24

Predicting Problem Difficulty in Chess

Ivan Bratko[1], Dayana Hristova[2], and Matej Guid[1]

[1] *University of Ljubljana,* [2] *University of Vienna*

24.1 Introduction

A question relevant to explainable AI and human-like computing is: How can we automatically predict the difficulty of a given problem for humans? The practical motivation for predicting task difficulty arises for example in intelligent tutoring systems and computer games. In both cases, the difficulty of problems has to be adjusted to the user. In general, understanding the difficulty for humans of problems that AI tries to solve is a relevant question for human-like computing. If AI systems find problems easy while humans find them hard, and vice versa, then this is evidence that the AI systems are solving the problems in a different way from humans. Also, for computation to be "human-like", it should be easy to understand by humans. Ideally, the system should be able to recognise when the problem or computation gets difficult for humans.

The difficulty of a problem for a human depends on the human's expertise in the domain of the problem, and consequently on how the human would go about solving the problem. The automatic prediction of difficulty could therefore involve a kind of simulation of human problem-solving, which would make prediction of difficulty particularly hard.

In this chapter we discuss an approach to the automatic prediction of difficulty for humans, of problems that are typically solved through informed search. Our experimental domain is the game of chess. Chess has often proved to be an excellent environment for research in human problem-solving. One reason for this, which is important for the present study, is the existence of the FIDE chess federation's rating system for registered players worldwide, and the Chess Tempo website with a large number of chess problems with measured difficulty ratings.

In this chapter we analyse experimental data of human chess players who attempted to solve tactical chess problems and also assess the difficulty of these problems. We carry out an experiment with an approach to predicting the difficulty of problems for humans automatically in this domain.

Ivan Bratko, Dayana Hristova, and Matej Guid, *Predicting Problem Difficulty in Chess* In: *Human Like Machine Intelligence.*
Edited by: Stephen Muggleton and Nick Charter, Oxford University Press.
DOI: 10.1093/oso/9780198862536.003.0024

Solving tactical problems in chess requires search among available alternative moves. The size of the search space is typically much too large for humans to search exhaustively. Good chess players therefore use pattern-based knowledge to guide their search extremely effectively. Problem-solving thus consists of detecting chess patterns—motifs, and the calculation of concrete chess variations trying to exploit these motifs to the player's advantage. What could be such motifs and how motifs are used in chess problem solving is explained in Section 3, where concrete examples of motifs and corresponding problem-solving are given. In our analysis we take into account players' comments on how they tackled individual problems.

Automated estimation of difficulty for humans in chess is hard because it requires the understanding of how humans solve chess problems. Strong chess players use large amount of pattern-based knowledge acquired through experience. To duplicate this vast amount of largely tacit knowledge in the computer is a formidable task that has never been accomplished. Therefore we are interested in alternative ways: estimating difficulty for humans without the use of chess-specific knowledge. In an experiment with such an approach, described in the second part of this chapter, we reduce this need for human players' pattern knowledge to a speculated equivalent: properties of game-tree search deemed to be carried out by strong players. We believe that this approach is applicable to estimating the problem difficulty in other domains where problems are solved through expert knowledge and search.

Related research into the issue of estimating problem difficulty of specific types of puzzles includes the following: Tower of Hanoi (Kotovsky *et al.*, 1985), Chinese rings (Kotovsky and Simon, 1990), 15-puzzle (Pizlo and Li, 2005), Traveling Salesperson Problem (Dry *et al.*, 2006), Sokoban puzzle (Jarušek and Pelánek, 2010), Sudoku (Pelánek, 2011), puzzle games played on grids (Van Kreveld *et al.*, 2015), mathematical puzzles (Sekiya *et al.*, 2019). Kegel and Haarh (2019) review techniques for procedural contents generation for games, paying attention to the difficulty of generated problems.

An early attempt at automated estimation of the difficulty of chess problems was made in Guid and Bratko, 2006. In that paper the authors analysed the quality of chess games played at world championship level. The positions in the analysed games were submitted to a strong chess-playing program, and the best moves (according to the program) were computed. For each player, the average difference per move between the value of the move suggested by the chess program and the value of the move actually played by the player (average loss per move) was computed. It would now be inappropriate simply to rank the players according to their average loss per move because the players' playing styles were different. Some players naturally tended towards quiet, simple positions, and others towards complex positions. In simple positions it is much easier to achieve a small loss than in complex positions. In order to allow a fair comparison, the difficulty of the positions had to be taken into account. An approach to automatic difficulty estimation of a position was therefore designed, essentially based on the amount of search required by the chess program to find the best move in the position. This made it possible to compute average loss per move for each player if all the players were faced with positions of equal difficulty. This approach to difficulty estimation was analysed in detail in Guid

and Bratko, 2013. However, it was found that this approach does not produce realistic estimates of the difficulty for humans in tactical chess problems. Therefore, in Stoiljkovikj *et al.* (2015) a more suitable approach for estimating the difficulty of tactical problems was developed, which will also be used in the experiment in the present chapter.

24.2 Experimental Data

In this study we used the data obtained in an experiment in which 12 chess players of various chess strengths were asked to solve 12 tactical chess problems (Hristova *et al.*, 2014*a*). A chess position is said to be *tactical* if finding the best move in the position requires the calculation of variations, and the solution typically leads to an obvious win after a relatively short sequence of moves.

