Approximate matching for Go board positions

Alonso GRAGERA[1,a)]

Abstract: Knowledge is crucial for being successful in playing Go, and this remains true even for computer programs where knowledge is used in combination with a search method such as Monte Carlo Tree Search. Although many progresses have been made towards acquiring and using this knowledge in several areas of the game, programs lack of an efficient ability for extracting and analyzing knowledge from similar situations as human players do, which is being considered a major weakness. This paper presents a systematic method that could be used to improve the usage of positional information databases and enhance algorithms for Go by using an approximate matching (similitude-based) to retrieve information instead of only the exact matching.

1. Introduction

Game playing has been a part of the core of artificial intelligence research since it became a field of study; and since the game of Go remains as one of its few grand challenges, it has a growing interest last years. Even after having studied computer Go for more than four decades now, the ultimate goal of defeating human professional players remains unachieved.

A very significant progress came from the introduction of Monte Carlo Tree Search (MCTS) algorithms in 2005[1], going from loosing while receiving 15 handicap stones against amateurs to 8 stones against professionals in a single year, and steady improving year by year till winning with only 4 stones handicap stones against a top professional in 2015[2].

Importance

This improvement has been possible not only for the introduction of MCTS technique in Computer Go, but also because the improvement in the management of Go knowledge. That knowledge can appear in several forms, ad-hoc rules, shape databases, opening books, statistical biases, etc; but up to very recently it has been constantly focused in using it only when the exact same situation appears (exact matching), and only in the past years a few attempts have been made to tackle this condition (approximate matching)[3].

Aim

The purpose of the paper is to introduce methods to measure the similitude between board positions with possible applications in different areas of the game from information retrieval in opening books [4] to traffic minimization in distributed algorithms [5], along with broad pattern matching [3] and winning-percentage evaluation [6].

[1] Graduate School of Information Science and Technology, The University of Tokyo, JAPAN
[a)] alonso@is.s.u-tokyo.ac.jp

The first one is the *a posteriori* similitude, that focus on the cases where there there are professional level information, or where we can assume an equivalent good move prediction.

And the second one is the *a priori* similitude, that focus on the cases where the previous condition is not true, situations where there is a severe lack of information or when that information can not be relied on.

Structure

The rest of the paper is organized as follows: Section II presents a brief introduction to the game of Go and its rules. Section III introduces some required preliminary works on influence modeling. Section IV defines formally the concept of similarity between positions and presents several way to estimate it. Section V provides some experimental results evaluating both measures, and shows some examples of its usage. And at last, Section VI gives a summary of the results, its applications and discusses future works.

2. The game of Go

The game of Go (also known as *Igo*, *Weiqi* or *Baduk*) is believed to have originated in Central Asia more than 2500 years ago [7], going up to more than 4000 years ago according to some sources [8], this makes it one of the oldest known board games.

Since the old times it has been regarded as an sign of intelligence; in China, it was one of the four arts that any scholar must master *(Music, Go, Calligraphy, Painting)*, in Japan, the best player of the country was given the position of a minister *(Godokoro)* in the government. And even nowadays still remains as the last board game where humans are significantly better than computers.

2.1 Rules

Although the game itself is very difficult to master, its rules are relatively simple and comprehensible. And since some essential knowledge of the game of Go is mandatory for subsequent discussions, a brief introduction to the rules of Go is stated below.

Rule 1 – Players

The game is played by two players, called *Black* and *White*.

Rule 2 – Board

The board is a grid of horizontal and vertical lines, usually of size 19 x 19.

Rule 3 – Stones

Black uses black stones and *White* uses white stones, which are placed in the board intersections.

Rule 4 – Turns

Players take alternate turns, starting with *Black*, to place their respective stones (or pass).

Rule 5 – Captures

If a player surrounds (adjacent intersections) the opponent's stone or stones completely, he captures those stones and removes them from the board.

Rule 6 – Ko

One may not play a move which repeats a previous board position.

