Interpretable Strategy Synthesis for Competitive Games

Oral Prelim Presentation

Abhijeet Krishnan

Department of Computer Science
North Carolina State University

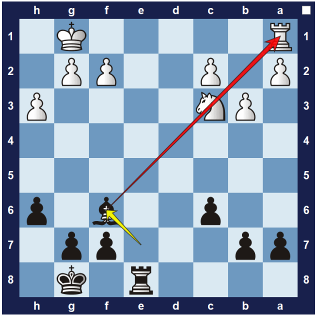
April 14, 2023

Outline

Outline

Overview

- Players tend to identify *strategies* in competitive games
- E.g., the *pin* in chess
- *Good*, *interpretable* strategies help players improve



Chessfox.com

Figure 1: An example of the *pin* tactic in chess

Overview

Motivating Question

How could we *automatically* learn good, interpretable strategies that help players improve?

- I formalize this question as the problem of *interpretable strategy synthesis*
- I show how we can approach it for *chess*
- I propose an approach for it for the game of *MicroRTS*

Outline

Outline

Motivation

Observation

Players tend to identify *strategies* that are a) good and b) communicable when trying to find ways to win in a competitive game.

- Can be communicated to another player

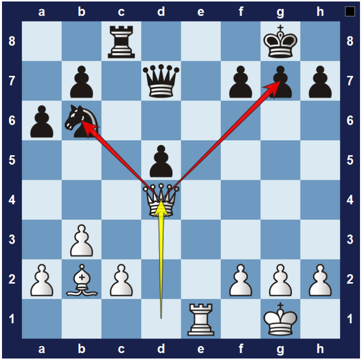
Motivation

Observation

Players tend to identify *strategies* that are a) good and b) communicable when trying to find ways to win in a competitive game.

- Can be communicated to another player
- Help a player decide which actions to take in-game

Examples



Chessfox

Figure 2: An example of the *fork* tactic in chess

Examples

Go Full Build

Figure 3: A *cannon rush* in progress against a Terran opponent

Examples

Unscriptd, The Times

Figure 4: Boris Becker's tennis service tic

Outcomes

- Esports is a massive industry

Tournament	Game	Prize Pool (USD)
World Blitz Chess Championship	Chess	350,000
IEM Katowice	StarCraft II	500,000

- Better strategies → higher player skill → more earning potential
- Could be used to coach players at all levels of skill
- Could be used to explain artificial agent decisions

Outline

Thesis Statement

Thesis Statement

A *computational model* of a game strategy, along with a *learning method*, could meet the goals of discovering good, communicable strategies and impact the fields of competitive esports and explainable AI.

RQ1

RQ1

How do we formally define the problem of *Interpretable Strategy Synthesis* (ISS)?

- Necessary to clarify terms like *strategy*

RQ1

RQ1

How do we formally define the problem of *Interpretable Strategy Synthesis* (ISS)?

- Necessary to clarify terms like *strategy*
- Need to agree on what “interpretable” means

RQ1

RQ1

How do we formally define the problem of *Interpretable Strategy Synthesis* (ISS)?

- Necessary to clarify terms like *strategy*
- Need to agree on what "interpretable" means
- Useful to compare and contrast approaches across game domains

RQ2

RQ2

How do we approach the problem of ISS for the game of *chess*?

- Well-studied domain with lots of player-derived strategies

RQ3

RQ3

How do we approach the problem of ISS for the game of *MicroRTS*?

- Less-studied domain with fewer player-derived strategies
- Qualitatively different from chess

Outline

Outline

RQ1 – ISS Framework

RQ1

How do we formally define the problem of *Interpretable Strategy Synthesis* (ISS)?

