

>> machine learning with



## >> Introduction

Machine Learning with CHOMP is a table-top activity used to demonstrate machine learning with no computers, maths or jargon. Through a simple game, participants will see how machines can be taught to solve problems and develop complex strategies, given only the “rules of the game” and some data from playing it.

The earliest analogue “Machine Learning” game was developed by Donald Michie in the 1960s. His “Machine Educable Noughts And Crosses Engine” (MENACE) machine learnt how to play noughts and crosses (tic-tac-toe) with 304 matchboxes filled with beads. Another popular example was developed by Martin Gardner and published in Scientific American – the game “Hexapawn” – played on a 3x3 chessboard reduced the the number of possible game states to 24, resulting in a much more manageable demonstration. An implementation of Michie’s MENACE was tested against the public by a team led by Matthew Scroggs at the Manchester Science Festival, and many examples of Hexapawn have also been demonstrated to the public.

In this paper, CHOMP (in uppercase) refers to the mechanical box-based computer, the “CHocolate Oriented Machine-learning Processor”<sup>1</sup> and Chomp (in lowercase) refers to the game itself, which will be described in the following section.

## >> How to play Chomp

Chomp is played on a 3x4 block of chocolate. The upper-left square is poisoned.

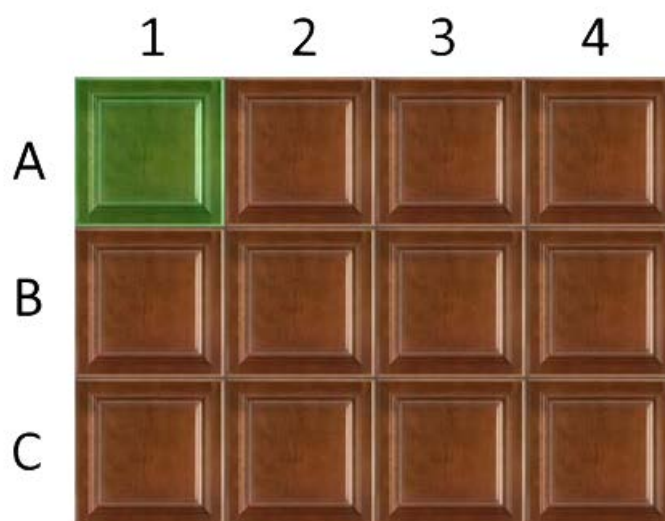










Figure 1: The starting position for every game of Chomp.

<sup>1</sup> Or for fans of recursive acronyms, CHOMP Has Outperformed Many People.

Players take turns to take bites out of the bar of chocolate, with the aim of leaving their opponent with the poison square. A “bite” in this case means removing one non-poisoned square and all remaining squares below and to the right. A game could progress as follows from the initial position in Figure 1. Pictures in table 1 represent the game board after the captioned move has been played.

Table 1: An example game of chomp

Player 1	Player 2
 B3	 C2
 A3	 C1
 B2	 B1
 A2 – Player 1 leaves only the poison squared	 A1 – Player 2 loses

## >> The Chocolate Oriented Machine-learning Processor

To make your own version of CHOMP, you'll need 33 small boxes, beads in 11 different colours, a representation of a chocolate bar with individual pieces (we 3D printed one) and some printed materials (A "Runway", colour maps, information sheets and a scoreboard).

