Observational Learning of Rules of Games

Debidatta Dwibedi
Y9227188
debidatt@iitk.ac.in

Advisors: Prof. Amitabha Mukerjee, Prof. K S Venkatesh

Dept. of Electrical Engineering
IIT Kanpur

June 2, 2014

Games and AI

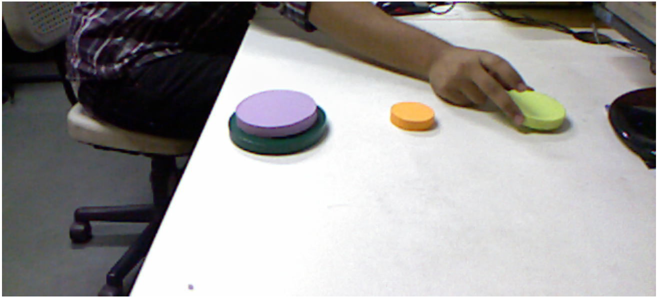
- Easy to model
  - discrete
  - deterministic
  - fully observable
- Computers have defeated the best players in many games

A General Game Playing System

- An AI program which plays more than one game successfully
- Devise strategies to win for multiple games
- Rules need to be provided in Game Description Language (GDL)
  - Game states as a series of facts
  - Game mechanics as logical rules
- Game learning system identifies states and discovers the rules

Problem Statement

Given a video of people playing a game, learn its rules.



Autonomous Game Learning

- Barbu et al.[2] learnt rules of Hexapawn and Tic-Tac-Toe
    - Fitted a 3×3 grid and use HSV values calculated at the time of calbration to decide game states
    - Use Inductive Logic Programming to learn rules
    - Lots of background knowledge like frame axioms, linearity test

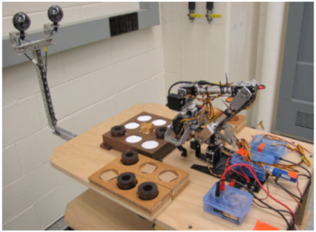


Figure: Setup of Game-playing robot(Barbu et al.)

Autonomous Game Learning

- Kaiser[6] learnt rules of two-player, grid-based games like Tic-Tac-Toe, Connect4
  - Canny edge detector and Hough transform to fit grid and decide occupied by X or O or unoccupied
  - Descriptive complexity to learn rules

Autonomous Game Learning

- Bjornsson [3] learns rules of simplified board games
  - Movement pattens of pieces by looking at change in coordinates
  - Prefix tree automata(PTA) from observation converted to minimum state deterministic finite automata(DFA).

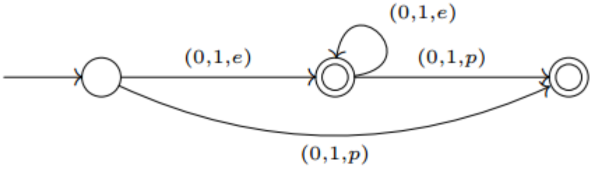


Figure: Example of a Rule in DFA for a piece that only moves forward

Objective

Objective

Use RGBD images to minimize game-specific background knowledge to an ILP to learn rules of games

Proposed Approach

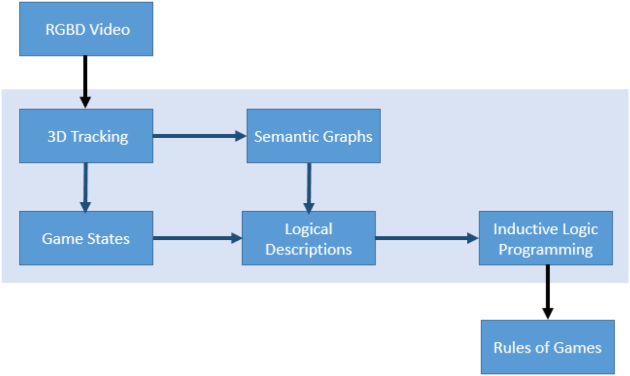


Figure: Pipeline of Proposed Algorithm

Point Clouds

- Collection of points in 3D
- RGB from a normal camera
- Depth from infrared laser

