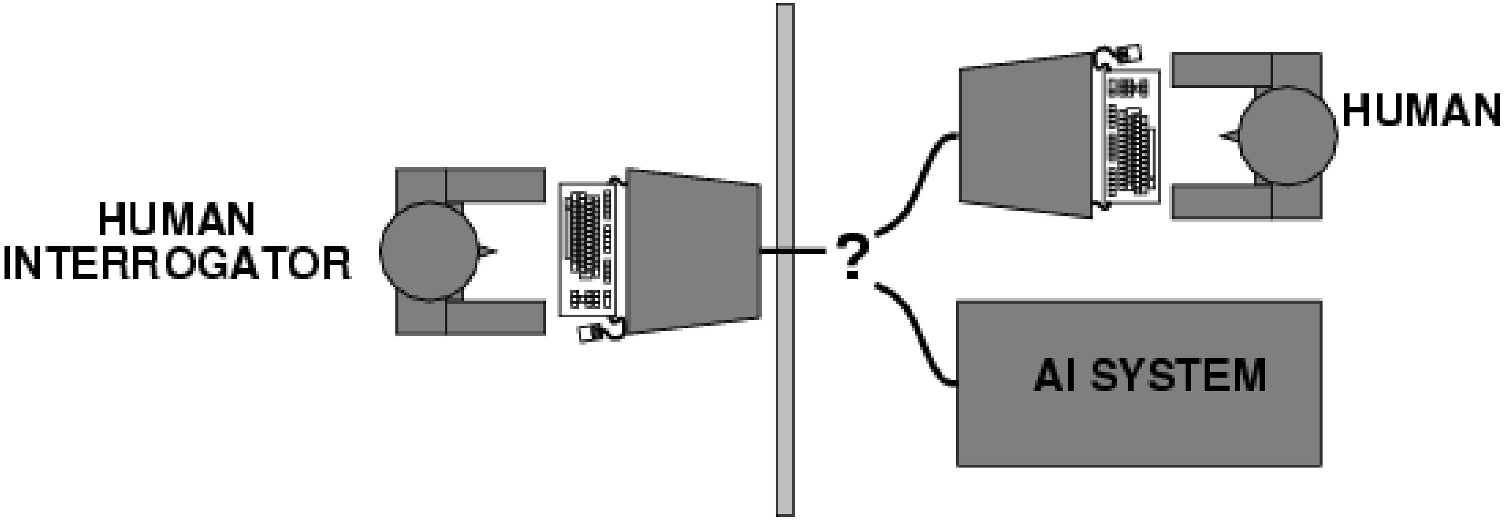
Artificial Intelligence CS365

Amitabha Mukerjee

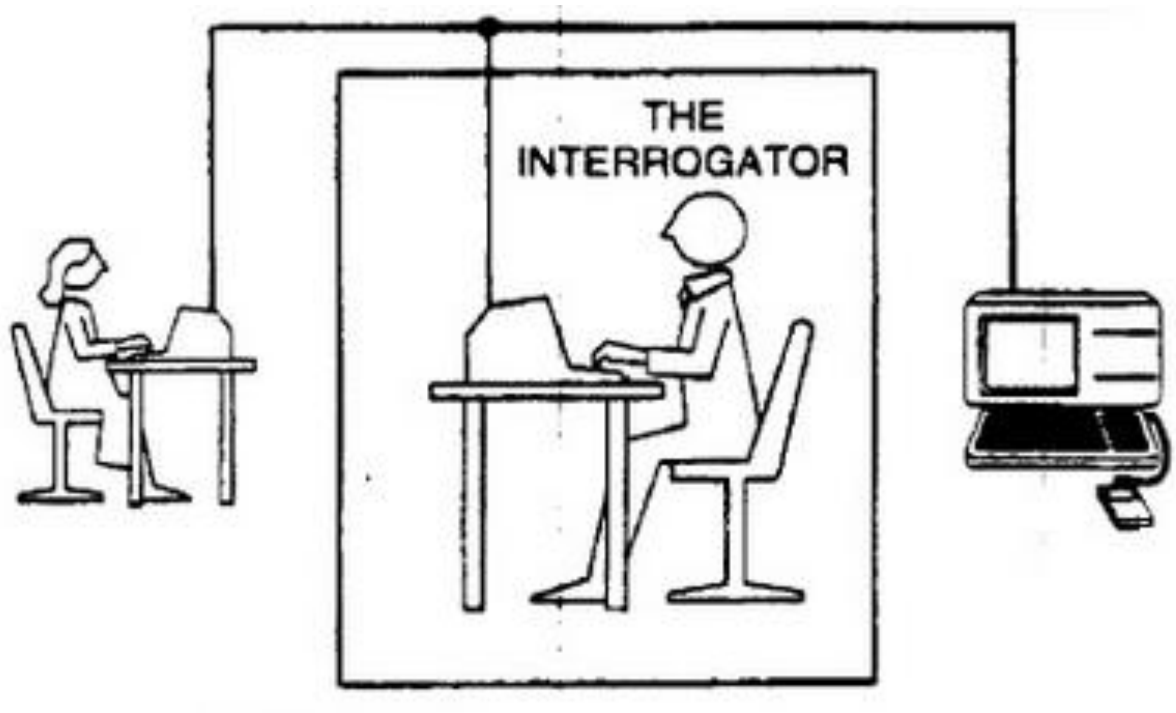
What is intelligence?

