Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 126 (2018) 831–840

Procedia Computer Science

www.elsevier.com/locate/procedia

22nd International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

Data analytics on the board game Go for the discovery of interesting sequences of moves in joseki

Carson K. Leung[a,*], Felix Kanke[a], Alfredo Cuzzocrea[b]

[a] *University of Manitoba, Winnipeg, MB, R3T 2N2, Canada*
[b] *University of Trieste, 34127 Triests (TS), Italy*

Abstract

In the current era of big data, high volumes of a wide variety of data of different veracity are generated at a high velocity in many real-life applications. Embedded in these big data is valuable information or knowledge. This calls for data science solution for discovering knowledge from the big data. A rich source of big data is game data. In this article, we focus on the board game of Go, which is a popular two-player strategic board game. Due to its popularity, many people are studying sequences of moves in games (i.e., joseki). However, with high volumes of the game data, manual solution or complex automatic solution for joseki may not be practical. Hence, in this article, we present a simple automatic data science solution for discovering interesting sequences of moves in joseki for the board game Go. Evaluation results show the benefits and practicality of using our solution in data analytics of the game.

Keywords: Data science; data mining; Go; joseki; board game; game mining; prefix tree; pruning

1. Introduction

Nowadays, we are living in the era of big data. Data are everywhere because huge volumes of a wide variety of valuable data of different veracities can be easily collected or generated at a high velocity in various real-life

* Corresponding author.
E-mail address: kleung@cs.umanitoba.ca

10.1016/j.procs.2018.08.017

applications. Embedded in these big data is useful information and knowledge. Hence, attention has been drawn to study and mine these big data. Rich sources of big data including the following:

- social media and social networks [1-7];
- wireless communication networks [8]
- bioinformatics and biomedicine applications, which may generate DNA sequences [9, 10]; and
- games, which include (i) sports games such as football games [11], (ii) card games [12], (iii) online video games [13, 14], and (iii) board games such as chess [15] and Go [16-19].

Go [20] is an old but popular two-player strategic board game. Although it is most commonly played in East Asian countries (e.g., China, Japan, and Korea), its popularity has grown in Europe and North America over the past few decades. It is played by millions of amateurs and thousands of professionals worldwide, and these players are thus constantly trying to improve their own play in a variety of ways (e.g., study joseki) [16-19] which go beyond simply playing the game.

Joseki [21] are studied standard sequences of moves, primarily played at the beginning of the game around the corners of the board, which have been evaluated by a consensus of professionals as locally optimal exchanges. It is important to note that these sequences are only joseki if they are *local* sequences. Global moves (i.e., moves that are made to counter multiple positions on the board at once) are not part of a joseki. The opening of the game is considered the hardest part of the game, both in theory and in practice. In professional championship games that last two days, the first day is usually spent on the first 50 moves and the second day finishing the rest, where games usually last around 200 moves depending on if the game is played out to the end or ends in resignation. Considering the large size of the 19 x 19 board and the simple ruleset as described in Section 2, the number of possibilities in any opening position are so vast that even professional players have to use instinct and feelings to guide them rather than objective analysis [22, 23]. Studying joseki is an excellent way to improve play in a more structured way than using instinct [24]. Existing studies usually involve either of the following:

- manual production of joseki dictionaries, which production may not be practical due to high volumes of the game data that can be collected at a high velocity; or
- automatic solutions, which may use complex artificial intelligence (AI) based or machine learning (ML) based programs (e.g., AlphaGo[†]—developed by Google's DeepMind—as first computer Go program to beat a human professional Go player without handicaps) and thus may not be easily accessible by most users.

Consequently, more practical and accessible solutions are needed. In this article, we design and develop a simple data science solution for studying joseki. Specifically, *key contributions of this article* is our solution that uses data mining methods, similarity measures based on Chebyshev distance, prefix trees, and pruning techniques, all of which avoid complex ML algorithms (e.g., artificial neural networks, deep learning) or programs (e.g., AlphaGo). As a preview, our solution mines a dataset of 89,587 games obtained from the summer 2017 edition of the Games of Go on Download (GoGoD)[‡]. These games were played by professional players dating back to the year 1800. Our data science solution aims for the following:

- finding as many standard joseki as we can, sorted by their frequency; and
- discovering how professional play has changed or improved over the centuries.

The remainder of this article is organized as follows: The next section provides some background information about the board game Go. Section 3 discusses related works. We then describe in Section 4 our data science solution for data analytics of Go. Evaluation and conclusions are given in Sections 5 and 6, respectively.

