Learning to Estimate Potential Territory in the Game of Go

Erik C. D. van der Werf, H. Jaap van den Herik, and Jos W. H. M. Uiterwijk

Search and Games Group, IKAT, Department of Computer Science,
Universiteit Maastricht, P.O. Box 616, 6200 MD Maastricht, The Netherlands
email{e.vanderwerf,herik,uiterwijk}@cs.unimaas.nl

Abstract. This paper investigates methods for estimating potential territory in the game of Go. We have tested the performance of direct methods known from the literature, which do not require a notion of life and death. Several enhancements are introduced which can improve the performance of the direct methods. New trainable methods are presented for learning to estimate potential territory from examples. The trainable methods can be used in combination with our previously developed method for predicting life and death [25]. Experiments show that all methods are greatly improved by adding knowledge of life and death.

1 Introduction

Evaluating Go positions is a difficult task [7, 17]. In the last decade the game of Go[1] has received significant attention from AI research [5, 16]. Yet, despite all efforts, the best Go programs are still weak. An important reason lies in the lack of an adequate full-board evaluation function. Building such a function requires a method for estimating potential territory. At the end of the game territory is defined as the intersections that are controlled by one colour. Together with the captured or remaining stones, territory determines who wins the game. For final positions (where both sides have completely sealed off the territory by stones of their colour) territory is determined by detecting and removing dead stones and assigning the empty intersections to their surrounding colour. Recently we have developed a system that learns to detect dead stones and that scores final positions at the level of at least a 7-kyu player [23]. Even more recently we have extended this system so that it is now able to predict life and death in non-final positions too [25].

In this paper we focus on evaluating non-final positions. In particular we deal with the task of estimating potential territory in non-final positions, which is much more difficult than determining territory in final positions. We believe that for both tasks predictions of life and death are a valuable component. We investigate several possible methods to estimate potential territory based on the

[1] For general information about the game including an introduction to the rules readers are referred to gobase.org [22].

predictions of life and death and compare them to other approaches, known from the literature, which do not require an explicit notion of life and death.

The remainder of this paper is organised as follows. First, in section 2 we define potential territory. Then, in section 3 we discuss five direct methods for estimating (potential) territory as well as two enhancements for supplying them with information about life and death. In section 4 we describe trainable methods for learning to estimate potential territory from examples. Section 5 presents our experimental setup. Then, in section 6 we present our experimental results. Finally, section 7 provides our conclusion and suggestions for future research.

2 Defining potential territory

The game of Go is played by two players, Black and White, who consecutively place a stone of their colour on an empty intersection of a square grid. At the start of the game the board is empty. During the game the moves gradually divide the intersections between Black and White. In the end the player who controls most intersections wins the game. Intersections that are controlled by one colour at the end of the game are called territory.

During the game human players typically try to estimate the territory that they will control at the end of the game. Moreover, they often distinguish between *secure territory*, which is assumed to be safe from attack, and *regions of influence*, which are unsafe. An important reason why human players like to distinguish secure territory from regions of influence is that, since the secure territory is assumed to be safe, they do not have to consider moves inside secure territory, which reduces the number of candidate moves to choose from.

In principle, secure territory can be recognised by extending Benson's method for recognising unconditional life [2], such as described in [15] or [24]. In practice, however, these methods are not sufficient to predict accurately the outcome of the game until the late end-game because they aim at 100 per cent certainty, which is assured by assumptions like losing all ko-fights, allowing the opponent to place several moves without the defender answering, and requiring completely enclosed regions. Therefore, such methods usually leave too many points undecided.

An alternative (probably more realistic) model of the human notion of secure territory may be obtained by identifying regions with a high confidence level. However, finding a good threshold for distinguishing regions with a high confidence level from regions with a low confidence level is a non-trivial task and admittedly always a bit arbitrary. As a consequence it may be debatable to compare heuristic methods to methods with a 100 per cent confidence level. Subsequently the debate continues when comparing among heuristic methods, e.g., a 77 per cent versus a 93 per cent confidence level (cf. Figure 1).

In this paper, our main interest is in evaluating positions with the purpose of estimating the score. For this purpose the distinction between secure territory and regions of influence is relatively unimportant. Therefore we combine the two notions into one definition of *potential territory*.

Definition 1. *In a position, available from a game record, an intersection is defined as potential territory of a certain colour if the game record shows that the intersection is controlled by that colour at the end of the game.*

Although it is not our main interest, it is possible to use our estimates of potential territory to provide a heuristic estimate of secure territory. This can be done by focusing on regions with a high confidence level, by setting an arbitrarily high threshold. In subsection 6.3 we will present results at various levels of confidence so that our methods can be compared more extensively to methods that are designed for regions with a high confidence level only.

3 Direct methods for estimating territory

In this section we present five direct methods for estimating territory (subsections 3.1 to 3.5). They are known or derived from the literature and are easy to implement in a Go program. All methods assign a scalar value to each (empty) intersection. In general, positive values are used for intersections controlled by Black, and negative values for intersections controlled by White. In subsection 3.6 we mention two immediate enhancements for adding knowledge about life and death to the direct methods.

