Biologically Inspired Cognitive Architectures (2014) **10**, 51–60

Available at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/bica

Alleviating the curse of dimensionality — A psychologically-inspired approach

Vladislav D. Veksler *, **Kevin A. Gluck, Christopher W. Myers, Jack Harris, Thomas Mielke**

Air Force Research Laboratory, Wright-Patterson AFB, USA

Received 7 November 2014; accepted 7 November 2014

KEYWORDS
Combinatorics;
Function approximation;
Chunking;
Unitization;
Configural-cue;
EPAM;
ACT-R;
Reinforcement learning;
Memory activation

Abstract
Various combinations of perceptual features are relevant for learning and action-selection. However, the storage of all possible feature combinations presents computationally impractical, and psychologically implausible, memory requirements in non-trivial environments due to a state-space explosion. Some psychological models suggest that feature combinations, or chunks, should be generated at a conservative rate (Feigenbaum and Simon, 1984). Other models suggest that chunk retrieval is based on statistical regularities in the environment, i.e. recency and frequency (Anderson and Schooler, 1991). We present a computational model for chunk learning based on these two principles, and demonstrate how combining these principles alleviates state-space explosion, producing exponential memory savings while maintaining a high level of performance.

Introduction

Decision making depends on the set of features perceived at decision time. Heart attack diagnoses depend on patient symptoms, such as chest pain and electrocardiogram readings; cyber attack detection and response depends on features of network activity; and the decision to break or turn in a critical driving situation depends on the sizes, locations, and direction vectors of nearby cars, pedestrians, and other obstacles. Whether the goal is to understand, predict, or aid human decision-making, or whether it is to achieve human-level performance in complex environments, inferring state-representation from perceived features is an important problem in Cognitive Science and Artificial Intelligence.

Cognitive architectures are often based on production systems (Anderson, 1993; Laird, 2012), where each production is a rule that specifies a condition (state) and an action to be fired whenever this condition is met. One of the greatest difficulties in the development of a cognitive model is in accounting for all the states that the model could encounter in a given task-environment. The inability to account for all

* Corresponding author.
E-mail address: vdv718@gmail.com (V.D. Veksler).

http://dx.doi.org/10.1016/j.bica.2014.11.007
2212-683X/

potential states generates brittle models that halt in the face of error.

Autonomous agents in machine learning are often based on Reinforcement Learning (RL) (Sutton & Barto, 1998), taking a similar approach to production systems, but assuming all possible rules, or state-action pairs. That is, rather than specifying which rules are appropriate in a given environment, RL agents contain all possible state-action pairs in a lookup table, and select which actions to fire based on prior reward feedback recorded for each state-action pair. However, in environments where each state comprises numerous perceptual features, treating each unique combination of percepts as a state (lookup-based RL) is extremely inefficient. As Sutton and Barto (1998) point out, ''The problem is not just the memory needed for large tables, but the time and data needed to fill them accurately. In other words, the key issue is that of *generalization*.'' For example, a lookup-based RL agent may learn that eating red and green apples is rewarding, but will fail to generalize, and will produce random-level behavior when encountering a yellow apple.

Rather than treating each unique input as a state, it is possible to treat each perceptual feature as a state (feature-based RL). Assuming a world where perceptual features may be {red, green, yellow, brown, apple, chocolate, ...}, rather than reinforcing the action of eating for red apples and green apples, a feature-based RL agent would reinforce the red-eat, green-eat, and apple-eat state-action pairs. This would work well for the ability to generalize to yellow apples, having learned positive reinforcement for the apple-eat rule. However, such an agent could not learn about rule exceptions. For example, let us assume that the apple category has an exception, and brown apples in this world do not taste good (whereas other brown objects, such as chocolate, do taste good). Feature-based RL would fail to learn about the brown apple instance, or any such feature combination (e.g. the XOR problem).

