New York, USA, 1987.

RÉSUMÉ

Deux stratégies de stochastic sont présentées pour le Mastermind de jeu logique. La première stratégie a été dérivée utilisant une pente un algorithme décent sur l'Importance Sampling fonction pour le nombre moyen de virages exigés gagner. La deuxième stratégie cherche l'espace de codes par un feuilleton de propositions de Metropolis-Hastings.