3.1 Explicit control

The explicit-control function is obtained from the ‘concrete evaluation function’ as described by Bouzy and Cazenave [5]. It is probably the simplest possible evaluation function and is included here as a baseline reference of performance. The explicit-control function assigns $+1$ to empty intersections which are completely surrounded by black stones and -1 to empty intersections which are completely surrounded by white stones, all other empty intersections are assigned 0.

3.2 Direct control

Since the explicit-control function only detects completely enclosed intersections (single-point eyes) as territory it performs quite weak. Therefore we propose a slight modification of the explicit-control function, called direct control. The direct-control function assigns $+1$ to empty intersections which are adjacent to a black stone and not adjacent to a white stone, -1 to empty intersections which are adjacent to a white stone and not adjacent to a black stone, and 0 to all other empty intersections.

3.3 Distance-based control

Both the explicit-control and the direct-control functions are not able to recognise larger regions surrounded by (loosely) connected stones. A possible alternative is the distance-based control (DBC) function. Distance-based control uses the Manhattan distance to assign $+1$ to each empty intersection which is closer to a black stone, -1 to each empty intersection which is closer to a white stone, and 0 to all other empty intersections.

3.4 Influence-based control

Although distance-based control is able to recognise larger territories a weakness is that it does not take into account the strength of stones in any way, i.e., a single stone is weighted equally important as a strong large block at the same distance. A way to overcome this weakness is by the use of influence functions, which were already described by the early researchers in computer Go Zobrist [26] and Ryder [20], and are still in use in several of today's Go programs [8, 9].

In this paper we adopt Zobrist's method to recognise influence; it works as follows. First, all intersections are initialised by one of three values: $+50$ if they are occupied by a black stone, -50 if they are occupied by a white stone, and 0 otherwise. (It should be noted that the value of 50 has no specific meaning and any other large value can be used in practice.) Then the following process is performed four times. For each intersection, add to the absolute value of the intersection the number of neighbouring intersections of the same sign minus the number of neighbouring intersections of the opposite sign.

3.5 Bouzy's method

It is important to note that the repeating process used to radiate the influence of stones in the Zobrist method is quite similar to the dilation operator known from mathematical morphology. This was remarked by Bouzy [3] who proposed a numerical refinement of the classical dilation operator which is similar (but not identical) to Zobrist's dilation.

Bouzy's dilation operator D_z works as follows. For each non-zero intersection which is not adjacent to an intersection of the opposite sign, take the number of neighbouring intersections of the same sign and add it to the absolute value of the intersection. For each zero intersection with positive adjacent intersections only, add the number of positive adjacent intersections. For each zero intersection with negative adjacent intersections only, subtract the number of negative adjacent intersections.

Bouzy argued that dilations alone are not the best way to recognise territory. Therefore he suggested that the dilations should be followed by a number of erosions. This combined form is similar to the classical closing operator known from mathematical morphology.

To do this numerically Bouzy proposed the following refinement of the classical erosion operator E_z. For each non-zero intersection subtract from its absolute value the number of adjacent intersections which are zero or have the opposite sign. If this causes the value of the intersection to change its sign the value becomes zero.

The operators E_z and D_z are then combined by first performing d times D_z followed by e times E_z. Bouzy suggested the relation $e = d(d-1)+1$ because this becomes the unity operator for a single stone in the centre of a sufficiently large board. He further recommended to use the values 4 or 5 for d.

The reader may be curious why the number of erosions is larger than the number of dilations. The main reason is that (unlike in the classical binary case)

Bouzy's dilation operator propagates faster than his erosion operator. Furthermore, Bouzy's method seems to be more aimed at recognising secure territory with a high confidence level than Zobrist's method (the intersections with a lower confidence level are removed by the erosions). Since Bouzy's method leaves many intersections undecided it is expected to perform sub-optimal at estimating potential territory, which also includes regions with lower confidence levels (cf. subsection 6.3). To improve the estimations of potential territory it is therefore interesting to consider an extension of Bouzy's method for dividing the remaining empty intersections. A natural choice to extend Bouzy's method is to divide the undecided empty intersections using distance-based control. The reason why we expect this combination to be better than only performing distance-based control directly from the raw board is that radiating influence from a (relatively) safe base, as provided by Bouzy's method, implicitly introduces some understanding of life and death. (It should be noted that extending Bouzy's method with distance-based control is not the only possible choice, and extending with for example influence-based control provides nearly identical results.)

3.6 Enhanced direct methods

The direct methods all share one important weakness: the lack of understanding life and death. As a consequence, dead stones (which are removed at the end of the game) can give the misleading impression of providing territory or reducing the opponent's territory. Recognising dead stones is a difficult task, but many Go programs have available some kind of (usually heuristic) information about the life-and-death status of stones. We have this information, provided by our recently developed system which has been trained to predict life and death for non-final positions [25].

Here we mention two immediate enhancements for the direct methods. (1) The simplest approach to use information about life and death for the estimation of territory is to remove dead stones before applying one of the direct methods. (2) An alternative sometimes used is to reverse the colour of dead stones [4].