Acting humanly: Turing Test

- Turing (1950) "Computing machinery and intelligence":
  - "Can machines think?"
- Imitation Game



Acting humanly: Turing Test



What is AI?

four views:

Think like a human	Think rationally
Act like a human	Act rationally

Are humans rational?

Perception

Kanisza triangle

Thinking rationally: "laws of thought"

- Aristotle: what are correct arguments/thought processes?
- Greek philosphers: forms of *logic*:
  3-step *syllogism*
- Indian philosophy: 5-step inference
- Problem:
  - Most intelligent behavior does not rely on logical deliberation

Thinking rationally: Boolean vs Probabilistic

- Q. Do we think in terms of True/False ?
  - e.g. what concepts have sharply defined boundaries?

- Deterministic vs. Probabilistic problems

- Are real-life problems deterministic

Subject matter in AI

- Get machines to do what humans do but machines can’t
  - AI: The study of how to make computers do things at which, at the moment, people are better.
    \- Rich and Knight, 1991

Problems in AI

Recognition

images: 100 x 100 pixels

Ack: A. Efros, original images from hormel corp.

Structured data

Features already extracted as Data + tags;
(Relational Databases)

e.g. Movie Preference matrix (Netflix)

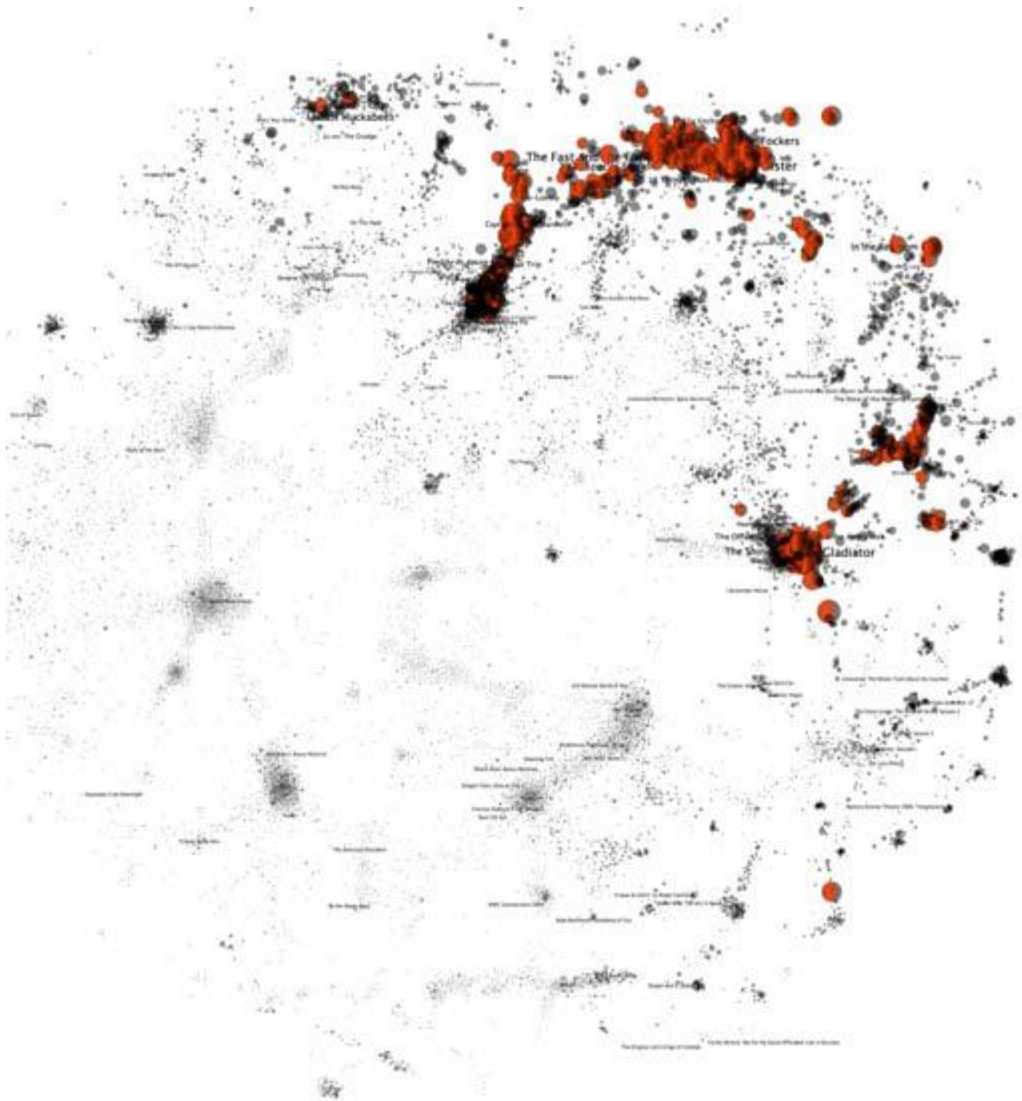
99 mn movie ratings

18K movies x 500K clients

e.g. facebook event logs – terabytes / day

- unstructured data (text / images) >> relational data

Netflix Movie model



Unstructured data

Text: Newspapers, blogs, technical papers

Images: ImageNet, LFW

Q. What are the objects and their relations?

Video : Hollywood2, UCF sports;