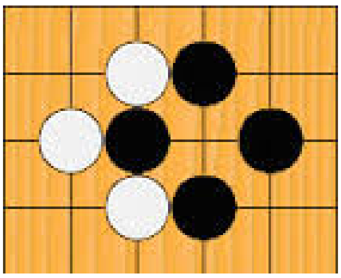

Fig. 1: Example of a ko situation

Rule 7 – Score

There are several alternative rule sets to score, but the most used one is the Japanese rules [9]. In this rule set the game ends after two consecutive passes, then each surrounded intersection and captured stones counts as a point, and *White* is given a 6.5 points compensation for not moving first.

3. Influence models

Influence models, also known as *influence functions* or *influence maps* are intended to be a representation of how each element of the model exert some effect to its surroundings. This idea was first introduced by Albert Zobrist [11], and it has become a very extended AI technique in modern games.

In the early computer Go research literature this method was very popular and broadly used in combination with expert knowledge, but over the years it has been replaced as new techniques were introduced, finally remaining used only as an score estimator; but nowadays even that function has been take over by the MCTS algorithms. Neither the less we will show that this methods are still useful in situations where expert knowledge is required.

In the past several authors have proposed different models, that are presented below, each one emphasizing a different aspect of the board position.

3.1 Stones influence model

The first idea that comes to mind is the trivial case where the stones themselves are used as the only relevant things of the model, without exerting any effect to their surroundings. This is obtained by just initializing the black stones to $+1$, the white stones to -1 and the empty intersection to 0.

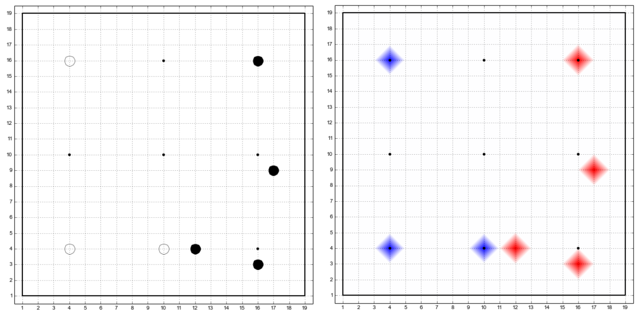

Fig. 2: Example of the stones influence model

This model results in a very simple, yet quite powerful function, that to detect trivial changes and it can be computed in an extremely efficient way.

3.2 Zobrist's influence model

Introduced by Albert L. Zobrist as part of its PhD thesis, and used in the first Go playing program [11], this model intended to capture the nature of influence in the game of Go. It studied the idea of influence as a concept emerging from the stones, with a limited range of 4 spaces and a decreasing strength, and gave them the ability to create synergies (additives when two influences of the same color collide, or canceling when the collision is between opposites, in a similar way to a magnetic field model).

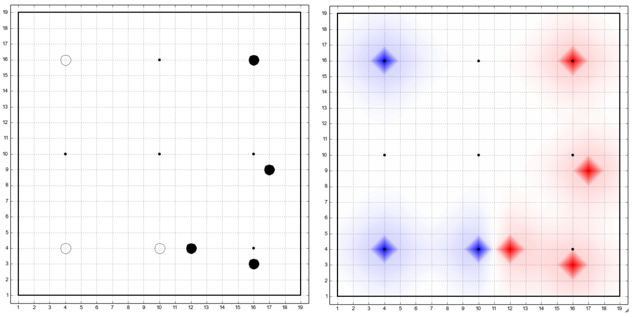

Fig. 3: Example of the Zobrist's influence model

This model is obtained by initialization the black stones to +64, the white stones to -64 and the empty intersections to 0. Then each intersection positive gives +1 to its neighbors (except for the for the ones occupied by stones), and similarly each negative intersection gives -1. The computation is completed after this transmission process is repeated exactly for times.

3.3 Ryder's influence model

This model was done by Jonathan L. Ryder also as part of its PhD thesis [12], inspired on the previous work done by Zobrist. Like Zobtist, Ryder also used an influence function to provide a numeric number to indicate the degree of tactical control each stone exert over its neighbors. His influence function is also similar to Zobrist in that black influence is positive and white influence is negative and the influence value at each intersection is the sum of the influence values propagated by its neighbors.