The Need for a Framework

Gaps in existing work on strategy synthesis –

- No comparison between multiple algorithms

The Need for a Framework

Gaps in existing work on strategy synthesis –

- No comparison between multiple algorithms
- No comparison between multiple strategy representations

The Need for a Framework

Gaps in existing work on strategy synthesis –

- No comparison between multiple algorithms
- No comparison between multiple strategy representations
- No comparison across multiple games

The Need for a Framework

Gaps in existing work on strategy synthesis –

- No comparison between multiple algorithms
- No comparison between multiple strategy representations
- No comparison across multiple games
- No definition or evaluation of interpretability

The Need for a Framework

Paper	#Domains	#Models	#Algorithms	Interpretability
Spronck, Sprinkhuizen-Kuyper, and Postma [1]	2	1	1	✗
Mesentier Silva, Isaksen, Togelius, *et al.* [2]	1	1	4	✓
Butler, Torlak, and Popović [3]	1	1	1	✗
Canaan, Shen, Torrado, *et al.* [4]	1	1	1	✗
Freitas, Souza, and Bernardino [5]	1	1	1	✗
Mariño, Moraes, Oliveira, *et al.* [6]	1	1	1	✗
Krishnan and Martens [7]	1	1	1	✗
Mariño and Toledo [8]	1	1	1	✗
Medeiros, Aleixo, and Lelis [9]	2	1	2	✗

Table 1: List of works in ISS

Strategy

Definition (strategy)

Given a game environment $\mathcal{G}$ modeled as a finite, episodic MDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$, a *strategy* σ is —

$$\sigma(a|s) \doteq \mathbb{P}[A_t = a | S_t = s], \forall s \in \mathrm{A}_\sigma, a \in \mathcal{A}(s)$$

A_σ: set of *applicable* states

- A strategy is similar to an RL policy
- Not necessarily applicable to all states
- Describes a pattern or feature that comes up often in gameplay

Interpretable Strategy Synthesis (ISS)

Definition (ISS)

Given a —

- Game environment $\mathcal{G}$
- Strategy model $\mathcal{M}$
- Performance measure $\mathcal{R}_{\mathcal{G}} : \mathcal{M} \to \mathbb{R}$
- Interpretability measure $\mathcal{I}_{\mathcal{G}} : \mathcal{M} \to \mathbb{R}$

The problem of ISS is to find a strategy σ^* s.t. —

$$\sigma^* \doteq \arg\max_{\sigma} \mathcal{R}_{\mathcal{G}}(\sigma)\mathcal{I}_{\mathcal{G}}(\sigma), \sigma \in \mathcal{M}$$

Outline

RQ2 – Chess

RQ2

How do we approach the problem of Interpretable Strategy Synthesis for the game of *chess*?

Why Chess?

- Popular "esport" with a *long* competitive history
- Large number of player-discovered strategies to compare against
- Extensive use as a testbed for AI
- Potential for discovery of new strategies using artificial agents [10]

Validating ISS using Chess

- Using our framework, could we measure how "good" existing chess strategies are?

RQ2(a)

Could we represent known chess tactics as a strategy model for chess and develop metrics to show that they suggest better moves than a random baseline?

Validating ISS using Chess

- Using our framework, could we measure how "good" existing chess strategies are?

RQ2(a)

Could we represent known chess tactics as a strategy model for chess and develop metrics to show that they suggest better moves than a random baseline?

Strategy Model for Chess

$$\texttt{tactic}(\text{Position}, \text{Move}) \leftarrow \texttt{feature\_1}(\cdots), \texttt{feature\_2}(\cdots), \vdots \texttt{feature\_n}(\cdots)$$

Figure 5: Our chess strategy model expressed in Prolog pseudocode

Strategy Model for Chess

- `Position`: list of predicate features encoding a board state

```
tactic(Position,
      Move) ←
  feature_1(···),
  feature_2(···),
  ⋮
  feature_n(···)
```

Figure 5: Our chess strategy model expressed in Prolog pseudocode

Strategy Model for Chess

- `Position`: list of predicate features encoding a board state
- `Move`: predicate encoding of a move

```
tactic(Position,
     Move) ←
  feature_1(···),
  feature_2(···),
  ⋮
  feature_n(···)
```

Figure 5: Our chess strategy model expressed in Prolog pseudocode

Strategy Model for Chess

- `Position`: list of predicate features encoding a board state
- `Move`: predicate encoding of a move
- `feature_i`: board state feature encoded as a predicate

```
tactic(Position,
    Move) ←
  feature_1(···),
  feature_2(···),
  ⋮
  feature_n(···)
```

Figure 5: Our chess strategy model expressed in Prolog pseudocode

Example

```
fork(Position,Move) ←
    legal_move(Position,Move),
    move(Move,_,To,_),
    make_move(Position,Move,NewPosition),
    can_capture(NewPosition,To,ForkSquare1),
    can_capture(NewPosition,To,ForkSquare2),
    different(ForkSquare1,ForkSquare2).
```

Figure 6: An interpretation of the *fork* tactic from the chess literature using our predicate vocabulary.