2. Background

In Go, two players take turns playing either black or white pieces (i.e., "stones") on the intersections of the board. The goal is to surround sections of the board with your own stones. At the end of the game, the player that has the most territory (i.e., playable points within his own border) wins. The edges of the board count as borders for both

[†] https://deepmind.com/research/alphago/
[‡] https://gogodonline.co.uk/

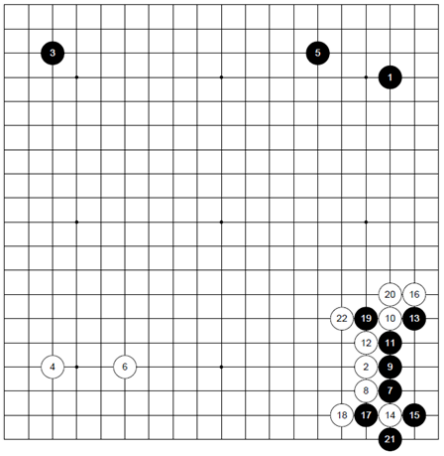


Fig. 1 The first 22 moves of a game between KE Jie and AlphaGo.

players. As such, in the beginning of the game, it is standard for a player to start placing stones around the corners of the board because it is easiest to enclose an area with the two edges already helping his border. Play generally spreads out to the sides of the boards because there is still one side of the board helping. Finally, games move into the center where all four sides of territory have to be created of the player's own stones. Of course, the opponent will try to prevent the player from creating territory by trying to interrupt the player as the player builds while trying to build something for himself. Many of these encounters result in local exchanges for both sides. These sequences can become joseki if they are played out on the professional level.

Fig. 1 shows the first 22 moves of a game between KE Jie (the top player in the world at the time of the game) and AlphaGo. The figure illustrates that the four corners being played first, followed by a joseki being played out in the bottom right. Although it is not necessary for these many consecutive stones to be played in one corner, it is common to play out parts of a joseki and return to them once other areas of the board have become more developed. Professionals have to assess (i) each section of the board and (ii) how does it relate to other sections. Joseki themselves are local phenomena primarily restricted to the corner. Players have to be able to evaluate which sequences are most appropriate with regard to the whole board.

3. Related works

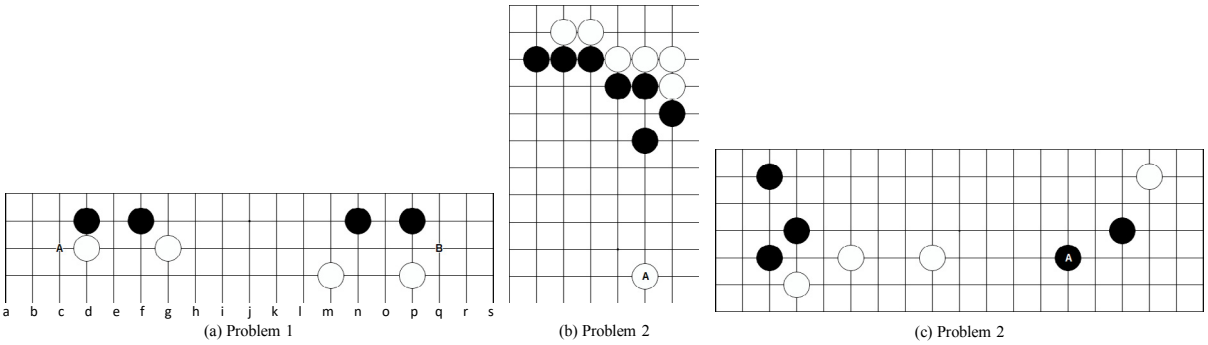
There is a general lack of data mining research with respect to the study of joseki. However, many game database-searching tools, electronic dictionaries, physical books and encyclopedias exist that list common joseki. The joseki in books are selected manually, primarily based on the experience of professional players that write these books.

3.1. Application of data mining to the study of joseki

Helvensteijn [21] applied data mining techniques to find and categorize common joseki. His algorithm consists of two phases. In the first phase, he traversed 13,325 games in a loop that iterates over every move in the games. He compared each move to all other moves within a game, and assigned them to sequences based on their Manhattan distance. Note that the *Manhattan distance* between two points (x_1, y_1) and (x_2, y_2) is the summed difference of their x- and y-coordinates:

$$\text{Manhattan distance} = |x_1 - x_2| + |y_1 - y_2|$$

Any move within a maximum distance of 5 to one or more moves in an existing sequence was considered part of those sequences and was added to the list. It is unclear whether more than four sequences (one per corner) were allowed. The algorithm stopped adding stones to sequences after at least 20 stones have been played in each corner. Then, these sequences are added to a prefix tree where (i) the root node represents the empty board and (ii) each node represents



(a) Problem 1 (b) Problem 2 (c) Problem 2

Fig. 2 Illustrations of potential problems associated with Helvensteijn's application of data mining to the study of joseki.

the next move in a given sequence. Each node contains the current move, a list of continuations, and the frequency of the move.