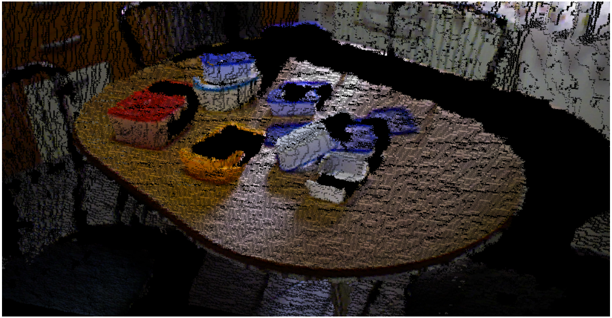


Figure: A sample point cloud from http://pointclouds.org

Game Piece Segmentation

Figure: Extract all game pieces as separate clusters.

- Difficult to segment from images
- Separated in space in point clouds

Euclidean Clustering of Game Pieces

- Cluster points that are close to each other in 3D space
- Model free detection of game pieces
- We assume that game pieces have either perceptually different colours or are separated in space
- Remove hand using HSV filter

Segmentation Results

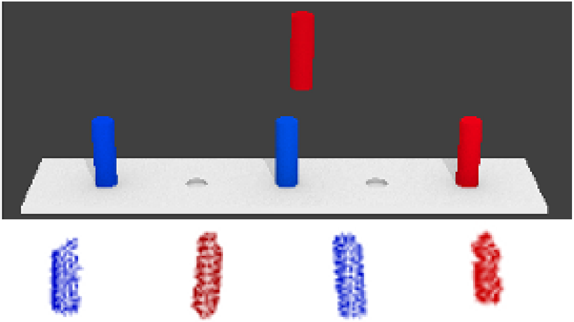


Figure: 4 game pieces in 1D Peg Solitaire

Segmentation Results

Figure: 4 game pieces in Animated Towers of Hanoi

Segmentation Results

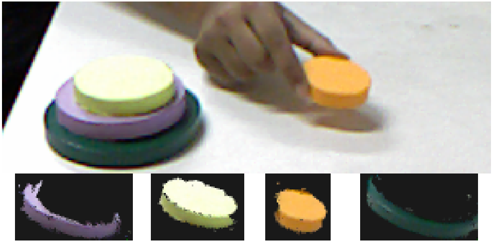


Figure: 4 game pieces in Real Towers of Hanoi

Problems in Multi-object Tracking

- Can't track by detection
- Occlusion by hand and other objects
- Centroid tracking might fail if more than one object is moving simultaneously and their paths cross

Multi-object Tracking = Assignment Problem

- Multi-object tracking problem is equivalent to assignment of labels from one frame to the next

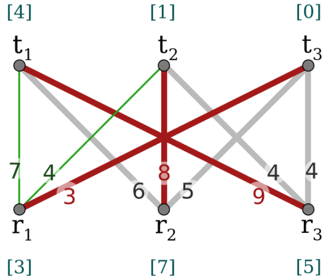


Figure: The Assignment Problem. Taken from http://www.frc.ri.cmu.edu/ lantao/

Octree Overlap

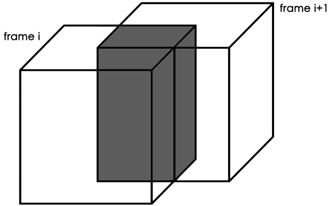
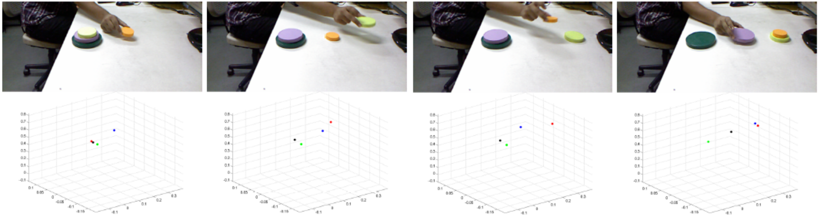


Figure: Overlap between an object in one frame and the same object in the next frame.

