Visualisation of Influence in Go

0810853

Abstract

In this paper I detail my implementation of a Go client using the CADENCE tool for Empirical Modelling. My model has features that are useful for teaching abstract concepts such as 'influence'.

1 Introduction

Current research into modelling the game Go tends to be in the field of AI, focusing on the development of bots that can play the game well (a difficult task due to its complexity). In the field of Empirical Modelling, however, we can make models that can be very effective in helping users (as opposed to machines) learn more about the game.

This paper focuses in particular on modelling the concept of 'influence' within the game. Influence is an abstract concept and can be somewhat subjective - for these reasons it can be difficult for a novice player to understand. Through empirical modelling we can create an interactive visual representation to aid in teaching.

2 Go

Go is an ancient abstract strategy board game which is thought to have originated in China several thousand years ago. The two players, black and white, take turns to place stones on the intersections of lines on the board. Every unoccupied intersection which is connected to a stone is called a liberty. Neighbouring stones of the same colour share liberties and are known as groups. If when a stone is placed, it removes the final liberty of an enemy group (this can be a single stone), the enemy group is removed.

The objective of the game is to obtain as much 'territory' as possible. Simply put, territory is the amount of space surrounded by stones of a player's colour, and is agreed upon at the end of the game. Essentially, a space on the board is a player's territory if, if his opponent plays there, then his opponent's stone will be captured. The game ends when all of the spaces are territory belonging to the players.

These have been simplified[1], but a full understanding of the rules is not necessary for this report. From the simple rules mentioned, complicated strategies are formed as opponents try to capture territory whilst preventing stones from being captured.

3 Teaching

3.1 Difficulties for beginners

The size of the board itself can be overwhelming, and beginners often spend a long time thinking about where to play. The game has a very large jargon which includes important concepts such as 'shape' and 'life' as well as 'influence'. These are often learnt intuitively after playing many games, as well as reading books with examples.

Traditional teaching methods for teaching beginners new concepts often rely on rote learning - a teacher will show a beginner examples of a concept that they will be expected to memorise. After memorising these, players gain an understanding of them by identifying and using them in real games. The initial memorisation stage can be off-putting, but it arises if the teacher cannot fully explain the concept to the learner in a way he understands - there likely exists a dichotomy between the teacher's mental model of the concept and the learner's.

Modern teaching methods involve the use of computer technology, most noticeable interactive boards designed to display problems, along with comments on moves that a learner attempts. These are, however, essentially the same as traditional Go books. An interesting development recently has been a system

---

[1]A link to the detailed rules of the game can be found on the British Go Association's website: http://britgo.org.uk/intro/intro2.html

built by students at Waseda University(Iwata, 2010). It uses a projector to display images on top of a Go board enabling augmented reality during a game. One of its more interesting features is a real-time display of territory estimation as well as 'links' between stones. This kind of interactive visualisation offers a new experience that learners can benefit from. Another, much simpler example, is Wolfram's model for learning Go that displays useful statistics on the side of the board.

3.2 EM in teaching

Empirical Modelling is useful for educational technology because of its philosophical ties with theories of learning (). In the case of teaching Go, experiential learning, as well as Papert's Constructionism are prominent examples. Empirical Modelling can be informally described as 'thinking with computers', which is done in the process of modelling. Constructionism builds on Constructivism (the theory that learners learn by making mental models) in saying that learners learn best by making things in the real world. Empirical Modelling allows us to create and explore tangible models of things based on the real world, thus facilitating constructionist learning. If a constructionist method were used to initially teach Go beginners new concepts, then it would be less off-putting for them, and they would not be faced with the challenge of trying to understand their teacher's mental model, instead making their own. Unfortunately, for abstract ideas with little tangible representation this can be difficult, which is the case in Go.

Experiential learning is simply learning from direct experience - this form of learning is active, and driven by the learner. Experimentation with an empirical model could be an example of this, as well as learning Go concepts by playing games. An interactive model on the computer that offers a tangible representation of influence should therefore increase the rate at which a beginner fully learns it.

There is no perfect mathematical model for influence, so any model will be inherently flawed - spotting these flaws and trying to improve the model by changing variables is an important learning activity facilitated by Empirical Modelling.