In the second phase, Helvensteijn pruned the tree and printed the final tree as a file in the smart game format (SGF), which is a standard for electronic recordings of Go games. A sequence is pruned based on the frequency of a sequence (say, a minimum frequency of 150 was the requirement to keep a sequence).

Observed from these two phases, four potential limitations were identified. Three of them are associated with the first phase and the fourth problem is associated with the second phase. These four problems are as follow:

1. The use of the Manhattan distance;
2. The use of a single measure for the distance to consider a stone as part of a sequence;
3. The addition of the same move to multiple sequences.
4. The requirement of high support for a sequence to be considered as significant (i.e., not to be pruned).

To elaborate, a potential issue with Problem 1 of using the Manhattan distance as the closeness measure is that diagonal plays are considered as significantly further away than horizontal or vertical plays. For example, in Fig. 2(a), the pair of stones in column d is directly connected and thus incurs a Manhattan distance of 1, whereas the pair of stones in columns f & g incurs a Manhattan distance of 2. Both moves are *direct threats* to the sequences, but the latter is considered to be twice as far apart in Manhattan distance and thus not parts of a sequence. Similarly, the pairs of stones in columns n & o (i.e., a "knight's move" or "*keima*") is considered further away than the pair of stones in column q (i.e., a "one-point jump"), despite that both moves are equally threatening with consideration to other stones in the area.

A potential issue with Problem 2 of using a single measure for the distance to consider a stone as part of a sequence is that adding a stone to sequence with the fixed distance of 5 works with the first few moves of a sequence. However, once a sequence becomes more defined, playing 5 steps away are certainly not part of a joseki anymore because there will be decisions made based on the whole board but not local positions. For example, in Fig. 2(b), the white stone marked A is not considered as a part of the settled joseki at the top. Furthermore, as stones stretch across the board, a stone becomes more likely to be within range of multiple sequences. These moves will again not be made as part of a joseki but rather a play against multiple positions. For example, in Fig. 2(c), the black stone marked A is not considered as a part of the joseki of the left.

A potential issue with Problem 3 of adding stones to multiple positions is that stones close enough to multiple joseki always have to take into account the state of both, and thus not local decisions and not joseki. These moves and their continuations will likely be pruned in the second phase because they are unlikely to be common. However, this means that, after such a move is added and subsequently removed from a sequence, all consecutive moves are lost. This can lead to sequences that are common being pruned away.

Moreover, a potential issue with Problem 4 of the high minimum frequency requirement is that (i) less common but well-known joseki, as well as (ii) common joseki with a high number of variations, will be pruned away. For example, a joseki with a frequency of 500 could have 4 standard continuations, each of which will have a frequency of less than 150. The minimum frequency of 150 (i.e., > 1% of the total games mined) is also relatively high. Go games are highly varied, and many sequences exist that occur regularly but still in less than 1% of games.

3.2. Joseki dictionaries and corner dictionary

In addition to Helvensteijn's application of data mining to the study of joseki, *Josekipedia* [25] is a commonly used online dictionary containing a large number of joseki which can be played interactively. Similarly, *Kogo's Joseki Dictionary* (*KJD*)[§] is a large SGF file containing many joseki [26]. In both joseki dictionaries, moves were added *manually*.

Besides joseki dictionaries, a popular SGF viewing and searching tool—called Kombilo—uses a program to search 60,000 pro games and 1.6 million games played by Amateur 4 Dan level and higher for common sequences. By searching for common sequences in an 11x11 search region at the lower right corner of the board, a *corner dictionary*[**] (i.e., an overview of common corner moves) was created. However, there are some potential problems associated with using this corner dictionaries for joseki study:

- Including amateur games and high-ranked games will lead to lower quality sequences being selected. This becomes a serious problem when the number of amateur games outnumbers those of professionals by almost two orders of magnitude. Note that, as studying joseki is mainly used for improving one's play, studying sub-optimal games would be a detriment.
- Searching only one corner misses approximately 75% of all joseki.
- Simply finding all sequences in an 11x11 grid is not the same as finding joseki because the former does not check if sequences are related at all. It simply checks if sequences are within the same grid area.