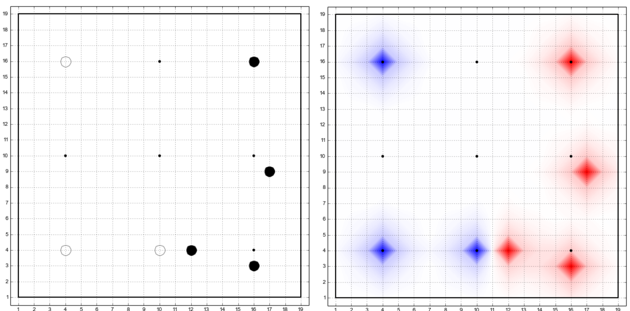

Fig. 4: Example of the Ryder's influence model

But in the case of Ryder's function, it is simpler than the Zobrist's one; each stone contributes a fixed value to its neighbors (without the need of an iterative process) according to the following pattern:

$$\begin{array}{ccccccc} \multicolumn{7}{c}{\text{Black}} \\ & & & 1 & & & \\ & 1 & 2 & 5 & 2 & 1 & \\ & 2 & 6 & 13 & 6 & 2 & \\ 1 & 5 & 13 & 64 & 13 & 5 & 1 \\ & 2 & 6 & 13 & 6 & 2 & \\ & 1 & 2 & 5 & 2 & 1 & \\ & & & 1 & & & \end{array} \qquad \begin{array}{ccccccc} \multicolumn{7}{c}{\text{White}} \\ & & & -1 & & & \\ & -1 & -2 & -5 & -2 & -1 & \\ & -2 & -6 & -13 & -6 & -2 & \\ -1 & -5 & -13 & -64 & -13 & -5 & -1 \\ & -2 & -6 & -13 & -6 & -2 & \\ & -1 & -2 & -5 & -2 & -1 & \\ & & & -1 & & & \end{array}$$

3.4 Spight's influence model

In 2002, the researcher William Spight introduced the idea behind his non-numerical influence model inspired in a waves coming put of the stones (therefore also known as *Spight's wavefronts analysis*) [13]. In practice, this model seems to find the equidistant boundaries between groups with opposite colors.

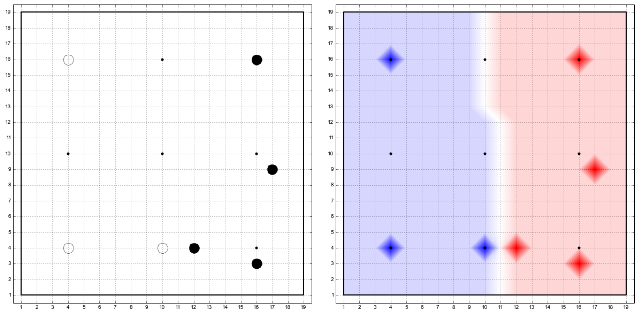

Fig. 5: Example of the Spight's wavefront analysis

3.5 Bouzy's influence model

In 2003, Bruno Bouzy published a refinement of the Zobrist influence model to accurately predict territory in the same way that human players do, therefore this model is better known as *Bouzy's territory model* [14]. It is based in the application of mathematical morphology given by the following operators:

Dilation	**External boundary**	**Closing set**
$D(A) = A + N(A)$	$ExtBound(A) = D(A) - A$	$Closing(A) = E(D(A))$

Erosion	**Internal boundary**	**Opening set**
$E(A) = A - N(UA)$	$IntBound(A) = A - E(A)$	$Opening(A) = D(E(A))$

Table 1: Basic operators

To redefine initial model by using the following Zobrist-like operators:

Zobrist Dilation	E_z = Add the #neightbours of the same color
Zobrist Erosion	D_z = Subtract the #neightbours of the opposite color or empty
Zobrist Territory	$X_z(e, d) = E_z^e \circ D_z^d$

Table 2: Zobrist-like operators

Some of the most common used ones are $X_z(13, 4)$ and $X_z(21, 5)$.