4 Influence

4.1 What is influence?

In this paper I shall consider the influence in an area to be the likelihood of a stone being able to live (avoid capture) in that area. This comes from the fact that each stone on the board can 'help' other stones (the other stones are less likely to get surrounded if they can connect onto another nearby). Therefore groups of stones close together have a high influence. Influence can be seen as something that 'radiates' from a stone(Sensei's Lib). There are various definitions that players use, and despite being open to interpretation, generally mean the same thing.

From my definition, we can see that influence is a desirable property, thus players should aim to create influence when choosing where to play a stone. It is of particular importance at the start of the game when there is more free space. Influence is closely linked with territory, and can be a good indication of which players will end up gaining territory in areas of the board. For example, if black has a lot of influence in an area, he is most likely to gain territory there, because if white plays in that area then white's stone will likely be captured.

4.2 Existing models

The idea of an 'influence function' that a computer could use to divide the board between black and white influence was first introduced by Albert Zobrist (1970). Existing models of interest were examined and from these I developed my own.

4.2.1 GNUGo

This free open source Go-playing program is uses an influence function that models influence as decaying exponentially away from stones. This model also adds extra influence to specific points on the board that are more important (such as a point where an invasion can easily be done). These points are determined through pattern recognition and relies on a database. The main purpose of this function is to predict areas of territory, so this added complexity is important in making these estimations more accurate. It is possible to view a coloured representation of influence in a game via a terminal.

4.2.2 Monte Carlo

There are Monte Carlo algorithms that involve playing many random games; for an area on the board, if one player plays there and wins more often than not, then the player has influence in that area. This method can be unreliable if too few games are played.

4.2.3 Meckes' Influence Simulation (Sensei's lib)

I discovered this model through Youtube video (Youtube). The visualisation of influence makes the game much easier to follow and the changes in colour are a good indication of when significant

moves are made. The software was developed by an amateur player, and has come under criticism from more experienced members of the Western Go community, but for more technical reasons regarding the rules of the game. His models for 'tension' are not applicable to Go, but his influence one worked well because it was so easy for a beginner to understand. Unfortunately the software is no longer available, so I was unable to test it.

Game development teacher Niklas Hansson used this visualisation software in a blog (Game Sch) to demonstrate ideas of influence applied to other games, specifically Real Time Strategy on computers. In this context, influence appears to refer to the overall area in which units can act. The primary difference in the case of RTS is that the units are able to move, unlike Go stones. This demonstrates how this style of visual representation can be accessible and intuitive for people who do not play Go - this is important for constructivist learning as we make sense of something by relating it to our previous knowledge.

4.2.4   My model

I decided to use exponential decay, as in the other models. I did not however make it any more complex than this. I chose to call the decay 'dissipation' as it reflects how influence can be seen as 'light' radiating from each stone, gradually losing its brightness and energy over distance.

The basic 'influence value' of a point is the sum of the 'effect' of all points on this point. Empty points have no effect. Points with a black stone have an effect of $+1 * (2)^{-distance}$. Points with a white stone have an effect of $-1 * (2^{-distance})$.

In this case the distance Chebyshev (chessboard) distance where a diagonally adjacent points are considered to be 1 unit length away. Below is a diagram:

1	1	1	2
1	0	1	2
1	1	1	2
2	2	2	2

This is the same diagram after working out the 'effect' of a stone:

1/2	1/2	1/2	1/4
1/2	0	1/2	1/4
1/2	1/2	1/2	1/4
1/4	1/4	1/4	1/4

This value is then turned into a colour for the background of the corresponding cell in the board and multiplied by the influence strength. Unlike some other models, the intensity of influence is taken into account enabling the model to show areas where influence is uncertain.

5 Modelling in Cadence

An initial prototype was made manually for a 3x3 board, before a Python script was used to automatically generate code for a larger 9x9 board. The subsections in this section indicate the components of the model that were developed in chronological order.

5.1 Board

I decided to model the board as a structure containing almost everything necessary for the game. In real life, this isn't the case - the board doesn't really store any information - but in the context of a computer program, the board is the only form of interface available to the player.