Why This Model?

- Chess tactics are used by human players to think about chess [11] and useful in chess education [12]
- FO-logic used extensively to model chess patterns [13]–[19]
- Logic rules are acknowledged to be interpretable and have a long history of research [20]

Validating ISS using Chess

- Using our framework, could we measure how "good" existing chess strategies are?

RQ2(a)

Could we represent known chess tactics as a strategy model for chess and develop metrics to show that they suggest better moves than a random baseline?

Metrics

Definition

Divergence of a tactic from a set of examples P is the average difference in *evaluation* between the moves suggested by the tactic and the ground truth move.

$$\text{Divergence}_E(\sigma, P) \doteq \frac{1}{|P_{\mathrm{A}}|} \sum_{(s,a_1)\in P_{\mathrm{A}}} \sum_{a_2\in\mathcal{A}(s)} \sigma(a_2|s)\, d_E(s, a_1, a_2) \tag{1}$$

Validating ISS using Chess

- Using our framework, could we measure how "good" existing chess strategies are?

RQ2(a)

Could we represent known chess tactics as a strategy model for chess and develop metrics to show that they suggest better moves than a random baseline?

Evaluating Chess Tactics [21]

- Obtain tactics learned using PAL [19]
- Translate to chess strategy model
- Obtain example set of positions and moves from online games played by human beginners
- Measure divergence using strong and weak chess engines
- Compare to random baseline

Results

- Tactics have lower divergence when measured using a weak chess engine, higher divergence when measured with a strong engine
- Tend to suggest moves similar to a human beginner

Learning Chess Strategy Models

- How do we automatically learn “good” chess strategy models?

RQ2(b)

Do the chess strategies learned using inductive logic programming outperform a random baseline in how closely their divergence scores approximate a beginner player?

Learning Chess Strategies using ILP [7]

- Inductive Logic Programming (ILP) is a symbolic machine learning technique
- Translate the ISS for chess problem into an ILP problem
- Modify an existing ILP system to learn chess strategies
- Evaluate the learned chess strategies using divergence metrics
- Compare to random, strong/weak engine baselines

Results

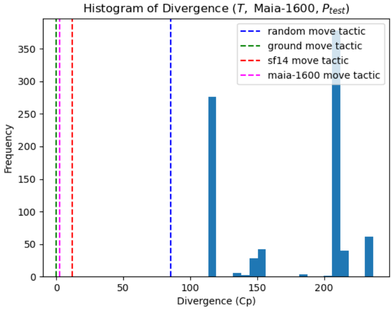


Figure 7: Divergence histogram for T evaluated using Maia-1600

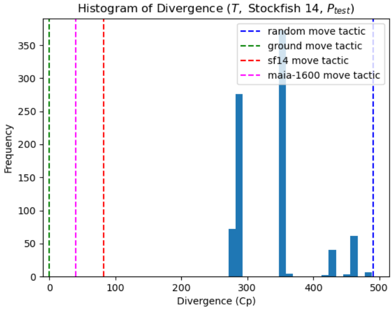


Figure 8: Divergence histogram for T evaluated using Stockfish 14

Conclusion

- The chess strategies learned are better than a random baseline when evaluated using a strong engine

Improving the ILP Learning Method

- The learned chess strategies merely outperform a random baseline, could we improve upon them?

RQ2(c)

Do the chess strategies learned by an ILP system incorporating the changes of the new predicate vocabulary and precision/recall-based constraints produce moves better than those learned by an ILP system without these modifications?