The chess strength of our players, measured by the FIDE chess ratings, was in the range between 1845 and 2279 rating points. The strength of the registered chess players is officially computed by (World Chess Federation) using the Elo rating system. This rating system was designed by Arpad Elo (1978). This rating is calculated for each player and updated regularly according to the tournament results of the players. The rating range of our players, between 1845 and 2279, means that there were big differences in chess strength between the players. The lowest end of this range corresponds to club players, and the highest end to chess masters (to obtain the FIDE master title, the player must reach at least 2300 points at some point in his career). Among our participants there were actually two chess masters, one of whom also had the title of a female grandmaster. The expected result in a match between the top ranked player in our experiment and our lowest ranked player would be about 92% against 8% (the stronger player winning 92% of all possible points). According to the definition of the Elo rating system, the expected outcome between two players is determined only by the *difference* between their ratings, and not by the ratings themselves. For example, consider two players with ratings 2200 and 2000. The difference is 200 rating points, which determines that the expected success rate of the higher rated player playing against the lower rated player is 76% and the expected success rate of the lower rated player is 24%. The same success rates could be expected if the players' ratings were, say, 2350 and 2150.

In addition to the differences in chess strength expressed by chess ratings, other differences between players could also be taken into account. One such factor might be the chess school where a player was taught, or the particular instructor who trained the player. However, in this chapter we did not explore the effects of such additional factors.

The 12 chess problems were selected from the Chess Tempo website,[1] which is intended for tactical chess training. At Chess Tempo, the problems are rated according to their difficulty. Chess problems are rated in a similar way as the players, except that the evidence does not come from chess games played, but from attempts by chess

[1] The website Chess Tempo is at www.chesstempo.com.

players to solve problems at Chess Tempo. Thus at Chess Tempo a problem's rating is determined by the success of the players in solving the problem. The principle is as follows: if a weak player has solved a problem, this is considered a strong indication that the problem is easy. So the problem's rating goes down. If a stronger player has solved the problem, the rating of the problem still decreases, but not as much as with a weak player. On the contrary, if a strong player failed to solve the problem, this is considered a strong evidence that the problem is hard, and the rating of a problem increases. More specifically, a problem's rating in Chess Tempo is determined by the Glicko rating system (Glickman, 1999), which is similar to the Elo system. Unlike Elo, the Glicko system takes into account the time a player has been inactive. In cases of prolonged inactivity, the player's rating becomes uncertain. It should be noted that the ratings of players—Chess Tempo users—are determined by the evidence of their success in solving problems, and not by their chess-playing results. Otherwise, the meaning of ratings in Chess Tempo is similar to the FIDE ratings of players. So a player with rating 2000 has a 50% chance of correctly solving a problem with rating 2000. The same player has a 76% chance to solve a problem rated 1800, and a 24% chance to solve a problem rated 2200.

In our selection of 12 chess problems we ensured a mixture of problems that largely differ in their difficulty. The problems were randomly selected from Chess Tempo according to their difficulty ratings. Based on their Chess Tempo ratings, our problems can be divided into three classes of difficulty: 'easy' (2 problems; their average Chess Tempo rating was 1493.9), 'medium' (4 problems; average rating 1878.8), and 'hard' (6 problems; average rating 2243.5). While the problems within the same difficulty class have very similar difficulty rating, each of the three classes is separated from their adjacent classes by at least 350 Chess Tempo rating points. Some problems have more than one correct solution. To ensure correctness, all the solutions were verified by a chess-playing program.

The experimental set-up was as follows. Chess problems, that is chess positions, were displayed to a participating player one after the other as chess diagrams on a monitor. For each problem, the player was asked to find a winning move, and the player's solution moves were recorded. The problem-solving time per position was limited to three minutes.

While the player was solving the problem, the player's eye movements were tracked with an eye-tracking device, EyeLink 1000, and recorded in a database. The processing of recorded eye movements roughly reveals on which squares of the chessboard the participant was focussing at any time during the problem-solving process. Observing eye movements has often been used in chess decision-making (Sheridan and Reingold, 2017).

After the player had finished with the 12 problems, a retrospection interview was conducted in which the player described how he or she approached the problem. From these retrospections, one could see which motifs were considered by the player, and roughly how the calculation of variations driven by the motifs was carried out. Finally, the players were asked to sort the 12 problems according to the difficulty of the problems perceived by the players. Further details of the experiment are described in (Hristova *et al.*, 2014*a*, *b*).

The relevant experimental data include the following. For every player and every position we have: (1) the correctness of the solution proposed by the player, (2) the motifs considered by the player compared to the motifs required to solve the problem, and (3) the correctness of the calculation of variations. The motifs considered were found through the players' retrospections, and to some extent verified by the eye movement data, although this verification cannot be done completely reliably.

The data concerning the correctness of the recognition of motifs and the calculation of variations were mostly constructed manually from the submitted solutions of the players and from their retrospections. To decide whether the player detected a complete set of motifs needed to carry out correct calculation, we defined for each position and each possible solution of the position, the 'standard' set of motifs necessary and sufficient to find the solution. In defining the standard sets of relevant motifs, we took into account all the motifs mentioned by all the players. In very rare cases when needed, we had to add motifs that fully enabled correct calculation for each possible solution. In doing so, we used our own chess expertise (two of us have chess ratings over 2300 and 2100 respectively). We verified all the solutions and corresponding chess variations by a chess program, and we believe that it would be hard to come up with reasonable alternative standard sets of motifs.