- Use octree overlap between same object in consecutive frames as a metric in he assignment problem
- Maximize sum of overlaps during assignment

Multi-object Tracking Results



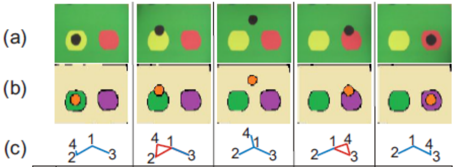
(a) Tracked Objects in Frame 1 (b) Tracked Objects in Frame 75 (c) Tracked Objects in Frame 150 (d) Tracked Objects in Frame 250

Figure: Tracking game pieces in Real Towers of Hanoi

Semantic Graphs to Represent Scenes

- Graph to represent scene(Aksoy et al.)[1](Yang et al.)[7]
- Objects as nodes
- Relationship between each other as edges

Semantic Graphs to Represent Scenes



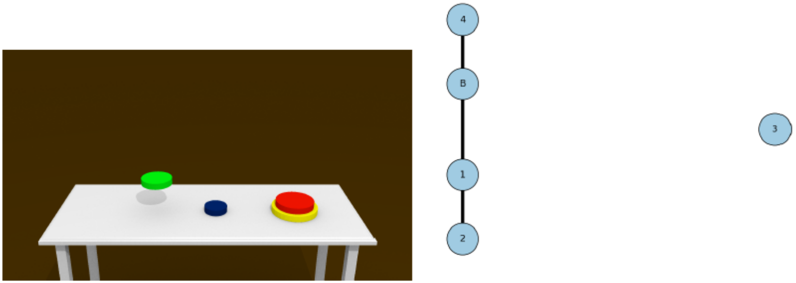
(d)

$\rho_{2,1}$	1	2	1	1	1
$\rho_{3,1}$	1	1	1	2	1
$\rho_{4,1}$	0	2	1	2	0
$\rho_{3,2}$	0	0	0	0	0
$\rho_{4,2}$	1	2	0	0	0
$\rho_{4,3}$	0	0	0	2	1

ξ_o^1

Figure: Semantic Event Chain from Aksoy et al.[1]

Semantic Graphs of Game Scenes

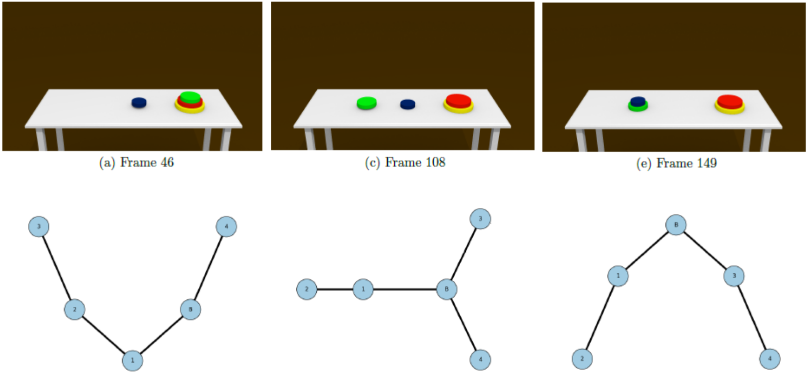


(a) Frame 93 from the animated *Towers of Hanoi* dataset

(b) Corresponding Semantic Graph for *contact*

Figure: B - Board, 1 - Yellow, 2 - Red, 3 - Green, 4 - Blue

Unsupervised Discovery of Game States



(a) Frame 46 (c) Frame 108 (e) Frame 149

(b) Corresponding Semantic Graph (d) Corresponding Semantic Graph (f) Corresponding Semantic Graph

Figure: B - Board, 1 - Yellow, 2 - Red, 3 - Green, 4 - Blue

Discover Valid Locations for Game Pieces

- SVD on positions of game pieces during game states
- Significant number of eigen vectors gives number of dimensions game is being played in
- Transform from the camera to the board

Discover Valid Locations for Game Pieces

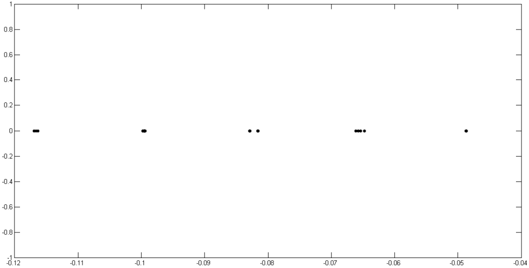
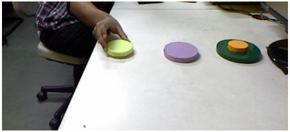