4 Trainable methods

Although the direct methods can be improved by (1) removing dead stones, or (2) reversing their colour, neither approach seems optimal, especially because both lack the ability to exploit the more subtle differences in the strength of stones, which would be expressed by human concepts such as ‘aji’ or ‘thickness’. However, since it is not well understood how such concepts should be modelled, it is tempting to try a machine-learning approach to train a general function approximator to provide an estimation of the potential territory. For this task we have selected the Multi-Layer Perceptron (MLP). The MLP has been used on similar tasks by several other researchers [10–12, 21], so we believe it is a reasonable choice. Nevertheless it should be clear that any other general function approximator can be used for the task.

Our MLP has a feed-forward architecture which estimates potential territory on a per intersection basis. The estimates are based on a local representation which includes features that are relevant for predicting the status of the intersection under investigation. In this paper we test two representations, first a simple one which only looks at the raw configuration of stones, and second an enhanced representation that exploits additional information about life and death.

For our experiments we exploit the fact that the game is played on a square board with eight symmetries. Furthermore, positions with Black to move are equal to positions with White to move provided that all stones reverse colour. To simplify the learning task we remove the symmetries in our representation by rotating the view on the intersection under investigation to one canonical region in the corner, and reversing the colours if the player to move is White.

4.1 The simple representation

The simple representation is characterised by the configuration of all stones in the region of interest (ROI) which is defined by all intersections within a predefined Manhattan distance of the intersection under investigation. For each intersection in the ROI we include the following feature:

- *Colour:* +1 if the intersection contains a black stone, −1 if the intersection contains a white stone, and 0 otherwise.

The simple representation will be compared to the direct methods because it does not use any explicit information of life and death (although some knowledge of life and death may of course be learned from examples) and only looks at the local configuration of stones. Since both Zobrist's and Bouzy's method (see above) are diameter limited by the number of times the dilation operator is used, our simple representation should be able to provide results which are at least comparable. However, we actually expect it to do better because the MLP might learn some additional shape-dependent properties.

4.2 The enhanced representation

We enhanced the simple representation with features that incorporate explicit information about the life-and-death status of stones. Of course, we used our recently developed system for predicting life and death [25]. The most straightforward way to include the predictions of life and death would be to add these predictions as an additional feature for each intersection in the ROI. However, preliminary experiments showed that this was not the best way to add knowledge of life and death. (The reason is that adding features reduces performance due to peaking phenomena caused by the curse of dimensionality [1, 14].) As an alternative which avoids increasing the dimensionality we decided to multiply the value of the colour feature in the simple representation with the estimated probability that the stones are alive. (This means that the sign of the value of an intersection indicates the colour, and the absolute value indicates some kind of strength.) Consequently, the following three features were added.

- *Edge:* encoded by a binary representation (board=0, edge=1) using a 9-bit string vector along the horizontal and vertical line from the intersection under investigation to the nearest edges.
- *Nearest colour:* the classification for the intersection using the distance-based control method on the raw board (black=1, empty=0, white=−1).
- *Nearest alive:* the classification for the intersection using the distance-based control method after removing dead stones (black=1, empty=0, white=−1).

5 Experimental setup

In this section we discuss the data set used for training and evaluation (5.1) and the performance measures used to evaluate the various methods (5.2).

5.1 The data set

In the experiments we used our collection of 9×9 game records which were originally obtained from NNGS [18]. The games, played between 1995 and 2002, were all played to the end and then scored. Since the original NNGS game records only contained a single numeric value for the score, the fate of all intersections was labelled by a threefold combination of GnuGo [13], our own learning system, and some manual labelling. Details about the data set and the way we labelled the games can be found in [23].

In all experiments, test examples were extracted from games played in 1995, training examples were extracted from games played between 1996 and 2002. In total the test set contained 906 games, 46,616 positions, and 2,538,152 empty intersections. It is remarked that the 1995 games were also left out of the training set used for learning to predict life and death [25].

5.2 The performance measures

Now that we have introduced a series of methods (combinations of methods are possible too) to estimate (potential) territory, an important question is: how good are they? We attempt to answer this question (in section 6) using several measures of performance which can be calculated from labelled game records (see section 5.1). Although game records are not ideal as an absolute measure of performance (because the people who played those games surely have made mistakes) we believe that the performance averaged over large numbers of unseen game records is a reasonable indication of strength.

Probably the most important question in assessing the quality of an evaluation function is how well it can predict the winner at the end of the game. By combining the estimated territory with the (alive) stones we obtain the so-called area score, which is the number of intersections controlled by Black minus the number of intersections controlled by White. Together with a possible komi (which compensates the advantage of the first player) the sign of this score determines the winner. Therefore, our first performance measure P_{winner} is the percentage of positions in which the sign of the score is predicted correctly.

Our second performance measure P_{score} uses the same score to calculate the average absolute difference between the predicted score and the actual score at the end of the game.

Both P_{winner} and P_{score} combine predictions of stones and territory in one measure of performance. As a consequence these measures are not sufficiently informative to evaluate the task of estimating potential territory alone. To provide more detailed information about the errors that are made by the various methods we also calculate the confusion matrices (see section 6.1) for the estimates of potential territory alone.