24.3 Analysis

24.3.1 Relations between player rating, problem rating, and success

We first consider some correlations between success in solving a problem, a player's chess rating, and problem's Chess Tempo rating. We represent the success in solving a problem by 1, and the failure to solve by 0. The total number of data points of the form (Rating, Success) in our experimental data was 142. For 12 problems and 12 players, there are altogether 12*12 = 144 such pairs, however due to misunderstandings during the experiment in two cases invalid results were obtained, so that they were excluded, which finally results in 142 data points.

Sample correlation coefficient between problem's Chess Tempo rating and success was:

```
r(ProblemRating, Success) = -0.345 (P = 0.000027)
```

This is basically as expected: higher problem rating means lower chances of success. Sample correlation between player's chess rating and success was:

```
r(PlayerRating, Success) = 0.077 (P = 0.36)
```

This result is not statistically significant. According to this, there is almost no correlation between player's Elo rating and success in solving a problem, which appears rather surprising. Success depends much more on the Chess Tempo difficulty of the problem than on the players' rating. One attempt at explaining this difference can be

that the differences in Chess Tempo ratings were higher than differences between players' ratings. The ranges were:

```
players' ratings: 2279 - 1845 = 434;
problems' ratings: 2243 - 1492 = 751.
```

Another, more plausible explanation is based on the observation of an important difference between the players' FIDE rating and Chess Tempo problem ratings. Chess Tempo problem ratings and players' FIDE ratings measure different things. The first measures success of individual moves, and the second measures success over long sequences of moves. The players' FIDE ratings are based on the results of complete games (won, or drawn, or lost). Winning a game is the outcome of a sequence of moves (usually about 40 moves). The success in winning a game depends on the sum of the correctness of all moves in the game, and not on the correctness of a single move in the game. A 40 moves game is typically decided by one or two moves where decisive mistakes are made, while the rest of the moves by the two players are of very similar quality. On the other hand, to solve a (single) problem successfully in our experiment, just a correct single first move of a tactical combination was required. This is similar to scoring a correct solution in Chess Tempo, although, to be precise, not exactly the same. The Chess Tempo system, to accept an answer as correct, requires from a player a correct first move, possibly followed by one or more moves in the main variation of the combination. The point of requiring additional moves is to verify that the player actually saw the whole variation and indeed played the first move for the right reasons (and was not just lucky). So, what counted as success in Chess Tempo was not exactly the same as what counted as success in our experiment. Nevertheless, both notions of success refer to solving a single position, which is considerably different from success in winning a game.

24.3.2 Relations between player's rating and estimation of difficulty

The next question of interest is how good chess players are at estimating the difficulty of problems. We take the Chess Tempo (CT for short) difficulty ratings as the gold standard, because they are based on observing large numbers of players' attempts at solving these problems, and the ratings are computed from these observations using an accepted method. So we compare the players' rankings of problems with the CT rankings.

In the experiment, the players did not directly estimate the difficulty ratings of the problems, but each player was asked to rank the 12 problems according to his or her perceived difficulty of the problems. We used Kendall's Tau rank correlation coefficient as a statistical measure of agreement between rankings. Given two rankings, Kendall's Tau is defined as:

$$\tau = (n_c - n_d)/(n_c + n_d) \tag{24.1}$$

Here n_c and n_d are the number of concordant pairs and discordant pairs, respectively. A pair of chess positions is concordant if their relative rankings are the same in both rankings; that is, given two problems, the same problem precedes the other one in both rankings. Otherwise the pair is discordant. In our data, subsets of the positions were, according to Chess Tempo, of very similar difficulty. Such positions belong to the same difficulty class, either 'easy' (CT-rating between 1492 and 1495) or 'medium' (between 1875 and 1883) or 'hard' (between 2231 and 2275). Within the three difficulty classes, we consider any ordering by the players to be acceptable. To account for this, we used a variation of the Tau formula above. When determining n_c and n_d, we only counted the pairs of positions that belong to different Chess Tempo classes. In view of the distribution of problems over the three classes (easy: 2, medium: 4, hard: 6), we have therefore only considered 2*4 + 2*6 + 4*6 = 44 problem pairs.

In (Hristova *et al.*, 2014*a*), we computed Kandall's Tau in this way for each of the 12 players. Then we computed sample correlation between the players' Tau and the players' FIDE ratings. There was a moderate positive relationship (not statistically significant) between Kendall's Tau and the FIDE ratings. We can strengthen this result by considering separately all pairs of positions of different difficulty class, and correctness of the relative rankings of these pairs in the 12 players' difficulty rankings. We represented correctly ordered pairs by 1, and incorrectly by 0. This way we have 44 pairs of problems and 12 players, which gives 12 * 44 = 528 data points. Each data point is of the form (PlayerRating, OrderCorrect), where OrderCorrect is 1 or 0 as stated above. Sample correlation coefficient for this data set is r = −0.196, which is significant (P = 0.000095). This result is as one would expect. It indicates that stronger players are indeed better able to assess the difficulty of problems than weaker players, although the correlation is quite weak, indicating that this relation is rather noisy. Overall, in all the difficulty rankings by all the players, 72.5% of relevant pairs are concordant (a 'relevant pair' is a pair of problems of different Chess Tempo difficulty class).