Perhaps a more realistic model would involve me modelling the board as just a surface on which stones could be placed. This could even be done by clicking and dragging to mimic the action of handling a stone.

The board node most importantly stores every cell; every cell has a state and a background colour that is determined by influence. Cells change state when clicked on.

Go is normally played on a board of size 19 x 19, but beginners may often wish to play on a smaller 9 x 9 board as games are faster and require less strategy. Basic tactical play is an important skill for beginners to learn before they move onto larger board sizes. Visualising influence on a larger board would be more useful and look nicer because of the increased resolution, but due to technical reasons was not featured in the model.

5.2 Stones

Each stone is the clone of the image template from the WGD library. This was chosen because it made it simple to set the positions of the stones and I could set the image as an external PNG file. I tried updating the 'filename' parameter to change the colour of stones (by loading an entirely different image) but this did not actually take effect. Instead, all the stones were based on an image of a white stone that is tinted black using the convenient 'tint' parameter in the image template.

5.3 Influence

The influence model I have chosen to use is fairly simple. The use of dependency in the DASM notation is particularly useful, as it is possible to define the relationship between the state of a point (that is, what kind of stone is placed on it) and all of the other points.

5.4 Code Generation Script

Copying and pasting the same code for every single cell would have been too time-consuming, particularly for the part that computes influence, as for each point, the distances to every other point need to be considered. Cadence lacks an exponential operator, so it was necessary to perform this operation in Python. All other arithmetic, however, is done in the DASM code.

6 Evaluation

6.1 Teaching

The model functioned as expected, and produced good visualisations in most cases. Below are some examples of how they could be used to explain the role of influence in choosing moves:
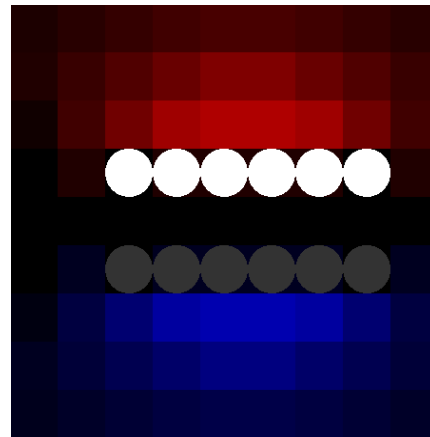
1. Stones too close together. Playing further apart allows one to gain influence over a wider area.
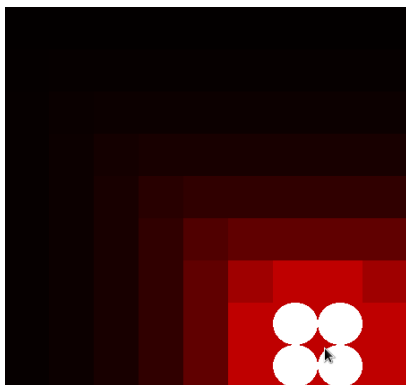
2. Stones are likely to be captured - shown by the lack of influence around them.
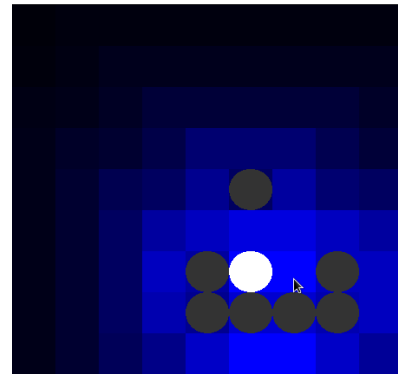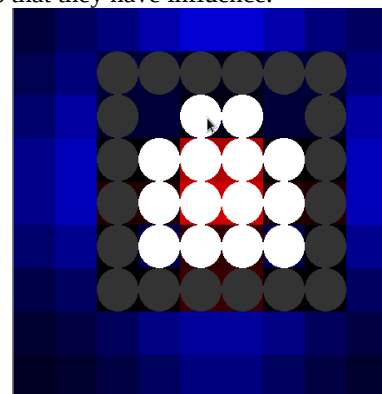
3. Influence cancels out leaving a neutral area in the middle of the board.

There are however, cases where it is not perfect:

1. Large clump of stones surrounded by opponent - the stones are about to be captured, but the board shows that they have influence.