Since some methods leave more intersections undecided (by assigning empty) than others, for example because they may have been designed originally for estimating secure territory only, it may seem unfair to compare them directly using only P_{winner} and P_{score}. As an alternative the fraction of intersections which are left undecided can be considered together with the performance on intersections which are decided. This typically leads to a trade-off curve where performance can be improved by rejecting intersections with a low confidence. The fraction of intersections that are left undecided, as well as the performance on the decided intersections is directly available from the confusion matrices of the various methods.

6 Experimental results

We tested the performance of the various methods and in this section we present our experimental results. They are subdivided as follows: performance of direct methods in 6.1; performance of trainable methods in 6.2; comparing different levels of confidence in 6.3; and performance over the game in 6.4.

6.1 Performance of direct methods

The performance of the direct methods was tested on all positions from the labelled test games. The results for P_{winner} and P_{score} are shown in Table 1. In this table the columns 'remain' represent results without using knowledge of life and death, the columns 'remove' and 'reverse' represent results with predictions of life and death used to remove or reverse the colour of dead stones.

To compare the results of P_{winner} and P_{score} it is useful to have a confidence interval. Unfortunately, since positions of the test set are not all independent, it is non-trivial to provide exact results. Nevertheless it is easy to calculate lower and upper bounds, based on an estimate of the number of independent positions. If we pessimistically assume only one independent position per game an upper bound (for a 95% confidence interval) is roughly 3% for P_{winner} and 1.2 points for P_{score}. If we optimistically assume all positions to be independent a lower bound is roughly 0.4% for P_{winner} and 0.2 points for P_{score}. Of course this is only a crude approximation which ignores the underlying distribution and the fact that the accuracy increases drastically towards the end of the game. However, given the fact that the average game length is around 50 moves it seems safe to

Table 1. Average performance of direct methods.

	P_{winner} (%)			P_{score} (points)		
Predicted dead stones	remain	remove	reverse	remain	remove	reverse
Explicit control	52.4	60.3	61.8	16.0	14.8	14.0
Direct control	54.7	66.5	66.9	15.9	12.9	12.7
Distance-based control	60.2	73.8	73.8	18.5	13.8	13.9
Influence-based control	61.0	73.6	73.6	17.3	12.8	12.9
Bouzy(4,13)	52.6	66.9	67.5	17.3	12.8	12.8
Bouzy(5,21)	55.5	70.2	70.4	17.0	12.3	12.4
Bouzy(5,21) + DBC	63.4	73.9	73.9	18.7	14.5	14.6

assume that the true confidence interval will be somewhere in the order of 1% for P_{winner} and 0.4 points for P_{score}.

More detailed results about the estimations (in percentages) for the empty intersections alone are presented in the confusion matrices shown in Table 2. The fraction of undecided intersections and the performance on the decided intersections, which can be calculated from the confusion matrices, will be discussed in subsection 6.3. (The rows of the confusion matrices contain the possible predictions which are either black (PB), white (PW), or empty (PE). The columns contain the actual labelling at the end of the game which are either black (B), white (W), or empty (E). Therefore, correct predictions are found on the trace, and errors are found in the upper right and lower left corners of the matrices.)

The difference in performance between (1) when stones remain on the board and (2) when dead stones are removed or reversed colour underlines the importance of understanding life and death. For the weakest direct methods reversing the colour of dead stones seems to improve performance compared to only removing them. For the stronger methods, however, it has no significant effect.

The best method for predicting the winner without understanding life and death is Bouzy's method extended with distance-based control to divide the remaining undecided intersections. It is interesting to see that this method also has a high P_{score} which would actually indicate a bad performance. The reason for this is instability of distance-based control in the opening, e.g., with only one stone on the board it assigns the whole board to the colour of that stone. We can filter out the instability near the opening by only looking at positions that occur after a certain minimal number of moves. When we do this for all positions with at least 20 moves made, as shown in Table 3, it becomes clear that Bouzy's method extended with distance-based control also achieves the best P_{score}. Our experiments indicate that radiating influence from a (relatively) safe base, as provided by Bouzy's method, outperforms other direct methods probably because it implicitly introduces some understanding of life and death. This conclusion is supported by the observation that the combination does not perform significantly better than for example influence-based control when knowledge about life and death is used.

At first glance the results presented in this subsection could lead to the tentative conclusion that for a method which only performs N dilations to estimate potential territory the performance keeps increasing with N; so the largest possible N might have the best performance. However, this is not the case and N should not be chosen too large. Especially in the beginning of the game a large N tends to perform significantly worse than a restricted setting with 4 or 5 dilations such as used by Zobrist's method. Moreover, a too large N is a waste of time under tournament conditions.

Table 2. Confusion matrices of direct methods.