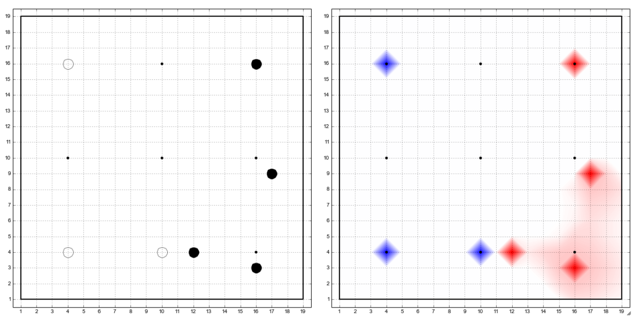

Fig. 6: Example of the Bouzy's territory model with $X_z(21, 5)$

4. Approximate matching of board positions

The necessity of extracting new knowledge from the program existing data, as mentioned in the section I, not only exists but it is also one of the current major weaknesses. In order to provide new tools to help in this task, in this paper, some methods to calculate similitude measures, between two board positions are introduced. This problem itself is extremely hard, since experts are based in their intuition do decide when two positions are similar or not.

4.1 Similitude *a posteriori*

The most reasonable approach in this case would to consider two board positions closer the more professional-level follow-up moves they have in common, after being rotated/mirrored to their respective canonical forms.

Definition 1 Let $next(x)$ be set of follow up moves of the position $x \in \mathcal{P}$, we can define the similitude as

$$\hat{s}_{pos}(x, y) = \frac{2\ |next(x) \cap next(y)|}{|next(x)| + |next(y)|}$$

4.2 Similitude *a priori*

But since the previous definition of similitude is impossible to compute without previous knowledge of the possible good moves for the given position, and most compelling problems on computer Go are related to the decision of the next good move, an alternative measure only dependent of the board position directly is required.

Definition 2 Let $\hat{S}(\mathcal{P}, \mathcal{P}, \mathcal{F})$ be a family of similitude measures between two board positions $x, y \in \mathcal{P}$, under a given influence model $f \in \mathcal{F}$, defined by

$$\hat{s}_f(x, y) = \begin{cases} 1 & if\ x = y \\ \left|1 - \dfrac{2}{1 + e^{\dfrac{1}{\alpha \sum\sum |f(x) - f(y)| + \beta}}}\right| & otherwise \end{cases}$$

where α, β are a configurable model-dependent parameters.

This family can me easily extended just by defining new influence models that capture the desired characteristic to emphasize.

5. Experiments

For the experiments in this section the BadukMovies collection[*1] consisting on $52,055$ game records between professionals. The positions after the 6th move for each game was selected, and all the different professional replies where clustered, there are 750 unique board positions (excluding with rotations and symmetry). Then, these $562,500$ pairs of positions where compared.

***A posteriori* similitude**

With the *a posteriori* similitude is easy to see, in the Figure 7, that most of the possible combinations are very different, still a small but relevant fraction is very similar.

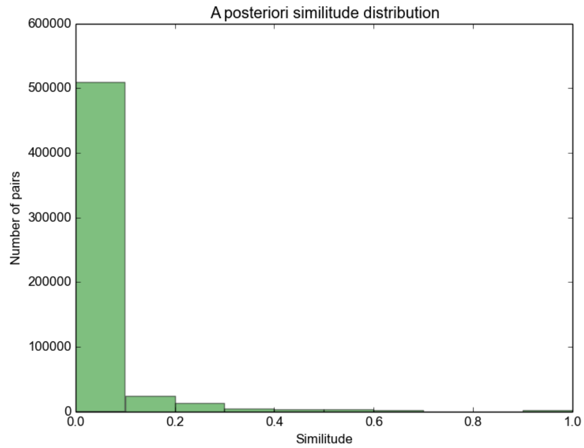

Fig. 7: Distribution of a posteriori similitude

And meanwhile certainly some of the positions are very unique, a relevant subset of the total have nice similitude relations among each others, for example Figure 8 shows a well related subset of all possible combinations.