Q. What is the action? Who are the agents?

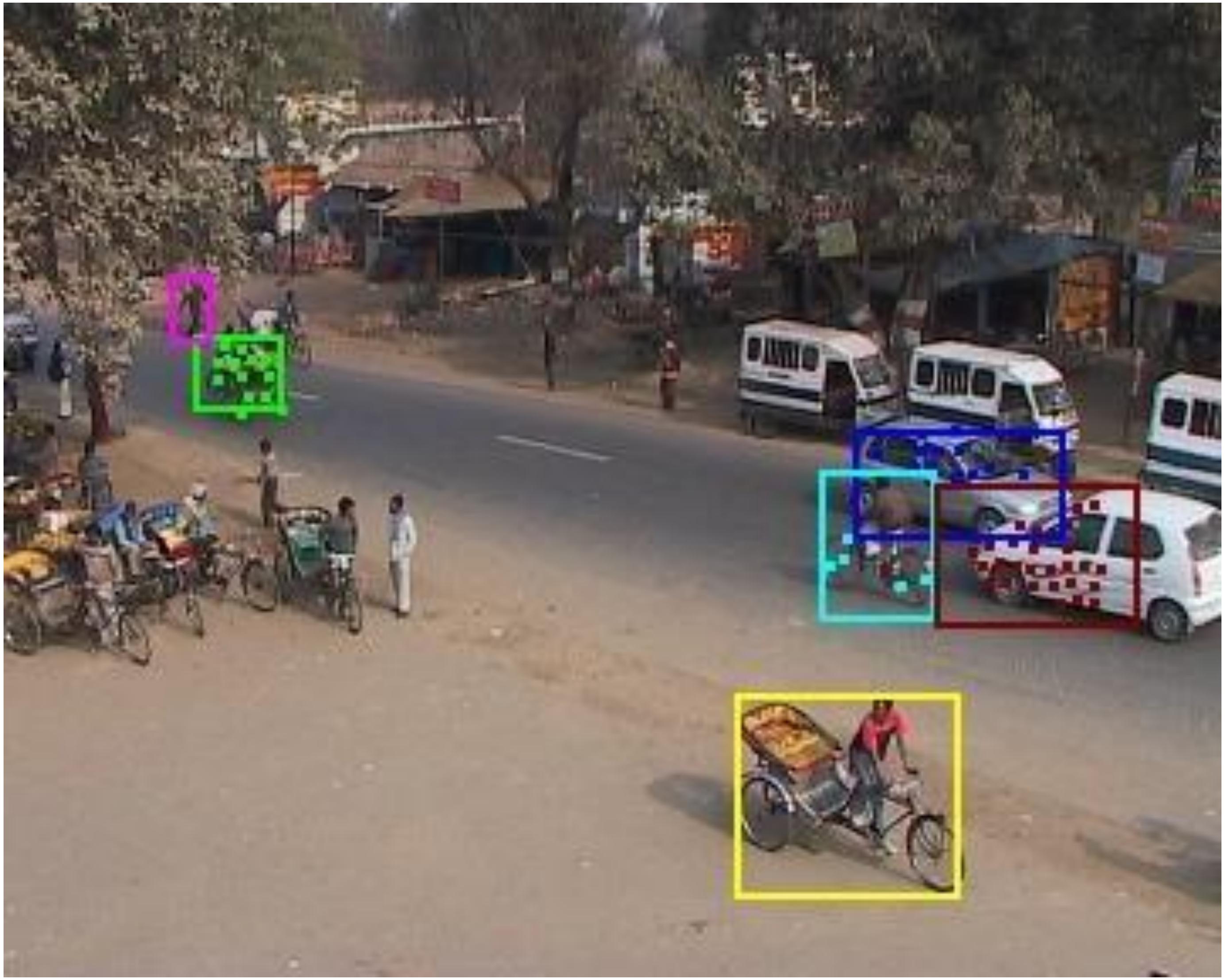
Multimedia : Audio + Video;

Label + image + preferences

Example : Face Recognition

which features to use?

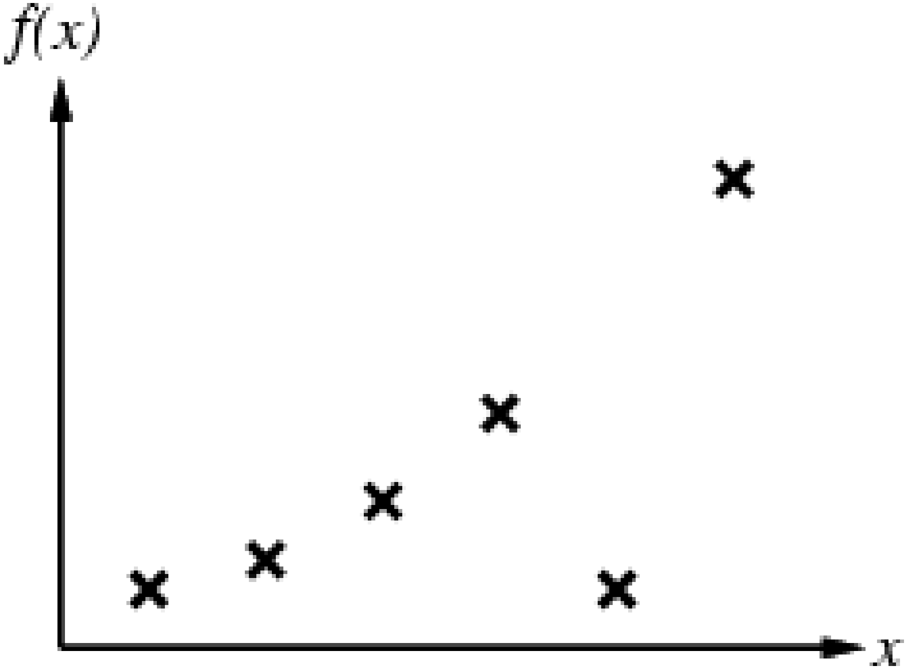
Events in Video



Mukerjee Satish and Guha 07

Constructing a model

- Construct hypothesis *h()* to agree with data f(x)
- (*h* is consistent if it agrees with *f* on all examples)
- E.g., [feature space : often very high-dimensional]