Improvements using Precision/Recall-based Constraints [22]

- Limit the search space of chess strategies using constraints based on precision and recall of the synthesized strategies
- Introduce a new predicate vocabulary
- Conduct ablative study to measure impact of contributions
- Use similar evaluation setup to previous work

Results

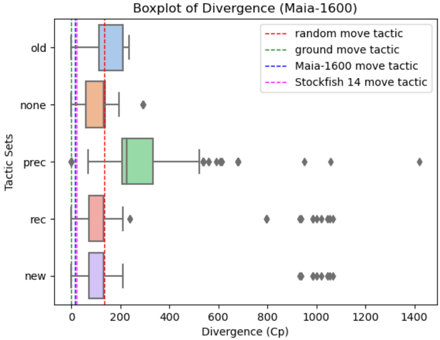


Figure 9: Boxplot of tactic divergence (evaluated using Maia-1600) for each system

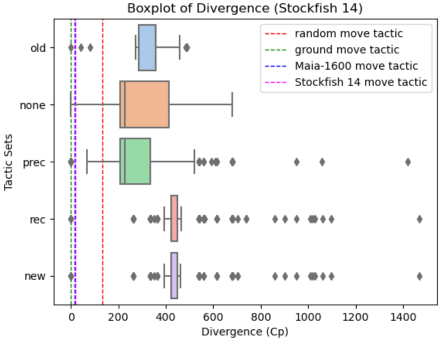


Figure 10: Boxplot of tactic divergence (evaluated using Stockfish 14) for each system

Conclusion

- Statistically significant ($p < 0.01$) improvement in divergence due to new predicate vocabulary
- Mixed results for precision and recall constraints
  - Recall constraint improves divergence when measured using weak engine
  - Precision constraint improves divergence when measured using strong engine

Outline

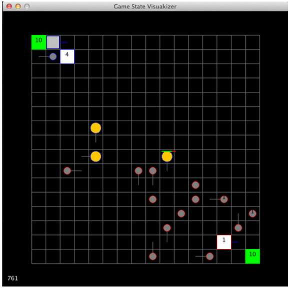
RQ3 – MicroRTS

RQ3

How do we approach the problem of Interpretable Strategy Synthesis for the game of *MicroRTS*?

Why MicroRTS?

- Simplified real-time strategy game for AI research [23]
- Popular genre for esport titles
- Active research community
- Qualitatively different from chess – *real-time*, *partially observable*



Google Code Archive

Figure 11: A MicroRTS game in progress

Learning MicroRTS Strategies using ASP

RQ3(a)

How does an ASP-based approach towards developing a synthesizer for the *SynProS competition* compare to other synthesizers in this competition?

SynProS Competition

- SynProS: **Syn**thesis of **Pro**grammatic **S**trategies
- Research competition [24] to test ISS approaches for MicroRTS with a fixed strategy model
- MicroRTS strategy model represented as a production in a **c**ontext-**f**ree **g**rammar (CFG)

$$
\begin{aligned}
S_1 &\rightarrow C\ S_1 \mid S_2\ S_1 \mid S_3\ S_1 \mid \epsilon \\
S_2 &\rightarrow \text{if } (S_5) \text{ then } \{C\} \mid \text{if } (S_5) \text{ then } \{C\} \text{ else } \{C\} \\
S_3 &\rightarrow \text{for (each unit } u) \ \{S_4\} \\
S_4 &\rightarrow C\ S_4 \mid S_2\ S_4 \mid \epsilon \\
S_5 &\rightarrow \text{not } B \mid B \\
B &\rightarrow b_1 \mid b_2 \mid \cdots \mid b_m \\
C &\rightarrow c_1\ C \mid c_2\ C \mid \cdots \mid c_n\ C \mid c_1 \mid c_2 \mid \cdots \mid c_n \mid \epsilon
\end{aligned}
$$

ASP-based Approach

- **A**nswer **S**et **P**rogramming (ASP) is a declarative programming paradigm (like Prolog)
- Has been used to model and generate game levels [25], [26]
- Has been used to model visualization design preferences and optimize data layouts based on them [27]

ASP-based Approach

- Proposal to replicate approach in [27]
- Model MicroRTS strategy using ASP
- Train a regression model to predict strategy win rate given feature encoding
- Convert regression function to ASP to optimize newly generated strategies

Interpretability Factors for MicroRTS Strategies

- How do we define the interpretability of a MicroRTS strategy?
- How do we make MicroRTS strategies *more* interpretable?

RQ3(b)

Which features of a MicroRTS strategy model have a statistically significant correlation with the interpretability of said strategy?