Now let us consider values of Tau for individual players. Tau is between −1 and 1. Tau = 1 indicates a perfect ranking, and Tau = −1 indicates a ranking which is "as wrong as possible'. Kendall's Tau coefficients of the players were in the large interval between −0.18 and 0.95 (two payers actually ordered more of the relevant pairs of positions incorrectly than correctly). It is interesting to consider the 'average ranking' by all 12 players. This can be obtained by a kind of players' voting, considering the average rank of each problem over all the players' rankings. The obtained ranking order of positions was: 2, 3, 1, 6, 10, 7, 4, 5, 9, 8, 12, 11; that is, overall, position 2 was perceived as the easiest, followed by position 3, etc., with position 11 perceived as the hardest. According to Chess Tempo ratings, the sets of positions belonging to the three difficulty classes are as follows:

```
easy: {1, 2}
medium: {3, 4, 5, 6}
hard: {7, 8, 9, 10, 11, 12}
```

Kendall's Tau for the joint ranking by the players is 0.77. This can be compared with the individual players' Tau coefficients. The highest player's Tau was 0.95, and the second

highest 0.68. Average Tau over the 12 players was 0.45. This is also in agreement with the result that overall, the players correctly ordered 72.5% of pairs of problems that were taken into account.

The results regarding the players' difficulty estimation require careful interpretation. The task of the players in the experiment was stated simply as follows: rank the 12 given problems according to their difficulty, from the easiest to the most difficult. The players were not told that there were essentially three difficulty classes, and that there were many pairs of positions of practically equal difficulty. Given this circumstance, the following cases were possible regarding players' rankings. Consider a pair of positions A and B. If position A was easier than B according to Chess Tempo then: if the player ranked A before B then this counted as a concordant pair, otherwise this counted as a discordant pair (incorrect order). If A and B belonged to the same Chess Tempo difficulty class then this pair was not included in the calculation of Tau, so it did not matter whether the player ordered the problems A before B or B before A. Both cases were treated as acceptable, and did not affect the player's evaluated ranking performance. This is reasonable because in this case there is no evidence of ranking error. However, in such cases we do not actually know. Suppose that the player ranked A before B. In this case, there are two possibilities: (1) the player actually considered both problems to be equally difficult, and arbitrarily ordered A before B (just because a total ordering was required, and he had to order them one way or the other); or (2) the player actually believed that A was easier than B; in this case he was wrong, but there is no way to detect this from experimental data. Our modified Tau measure can therefore be interpreted as a potentially optimistic assessment of the player's ranking accuracy.

24.3.3 Experiment in automated prediction of difficulty

In this section we carry out an experiment, using our 12 experimental positions, with a program for automatically estimating the difficulty of tactical chess positions. We used the approach to estimating difficulty proposed in (Stoiljkovikj *et al.*, 2015), which will be referred to as the SBG method. This method is based on machine learning about difficulty for humans, using features of search trees that are searched by good human players when solving a tactical chess problem.

The size of the combinatorial search space involved in solving the problem is the most obvious source of difficulty. For an uninformed problem-solver without problem-specific knowledge, the size of the search space would indeed be a useful indicator of difficulty. For experienced chess players the situation is quite different. Such players employ their knowledge to search this space very selectively so that only a small fraction of the entire space is actually searched.

When solving tactical chess problems, human players use a repertoire of common motifs that allow such a highly selective and effective search. Figure 24.1 illustrates this. In this example, the solution consists in spotting the well-known motif of a pinned piece. Brute force search, realized as, say, iterative deepening to depth 5 (which would be an adequate depth in this example) would in this implementation require searching millions of positions. Using the chess-specific motif of a pin, this is reduced to the order of

Figure 24.1 *White to move and win. An experienced player will immediately notice the motif that Black king and Black knight on e4 are on the same file. This gives rise to the motif of pinning Black knight with the move 1.Re1 (green arrow). Knight on e4 is now attacked and cannot escape due to the pin. Black may try to defend knight e4 with moving the other knight: 1...Na4-c5. Now a common mechanism of exploiting a pin is used by White: attack the pinned piece with yet another piece. In this case this can be accomplished by White pawn move to f3. On the next move, Black knight on e4 will be captured, giving White a decisive advantage.*

10 or 20 positions. It is this latter number that is indeed relevant for the difficulty of the position for good players.

We will be referring to such a reduced search space as 'meaningful search tree'. It should be noted that all the players in our experiment easily had enough knowledge to solve the position of Figure 24.1 quickly, exploring a small meaningful tree, as explained in the caption of Figure 24.1. In this position, there is another common motif for White: double attack with White rook move to b4, simultaneously attacking both Black knights. However, a trivial search shows that in this position Black knights can defend each other with the move Ne4 c5, so double attack motif does not work in this case.

To estimate the difficulty for an expert human player, the estimation program would ideally simulate the search actually performed by the player and predict the difficulty based on this simulated search. To simulate such a search, the program would have to possess similar chess knowledge as the player. However, this knowledge consists of a very large library of chess motifs, or patterns, of the kind illustrated in Figure 24.1. Some of this knowledge is acquired by players through explicit instruction, and that part can be found in chess books. The larger part of that pattern-based knowledge is however tacit knowledge that a player has acquired through experience, but does not exist in formalized and documented form. The difficulty in predicting the difficulty for experts lies in the question: how to take into account such tacit knowledge?