	B	W	E
PB	0.78	0.16	0
PW	0.1	0.88	0
PE	48.6	49.3	0.13

Explicit control

	B	W	E
PB	0.74	0.04	0
PW	0.04	0.82	0
PE	48.7	49.5	0.14

dead stones removed

	B	W	E
PB	0.95	0.09	0.01
PW	0.07	1.2	0.01
PE	48.4	49.0	0.12

dead colour reversed

	B	W	E
PB	15.4	4.33	0.02
PW	3.16	14.3	0.01
PE	30.9	31.6	0.1

Direct control

	B	W	E
PB	16.0	3.32	0.02
PW	2.56	15.2	0.02
PE	30.9	31.7	0.09

dead stones removed

	B	W	E
PB	16.6	3.44	0.02
PW	2.75	16.3	0.03
PE	30.0	30.6	0.08

dead colour reversed

	B	W	E
PB	36.0	11.6	0.05
PW	6.63	31.0	0.03
PE	6.86	7.71	0.06

Distance-based control

	B	W	E
PB	38.2	10.4	0.06
PW	6.21	34.3	0.06
PE	5.09	5.55	0.02

dead stones removed

	B	W	E
PB	38.0	10.3	0.05
PW	6.21	34.1	0.05
PE	5.29	5.87	0.04

dead colour reversed

	B	W	E
PB	37.4	12.2	0.07
PW	7.76	33.3	0.04
PE	4.25	4.79	0.03

Influence-based control

	B	W	E
PB	38.4	10.5	0.06
PW	7.02	35.3	0.06
PE	4	4.47	0.02

dead stones removed

	B	W	E
PB	38.4	10.5	0.06
PW	7.11	35.4	0.06
PE	3.98	4.46	0.02

dead colour reversed

	B	W	E
PB	17.7	1.82	0.02
PW	0.83	15.4	0.01
PE	30.9	33.1	0.11

Bouzy(4,13)

	B	W	E
PB	21.2	1.86	0.03
PW	1.06	19.9	0.03
PE	27.2	28.5	0.07

dead stones removed

	B	W	E
PB	21.2	1.9	0.03
PW	1.16	20.1	0.03
PE	27.0	28.3	0.08

dead colour reversed

	B	W	E
PB	19.0	1.87	0.02
PW	0.81	15.8	0.01
PE	29.6	32.6	0.1

Bouzy(5,21)

	B	W	E
PB	23	1.98	0.03
PW	1.13	20.8	0.04
PE	25.3	27.5	0.07

dead stones removed

	B	W	E
PB	22.9	1.99	0.03
PW	1.17	20.7	0.03
PE	25.3	27.6	0.08

dead colour reversed

	B	W	E
PB	37.9	12.1	0.05
PW	6.51	32.4	0.03
PE	5	5.73	0.05

Bouzy(5,21) + DBC

	B	W	E
PB	39.0	11.0	0.06
PW	6.3	34.8	0.06
PE	4.12	4.5	0.02

dead stones removed

	B	W	E
PB	38.9	10.9	0.05
PW	6.32	34.6	0.05
PE	4.28	4.74	0.03

dead colour reversed

Table 3. Average performance of direct methods after 20 moves.

	P_{winner} (%)			P_{score} (points)		
Predicted dead stones	remain	remove	reverse	remain	remove	reverse
Explicit control	55.0	66.2	68.2	16.4	14.5	13.3
Direct control	57.7	74.9	75.3	16.1	11.6	11.3
Distance-based control	61.9	82.1	82.1	16.4	9.6	9.7
Influence-based control	63.6	82.1	82.2	16.0	9.5	9.6
Bouzy(4,13)	56.7	77.9	78.3	17.2	10.4	10.5
Bouzy(5,21)	58.6	80.3	80.5	16.9	9.9	10
Bouzy(5,21) + DBC	66.7	82.2	82.3	15.5	9.6	9.7

6.2 Performance of trainable methods

Below we present the results of the trainable methods. All architectures were trained with the resilient propagation algorithm (RPROP) developed by Riedmiller and Braun [19]. The non-linear architectures all had one hidden layer with 25 units using the hyperbolic tangent sigmoid transfer function. (Preliminary experiments showed this to be a reasonable setting, though large networks may still provide a slightly better performance when more training examples are used.) For training, 200,000 examples were used. A validation set of 25,000 examples was used to stop training. For each architecture the weights were trained three times with different random initialisations, after which the best result was selected according to the performance on the validation set. (Note that the validation examples were taken, too, from games played between 1996 and 2002.)

We tested the various linear and non-linear architectures on all positions from the labelled test games. Results for P_{winner} and P_{score} are presented in Table 4, and the confusion matrices are shown in Table 5. The enhanced representation, which uses predictions of life and death, clearly performs much better than the simple representation. We further see that the performance tends to improve with increasing size of the ROI. (A ROI of size 24, 40, and 60 corresponds to the number of intersections within a Manhattan distance of 3, 4, and 5 respectively, excluding the centre point which is always empty.)

Table 4. Performance of the trainable methods.

Architecture	Representation	ROI	P_{winner} (%)	P_{score} (points)
linear	simple	24	64.0	17.9
linear	simple	40	64.5	18.4
linear	simple	60	64.6	19.0
non-linear	simple	24	63.1	18.2
non-linear	simple	40	64.5	18.3
non-linear	simple	60	65.1	18.3
linear	enhanced	24	75.0	13.4
linear	enhanced	40	75.2	13.3
linear	enhanced	60	75.1	13.4
non-linear	enhanced	24	75.2	13.2
non-linear	enhanced	40	75.5	12.9
non-linear	enhanced	60	75.5	12.5

Table 5. Confusion matrices of trainable methods.

