CS 440: Introduction to Artificial Intelligence

Lecture 12

Matthew Stone

October 14, 2015

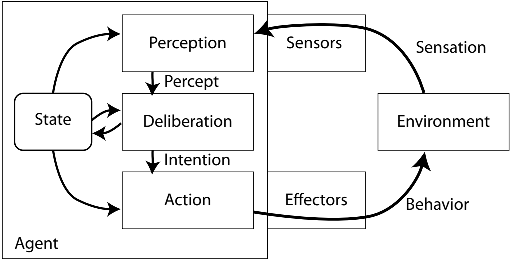
Recap— Search

- Initial state
- Possible actions in each state
- Transition model:
  Takes state and action and gives new state
- Goal test
  Describes whether state is what you want
- Path cost
  Says how easy or hard action sequence is

Search— Approximate Techniques

- May be better to give up optimality
- Settle for solution that's "good enough"
- Hill climbing
- Simulated annealing
- Genetic search

Search in Context



Fundamental Challenge

Applying knowledge in new situations

- Actual situations are *complex*.
  You usually haven't seen exactly the same thing before.
  Often: a novel mix of familiar features.
- Useful knowledge needs to be *generalizable*
  It describes large classes of situations
  in terms of underlying features.
- Consequence:
  New situations require a creative synthesis of existing knowledge

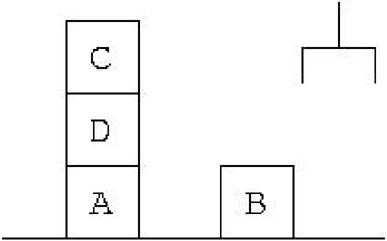
Search in Model-Based Agents

Relevant in all parts of agent architecture

- Planning
- Perception
- Learning

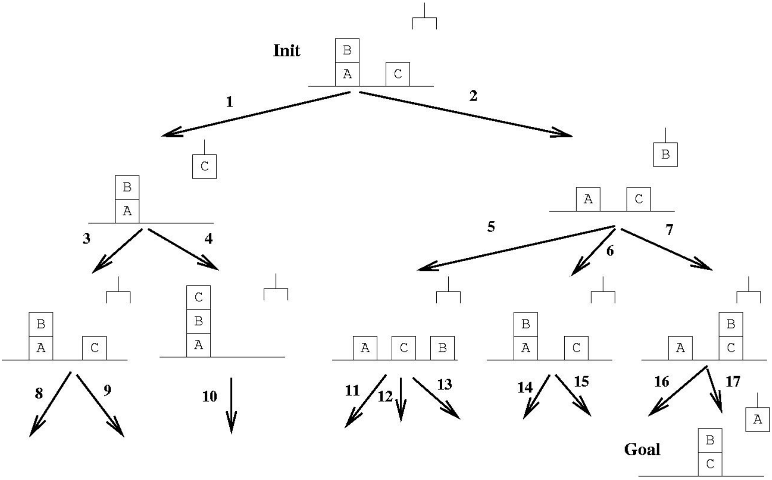
Planning as Search

Blocks world



from Manfred Kerber, `http://www.cs.bham.ac.uk/~mmk/Teaching/AI/l8.html`

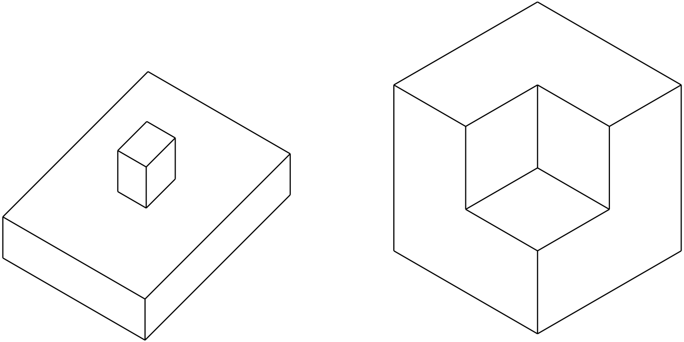
Planning as Search



from Manfred Kerber, `http://www.cs.bham.ac.uk/~mmk/Teaching/AI/l8.html`

Perception as Search

Understanding Line Drawings

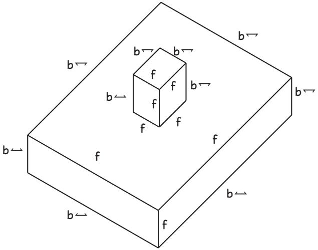


from Peter Varley's thesis `http://ralph.cs.cf.ac.uk/Data/Sketch.html`

Classifying Edges

Edges can be:

- folds where two surfaces come together
- blades where surface turns away
- visible surface can be on either side of blade



Formal Search Problem

Assign labels to each edge: Analogous to map coloring

- State: consistent partial line labeling
- Transition: label the next line
- Goal: all lines are labeled

Consistency means junctions where edges meet make sense in terms of a 3D object.