The main idea of the SBG method is the concept of a 'meaningful search tree' (defined later). This is based on the assumption that the search of the human chess expert can be

simulated by a standard chess-playing program without knowing the extensive pattern knowledge. Hopefully, for a given chess position, the meaningful tree approximates the tree actually searched by a chess expert when searching for the best move in a position. Accordingly, a meaningful tree is formally defined with this aim in mind. For a given position P, the meaningful tree is a subtree of the game tree rooted in P. Suppose that a player is given position P and is asked to find a winning move in P. The player will try to solve the problem by economical search, so he or she will only investigate moves that come into consideration and discard other moves. The player's pattern knowledge and detected motifs will help the player to identify promising moves. The idea is to use a standard chess engine like Stockfish to carry out a relatively shallow search (e.g., 10 ply) and evaluate the positions in the corresponding game tree by backing-up heuristic values of the positions in the leaves of this search tree. These backed-up heuristic evaluations are hopefully indicative of what an expert player can (approximately) evaluate without search, just by using his or her pattern knowledge. The meaningful tree is the game tree up to a chosen depth limit (in our experiments set to 5 ply), with 'unpromising' moves removed from the tree (unpromising from the player's point of view, or from the opponent's point of view, depending on whose move it is). Formally, for the task of winning in P, the meaningful tree consists of the root position P, and all the player-to-move positions whose backed-up heuristic value exceeds w ('winning threshold'), and the opponent-to-move positions whose value differs from the value of the best sibling (from the opponent's point of view) by no more than m ('margin'). These parameters were set to $w = 200$ centipawns, $m = 50$ centipawns in our experiment.

This design can be debated in the light of the question: how well do so defined meaningful trees approximate trees that are actually searched by chess players? Another question can be: the SBG approach is mainly concerned with the 'meaningful complexity', and ignores some other sources of difficulty discussed in the next section, such as 'invisible moves' (Neiman and Afek, 2011). Another contentious issue could be: is it appropriate to assume that all the players (at least of the chess strength comparable to our group of players) search more or less the same search tree? Or does this depend on certain players and their chess knowledge, especially on their specific repertoire of chess motifs? The classical study by De Groot (1965) on human problem-solving in chess suggests that players solve chess problems in a similar way over large ranges of chess rating (such as 400 rating points, as in the case of our 12 players). Experiments in a related study (Gobet, 1998) also generally confirm this. The following is a relevant result concerning this latter question. A quantitative model of chess problem-solving of tactical problems as a Bayesian network was proposed in (Bratko *et al.*, 2016). The network is structured according to the classical chess problem-solving model in (De Groot, 1965). Standard sets of chess motifs required to solve the 12 experimental positions were defined and were needed to solve each problem. In most positions, more motifs than one are relevant. Also, relevant chess moves to be searched by players that corresponded to the positions' motifs were defined. It was possible to observe success of the players at detecting relevant motifs, and also at carrying out the calculations. The players successfully detected relevant motifs in 88% of all the cases (Bratko *et al.*, 2016). Here we add how this percentage depends on the players' ratings. This percentage

was somewhat higher for the top-half ranked players (92%, average rating 2198), and somewhat lower for the bottom-half ranked players (84%, average rating 1980). In spite of this difference, this indicates that a large majority of our players were able to detect relevant motifs correctly. The fact that the large majority of players detected relevant motifs (standard sets for the experimental positions) supports the assumption that, at least roughly, the players searched similar trees.

Some properties of a meaningful tree are naturally indicative of difficulty. For example, the total number of nodes in a meaningful tree or the branching factors at different levels of the tree. A more sophisticated indication of difficulty, is the attribute of a tree denoted by *NarrowSolution(L)*. This is defined as the number of opponent's moves at level L in the tree for which the winning player has only one good reply. A high value of *NarrowSolution* indicates situations where the opponent has many promising moves, and each of them requires to be met by the player with a unique reply.

In Stoiljkovikj *et al.* (2015), 10 attributes of a meaningful tree of this kind were defined. Another 10, chess-specific attributes of a position were defined, such as the number of chess pieces in the position or the existence of 'long moves' in the meaningful tree. Long moves are moves in which a chess piece moves by a long distance on the board, and sometimes such moves are suspected of being harder to notice by chess players, so they are one kind of 'invisible moves'. They contribute to the difficulty. Definitions of all the attributes can be found in (Stoiljkovikj *et al.*, 2015).

These 20 attributes of a position define a space for machine learning, and the problem of learning to predict the difficulty of chess positions can be formulated as follows. The learning data consists of a set of chess positions together with their difficulty class, where each position is described by the 20 attributes.

An experiment with learning to predict problem difficulty using this setting was carried out by (Stoiljkovikj *et al.*, 2015). Nine-hundred chess problems from Chess Tempo were randomly selected for learning. The difficulty class (easy, medium, or hard) was determined according to the Chess Tempo ratings of the problems, resulting in a balanced learning set with 300 examples of each class. In that experiment, the average Chess Tempo ratings of problems in the three learning subsets belonging to the three classes were as follows: easy: 1254.6, medium: 1669.3, hard: 2088.8. The reported classification results were very high (up to 83%, depending on the learning method used). However, these results cannot be trusted due to a suspected methodological slippage, which became apparent later when these experimental results could not be completely reproduced. In this chapter we repeat the learning experiment with the same set of learning problems (not including our 12 experimental positions), and the same positions' attribute values. However, to make the trained classifiers applicable to the 12 experimental problems of this chapter, we redefined the difficulty classes in the learning data, so that the new classes are appropriate for the three difficulty classes in our 12 positions. To this end we moved the thresholds for class separation to the midpoints between the Chess Tempo ratings of the three classes in the present experimental set, as given in Table 24.1.