	B	W	E
PB	40.5	13.6	0.08
PW	6.7	33.8	0.05
PE	2.3	2.9	0.01

Simple, linear, roi=24

	B	W	E
PB	41.5	14.2	0.08
PW	5.8	33.2	0.05
PE	2.1	2.9	0.01

Simple, linear, roi=40

	B	W	E
PB	42.0	14.6	0.08
PW	5.3	32.6	0.04
PE	2.2	3.1	0.01

Simple, linear, roi=60

	B	W	E
PB	40.5	13.6	0.07
PW	6.4	33.3	0.05
PE	2.6	3.5	0.02

Simple, non-linear, roi=24

	B	W	E
PB	41.4	14.0	0.08
PW	5.8	33.0	0.04
PE	2.4	3.3	0.02

Simple, non-linear, roi=40

	B	W	E
PB	41.8	14.2	0.0759
PW	5.5	33.0	0.04
PE	2.2	3.2	0.01

Simple, non-linear, roi=60

	B	W	E
PB	40.5	11.7	0.06
PW	7.0	36.4	0.07
PE	2.0	2.3	0.01

Enhanced, linear, roi=24

	B	W	E
PB	40.6	11.6	0.06
PW	6.8	36.3	0.07
PE	2.1	2.5	0.01

Enhanced, linear, roi=40

	B	W	E
PB	40.6	11.6	0.06
PW	6.6	36.2	0.07
PE	2.2	2.6	0.01

Enhanced, linear, roi=60

	B	W	E
PB	40.3	11.4	0.06
PW	6.8	36.2	0.06
PE	2.4	2.7	0.01

Enhanced, non-linear, roi=24

	B	W	E
PB	40.4	11.3	0.06
PW	6.8	36.4	0.06
PE	2.4	2.7	0.01

Enhanced, non-linear, roi=40

	B	W	E
PB	40.2	10.9	0.06
PW	6.8	36.6	0.07
PE	2.5	2.9	0.01

Enhanced, non-linear, roi=60

It is interesting to see that the non-linear architectures are not much better than the linear architectures. This seems to indicate that, once life and death has been established, influence spreads mostly linearly.

6.3 Comparing different levels of confidence

The MLPs are trained to predict positive values for black territory and negative values for white territory. Small values close to zero indicate that intersections are undecided and by adjusting the size of the window around zero, in which we predict empty, we can modify the confidence level of the non-empty classifications. If we do this we can plot a trade-off curve which shows how the performance increases at the cost of rejecting undecided intersections.

In Figure 1 two such trade-off curves are shown for the simple MLP and the enhanced MLP, both non-linear with a ROI of size 60. For comparison, results for the various direct methods are also plotted. It is shown that the MLPs perform well at all levels of confidence. Moreover, it is interesting to see that at high confidence levels Bouzy(5,21) performs nearly as good as the MLPs.

Although Bouzy's methods and the influence methods provide numerical results, which could be used to plot trade-off curves, too, we did not do this because they would make the plot less readable. Moreover, for Bouzy's methods the lines would be quite short and uninteresting because they already start high.

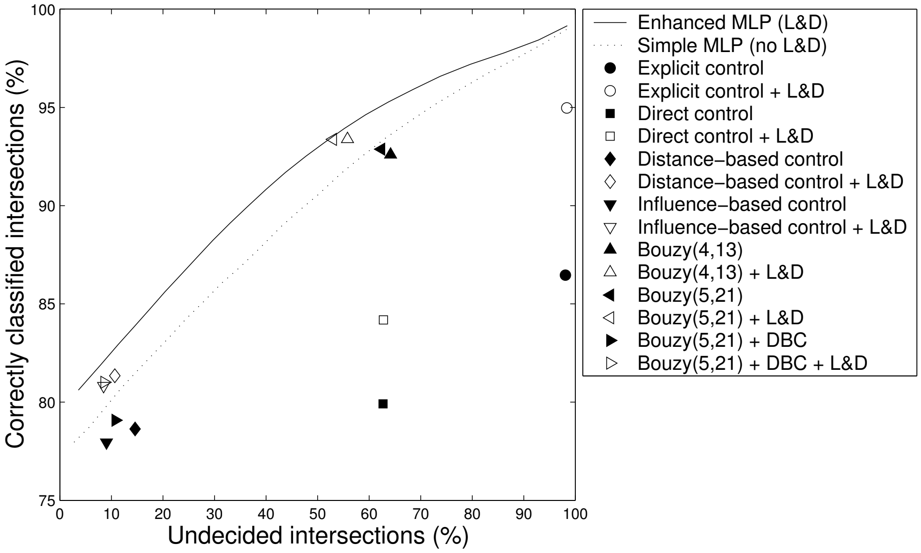


Fig. 1. Performance at different levels of confidence.