Task Design

- Conduct a *human-grounded* [28] evaluation using a simplified task to quantify interpretability
- Use a forward simulation/prediction task
- Subjects presented with —
  - Strategy
  - Game state
  - Options for future states (1 correct, 3 incorrect)
- Task is to predict the expected future state from current state if strategy is followed
- Generate tasks using ASP model of MicroRTS strategy

Obtaining Significant Factors

- Train decision tree model to predict whether strategy will be correctly simulated
- Obtain significant factors by measuring Gini index [29]

Outline

Proposed Work & Timeline

- Spring 2022
  - RQ1 Framework (EAAI-22) - *Completed*
  - RQ2(a) Validity (EAAI-22) - *Completed*
- Fall 2022
  - RQ2(b) Study (AIIDE-22 SG Workshop) - *Completed*
- Spring 2023
  - RQ2(c) Study (under review) - *Completed*
  - RQ3(a) Dataset Assembly - *In progress*
  - RQ3(a,b) ASP Modeling

Proposed Work & Timeline

- Summer 2023

RQ3(a) Study (ACG-23)
RQ3(b) Obtaining Strategies

- Fall 2023

RQ3(b) Task Design
RQ3(b) Study Approval

- Spring 2024

RQ3(b) Analysis (EXTRAAMAS ’24)
  - Complete dissertation (early Spring ’24)
  - Defend thesis (Mid-late Spring ’24)
  - Graduate (May ’24)

Outline

Conclusion

- Goal: investigate approaches to the problem of interpretable strategy synthesis for games
- Defined a framework for ISS
- Approached ISS using ILP for chess
- Proposal to approach ISS for MicroRTS using ASP
- Outcomes –
  - Benefit esports industry in providing better analytics for player performance
  - Benefit explainable AI research in investigating new approaches to generate policy explanations

Thank You!

Questions?

References I

[1] P. Spronck, I. Sprinkhuizen-Kuyper, and E. Postma, “Online adaptation of game opponent ai with dynamic scripting,” *International Journal of Intelligent Games and Simulation*, vol. 3, no. 1, pp. 45–53, 2004.

[2] F. de Mesentier Silva, A. Isaksen, J. Togelius, and A. Nealen, “Generating heuristics for novice players,” in *2016 IEEE Conference on Computational Intelligence and Games (CIG)*, IEEE, 2016, pp. 1–8.

References II

[3] E. Butler, E. Torlak, and Z. Popović, "Synthesizing interpretable strategies for solving puzzle games," in *Proceedings of the 12th International Conference on the Foundations of Digital Games*, ser. FDG '17, Hyannis, Massachusetts: Association for Computing Machinery, 2017, ISBN: 9781450353199. DOI: 10.1145/3102071.3102084. [Online]. Available: https://doi.org/10.1145/3102071.3102084.

References III

[4] R. Canaan, H. Shen, R. Torrado, J. Togelius, A. Nealen, and S. Menzel, "Evolving agents for the hanabi 2018 cig competition," in *2018 IEEE Conference on Computational Intelligence and Games (CIG)*, 2018, pp. 1–8. DOI: 10.1109/CIG.2018.8490449.

References IV

[5] J. M. de Freitas, F. R. de Souza, and H. S. Bernardino, "Evolving controllers for mario ai using grammar-based genetic programming," in *2018 IEEE Congress on Evolutionary Computation (CEC)*, 2018, pp. 1–8. DOI: 10.1109/CEC.2018.8477698.

References V

[6] J. R. H. Mariño, R. O. Moraes, T. C. Oliveira, C. Toledo, and L. H. S. Lelis, "Programmatic strategies for real-time strategy games," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 1, pp. 381–389, May 2021. DOI: 10.1609/aaai.v35i1.16114. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/16114.

References VI

[7] A. Krishnan and C. Martens, “Synthesizing interpretable chess tactics from player games,” in *Proceedings of the Workshop on Artificial Intelligence for Strategy Games (SG) and Esports Analytics (EA), 18th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, American Association for Artificial Intelligence, Oct. 2022.

References VII

[8] J. R. Mariño and C. F. Toledo, "Evolving interpretable strategies for zero-sum games," *Applied Soft Computing*, vol. 122, p. 108 860, 2022.