After this redefinition of the thresholds between classes, the class distribution became imbalanced (which is less favourable for learning), as follows. Class easy: 479 examples,

Table 24.1 *The difficulty classes were determined according to the Chess Tempo ratings.*

1	rating $<$ 1685	easy
2	1685 $\leq$ rating $<$ 2055	medium
3	2055 $\leq$ rating	hard

medium: 231 examples, hard: 190 examples. We used several learning methods implemented in the scikit-learn machine learning library.

The best classification accuracy was obtained with Gradient Boosting Trees learning method (60%, measured by 10-fold cross-validation). We will refer to this predictor of difficulty as SBG2020. We applied this classifier to our 12 experimental problems. The results are given in Table 24.1. For each position, the table also gives the position's ranks according to average players' rankings, and the number of players that successfully solved the position.

Here are some quick observations from the table. The actual success rates by our players do not correlate very well with CT classes. Success rates of 100% (solved by all 12 players, see the column Success in Table 24.1) for problems 3 and 6, both medium difficulty by Chess Tempo, are surprising. A closer look at position 3 gives a likely explanation for what happened with this position. There are several winning moves in position 3 which all counted as success in our study, while for an unclear reason Chess Tempo only accepted as correct one of these alternative solutions. A similar explanation is possible for position 6. A closer look at position 6 suggests that this position is in fact relatively easy. According to this, the predicted class 'easy' by the SBG2020 classifier seems to be more appropriate. There are other discrepancies: problem 4, and some problems in CT class hard. But for these we could not find any simple explanation other than chance.

The sample correlation coefficients between the variables in Table 'classes' were computed by representing the three classes easy, medium and hard with 1, 2, and 3 respectively. The correlations are as follows:

```
r( CT-class, Success) = -0.60 (P = 0.0383)
r( SBG2020-class, Success) = -0.79 (P = 0.0023)
r( PlayersRanking, Success) = -0.78 (P = 0.0029)
r( CT-class, SBG2020-class) =  0.79 (P = 0.0024)
```

Also of interest are relations between the perceived difficulty of the positions by the players, represented by the joint players' rankings (average ranking of the positions), and the measured difficulty (CT-class) and automatically estimated difficulty (SBG2020-class):

```
r( Rank-by-players, CT-class) = 0.74
r( Rank-by-players,  SBG2020-class) = 0.94
```

Table 24.2 *Basic description of the Chess Tempo problem set.*

Position	*CT class*	*SBG2020*	*Rank*	*Success*
1	easy	easy	3	11
2	easy	easy	1	12
3	medium	easy	2	12
4	medium	medium	7	4
5	medium	medium	8	5
6	medium	easy	4	12
7	hard	medium	6	3
8	hard	hard	10	7
9	hard	medium	9	8
10	hard	medium	5	9
11	hard	hard	12	4
12	hard	hard	11	4

This is surprising as it suggests that the difficulty, as perceived by the human players, in fact better correlates with the automatically predicted difficulty by the SBG2020 approach, than with the actually measured Chess Tempo difficulty. This can be however at least partially explained by the problems mentioned above with positions 3 and 6, whose solutions seem to have been treated too harshly in Chess Tempo. The average ranking of the 12 positions by the 12 chess players is, interestingly, completely consistent with the SBG2020 classification.

Finally, we can try to compare the appropriateness of SBG2020 classification with respect to Chess Tempo classification by using Kendall's Tau coefficient. This is useful for comparison of the individual players' rankings (earlier assessed by Kendall's Tau) with SBG2020 rankings. There is a difficulty in that players' rankings are complete orderings, whereas SBG2020 classes only define a partial ordering. There are many total orderings consistent with the SBG2020 partial ordering. Now imagine a human player whose perceived position difficulties were exactly as by SBG2020. When asked to produce a total ordering, as in our experiment, this player could answer with any of the total rankings consistent with SBG2020. Assuming that all these rankings are equally likely, the expected value Tau over all these rankings is 0.78. Over all consistent rankings, Tau is between 0.56 and 1, with standard deviation 0.106. This is practically equal to Kendall's Tau of the average players' ranking. Even if the SBG approach is based on a very crude approximation to human players' game-tree search, it does seem to capture well the difficulty of problems as perceived by humans.

24.4 More Subtle Sources of Difficulty

There are some other sources of difficulty in chess problems, in addition to the size and other properties of meaningful trees, which were used in the SBG method in the previous section. In this section we point out other sources of difficulty, illustrated by examples from our experimental positions. These sources of difficulty were not considered in the SBG method.