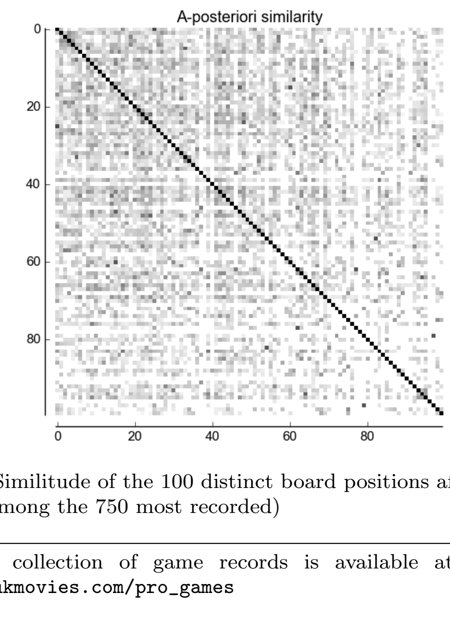

Fig. 8: Similitude of the 100 distinct board positions after the 6th move (among the 750 most recorded)

[*1] The collection of game records is available at `https://badukmovies.com/pro_games`

Stones' *a priori* similitude

The α and β parameters can be easily optimized to get a good balance between the precision and recall, using the simplest of the influence models. Figure 9 shows that the *a priori* looks like an smoothened version of the *a posteriori* similitude.

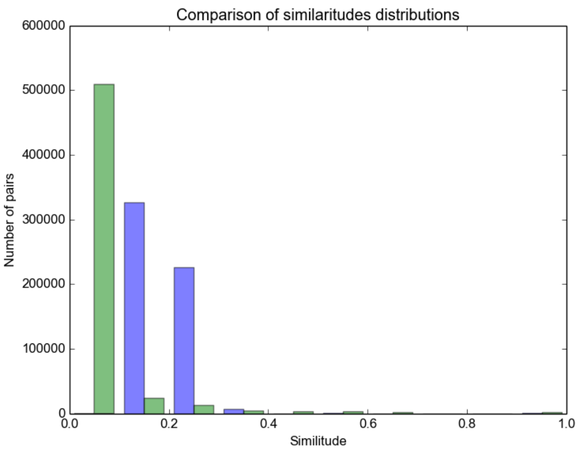

Fig. 9: Comparison of distributions of similitude of Zobrist's *a posteriori* (with $\alpha = 0.25$ and $\beta = -0.2$) (green) and *a priori* (blue)

This can be used for classify highly similar, meaning *a posteriori* similitude ≥ 0.75, to easily find around 1/3 of the similar positions.

Precision	Recall	Accuracy	Average error
0.46	0.32	0.995	0.19

Table 3: Classification of highly similar positions using Stones' *a posteriori* similitude (with $\alpha = 0.25$ and $\beta = -0.2$)

Zobrist's *a priori* similitude

With just a little of optimization the same can be applied with the Zobrist's inflence model.

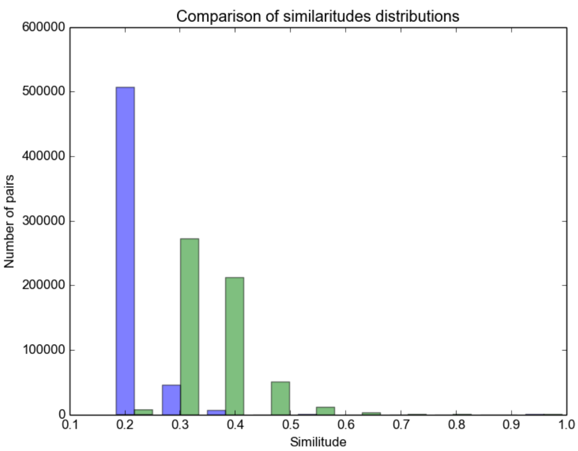

Fig. 10: Comparison of distributions of similitude of Zobrist's *a posteriori* (with $\alpha = 0.00082$ and $\beta = 0.035$) (green) and *a priori* (blue)

This can again be used for classify highly similar, but with emphasis in the Zobrist influence characteristics.