Daily Joseki[††] is another compilation of corner moves based on the collection of games from GoGoD. So, like the aforementioned corner dictionary, Daily Joseki does not look for joseki specifically. It searches for common corner moves without any similarity considerations.

Moreover, Waltheri's go pattern search [27] is a web application that allows users to search for positions of common joseki. It is built upon a database of over 70,000 games. It finds the most common moves played from given positions instead of specifically finding joseki. It also does not do a complete search of the entire database every time.

4. Our data science solution

Our data science solution is set up in three phases. In the first phase, our solution traverses through every game, reads each move, and adds the move to a joseki sequence if the move meets the user-specified criteria. Then, in the second phase the discovered joseki are added to a prefix tree structure, where each node contains the move, child sequences and frequency. In the third and last phase, the tree paths are pruned and sorted.

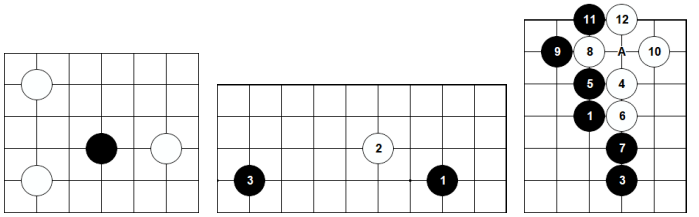
4.1. Phase 1: Joseki discovery

The first phase of our data science solution creates four potential joseki per game, one for each corner. After reading a file, our data science solution goes through every move sequentially, and adds it to one of the corner joseki if it is close enough. Recall from Section 3.1 about Problem 1 of using the Manhattan distance as the closeness measure in Helvensteijn's application of data mining to the study of joseki is that diagonal plays are considered as significantly further away than horizontal or vertical plays. Hence, in our data science solution, closeness or similarity is determined by using the Chebyshev distance. Note that the Chebyshev distance between two points (x_1, y_1) and (x_2, y_2) can be computed as follows:

$$\text{Chebyshev distance} = \max(|x_1 - x_2|, |y_1 - y_2|)$$

The Chebyshev distance appears to be a more accurate measure of closeness in terms of moves on a Go board. As only the maximum is used instead of combination of the x and y distances, diagonal moves are treated as equally distant as horizontal moves, see Fig. 3(a). This solves Problem 1.

[§] https://waterfire.us/joseki.htm
[**] https://www.u-go.net/2016/cornerdict/
[††] http://dailyjoseki.com/browse



(a) All white stones are at a Chebyshev distance of 2 from the black stone

(b) A common long pincer

(c) A potential "ko" sequence at position A. (2 is played at a different corner.)

Fig. 3 Examples of (a) equal Chebyshev distances for diagonal, horizontal & vertical plays; (b) long pincer; and (c) a potential "ko" sequence.

Recall from Section 3.1 about Problem 2 of using a single measure for the distance to consider a stone as part of a sequence is that adding a stone to sequence with the fixed distance of 5 may not always work due to the decisions made based on local positions than the whole board. The exact distance required to be considered as part of the joseki is (i) dynamic, (ii) depends on the number of moves already in the sequence, and (iii) other moves played on the board. When a joseki has fewer than six moves, moves that are within a distance of 5 are accepted. This allows for long pincers, which are not uncommon. One such example is shown in Fig. 3(b). After six moves have been added to a joseki, our data science solution reduces the maximum distance to 3. After 10 moves, it reduces the maximum distance to 2. These numbers were manually chosen after manually reviewing several games, including ones with uncommon openings and how our solution handled them. By gradually shortening the accepted distance, our solution reduces unrelated moves being added to joseki. This solves Problem 2, and prevents excessive pruning of branches.

Recall from Section 3.1 about Problem 3 of adding stones to multiple positions is that stones close enough to multiple joseki always have to take into account the whole board including non-joseki. Hence, in addition to the four joseki lists, our data science solution adds a fifth non-joseki list (which keeps track of moves that are not part of any joseki). Moves are added to this list if they meet one of the following criteria:

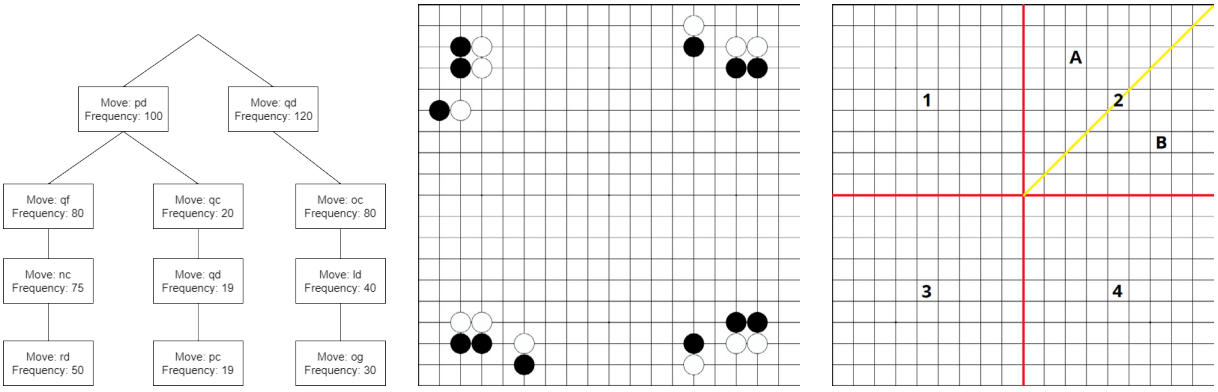
- A move is further than the accepted distance from all joseki.
- For simplicity, we limited the length of a joseki to 20. After a corner has more than 20 moves (which are considered as part of a joseki), it becomes "settled" and any further moves will be added to the non-joseki list even if they are within range.
- If a move is equidistant to multiple joseki, it cannot be considered as part of either because it will have to be played according to global conditions, not local ones specific to a single corner.

This solves Problem 3.

Beside the maximum length, a corner can also become settled if a "*ko*" (i.e., a situation where two alternating single stone captures would repeat the original board position) appears in the game. These alternating captures could repeat indefinitely, preventing the game from ending. An example is shown in Fig. 3(c). Since this is once again a matter that involves global decision making, the subsequent "*ko*"-fight is not considered part of a joseki, and the sequence ends once one is detected. Once all corners are settled or the non-joseki list gathers 50 moves, the game is considered to have reached midgame (i.e., middle part of the game) and no more local joseki moves will be played. At this stage, discovered joseki are added to a prefix tree, to which the color of a move is not added because the sequences of moves will be the same regardless of which of the two players initiates the joseki. Whenever two consecutive moves of the same color are added to a joseki, our solution first adds a "*tenuki*" (i.e., a situation where a player adds a "play-away" move in another part of the board instead of answering the opponent's last move locally). By doing so, identical joseki can be added to the same branch in a tree.

4.2. Phase 2: Creation of the prefix tree and SGF file

The second phase of our data science solution adds the joseki found in the first phase to a global tree structure. Each node in the tree contains (i) a move, (ii) its frequency, and (iii) a list of child moves. By doing so, each branch represents a possible variation. Duplicate sequences only increment the frequency; they do not create multiple branches. Hence, this is an efficient way to store sequences of moves. If a joseki starts off the same as a previous



(a) General layout of our prefix tree structure (b) Four identical joseki (c) Symmetry of a Go board

Fig. 4 Examples of (a) our prefix tree; (b) four identical joseki; (c) symmetry of a Go board.

sequence but diverges at some point, a new branch will only be created at that point of divergence. It is also important to note that, when using this method, the frequency of a parent node will always be greater than or equal to the sum of frequencies of their direct child nodes. Hence, an infrequent variation can be pruned away without the need to check any further children.

Fig. 4(a) shows a small example of a tree with 3 different variations. Moves are denoted by their SGF coordinates on the board, with letters a-s. Note that the frequency of each child node is at most that of the parent. To prune out sequences with a frequency less than 20, a top-down traversal can be performed to look for the highest node with a lower frequency and prune the entire branch without the loss of frequent sequences.

Recall from Section 4.1 that our tree structure does not store the color of the move played. Colors are added only when the final output SGF file is created.

In addition, another critical point to consider is the symmetry inherent to a Go board. Consider Fig. 4(b). Each of the four corners has an identical joseki played out. When these sequences are added to the tree, it is necessary to merge them into one branch. In addition to color being swapped, x- and y-coordinates will have to be mirrored into a single corner. There are two main lines of symmetry. Consider Fig. 4(c), in which every move in Quadrant 2 are to be mirrored. The first line of symmetry is shown by the red lines. A joseki in Quadrant 1 is mirrored horizontally, that in Quadrant 4 is mirrored vertically, and that in Quadrant 3 is mirrored both horizontally and vertically. However, another more difficult line of symmetry is shown by the yellow line in Quadrant 2. A first move played at (x, y)-coordinate (16, 3) is equivalent to a move played at position (17, 4). They are mirrored diagonally within the same quadrant. In other words, every sequence is mirrored to side B. To manage this, we set up coordinates for the "incorrect" triangle, and solved the following equation system to determine the barycentric coordinates:

$$p = p_0 + (p_1 - p_0) \times s + (p_2 - p_1) \times t$$

where

- p is the position of the move;
- p_0, p_1 and p_2 are the points of the triangle; and
- s and t are two of the barycentric coordinates. The third barycentric coordinate can be found by calculating $1–s–t$. If $(0 \leq s \leq 1)$ and $(0 \leq t \leq 1)$ and $(s+t \leq 1)$, then the point is within the triangle and needs to be mirrored.

Once the tree is completed, an unpruned SGF output file is created.

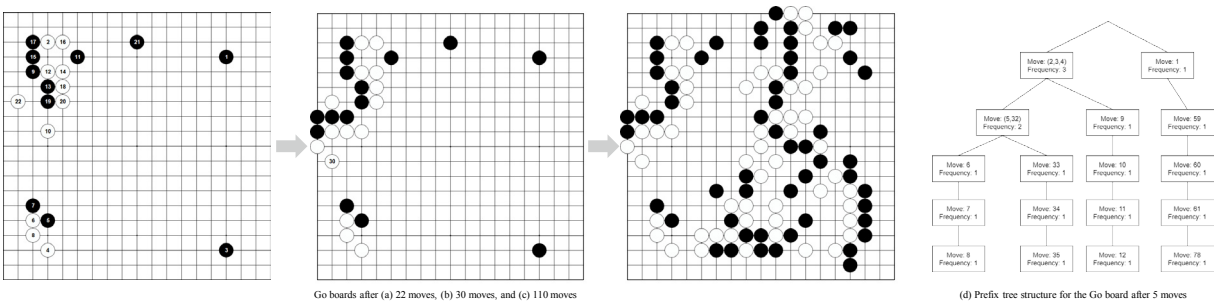
As our data science solution is developed in Python, we also create a "pickle" file using the standard Python pickle library, which stores the structure as a binary representation of the original Python structure. This file can then be used by "unpickling", which returns the original structure. This allows us to work with the output the need to parse the SGF file and recreating the tree structure. To create the SGF file, we perform a depth-first recursive traversal of the tree. The SGF file format is a structured text file with (i) an initial header and (ii) a body containing the moves with their variations as well as optional comments. Besides adding a color to each move, we add a comment that denotes the support (or frequency) of that move. Finally, we also add a comment to the top of the body to indicate the total number of moves and variations in the file.

4.2.1. Examples

Consider a dataset consisting of 89,587 professional games obtained from the summer 2017 edition of GoGoD. In Fig. 5(a), the first 22 moves of a game between two professionals is displayed. The first four moves are played in different corners and added to their joseki sequence. Moves 5 through 8 are played within a Chebyshev distance of 2 to the bottom-left joseki and are added to that sequence. Moves 9 and 10 form a case of a large pincer as described earlier. Move 10 is still considered a part of the top-left joseki because it is within range of 5 and fewer than 6 stones have been played in that sequence. The series continues adding stones without complication until move 21. It is a distance of 4 away from the nearest sequence in the top-left. However, since more than 6 stones have been added to that sequence, it is now considered too far away and not added to the joseki. Instead, it is added to the non-joseki list. Move 22, however, is added to the joseki because it is only 2 away. In contrast, with Helvensteijn's application, Move 22 and any future stones would be pruned away because 21 will not arise frequently enough. However, our solution just ignores 21 but still adds closer moves.

Fig. 5(b) shows the Go board after 30 moves. Since we have a limit of 20 moves, Move 30 is added to the non-joseki list. All further moves in that corner are added to the non-joseki list. The game continues with moves being played in other corners as well until at Move 110 when the non-joseki group contains 50 moves and the search is halted. Fig. 5(c) shows the Go board at Move 110.

Fig. 5(d) shows the prefix tree for the first 5 moves. For readability, we display the move numbers (instead of the coordinates of the move) in the tree. Moreover, Moves 2, 3 and 4 are merged into one because all three are mirrored to have the same coordinates and color at the top-right of the board.



Go boards after (a) 22 moves, (b) 30 moves, and (c) 110 moves

(d) Prefix tree structure for the Go board after 5 moves

Fig. 5 (a-c) Go board after 22, 30 and 110 moves, respectively; (d) Prefix tree for Go board after 5 moves.