24.4.1 Invisible moves

Some moves are hard to see by good chess players. Neiman and Afek (2011) investigated the properties of chess moves that are difficult to find and anticipate. It is precisely good chess player's knowledge which is so successfully used to make search more selective, that prevents the player from seeing such moves and is occasionally the cause of bad mistakes. For example, novice players are taught from the beginning that chess pieces should be developed as quickly as possible, therefore they have to move forward and preferably towards the centre where they are generally the most powerful. This cliché makes the players more likely to consider forward moves and sometimes automatically disregard moves away from the centre. Thus some moves become more difficult to see simply for geometrical reasons. Bent Larsen even points out that backward moves on diagonals are particularly difficult to detect 'except on the long diagonal' (Larsen, 2014).

Of all possible backward moves, those of the knight are the most difficult to find (Neiman and Afek, 2011). There is also a technical reason for this: As a short-range piece, the knight in particular has to be centralized. It takes too long to bring it back into the critical areas once it is out of play. There was an example of this kind of invisible move in our experimental position no. 4 (Figure 24.2), which was only solved by four players. Although the players who failed to solve it were in fact considering the right idea (described in players' retrospections), they simply could not see the winning move by a knight into the corner of the board.

24.4.2 Seemingly good moves and the 'Einstellung' effect

Clearly, difficulty should not be confused with complexity. Sometimes a problem may seem easy because the position does not seem complex at all. There may be an attractive move that seems to lead to victory, but in reality it does not. It is the presence of such a 'seemingly good' move (Stoiljkovikj *et al.*, 2015) that diverts the players attention from a truly good move and thus makes the problem difficult. In our experimental position no. 7 (Figure 24.3), there is a seemingly good move: 1 . . . Qd8-b6. But the real solution requires the insertion of the move 1 . . . Bf8-h6 before pinning the knight. It would be much easier to spot the correct move sequence if the above mentioned move with the queen did not look so attractive. In fact, only three players correctly solved this problem.

Figure 24.2 *There are two main motifs for White here. The first is to attack Black king via open e file, and part of this idea is the pinned Black bishop at e7. An obvious move to exploit that is by move 1.Qe3, increasing the pressure on Be7. However, this does not work. Black can successfully defend with 1...Ne5, which can be determined by relatively complex calculation. Another, completely different motif is triggered by a complex pattern: Black queen is surrounded by many White pieces and does not have any safe square to move to. This gives rise to the idea of trapping Black queen. To this end, queen has to be attacked, and White knight on c2 can do that, in two ways. One way is to move to d4 (red arrow). This move however disables the control of square c4 by White rook's on e4. So Black queen can now escape to c4. Now White has another familiar powerful pattern at disposal: discovered attack on Black queen with move 2.Ne6, also attacking Black rook d8. All that looks very strong for White, but as it turns out not sufficient for a clear win. This was calculated by many players who did play Nd4 and eventually failed to solve this problem. Much more straightforward and effective is the invisible move 1.Na1 (green arrow), immediately winning Black queen, but not seen by many players.*

When faced with a decision in chess, people are sometimes misled by familiar patterns and motifs, so that they miss better solutions. When we solve problems, our prior knowledge usually helps us by efficiently leading us to solutions that have worked for us in the past. However, if a problem requires a new solution, it can sometimes be surprisingly difficult to find the new solution because of our prior knowledge. This problem-solving effect was discovered by a psychologist Abraham Luchins (1942). He called this effect the 'Einstellung' effect. Bilalic *et al.*, (2008) experimentally confirmed that the Einstellung effect also exists in chess. A familiar pattern in a chess position drew the attention of the players to find a familiar solution (which did not work) and prevented them from finding a real solution that could be linked to a completely different pattern.

Figure 24.3 *Black to move wins. It is trivial for a good player to immediately notice the possibility of pinning White knight on d4 against White king with Qb6 (red arrow). The seemingly straightforward variation is thus 1... Qb6 2.Rfd1 Bh6 (another common method: attack the piece defending the pinned knight on d4) 3.Qd3 Nxd4 4.Qxd4 Be3+ winning White queen. This looks excellent for Black, but fails to notice that instead of 3.Qd3 White can unexpectedly strike back with 3.Nd5. After that it is no longer clear whether Black can win. The clear winning line for Black is 1...Bh6 (green arrow) 2.Qd3 Qb6 and now this indeed wins.*

24.5 Conclusions

This is a summary of the results of the analysis of our experimental data in expert problem-solving in chess. The results apply to solving tactical chess problems with Chess Tempo ratings roughly between 1500 and 2300, and players with FIDE ratings between 1800 and 2300:

1. A negative correlation was found between player's success in solving problems and the Chess Tempo rating of the problems, which is as expected.
2. There was no evidence of a correlation between the success of the players and the FIDE rating of the players. This is surprising. A plausible explanation is that success refers to finding a winning move in one position, whereas the FIDE rating measures success over entire games; that is, over a sequence of positions. Winning a game often means making a better decision than your opponent in only one or two positions in the whole game.
3. There is a statistically significant positive correlation between the players' ratings and the correctness of the ranking by the players of the 'relevant' position pairs according to their difficulty, although this relationship is quite weak.

We carried out an experiment in which the difficulty of the 12 experimental positions was automatically estimated using the SBG method. The main idea of the SBG is to use the properties of a 'meaningful' search tree as attributes for learning to estimate the difficulty of example positions, which are divided into difficulty classes. A meaningful search tree is defined as an attempt to automatically construct approximations to trees searched by human experts without knowing human expertise. The learned classifier was applied to our experimental positions, and the resulting classification of the positions compared well with the Chess Tempo difficulty classes, and also to the average perceived difficulty by the players. A question for future work is to explore why the SBG has done surprisingly well, even though it is based on a rather crude approximation to problem-solving by human experts.