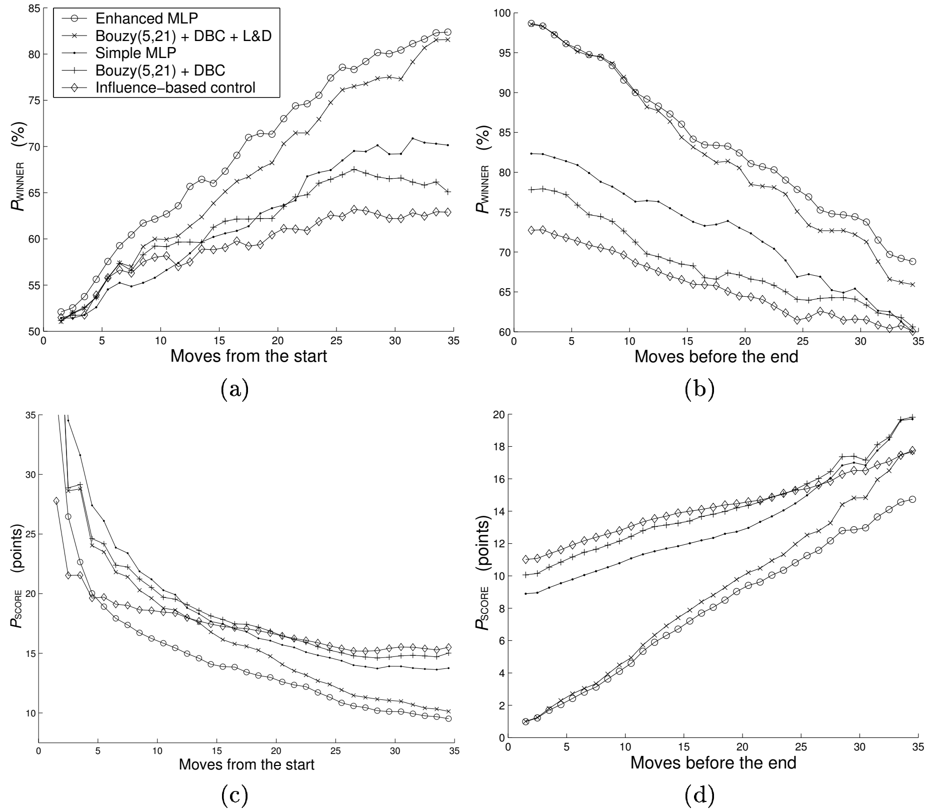
6.4 Performance over the game

In the previous subsections we looked at the average performance over complete games. Although this is interesting, it does not tell us how the performance changes as the game develops. Below we consider the performance changes and the adequacy of the MLP performance.

Since all games do not have equal length, there are two principal ways of looking at the performance. First, we can look forward from the start, and second, we can look backward from the end. The results for P_{winner} are shown in Figure 2a looking forward from the start and in Figure 2b looking backward from the end. We remark that the plotted points are between moves and their associated performance is the average obtained for the two directly adjacent positions (where one position has Black to move and the other has White to move). This was done to filter out some distracting odd-even effects caused by the alternation of the player to move. It is shown that the MLP using the enhanced representation performs best. However, close to the end Bouzy's method extended with distance-based control and predictions of life and death performs nearly as good. The results for P_{score} are shown in Figure 2c looking forward from the start and in Figure 2d looking backward from the end. Also here we see that the MLP using the enhanced representation performs best.

For clarity of presentation we did not plot the performance of DBC, which is rather similar to Influence-based control (IBC) (but over-all slightly worse). For the same reason we did not plot the results for DBC and IBC with knowledge of life and death, which perform quite similar to Bouzy(5,21)+DBC+L&D.

It is interesting to observe how good the simple MLP performs. It outperforms all direct methods without using life and death. Here it should be noted



(a) (b)

(c) (d)

Fig. 2. Performance over the game.

that the adequate performance of the simple MLP could still be improved considerably, if it would be allowed to make predictions for occupied intersections too, i.e., remove dead stones. (This was not done for a fair comparison with the direct methods.)

7 Conclusion and future research

We have investigated several direct and trainable methods for estimating potential territory in the game of Go. We tested the performance of the direct methods, known from the literature, which do not require an explicit notion of life and death. Additionally, two enhancements for adding knowledge of life and death and an extension of Bouzy's method were presented. From the experiments we may conclude that without explicit knowledge of life and death the best direct method is Bouzy's method extended with distance-based control to divide the remaining empty intersections. If information about life and death is used to remove dead stones this method also performs well. However, the difference with distance-based and influence-based control becomes small.

Moreover, we presented new trainable methods for estimating potential territory. They can be used as an extension of our system for predicting life and death. Using only the simple representation our trainable methods can estimate potential territory at a level outperforming the best direct methods. Experiments showed that all methods are greatly improved by adding knowledge of life and death, which leads us to conclude that good predictions of life and death are the most important ingredients for an adequate full-board evaluation function.

Future research

Although our system for predicting life and death already performs quite well, we believe that it can still be improved significantly. The most important reason is that we only use static features, which do not require search. Incorporating features from specialised life-and-death searches should improve predictions of life and death as well as estimations of potential territory.

Previous work on learning to score final positions [23] indicated that our system for predicting life and death scales up well to the 19×19 board. Although we expect similar results for estimating potential territory, additional experiments should be performed to validate this claim.