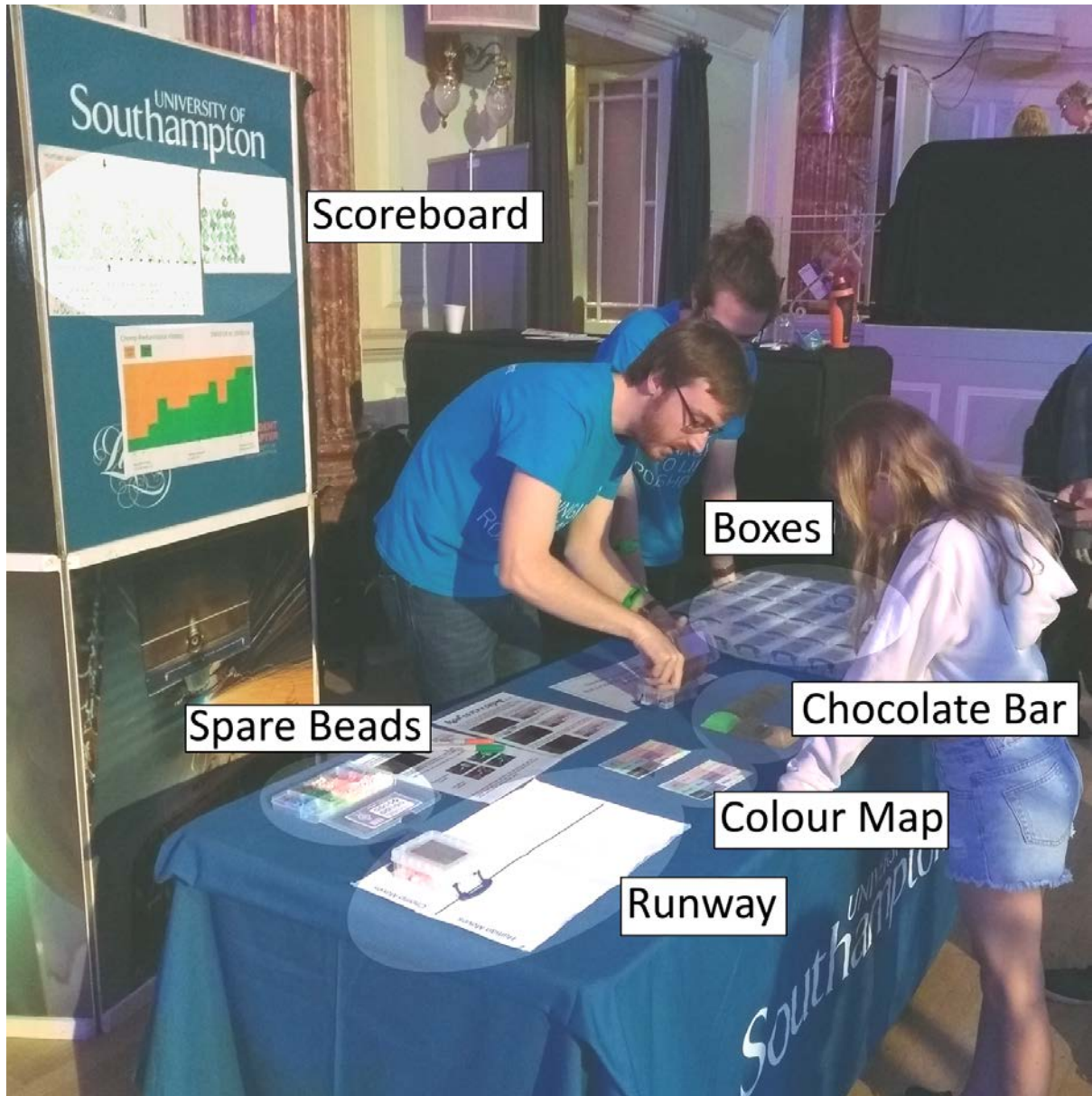


Figure 2: CHOMP in action at Cheltenham Science Festival 2018

As with the other analogue machine learning games described in the introduction, mechanising the game of Chomp requires a container for each possible game state and several coloured beads to represent each legal move from each state. The game of Chomp has 34 total states, and a maximum of 11 legal moves from the beginning. In the remainder of this paper, the moves denoted by coordinates in Figure 1 will be represented by the following coloured beads:

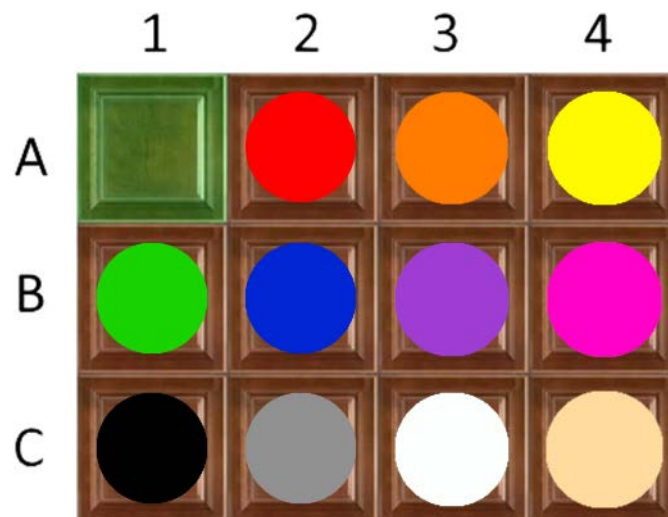


Figure 3: Move colours for CHOMP.

The various pieces of equipment required will be described by showing how a game is played between a human and CHOMP. In this example, CHOMP plays first, but this is an arbitrary choice.

- Assemble the **chocolate bar** into its complete 3x4 position with the poison square in the top left corner from the visitor's perspective
- CHOMP plays first – find the **box** which matches the current position of the chocolate bar and invite the participant to select a **bead** from the box at random.
- Find the colour on the **map** (See Figure 3) to determine which square CHOMP selected. Remove this square and all remaining squares below and to the right.
- Place the removed **bead** on the lid of the closed **box**, and place the **box** on the **runway** under “CHOMP moves”
- It is now the participant's turn. Find the **box** corresponding to the new pattern in the **chocolate bar**, and invite them to point to a square, ready to make their move.
- When the person removes their chosen square, and all squares below and to the right, open the **box** and remove a **bead** which corresponds to their chosen move, using the map.
- Place this **bead** on top of the closed **box** and place the **box** on the **runway** under “Human moves”. This “logs” or “records” their move.
- ...
- Play continues in a similar way, alternating between CHOMP and the human player.
- ...
- At the end of the game, either CHOMP or the participant will be forced to eat the poisoned square. Place the poisoned square at the end of the list of moves on the **runway**, according to whoever

What follows is the “reinforcement learning”. This is the take-home message, so it is worth making sure that the participant is listening at this stage!

