Artificial Intelligence 217 (2014) 92–116

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Information capture and reuse strategies in Monte Carlo Tree Search, with applications to games of hidden information

Edward J. Powley *, Peter I. Cowling, Daniel Whitehouse

Department of Computer Science, University of York, Heslington, York, YO10 5DD, UK

ARTICLE INFO

Article history:
Received 20 December 2012
Received in revised form 5 August 2014
Accepted 14 August 2014
Available online 23 August 2014

Keywords:
Game tree search
Hidden information
Information reuse
Machine learning
Monte Carlo Tree Search (MCTS)
Uncertainty

ABSTRACT

Monte Carlo Tree Search (MCTS) has produced many breakthroughs in search-based decision-making in games and other domains. There exist many general-purpose enhancements for MCTS, which improve its efficiency and effectiveness by learning information from one part of the search space and using it to guide the search in other parts. We introduce the *Information Capture And ReUse Strategy (ICARUS)* framework for describing and combining such enhancements. We demonstrate the ICARUS framework's usefulness as a frame of reference for understanding existing enhancements, combining them, and designing new ones.

We also use ICARUS to adapt some well-known MCTS enhancements (originally designed for games of perfect information) to handle information asymmetry between players and randomness, features which can make decision-making much more difficult. We also introduce a new enhancement designed within the ICARUS framework, *EPisodic Information Capture and reuse (EPIC)*, designed to exploit the episodic nature of many games. Empirically we demonstrate that EPIC is stronger and more robust than existing enhancements in a variety of game domains, thus validating ICARUS as a powerful tool for enhancement design within MCTS.

1. Introduction

Monte Carlo Tree Search (MCTS) is a decision tree search algorithm that has produced a huge leap in AI player strength for a range of two-player zero-sum games and proven effective in a wide range of games and decision problems [1]. In particular, MCTS is effective when it is difficult to evaluate non-terminal states so that traditional depth-limited search methods perform poorly. For example, MCTS has advanced the state of the art in computer Go from the level of weak amateur to approach that of professional players in only a few years [2,3]. MCTS has also produced state-of-the-art performance in many other domains, with over 250 papers published since the algorithm's invention in 2006 [1]. MCTS shows promise in real-time games, being the basis of winning competition entries for both Ms. Pac-Man [4] and the Physical Travelling Salesman Problem [5].

Generally speaking, MCTS algorithms heuristically build an asymmetric partial search tree by applying machine learning, using the weak reward signal given by randomly simulating a playout to the end of the game from nodes representing intermediate positions. The tree is descended by recursively applying a multi-armed bandit formula (such as UCB1 [6]) to each tree node's counts of simulation wins and visits.

* Corresponding author.
E-mail addresses: edward.powley@york.ac.uk (E.J. Powley), peter.cowling@york.ac.uk (P.I. Cowling), dw830@york.ac.uk (D. Whitehouse).

http://dx.doi.org/10.1016/j.artint.2014.08.002
0004-3702/

While MCTS has provided effective and even state-of-the-art decision-making in its "vanilla" form (particularly UCT [7]), it is often enhanced [1]. Some of these enhancements incorporate external knowledge into the search, whereas others are *general purpose* enhancements which can be applied to any domain without specific knowledge. In some cases these enhancements are crucial aspects of successful MCTS programs, for example the RAVE enhancement [8] used in champion Go [9] and Hex [10] programs. In vanilla MCTS, the only information retained from a playout is the terminal reward, and the only use for that information is to update the nodes visited during the playout. Many enhancements aim to extract more data from each playout and spread the influence of that data across more of the search tree, thus increasing the value of each playout.