Hand-coded models are brittle, lookup-based agents cannot generalize, and feature-based agents cannot learn exceptions. A brown apple is neither just brown, nor just an apple, nor just a brown apple. It is all of these things simultaneously, and any of these representations may be important for both learning and action-selection. Identifying the object as a brown-apple may be inefficient, and identifying it as just an apple may be misleading. Concurrent representation of all features combinations, a.k.a. chunks[1] or configural-cues (Gluck & Bower, 1988b; Wagner & Rescorla, 1972), would allow for learning of both generic rules (e.g. apples taste good) and exceptions to those rules (e.g. brown apples are spoiled).

Indeed, each set of perceptual features may potentially be recognized as all possible combinations of those features. Perceptual input $\{large, square, white\}$ may be represented as seven different states: $\{large\}, \{square\}, \{white\}, \{large, square\}, \{large, white\}, \{square, white\}$, and $\{large, square, white\}$. The problem with such representation is that too many memory chunks would be required in complex environments. A mere ten binary perceptual inputs (e.g. black vs white, large vs small) will require 59,048 chunks to be present in memory.[2] If each perceptual input allowed for five possible values (e.g. black, dark-gray, gray, light-gray, white), ten such input dimensions would result in almost ten million chunks. Twenty such perceptual dimensions would result in 95 trillion chunks. One hundred inputs with ten values per input would result in more chunks than there are atoms in the universe. The exponential growth of memory based on combinations of perceptual features is referred to as the state-space explosion problem, or the curse of dimensionality (Bellman, 1961).

To be clear, the problems with storing all possible chunk combinations, hand-coding models, or using lookup-based and feature-based agents are all well-known. These alternatives are presented here (1) to point out that ultimately we would like computational agents to learn generic situational rules, as well as exceptions to those situations, as well as exceptions to those exceptions, and so on, and (2) to highlight the difficulty with achieving this behavior. In practice, generalization for RL agents is done via one of many existing *function approximation* techniques.[3] These include neural networks, support vector machines, coarse coding (e.g. CMAC or tiling), decision trees, sparse distributed memory, radial basis function networks, and case-based reasoning (a.k.a instance-based or memory-based) methods (Kaelbling, Littman, & Moore, 1996; Sutton & Barto, 1998). Each of these methods provides advantages under specific conditions. Decision trees assume generic rules, and then gradually learn rule exceptions, but cannot learn in environments where feature-combinations, rather than features themselves, are predictive of performance (Kaelbling et al., 1996). Case-based reasoning methods (e.g. *k*-nearest neighbor) provide a way to account for all potential rules and exceptions, but their memory requirements approach those of lookup-based RL in persistent environments (Ratitch & Precup, 2004). Other methods greatly reduce memory demands, but require *a priori* knowledge about the task-environment (e.g. total number of rules needed to solve a task) (Kaelbling et al., 1996; Ratitch & Precup, 2004).

[1] We use the term chunk to refer to perceptual chunks, as is the case in EPAM/CHREST (Feigenbaum & Simon, 1984; Gobet et al., 2001). The term *chunk* has a slightly different use in the SOAR literature (Laird, 2012), referring to the creation of a production based on a resolved impasse — this is not the definition that we adopt in this paper.

[2] Given *n* features (e.g. $large, square, white$), we can create a chunk for every combination of feature presence and absence ($\{large\}, \{square\}, \{white\}, \{large, square\}, \{large, white\}, \{square, white\}$, and $\{large, square, white\}$). If we represent feature presence as a 1 and feature absence as a 0, we can represent each chunk as a binary number, and the total number of possible chunks is the total number of possible binary numbers, minus the blank chunk, which is $2^n - 1$. When each feature dimension can have two potential values, the total number of possible chunks is $3^n - 1$. With $k - 1$ possible values on *n* feature dimensions, we can have at most $k^n - 1$ possible chunks to represent all potential feature combinations.

[3] Function approximation has two purposes, dimensionality reduction and discretization of a continuous state-space. In this paper we assume a pre-discretized state-space. In the case of a continuos state-space, single-dimension function approximation (which is a much more tractable problem than multi-dimensional function approximation) may be done for each input dimension, and the conservative-rational mechanism proposed in this paper may be employed for dealing with the high dimensionality.