[9] L. C. Medeiros, D. S. Aleixo, and L. H. Lelis, "What can we learn even from the weakest? learning sketches for programmatic strategies," *arXiv preprint arXiv:2203.11912*, 2022.

References VIII

[10] M. Sadler and N. Regan, "Game changer: Alphazero's groundbreaking chess strategies and the promise of ai," *New in Chess*, 2019.

[11] A. Szabo, "Computer chess tactics and strategy," PhD thesis, University of British Columbia, 1984. DOI: http://dx.doi.org/10.14288/1.0051870. [Online]. Available: https://open.library.ubc.ca/collections/ubctheses/831/items/1.0051870.

References IX

[12] F. Gobet and P. J. Jansen, "Training in chess: A scientific approach," *Education and chess*, 2006.

[13] H. J. Berliner, "A representation and some mechanisms for a problem solving chess program," Carnegie-Mellon Univ Pittsburgh PA Dept of Computer Science, Tech. Rep., 1975.

[14] J. Pitrat, "A chess combination program which uses plans," *Artificial Intelligence*, vol. 8, no. 3, pp. 275–321, 1977.

References X

[15] D. E. Wilkins, *Using patterns and plans to solve problems and control search*. Stanford University, 1979.

[16] B. J. Huberman, “A program to play chess end games,” PhD thesis, Department of Computer Science, Stanford University., Aug. 1968.

[17] M. A. Bramer, “Representation of knowledge for chess endgames towards a self-improving system,” PhD thesis, Open University (United Kingdom), 1977.

References XI

[18] I. Bratko, "Knowledge-based problem-solving in al3," *Machine intelligence*, vol. 10, pp. 73–100, 1982.

[19] E. Morales, "First order induction of patterns in chess," PhD thesis, PhD thesis, The Turing Institute-University of Strathclyde, 1992.

References XII

[20] Y. Zhang, P. Tiňo, A. Leonardis, and K. Tang, "A survey on neural network interpretability," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 5, no. 5, pp. 726–742, Oct. 2021, ISSN: 2471-285X. DOI: 10.1109/TETCI.2021.3100641.

References XIII

[21] A. Krishnan and C. Martens, "Towards the automatic synthesis of interpretable chess tactics," in *Proceedings of the Explainable Agency in Artificial Intelligence Workshop, 36th AAAI Conference on Artificial Intelligence*, American Association of Artificial Intelligence, Mar. 2022, pp. 91–97.

[22] A. Krishnan, C. Martens, and A. Jhala, "Improving strategy synthesis for chess using precision and recall,", [Manuscript submitted for publication], Mar. 2023.

References XIV

[23] S. Ontanon, "The combinatorial multi-armed bandit problem and its application to real-time strategy games," *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 9, no. 1, pp. 58–64, Jun. 2021. DOI: 10.1609/aiide.v9i1.12681. [Online]. Available: https://ojs.aaai.org/index.php/AIIDE/article/view/12681.

References XV

[24] R. Moraes. (Jul. 2021), Synpros - synthesis of programmatic strategies, [Online]. Available: https://rubensolv.github.io/synpros-microrts/ (visited on 03/26/2023).

[25] A. M. Smith and M. Mateas, “Answer set programming for procedural content generation: A design space approach,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, no. 3, pp. 187–200, 2011.

References XVI

[26] A. M. Smith, E. Andersen, M. Mateas, and Z. Popović, "A case study of expressively constrainable level design automation tools for a puzzle game," in *Proceedings of the International Conference on the Foundations of Digital Games*, 2012, pp. 156–163.

References XVII

[27] D. Moritz, C. Wang, G. L. Nelson, H. Lin, A. M. Smith, B. Howe, and J. Heer, "Formalizing visualization design knowledge as constraints: Actionable and extensible models in draco," *IEEE transactions on visualization and computer graphics*, vol. 25, no. 1, pp. 438–448, 2018.

[28] F. Doshi-Velez and B. Kim, *Towards a rigorous science of interpretable machine learning*, 2017. arXiv: 1702.08608 [stat.ML].

References XVIII

[29] C. Molnar, "A guide for making black box models explainable," *URL: https://christophm. github. io/interpretable-ml-book*, p. 3, 2018.