If the model does not match a user's expectations, then tweaking the 'inflstrength' variable should be

of help. This is an important part of the learning process from a constructionist perspective. The interactivity of the board encourages the user to experiment in order to see how it reacts with a change of influence colours.

The model does, however, have some noticeable flaws. Changing or redefining variables defined by 'willbe' (:=) dependencies will most likely break the model, so the range of control the user has is limited. The model does not take into account any of the rules of Go (such as capturing), but this makes it easier for a user to set up particular board layouts (which is necessary when recreating a problem). Playing a full game using the model relies on the user's full knowledge of the rules, and therefore is not recommended for complete beginners. However, when played with a teacher it can be very useful in demonstrating the importance of influence at different stages of the game, by looking at how it changes. This could then be developed into a discussion on how influence earlier in the game relates to territory at the end of the game.

The colours are clear and easy to understand from a distance, so it could also be used in a group teaching exercise where the board is shown to many learners at once.

6.2 Technical

Cadence is still a prototype, so not all of its features had been implemented when the model was made. Some GUI elements, for example a taskbar to change influence strength had to be omitted from the model as well as textboxes. The GUI's text is actually part of the background image. Possibly due to large file size, Cadence crashed when I tried running the model with DASM files for 13x13 and 19x19 boards.

6.3 Further Improvements

These are nice additions to the project that could be made in the future:

6.3.1 SGF playback

SGF is a popular standard for recording games. They record the moves in the game, as well as branching variations and comments - these features are ideal for teaching. Comments can be made by teachers with detailed explanations that involve exploring alternate moves in a separate branch.

6.3.2 Sound

The importance of sound is easily overlooked in a board game like Go, but for many players the satisfying 'click' of a stone on a board has a simple appeal. Though seemingly trivial, it is one of the main reasons why many players still prefer to play on real boards. As part of the user interface sound provides useful feedback, and hearing a loud 'click' sound after making a move is a common feature in many Go clients.

6.4 Conclusion

A simple method of modelling influence was used to create a model in Cadence to be used to teach the concept of 'influence' in Go. Despite some flaws it proved to be successful. In most cases it highlights areas of influence correctly, after an appropriate influence strength is selected, providing novice players with a view of how an experienced player might see the game.

Acknowledgements

References

Takahiro Iwata, Tetsuo Yamabe, Mikko Polojärvi, and Tatsuo Nakajima. 2010. Traditional games meet ICT: a case study on go game augmentation. In Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction (TEI '10). ACM, New York, NY, USA, 237-240

Albert Lindsey Zobrist. 1970. Feature Extraction and Representation for Pattern Recognition and the Game of Go. Ph.D. Dissertation. The University of Wisconsin – Madison

Wolfram Demonstrations Project. 2012. Wolfram Demonstrations Project. [ONLINE] Available at: http://demonstrations.wolfram.com/Learn-TheGameOfGo/. [Accessed 31 January 2012]

Sensei's Library: Influence Map. 2012. Sensei's Library: Influence Map. [ONLINE] Available at: http://senseis.xmp.net/?InfluenceMap. [Accessed 31 January 2012]

Educational technology. 2012. Educational technology. [ONLINE] Available at:http://www2.warwick.ac.uk/fac/sci/dcs/research/em/applications/educationaltech/. [Accessed 31 January 2012]

GNU Go Documentation: 13. Influence Function. 2012. GNU Go Documentation: 13. Influence Function. [ONLINE] Available at:http://www.gnu.org/software/gnugo/gnugo_13.html#SEC156. [Accessed 31 January 2012]

Go Seigen - Sixteen Soldiers Game (Moves 1- 60) - YouTube . 2012. Go Seigen - Sixteen Soldiers Game (Moves 1- 60) - YouTube . [ONLINE] Available at:http://www.youtube.com/watch?v=UGDQc-cLdZWc. [Accessed 31 January 2012]

Game School Gems: Influence Maps I. 2012. Game School Gems: Influence Maps I. [ONLINE] Available at: http://gameschoolgems.blogspot.com/2009/12/influence-maps-i.html. [Accessed 31 January 2012]