Chess Game Analysis Tool Documentation

Donato Martinelli

March 14, 2025

1 Overview

This document explains a Python program that performs post-game analysis for chess matches stored in PGN files using the Stockfish engine. The program evaluates the game move-by-move, prints an evaluation table, plots the evaluation trend, and analyzes move errors to compute overall accuracy.

2 Program Structure

2.1 Engine Initialization

The program initializes a UCI-compatible chess engine (e.g., Stockfish) using a specified path:

```python
engine_path = "path/to/engine/executable"
engine = chess.engine.SimpleEngine.popen_uci(engine_path)
```

2.2 Loading a PGN Game

The function `load_game_from_pgn` opens a PGN file and returns the first game:

```python
def load_game_from_pgn(pgn_path):
    with open(pgn_path) as pgn_file:
        game = chess.pgn.read_game(pgn_file)
    return game
```

2.3 User Input for Color

The program asks the user whether they played as white or black, and sets the evaluation perspective accordingly:

```python
color_input = input("Did you play as white or black? ('w' or 'b'):
    ") or "w"
if color_input.lower() == 'w':
    my_color = chess.WHITE
elif color_input.lower() == 'b':
    my_color = chess.BLACK
else:
    print("Invalid input, using white as default.")
    my_color = chess.WHITE
```

2.4 Evaluation Functions

The program includes several helper functions:

- `get_evaluation_str`: Converts an engine evaluation (PovScore) to a formatted string.
- `average_evaluation`: Computes the average evaluation for a board position over multiple analyses.
- `format_eval_value`: Formats a centipawn score into a human-readable string.

2.5 Game Analysis and Visualization

The function `analyze_game_table_and_plot` processes the game move-by-move:

- It calculates average evaluations for white and black moves.
- It prints a table displaying move numbers, move notations, and their evaluations.
- It collects evaluation data to plot a graph of the evaluation trend over the game.

2.6 Move Error Analysis

The function `analyze_move_errors` compares each move played by the user to the engine's best move:

- It runs multiple analyses for each move to compute an average error.
- Moves with an error exceeding a defined threshold are printed.
- Overall accuracy (percentage of moves with error below the threshold) is calculated and displayed.

2.7 Analysis Mode Selection

The function `choose_analysis_mode` allows the user to select from different analysis modes:

- Fast: 2 seconds per move, 2 evaluations.
- In-depth: 5 seconds per move, 3 evaluations.
- Very In-depth: Fixed search depth (20 plies), 3 evaluations.
- Very Fast: 0.5 seconds per move, 2 evaluations.

3 Main Execution Flow

The program executes the following steps:

1. Load the game from the PGN file.
2. Prompt the user for their played color.
3. Let the user select an analysis mode.
4. Analyze the game, printing the evaluation table and plotting the evaluation trend.
5. Reload the game and perform move error analysis.
6. Shut down the engine after analysis.

4 Usage

To run the program:

1. Update the `engine_path` and `pgn_path` variables in the script with your own file paths.
2. Run the script from a terminal or command prompt:

   ```
   python chess_analysis.py

   ```

3. Follow the on-screen prompts to specify your played color and choose the analysis mode.

5 Example Usage and Output

After running the script, you will be prompted to select your color (`w` or `b`) and the analysis mode. Below is an example of the console output when the user chooses to play as white (`w`) and selects analysis mode `4` (Very Fast):

```
Did you play as white or black? ('w' or 'b'): w
Choose the analysis mode:
1. Fast (2 sec, 2 evaluations)
2. In-depth (5 sec, 3 evaluations)
3. Very In-depth (depth=20, 3 evaluations)
4. Very Fast (0.5 sec, 2 evaluations)
Enter 1, 2, 3 or 4: 4
Evaluation table and graph:
1. | d4 | +25 | d5 | +29 |
2. | Bf4 | +24 | c6 | +35 |
3. | e3 | +39 | Bf5 | +36 |
4. | Bd3 | +14 | Bxd3 | +15 |
5. | Qxd3 | +19 | Nd7 | +16 |
6. | Ne2 | +15 | Ngf6 | +13 |
```

```
7. | f3 | -16 | Qb6 | +20 |
8. | b3 | -19 | e6 | -22 |
9. | c3 | -30 | a5 | -15 |
10. | Nd2 | -10 | h6 | -5 |
11. | c4 | -38 | Qa6 | -18 |
12. | Rc1 | -65 | c5 | +7 |
13. | g4 | -61 | g5 | +52 |
14. | Bg3 | +53 | h5 | +78 |
15. | h3 | -81 | h4 | +112 |
16. | Bh2 | +120 | cxd4 | +124 |
17. | Nxd4 | +45 | Bd6 | +42 |
18. | Bxd6 | -11 | Qxd6 | -21 |
19. | Nb5 | -111 | Qg3+ | -138 |
20. | Kd1 | -165 | O-O | -161 |
21. | Kc2 | -204 | Ne5 | -68 |
22. | Qc3 | -115 | Kg7 | +400 |
23. | Rcg1 | +417 | Qf2 | +452 |
24. | Qxe5 | +443 | dxc4 | +472 |
25. | Qxg5+ | +405 | Kh8 | +584 |
26. | Qxf6+ | +601 | Kh7 | +638 |
27. | bxc4 | +653 | Rad8 | +640 |
28. | Nd4 | +481 | Qxe3 | +442 |
29. | Qxh4+ | +361 | Kg8 | +393 |
30. | N2b3 | +323 | a4 | +363 |
31. | Re1 | +388 | Qf4 | +472 |
32. | Re4 | +505 | axb3+ | +496 |
33. | axb3 | +571 | Qd6 | +654 |
34. | Qg5+ | +666 | Kh8 | +780 |
35. | Qh6+ | +704 | Kg8 | +709 |
36. | g5 | +619 | Qe7 | +9995 |
37. | g6 | +730 | f5 | +785 |
38. | Rh4 | +797 | Qg7 | +799 |
39. | Qh7+ | +694 | Qxh7 | +709 |
40. | gxh7+ | +709 | Kh8 | +869 |
41. | Rg1 | +566 | f4 | +932 |
42. | Nxe6 | +974 |  |  |

Move errors analysis:
13. g4 -> Nc3  avg. diff: 91
15. h3 -> Nc3  avg. diff: 163
17. Nxd4 -> exd4  avg. diff: 81
18. Bxd6 -> f4  avg. diff: 51
19. Nb5 -> Ne2  avg. diff: 100
21. Kc2 -> Qe2  avg. diff: 50
22. Qc3 -> Qe2  avg. diff: 67
28. Nd4 -> Qc3  avg. diff: 234
29. Qxh4+ -> Re1  avg. diff: 72
35. Qh6+ -> Qf6+  avg. diff: 229
37. g6 -> Rh4  avg. diff: 9307
39. Qh7+ -> Qxg7+  avg. diff: 93
41. Rg1 -> Nxe6  avg. diff: 203
Overall Accuracy: 69.05%
```

The script also produces a plot illustrating the evaluation trend throughout the game, similar to Figure 1 below.

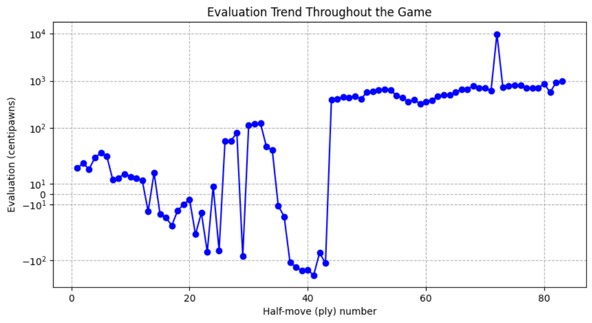

Figure 1: Example Evaluation Trend Graph

6 Conclusion

This tool is designed for chess enthusiasts to review and improve their performance by comparing their moves against engine recommendations. It provides both a move-by-move evaluation and an overall accuracy metric, along with a graphical representation of the evaluation trend throughout the game.