In this paper we estimated potential territory based on knowledge extracted from game records. An interesting alternative for acquiring such knowledge may be obtaining it by simulation using, e.g., Monte Carlo methods [6].

Acknowledgements

This work was funded by the Netherlands Organisation for Scientific Research (NWO), dossier number 612.052.003. We would like to thank the anonymous referees for their helpful comments that significantly improved this paper.

References

1. R. E. Bellman. *Adaptive Control Processes: A Guided Tour.* Princeton University Press, Princeton, New Jersey, 1961.
2. D. B. Benson. Life in the game of Go. *Information Sciences*, 10:17–29, 1976. Reprinted in Computer Games, Levy, D.N.L (editor), Vol. II, pp. 203-213, Springer Verlag, New York, 1988. ISBN 0-387-96609-9.
3. B. Bouzy. Mathematical morphology applied to computer Go. *International Journal of Pattern Recognition and Artificial Intelligence*, 17(2):257–268, March 2003.
4. B. Bouzy, 2004. Personal communication.
5. B. Bouzy and T. Cazenave. Computer Go: An AI oriented survey. *Artificial Intelligence*, 132(1):39–102, October 2001.
6. B. Brügmann. Monte Carlo Go, March 1993. Available at ftp://ftp.cse.cuhk.edu.hk/pub/neuro/GO/mcgo.tex.
7. K. Chen. Some practical techniques for global search in Go. *ICGA Journal*, 23(2):67–74, 2000.
8. K. Chen. Computer Go: Knowledge, search, and move decision. *ICGA Journal*, 24(4):203–215, 2001.

9. Z. Chen. Semi-empirical quantitative theory of Go part i: Estimation of the influence of a wall. *ICGA Journal*, 25(4):211–218, 2002.
10. F. A. Dahl. Honte, a Go-playing program using neural nets. In J. Fürnkranz and M. Kubat, editors, *Machines that Learn to Play Games*, chapter 10, pages 205–223. Nova Science Publishers, Huntington, NY, 2001.
11. M. Enzenberger. The integration of a priori knowledge into a Go playing neural network, September 1996. Available at http://www.markus-enzenberger.de/neurogo1996.html.
12. M. Enzenberger. Evaluation in Go by a neural network using soft segmentation. In H. J. van den Herik, H. Iida, and E. A. Heinz, editors, *Advances in Computer Games: Many Games, Many Challenges*, pages 97–108, Boston, 2003. Kluwer Academic Publishers.
13. GnuGo, 2003.
14. A. Jain and B. Chandrasekaran. Dimensionality and sample size considerations in pattern recognition practice. In P. R. Krishnaiah and L. N. Kanal, editors, *Handbook of Statistics*, volume 2, pages 835–855. North-Holland, Amsterdam, 1982.
15. M. Müller. Playing it safe: Recognizing secure territories in computer Go by using static rules and search. In H. Matsubara, editor, *Proceedings of the Game Programming Workshop in Japan '97*, pages 80–86. Computer Shogi Association, Tokyo, Japan, 1997.
16. M. Müller. Computer Go. *Artificial Intelligence*, 134(1-2):145–179, January 2002.
17. M. Müller. Position evaluation in computer Go. *ICGA Journal*, 25(4):219–228, 2002.
18. NNGS. The no name Go server game archive, 2002.
19. M. Riedmiller and H. Braun. A direct adaptive method for faster backpropagation: the RPROP algorithm. In H. Rusini, editor, *Proceedings of the IEEE Int. Conf. on Neural Networks (ICNN)*, pages 586–591, 1993.
20. J. L. Ryder. *Heuristic analysis of large trees as generated in the game of Go*. PhD thesis, Stanford University, 1971.
21. N. N. Schraudolph, P. Dayan, and T. J. Sejnowski. Temporal difference learning of position evaluation in the game of Go. In J. D. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing 6*, pages 817–824. Morgan Kaufmann, San Francisco, 1994.
22. J. van der Steen. Gobase.org - Go games, Go information and Go study tools, 2003.
23. E. C. D. van der Werf, H. J. van den Herik, and J. W. H. M. Uiterwijk. Learning to score final positions in the game of Go. In H. J. van den Herik, H. Iida, and E. A. Heinz, editors, *Advances in Computer Games: Many Games, Many Challenges*, pages 143–158, Boston, 2003. Kluwer Academic Publishers.
24. E. C. D. van der Werf, H. J. van den Herik, and J. W. H. M. Uiterwijk. Solving Go on small boards. *ICGA Journal*, 26(2):92–107, 2003.
25. E. C. D. van der Werf, M. H. M. Winands, H. J. van den Herik, and J. W. H. M. Uiterwijk. Learning to predict life and death from Go game records. In Ken Chen *et al.*, editor, *Proceedings of JCIS 2003 7th Joint Conference on Information Sciences*, pages 501–504. JCIS/Association for Intelligent Machinery, Inc., 2003.
26. A. L. Zobrist. A model of visual organization for the game Go. In *Proceedings of AFIPS 1969 Spring Joint Computer Conference*, volume 34, pages 103–112. AFIPS Press, 1969.