Figure: Clusters in positions in 1D Peg Solitaire

Discover Valid Locations for Game Pieces



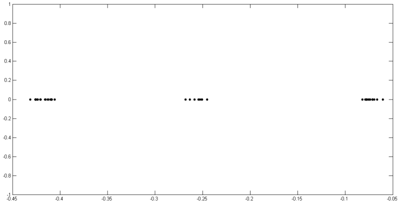


Figure: Clusters in positions in Real Towers of Hanoi

Discover Valid Locations

- Do k-Means by varying k
- Look for elbow in Sum Squared Error graph

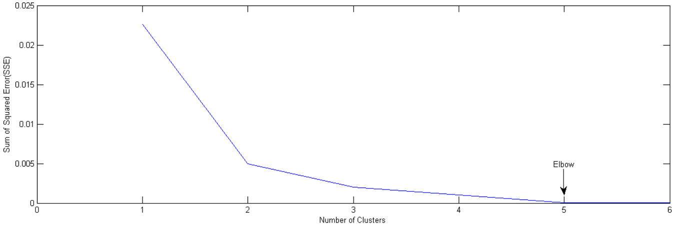


Figure: SSE v/s k for 1D Peg Solitaire

Discover Valid Locations

- Do k-Means by varying k
- Look for elbow in Sum Squared Error graph

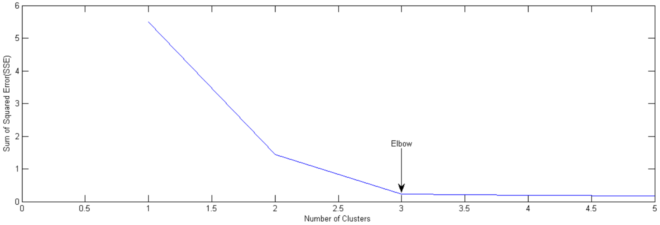


Figure: SSE v/s k for Real Towers of Hanoi

Respresenting Game States

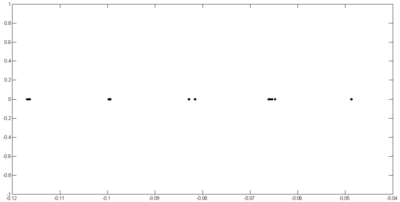


Figure: Game State:[{a},{b},{},{d},{e}]

Representing Game States

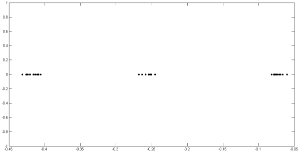


Figure: Game State:[{c},{b},{a,d}]

Inductive Reasoning

- Induce general rules by finding patterns from detailed facts
- Rules might not be always true but explain the observations
- Towers of Hanoi example:
  - Smaller piece is always being kept on bigger
- Learning rules of games can also be modeled using induction reasoning

Inductive Logic Programming(ILP)

- A logic program that comes up with hypotheses given:
  - background knowledge
  - positive example
  - negative example

Inductive Logic Programming(ILP)

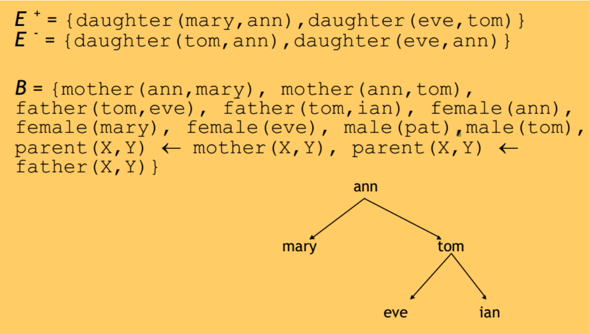


Figure: Slide from http://www-ai.ijs.si/SasoDzeroski/

- `daughter(X,Y) :- female(X), parent(Y,X)`

From Semantic Graphs to Logical Clauses

- Rule learning system takes input in form of logical clauses
- Three kinds of clauses:
  - Attributes of game pieces
  - Relationship between game pieces
  - Movement of game pieces

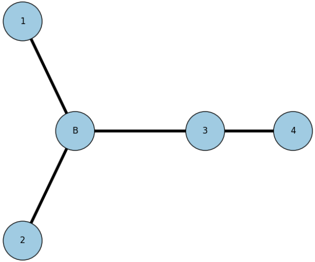
From Semantic Graphs to Logical Clauses

- Rule learning system takes input in form of logical clauses
- Three kinds of clauses:
  - Attributes of game pieces - from visual classifiers
  - Relationship between game pieces - from visual classifiers
  - Movement of game pieces - from game state changes