Understanding Learning

Items are always described with features

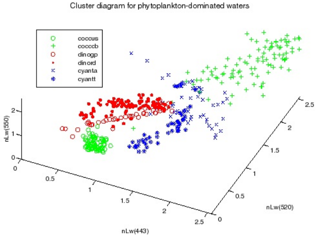
- Item is like a vector
- Each position in the vector represents a feature
- Feature value is given by element at that position

Example in spam domain:

$$\begin{array}{c} \\ \left[\begin{array}{cccccc} \text{madam?} & \text{promotion?} & \text{quality?} & \text{lisp?} & \text{paul?} & \text{ff0000?} \\ \text{YES} & \text{YES} & \text{YES} & \text{NO} & \text{NO} & \text{NO} \end{array} \right] \end{array}$$

Feature space

You can therefore visualize items as points in N-D space

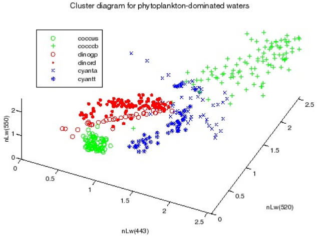


This is actually ocean color data during plankton blooms.
Data from Linda Traykovski, Woods Hole

Plot items together to get overview of system's experience

Feature space

You can therefore visualize items as points in N-D space

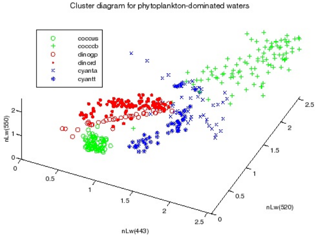


This is actually ocean color data during plankton blooms.
Data from Linda Traykovski, Woods Hole

Measure distance or similarity between points in feature space

Feature space

You can therefore visualize items as points in N-D space



This is actually ocean color data during plankton blooms.
Data from Linda Traykovski, Woods Hole

Expect nearby or similar points to have the same category

Decision boundary

A classifier assigns categories to regions of feature space

- Algorithms differ in shape of region
- More complex shapes can fit data more accurately
- However they require more data to do so

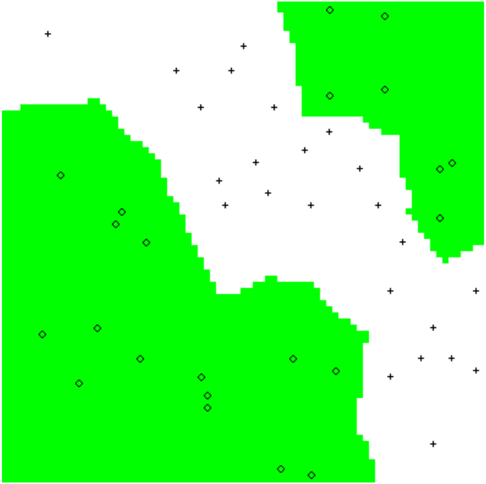
Demo: `http://www.cs.cmu.edu/~zhuxj/courseproject/knndemo/KNN.html`

Simplest Learner

Nearest neighbor

- Given
  - A collection of data points D
    Represented as vectors in n-dimensional space
    Labeled with categories
  - A similarity measure M in n-dimensional space
    e.g., Euclidean distance, cosine similarity etc
  - A new vector v in n-dimensional space
    not labeled with a category
- Find the $d \in D$ that is closest to v by M
- Predict that the category of v is the category of d.

Visualization



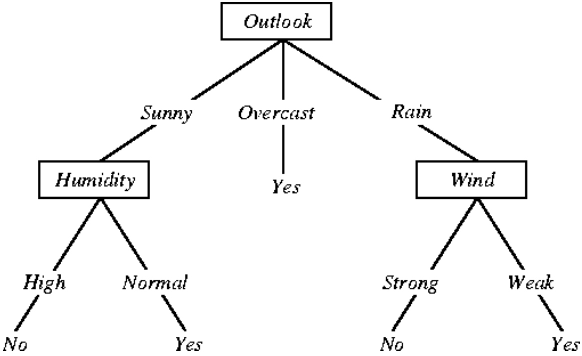
Learning as Search

Building decision trees

- Start with training data
  Set of objects with labeled with attributes and category
- Want to build a rule mapping attributes to category
  Agrees with the training data
  Hopefully generalizes to new instances
- Represent the rule as a tree
  Leaves say what class to predict
  Internal nodes do test and make conditional predictions

See `http://www.eecs.wsu.edu/~cook/dm/lectures/l4/mlbook.html`

Example: Predict if tennis will be fun today



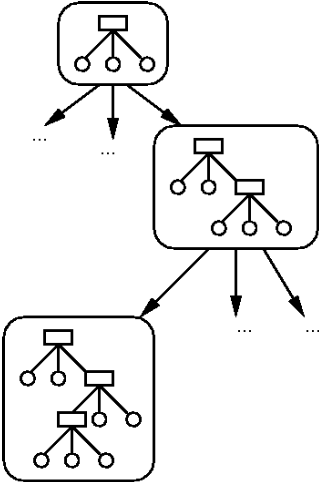
See `http://www.eecs.wsu.edu/~cook/dm/lectures/l4/mlbook.html`

Formal search problem

Construct a good tree top-down

- State: Decision tree
- Transition: Replace a leaf by a one-layer tree
  Do new test, predict based on result
- Goal: good prediction on training data
- Cost: size of the tree

Schematic view of learning search space



See `http://www.eecs.wsu.edu/~cook/dm/lectures/l4/mlbook.html`