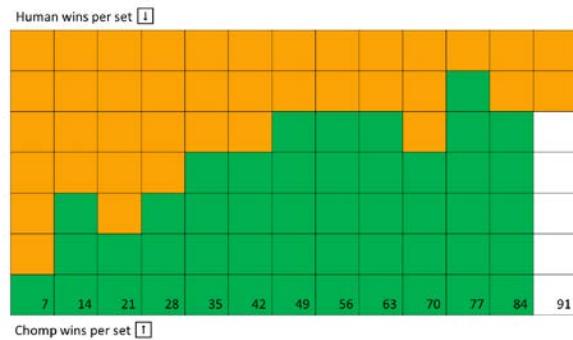
- The **runway** contains two lists of moves, one set which led to a loss, and one which led to a win.

- We want CHOMP to become more likely to win, so we want to increase the probability of winning moves being made and decrease the probability that losing moves are chosen.
- Take all the **beads** from the lids of the losing boxes and place them back in the **spare beads** box. This makes those losing moves less likely to re-appear and can be thought of as “learning from mistakes”.
- Put these losing boxes back in the pile of boxes.
- For each of the winning moves on the lids of the remaining boxes, take **two beads** from the **spare beads** box of the right colour for each box, and place them into each box.
- Three beads of the same colour should go back into each box in this “reinforcement” phase, the **original bead** and **two extras**. This increases the probability that CHOMP will choose this “winning” move again from this position.

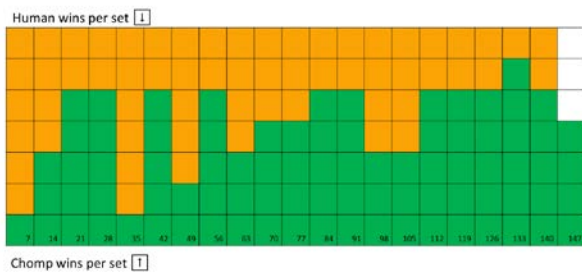
## >> Results

CHOMP is designed such that after about 100 games, it should reach a high level of proficiency against human opponents. At the beginning of each tournament, CHOMP should be “reset” with equal numbers of each of the possible moves. At the start CHOMP makes many obvious mistakes and loses consistently to human players with limited experience of the game. After 80-100 games, CHOMP’s win-rate increases to 70-80%.

Southampton Science and Engineering Day: March 2019



Cheltenham Science Festival: June 2018



Bringing Research to Life Roadshow: March-May 2018



Winchester Science Centre &amp; Science in Public Conference, Summer 2019

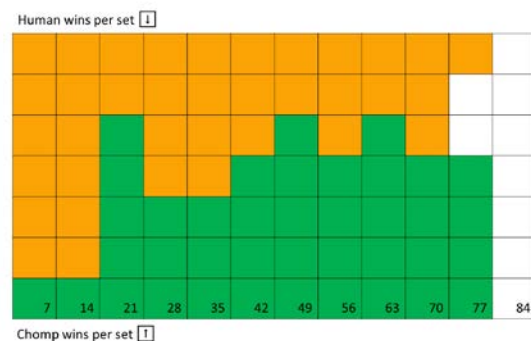


Figure 4: CHOMP vs Human players. Human wins are marked in orange from the top of each column, CHOMP wins are marked in green from the bottom until a “set” of 7 games have been played. Columns are filled from left to right, so early sets show human dominance, whereas later sets show CHOMP winning consistently.

## >> Teacher Notes

Machine learning is an alternative to conventional programming. Rather than directly programming an algorithm or strategy to solve a problem, machine learning systems analyse patterns of labelled data to determine the best way to achieve a certain outcome. We can draw parallels with a (fictional, but plausible) weather forecasting machine: STORM<sup>2</sup>

CHOMP		STORM	
Looks at patterns in the chocolate		Looks at patterns in the weather	
Makes a move based on this pattern		Makes a prediction of the weather based on this pattern, i.e. “it will rain”	
CHOMP Wins	CHOMP Loses	STORM is correct	STORM is wrong
Add beads to boxes to reinforce this move.	Remove this bead to decrease chance of making this move again.	Make this inference more likely to happen in future.	Make this inference less likely to happen in future.
Over time, CHOMP becomes more likely to win the game.		Over time (or based on existing data) STORM becomes more likely to predict rain correctly.	

<sup>2</sup> System That Observes Rain Maps