In this work we investigate the use of general purpose enhancements to improve the performance of MCTS. In some games[1] a move that is good in one state may be good in other similar states, and we argue that general purpose MCTS enhancements improve the performance of the algorithm by exploiting opportunities for learning in these situations. The enhancements in this paper bootstrap the learning of whether states and actions are good or bad by using analogy with similar states and actions elsewhere in the search tree. A substantial contribution of this work is to develop a framework which formalises the correlation between states and actions, and the effects that this has on the tree and default policies of MCTS. Further, we develop and empirically investigate combination operators for MCTS enhancements, and show how we can use our framework and operators to understand, categorise and invent new enhancements. Hence we can explain the effectiveness of MCTS enhancements by understanding how information is shared between states and actions and how this information is used to improve the MCTS selection and simulation policies. Additionally we show that enhancements developed for games of *perfect information* (where the state is fully observable to all players and state transitions are deterministic) can also be effective in games of *imperfect information* (where the state is partially observable with different observations for different players, and state transitions may be stochastic).

The framework in this paper aims to unify MCTS and its various enhancements, whereas other authors have sought to unify MCTS and related search techniques. Keller and Helmert [11] propose a framework for finite horizon Markov decision processes (i.e. single-player games). This framework can express UCT as well as other heuristic search and dynamic programming techniques. By interchanging the component parts of the methods within the framework, new methods are derived. Maes et al. [12] define a grammar over Monte Carlo search algorithms for single-player games (including UCT and Nested Monte Carlo Search [13]), and use this to evolve new algorithms. Saffidine [14] presents a framework for "best first search" methods in two-player games, which encompasses methods such as MCTS-Solver [15] and Proof-Number Search [16] and guarantees that methods expressible in this framework must converge to the minimax solution of the game.

The idea of enhancing an algorithm to better capture and reuse information as it executes is used in a number of search and learning algorithms. The efficiency of the α–β pruning strategy in minimax search is largely dependent on the order in which actions are visited in the tree [17]. Enhancements such as the killer heuristic [18], history heuristic [19] and iterative deepening [20] use information gathered during the search to refine this ordering as the search progresses. Even α–β pruning itself can be seen as an information reuse enhancement, as it uses information gathered in one part of the tree to influence the search in other parts (specifically, to prune other parts entirely). Machine learning algorithms can also bootstrap learning through reuse. In *transfer learning* [21] or *lifelong learning* [22], the learner uses information learned from previous problems to bootstrap learning for the present problem. In *multitask learning* [23], the system learns to solve several problems in parallel. In both cases the system can be thought of as "learning to learn", thus these approaches are often termed *meta-learning* [24]. Typically meta-learning systems work by learning reusable features or representations, or by adjusting the parameters of an underlying learning algorithm. Although the actual methods used are different, the idea of a learning system acquiring knowledge over its lifetime as it is confronted by different problems is similar to the idea of a tree search algorithm transferring knowledge from one part of the game tree to another over the "lifetime" of a single search.

Most general purpose MCTS enhancements derive knowledge by comparing and combining simulations from different states. We show that these general purpose enhancements do not always work and are sometimes detrimental to the performance of MCTS, adding to existing observations that certain enhancements which are effective in some domains fail to provide any benefit in other domains (e.g. [25,26]). The most effective enhancements correctly identify which states have correlated action values. This suggests that even if a general purpose enhancement is knowledge-free, there is implicit knowledge contained in the AI designer's decision of whether or not to use that enhancement.

As well as letting us choose between existing enhancements, consideration of correlated states allows us to design entirely new enhancements. In this paper we present a new enhancement, *EPisodic Information Capture and reuse (EPIC)*, that was designed by considering correlation between states in the card game Dou Di Zhu. Dou Di Zhu has an episodic structure, where a game consists of a sequence of somewhat independent rounds, and EPIC is designed to correlate states in analogous positions within different episodes. Many games have an episodic structure, and we demonstrate that EPIC is an effective general purpose enhancement for other games.

Capturing information in the correct way is important, but reusing it in the correct way is equally crucial. Our framework separates reuse from capture, enabling us to study the effectiveness of different information reuse techniques. In [27] we

[1] The word *games* in this paper includes multiplayer games, single player puzzles and decision problems, although most work to date is on two-player noncooperative games.