4.3. Phase 3: Pruning and sorting

Although the resulting SGF file from the second phase is already very useful, it may contain some sequences that appear only a handful of times. This indicates that the position was likely unique and may not be useful for general study. Hence, the third phase of our data science solution creates another SGF output file (as well as another pickle file), in which the tree is traversed again. Every branch with a frequency of less than 10 (which is arbitrarily low) is pruned. We chose such a low number to account for Problem 4 (in Section 3.1) of the high minimum frequency requirement that (i) less common but well-known joseki and (ii) common joseki with a high number of variations were pruned away in Helvensteijn's application. Due to the vast complexity and number of positions in Go, sequences that are not relevant or erroneous will be unique or come up only 2 to 3 times. Additionally, there are highly complex joseki with dozens (or even hundreds of) variations. One of them is "*taisha*", which is also known as "joseki of hundreds variations" [23]. This joseki is worth studying because it is one of the few variations (out of all of the 10^{170} possible variations on a Go board) to have a name and is mentioned in books written by professionals. However, each specific variation only occur once in many thousands of games. Hence, our choice of 10 is high enough to avoid irrelevant sequences, while low enough to account for highly variable but significant variations.

In addition to the pruning step, we also sort the variations in descending order. The resulting SGF file thus differentiates between common moves and less common variations by making the more common results more accessible.

5. Evaluation

To evaluate our data science solution, we used a database for all games between January 1800 and July 2017. The results are divided by century, with one additional file where all results are combined. The most interesting data for study are (i) the SGF file of all data and (ii) the one with data from the 2000s (which can be considered "modern"). Table 1 shows total variations and moves that the variations are made up of. The total and modern file have both the pruned and unpruned data for comparison.

Table 1. Evaluation results for various SGF files.

Years	Games mined	Total variations	Total moves
All games unpruned	89,587	347,456	1,971,337
All games pruned	89,587	14,120	27,086
2000s unpruned	45,638	182,341	1,038,168
2000s pruned	45,638	7,558	14,868
1900s pruned	40,965	6,541	12,719
1800s pruned	2,984	2,002	4,386

Despite the low minimum frequency of 10, a large percentage of total variations were removed in the pruning process. In the case of the combined SGF file, 96% of all variations were cut. This confirms that uncommon variations tend to have extremely low support.

For comparison, the corner dictionary had 105,406 moves, and Kogo's Joseki Dictionary (KJD) has 62,749 moves [26]. In contrast, our data science solution produced 27,086 moves. On the surface, both the corner dictionary and KJD may appear to be better solutions by producing more moves. However, it is important to note that the corner dictionary mined over 1.6 million games mostly from amateurs who may play sub-optimal sequences, allowed sequences up to 35 moves in length, and simply searched for moves played in an 11x11 grid without using a criterion to determine if a move is part of the joseki or not. In other words, the corner dictionary is a *dictionary of common moves* instead of the desired joseki dictionary. In contrast, our solution only considers moves that are parts of a joseki, producing a truly joseki dictionary.

Moreover, it is also important to note that KJD was hand-crafted by several people with input from professionals over a period of more than a decade. As KJD was *manually* produced over the course of many years, variations of moves could have been added without regard to real frequency. In contrast, our solution *automatically* produces a dictionary of joseki satisfying the real frequency.

To compare with Helvensteijn's data mining application [21] that only discovered 81 joseki variations in a data set of 13,325, our data science solution discovered 14,120 variations—even with century specific files—which is far more interesting for study.

6. Conclusions

Our simple but yet useful data science solution *automatically* discovered many *joseki* (rather than just *moves*) and their variations—including relatively rare but noteworthy variations—without the need to complex machine learning techniques (e.g., artificial neural networks, deep learning) or programs (e.g., AlphaGo). Here, we used data mining methods, similarity measures based on Chebyshev distance, creation of prefix trees, and pruning techniques. We produced several SGF files that can be considered useful for study and to help improve a player's skill.

As ongoing and future works, we are exploring the impact of different input parameters. For instance, a lower minimum frequency and a longer maximum joseki length (say, 20) is expected to yield more variations to *manually*

selected dictionaries. Moreover, we are also exploring the benefits of having input from professional players, which could be valuable in determining the accuracy and quality of the SGF files.

Acknowledgements

This project is partially supported by NSERC (Canada) and University of Manitoba.

References

[1] Azaouzi M, Romdhane LB. An evidential influence-based label propagation algorithm for distributed community detection in social networks. Procedia Computer Science 2017; 112: 407-416.