Precision	Recall	Accuracy	Average error
0.06	0.40	0.97	0.33

Table 4: Classification of highly similar positions using Zobrist's *a posteriori* similitude (with $\alpha = 0.00082$ and $\beta = 0.035$)

Even if the precision is low, depending of the intended usage, since it can be post filtered by a MCTS algorithm if the recall is acceptable and the accuracy is high it can be used.

6. Applications and future works

This paper has presented the concepts of *a posteriori* and *a priori* similitudes, and some ways of compute them. These methods described in section IV, are of a general nature and could be potentially applied as a search subroutine of other algorithms including among others: approximate opening book construction, node allocation for message traffic reduction in massive parallelizations, pro-games database clustering, winning-probability approximation by comparison, and combining results from local search of life and death subproblem with MCTS-based algorithms.

As mentioned in section V, the *a posteriori* similitude, but both a theoretical analysis of its characteristics, in order to proof that it's natural induced distance defines a pseudometric space; and a practical evaluation on the analysis of game record databases are required.

And nevertheless, although the other proposed method has the advantage of being really flexible, on the other hand, a proper optimization of each *a priori* similitude parameter is required to accommodate each of the concrete applications mentioned above; once this required is fulfilled, a more practical evaluation can be finally done.

In summary, we have introduced the new idea of similitude between Go board position (with and without the requirement of an evaluation function), and the next logic step is to apply it to proper Go subproblems and see its real performance. These are the next directions for our future research.

References

[1] Coulom R.: Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search. Proceedings of the 5th international conference on Computers and games, 2006

[2] IEEE CIG: Human vs. Computer Go Competition 2015. Source: http://oase.nutn.edu.tw/IEEECIG2015/result.htm

[3] Honda T., Viennot S., and Ikeda. K.: Broad Pattern Matching for the Opening Moves in Go. (Japanese) 17th Game Programming Workshop, 2012.

[4] Mullins, J., and Drake, P.: .Using Human Knowledge to Improve Opening Strategy in Computer Go. In Proceedings of the 2010 International Conference on Artificial Intelligence, CSREA Press, 2010.

[5] Yoshizoe K., Kishimoto A., Kaneko T., Yoshimoto H., and Ishikawa Y.: Scalable Distributed Monte-Carlo Tree Search. In Proceedings of the 4th Symposium on Combinatorial Search (SoCS'2011), pages 180-187, 2011.

[6] Takeuchi S., Kaneko T., and Yamaguchi K.: Evaluation of Game Tree Search Methods by Game Records. IEEE Transactions on Computational Intelligence and AI in Games, 2 (4), 288 - 302, Dec. 2010.

[7] Confucius: Analects, Book 17, Chapter 22, 500 BC.

[8] The Editors of Encyclopædia Britannica: Go game. In Encyclopædia Britannica (online). Source: http://global.britannica.com/topic/go-game

[9] Davis, J.: The Japanese Rules of Go [English translation]. Nihon Kiin and Kansai Kiin (1989). Source: http://www.cs.cmu.edu/ wjh/go/rules/Japanese.html

[10] Zobrist A.: A New Hashing Method with Application for Game Playing. Tech. Rep. 88, University of Wisconsin, Madison, Wisconsin, 1969.

[11] Zobrist A.: A model of visual organization for the game of Go. Ph.D. thesis, University of Wisconsin, Madison, Wisconsin, 1969.

[12] Ryder J.: Heuristic analysis of large trees as generated in the game of Go. Ph.D. thesis, Stanford University, 1971.

[13] Spight W.: Wave Front Analysis. Sensei's Library, 2003. Source: http://senseis.xmp.net/?WaveFrontAnalysis

[14] Bouzy B.: Mathematical morphology applied to Computer Go. International Journal of Pattern Recognition and Artificial Intelligence, 17(2):257268, 2003.