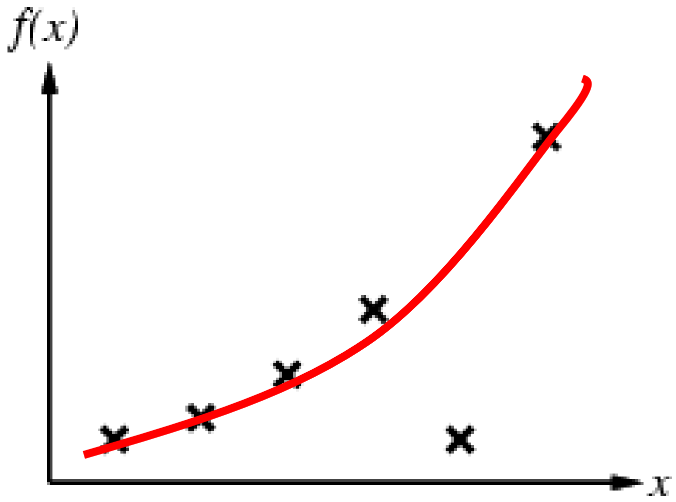


Regression vs Classification

$y = f(x)$

Regression:

y is continuous

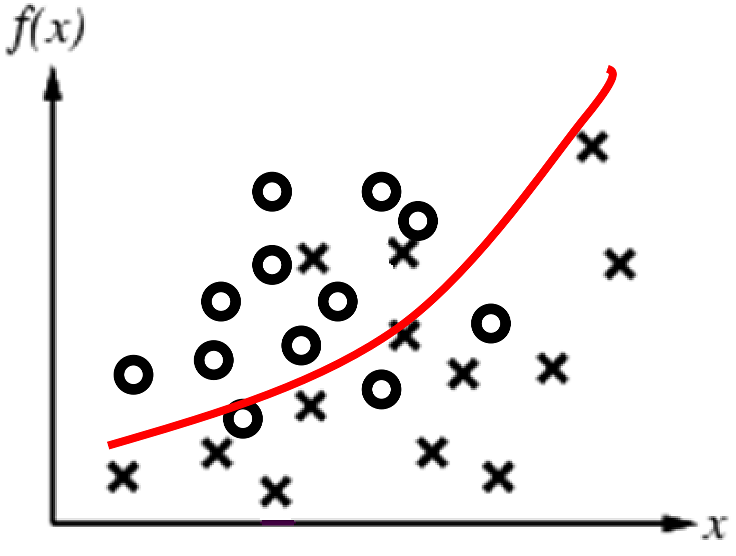


Classification:

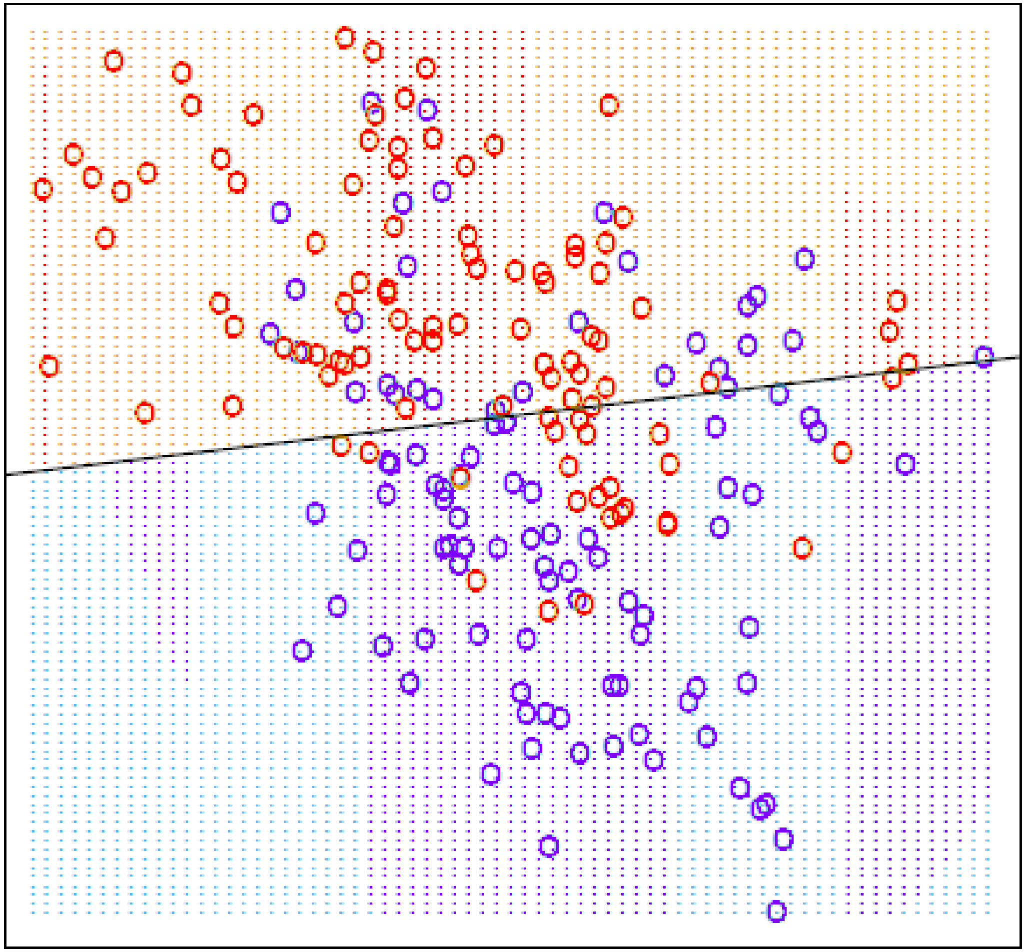
y : set of discrete values

e.g. classes C_1, C_2, C_3...