Attribute based Clauses

- `colour(a,orange).`
- `colour(d,green).`
- `size(a,10).`
- `size(d,2).`
- Working on shape classifiers for game pieces

Relationship based Clauses



- Relationship between objects in Semantic Graph converted to logical clauses
- `contact(a,d). contact(d,a).`
- `on(d,a).`

Move based Clauses

- Move from one valid location to another
  - `move(d,l1,l2).`
  - Piece *d* moves from location *l*1 to *l*2
- Effect of move on the location clusters
  - `fromto(d,[a,d],[c,d]).`
  - Piece *d* moves from the cluster whose previous state was `{a,d}` and the cluster where it moved is `{c,d}` after the move.

Meta-clause/Rule Generator

- Logical system should be able to compare numerical features like size or height
- Meta-clause generator creates comparators for any numerical feature
  - `greatersize(A,B) :- piece(A),piece(B), size(A,NA),size(B,NB), NA>NB.`
  - `greaterheight(A,B) :- piece(A),piece(B), height(A,NA),height(B,NB), NA>NB.`

Background Knowledge

- Attribute based clauses
- Meta-clause generated rules
- Rules built in system:
  - *diff* - difference between two locations
  - *absdiff* - absolute difference between two locations
  - *top* - top element of a stack
  - *bottom* - bottom element of a stack

Positive examples

- Relationship based clauses
- Move based rules

1D Peg Solitaire

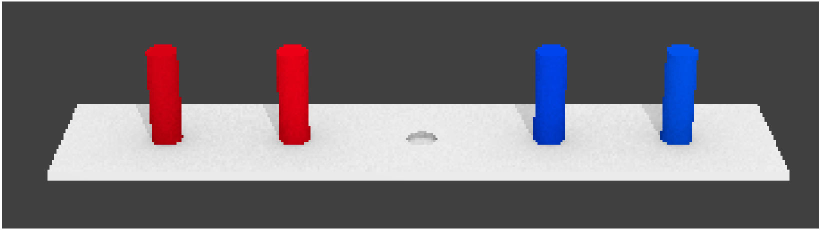


Figure: Start State

1D Peg Solitaire

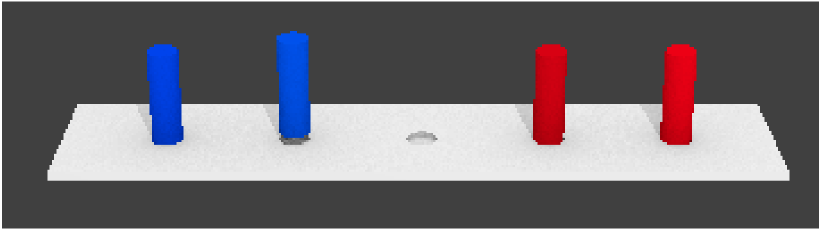


Figure: End State

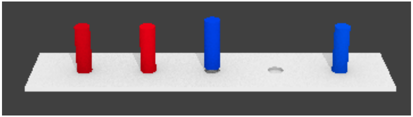
Rules of 1D Peg Solitaire

- Red pegs can take one step or two steps to the right into an empty hole.
- Blue pegs can take one step or two steps to the left into an empty hole.

Clauses Generated

- `piece(p1).`
- `piece(p2).`
- `piece(p3).`
- `piece(p4).`
- `color(p1,red).`
- `color(p2,red).`
- `colour(p3,blue).`
- `color(p4,blue).`

Clauses Generated



- `move(p3,l4,l3).`
- `fromto(p3,[p3],[p3]).`

Rules Learnt

- move(A,B,C) means piece *A* moves from *B* to *C*
- `move(A,B,C) :- diff(B,C,-2), color(A,blue).`
- `move(A,B,C) :- diff(B,C,-1), color(A,blue).`
- `move(A,B,C) :- diff(B,C,1), color(A,red).`
- `move(A,B,C) :- diff(B,C,2), color(A,red).`
- `fromto(A,[A],[A]).`

Towers of Hanoi

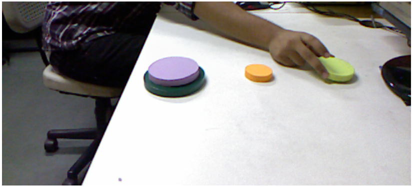
Figure: Start State

Figure: End State

Rules of Towers of Hanoi

- Only one disk can be moved at a time.
- Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack
- No disk may be placed on top of a smaller disk.