Acknowledgements

This work was in part supported by the Research Agency of Republic of Slovenia (ARRS), research program Artificial Intelligence and Intelligent Systems. The authors would like to thank Peter Cheng for pointing out relevant research, and anonymous reviewers for their comments and suggestions.

References

Bilalić, M., McLeod, P., and Gobet, F. (2008). Why good thoughts block better ones: The mechanism of the pernicious einstellung (set) effect. *Cognition*, **108**(3), 652–61.

Bratko, I., Hristova, D., and Guid, M. (2016). Search versus knowledge in human problem solving: a case study in chess, in *Model-Based Reasoning in Science and Technology*. Berlin: Springer, 569–83.

De Groot, A. D. (1965). *Thought and Choice in Chess*. The Hague: Mouton.

De Kegel, B. and Haahr, M. (2019). Procedural puzzle generation: A survey. *IEEE Transactions on Games*, **12**(1), 21–40.

Dry, M., Lee, M. D., Vickers, D. (2006). Human performance on visually presented traveling salesperson problems with varying numbers of nodes. *Journal of Problem Solving*, **1**(1), 20–32.

Elo, A. E. (1978). *The Rating of Chessplayers, Past and Present*. London: Arco Publications.

Glickman, M. E. (1999). Parameter estimation in large dynamic paired comparison experiments. *Applied Statistics*, **48**, 377–94.

Gobet, F. (1998). Chess players' thinking revisited. *Swiss Journal of Psychology*, **57**, 18–32.

Guid, M. and Bratko, I. (2006). Computer analysis of world chess champions. *ICGA Journal*, **29**(2), 65–73.

Guid, M. and Bratko, I. (2013). Search-based estimation of problem difficulty for humans, in H. Lane, K. Yacef, J. Mostow, *et al.*, eds, *Artificial Intelligence in Education*, Vol. 7926, Lecture Notes in Computer Science. Berlin: Springer, 860–3.

Hristova, D., Guid, M., and Bratko, I. (2014*a*). Assessing the difficulty of chess tactical problems. *International Journal on Advances in Intelligent Systems*, 7(3&4), 728–38.

Hristova, D., Guid, M., and Bratko, I. (2014*b*). Toward modeling task difficulty: the case of chess, in *Proceedings of the Sixth International Conference on Advanced Cognitive Technologies*

and Applications, Venice, Italy. Wilmington: International Academy Research and Industry Association (IARIA), 211–4.

Jarušek, P. and Pelánek, R. (2010). Difficulty rating of sokoban puzzle, in T. Agotnes, ed., *Proceedings of the Fifth Starting AI Researchers' Symposium (STAIRS 2010)*, Lisbon, Portugal. Amsterdam, Netherlands: IOS Press, 140–50.

Kotovsky, K., Hayes, J. R., and Simon, H. A. (1985). Why are some problems hard? Evidence from tower of Hanoi. *Cognitive Psychology*, **17**(2), 248–94.

Kotovsky, K., and Simon, H. A. (1990). What makes some problems really hard: Explorations in the problem space of difficulty. *Cognitive Psychology*, **22**(2), 143–83.

Larsen, Bent (2014). *Bent Larsen's Best Games: Fighting Chess with the Great Dane*. New in Chess.

Luchins, A. S. (1942). Mechanization in problem solving: the effect of Einstellung. *Psychological Monographs*, **54**(6), i.

Neiman, E. and Afek, Y. (2011). *Invisible Chess Moves: Discover Your Blind Spots and Stop Overlooking Simple Wins*. Amsterdam, Netherlands: New in Chess.

Pelánek, R. (2011). Difficulty rating of sudoku puzzles by a computational model, in R. Murray and P. McCarthy, eds, *Proceedings of Florida Artificial Intelligence Research Society Conference*, Palm Beach. New York, NY: AAAI Press, 434–9.

Pizlo, Z. and Li, Z. (2005). Solving combinatorial problems: the 15-puzzle. *Memory and Cognition*, **33**(6), 1069–84.

Sekiya, R., Oyama, S. and Kurihara, M. (2019). User-adaptive preparation of mathematical puzzles using item response theory and deep learning, in F. Wotawa, G. Friedrich, I. Pill, R. Koitz-Hristov, and M. Ali, eds, *Proceedings International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, Graz, Austria. Cham: Springer, 530–7.

Sheridan, Heather and Reingold, Eyal M. (2017). Chess players' eye movements reveal rapid recognition of complex visual patterns: Evidence from a chess-related visual search task. *Journal of vision*, **17**(3), 4–4.

Stoiljkovikj, S., Bratko, I., and Guid, M. (2015). A computational model for estimating the difficulty of chess problems, in A. Goel and M. Riedl, eds, *Proceedings of the Third Annual Conference on Advances in Cognitive Systems*, Atlanta, Georgia. Auckland: Cognitive Systems Foundation, 7.

Van Kreveld, M., Löffler, M., and Mutser, P. (2015). Automated puzzle difficulty estimation, in C. Lee, I. Wu, M. Wang, eds, *2015 IEEE Conference on Computational Intelligence and Games (CIG 2015)*, Tainan, Taiwan. New York, NY: IEEE Press, 415–22.