[2] Braun P, Cuzzocrea A, Doan LMV, Kim S, Leung CK, Matundan JFA, Singh RR. Enhanced prediction of user-preferred YouTube videos based on cleaned viewing pattern history. Procedia Computer Science 2017; 112: 2230-2239.

[3] Braun P, Cuzzocrea A, Leung CK, Pazdor AGM, Tran K. Knowledge discovery from social graph data. Procedia Computer Science 2016; 96: 682-691.

[4] Hoi CSH, Leung CK, Tran K, Cuzzocrea A, Bochicchio M, Simonetti M. Supporting social information discovery from big uncertain social key-value data via graph-like metaphors. In: ICCC 2018. DOI: 10.1007/978-3-319-94307-7_8

[5] Kawagoe K, Leung CK. Similarities of frequent following patterns and social entities. Procedia Computer Science 2015; 60: 642-651.

[6] Leung CK, Jiang F, Poon TW, Crevier P. Big data analytics of social network data: who cares most about you on Facebook? In: Highlighting the Importance of Big Data Management and Analysis for Various Applications, 2018; pp. 1-15.

[7] Leung CK, Tanbeer SK, Cuzzocrea A, Braun P, MacKinnon RK. Interactive mining of diverse social entities. KES Journal 2016; 20(2): 97-111.

[8] Cuzzocrea A, Grasso GM, Jiang F, Leung CK. Mining uplink-downlink user association in wireless heterogeneous networks. In: IDEAL 2016, pp. 533-541.

[9] Ayadi A, Zanni-Merk C, de Bertrand de Beuvron F, Krichen S. BNO: an ontology for describing the behaviour of complex biomolecular networks. Procedia Computer Science 2017; 112: 524-533.

[10] Jiang F, Leung CK, Sarumi OA, Zhang CY. Mining sequential patterns from uncertain big DNA data in the Spark framework. In: IEEE BIBM 2016, pp. 874-881.

[11] Leung CK, Joseph KW. Sports data mining: predicting results for the college football games. Procedia Computer Science 2014; 35: 710-719.

[12] Correia F, Alves-Oliveira P, Ribeiro T, Melo,FS, Paiva A. A social robot as a card game player. In: AAAI AIIDE 2017, pp. 23- 29.

[13] Bertens P, Guitart A, Perianez A. Games and big data: a scalable multi-dimensional churn prediction model. In: IEEE CIG 2017, pp. 33-36.

[14] Braun P, Cuzzocrea A, Keding TD, Leung CK, Pazdor AGM, Sayson D. Game data mining: clustering and visualization of online game data in cyber-physical worlds. Procedia Computer Science 2017; 112: 2259-2268.

[15] Brown JA, Cuzzocrea A, Kresta M, Kristjanson KDL, Leung CK, Tebinka TW. A machine learning system for supporting advanced knowledge discovery from chess game data. In: IEEE ICMLA 2017, pp. 649-654.

[16] Bossomaier T, Traish J, Gobet F, Lane PCR. Neuro-cognitive model of move location in the game of Go. In: IJCNN 2012, pp. 1-7.

[17] Lee C, Wang M, Wu M, Teytaud O, Yen S. T2FS-based adaptive linguistic assessment system for semantic analysis and human performance evaluation on game of Go. IEEE TFS 2015; 23(2): 400-420.

[18] Silver D, Huang A, Maddison CJ, Guez A, Sifre L, van den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M, Dieleman S, Grewe D, Nham J, Kalchbrenner N, Sutskever I, Lillicrap TP, Leach M, Kavukcuoglu K, Graepel T, Hassabis D. Mastering the game of Go with deep neural networks and tree search. Nature 2016; 529(7587): 484-489.

[19] Xiao C, Müller M. Factorization ranking model for move prediction in the game of Go. In: AAAI 2016, pp. 1359-1365.

[20] Shotwell P. The Game of Go: Speculations on its Origins and Symbolism in Ancient China. 2008.

[21] Helvensteijn M. Applying data mining to the study of joseki. In: IFIP AI 2008, pp. 87-96.

[22] Ishigure I. In the Beginning: The Opening in the Game of Go. Ishi Press; 1973.

[23] Kosugi K, Davies J. 38 Basic Joseki. Ishi Press; 1973.

[24] Bozulich R, Kazunari F. Get Strong at Joseki 3. Kiseido Publishing Company; 1996.

[25] Miller A. Josekipedia. http://www.josekipedia.com/info/overview.php

[26] Dinerchtein A. New Moves. Slate & Shell; 2010.

[27] Prokop J. Waltheri's go pattern search. http://ps.waltheri.net/about/