Clauses Generated



- `move(d,l2,l3).`
- `fromto(d,[d],[c,d]).`

Rules Learnt

- `on(A,B) :- greatersize(B,A).`
- `fromto(A,B,C) :- top(A,B), top(A,C).`
- fromto(A,B,C) means piece A moves from the stack B to the stack C

More Complex Games

- Modified Towers of Hanoi
  - `on(A,B) :- colour(B,red),colour(A,green), greatersize(B,A).`
- Chess with hundreds of manually written clauses
  - `move(bishop,pos(A,B),pos(C,D)) :- xdiff(B,D,E), ydiff(A,C,E).`
  - `move(knight,pos(A,B),pos(C,D)) :- xdiff(B,D,1), ydiff(A,C,2).`
  - `move(knight,pos(A,B),pos(C,D)) :- xdiff(B,D,2), ydiff(A,C,1).`
- ILP is powerful in inducing rules of games if the clauses encode the rule being followed

Conclusion

- Successfully learnt rules of two games with very different rules
- Semantic Graphs from point clouds
- Unsupervised discovery of game clusters and representation of game states

Future Work

- Convert rules, start and end states to GDL to play games
- 3d spatial puzzles and assembly
- Ground natural language descriptions in semantic graphs

References I

Eren Erdal Aksoy, Alexey Abramov, Johannes Dörr, Kejun Ning, Babette Dellen, and Florentin Wörgötter.
Learning the semantics of object–action relations by observation.
The International Journal of Robotics Research, 30(10):1229–1249, 2011.

Andrei Barbu, Siddharth Narayanaswamy, and Jeffrey Mark Siskind.
Learning physically-instantiated game play through visual observation.
In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 1879–1886. IEEE, 2010.

Yngvi Björnsson.
Learning rules of simplified boardgames by observing.
In *ECAI*, pages 175–180, 2012.

References II

Neil Dantam, Irfan Essa, and Mike Stilman.
Linguistic transfer of human assembly tasks to robots.
In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 237–242. IEEE, 2012.

Shyamanta M Hazarika and Alexy Bhowmick.
Learning rules of a card game from video.
Artificial Intelligence Review, 38(1):55–65, 2012.

Łukasz Kaiser.
Learning games from videos guided by descriptive complexity.
In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

Yezhou Yang, Cornelia Fermuller, and Yiannis Aloimonos.
Detection of manipulation action consequences (mac).
In *CVPR 2013*, 2013.

Thank You

How good are computers at playing games?

- Chinook beat Grandmaster Don Lafferty in Checkers (1995)
- Deep Blue defeated Kasparov in Chess (1997)
- Quackle beat Scrabble world champion David Boys (2010)
- MoGoTW defeated Go player Catalin Taranu (2010)
- Watson defeat previous winners on Jeopardy! (2011)

Octree

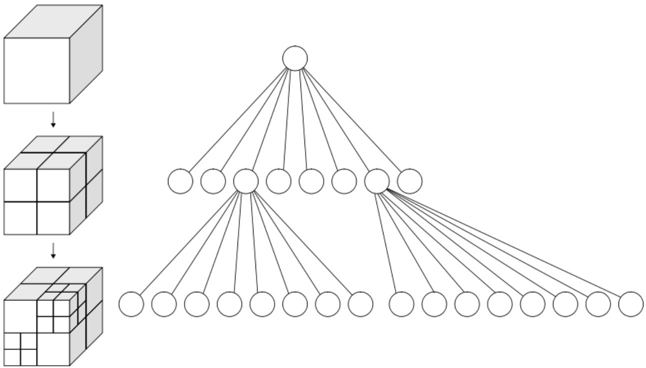


Figure: Octree from http://wikipedia.org

Autonomous Game Learning

- Hazarika and Bhowmick[5] learn rules of card games from videos
    - Visual features to classify cards
    - Decision trees to learn rules