$y \in \{1,2,3...\}$



2-class (binary) classification



[hastie tibshirani 2009]: elements of statistical learning

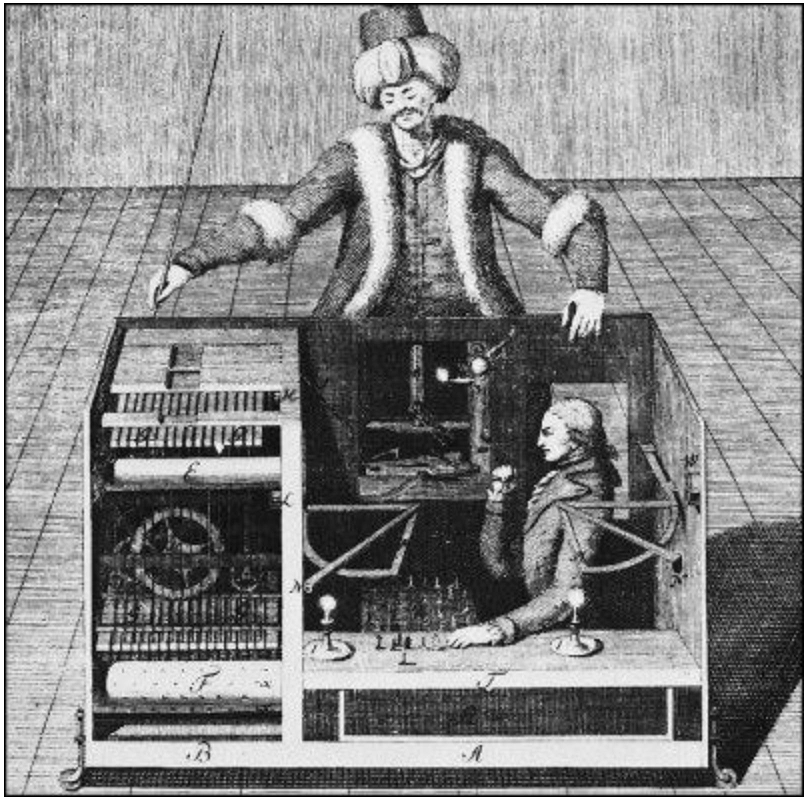
AI history

Timeline : Prehistory / Early AI

- Pre-history: Pascal, Leibniz

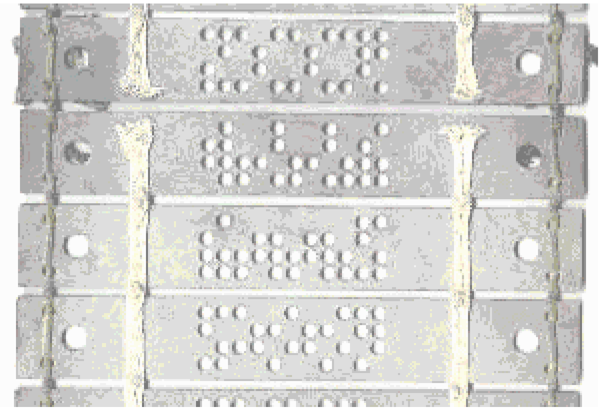
  hoaxes

  Babbage
- 1943 McCulloch & Pitts: Boolean circuit model of neuron
- 1950 Turing's "Computing Machinery and Intelligence“
- 1956 Dartmouth meeting: “Artificial Intelligence“ name

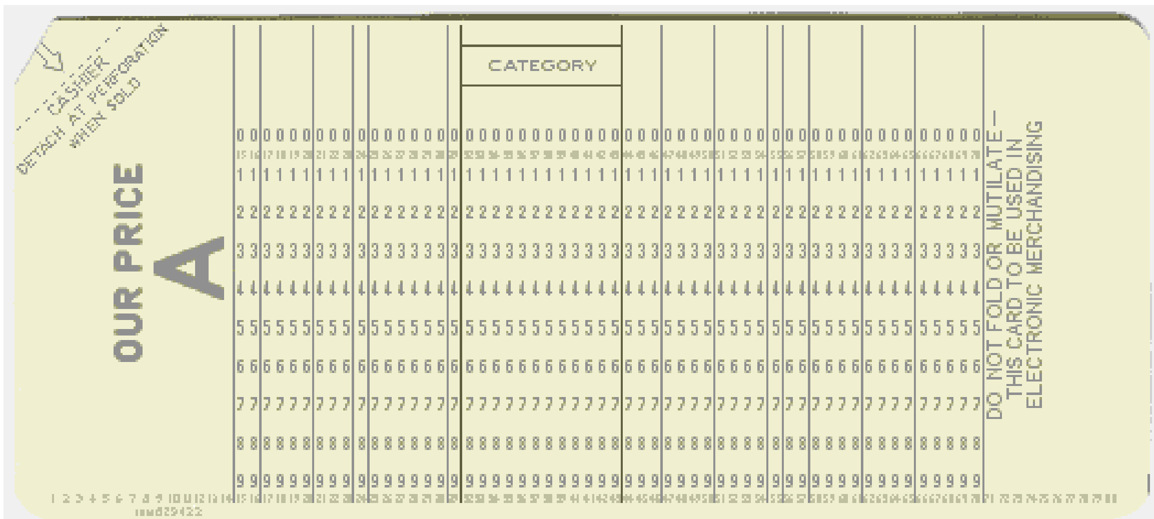


von kempelen’s chess-playing turk, 1769 (hoax)

Timeline : Prehistory / Early AI



- Punched cards for weaving looms (1805)



- Hollerith Punched Cards (IBM) (upto 1990s)

1955: coining the name "Artificial Intelligence"

John McCarthy,
Marvin Minsky,
N Rochester, and
Claude Shannon:
(1955) :

"the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it."

A PROPOSAL FOR THE DARTMOUTH SUMMER RESEARCH PROJECT ON ARTIFICIAL INTELLIGENCE

J. McCarthy, Dartmouth College
M. L. Minsky, Harvard University
N. Rochester, I.B.M. Corporation
C.E. Shannon, Bell Telephone Laboratories

August 31, 1955

We propose that a 2 month, 10 man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer.

“Artificial Intelligence”

- artificial :

  artifice *ars* (method, technique) + *facere* (to do)

  → man made (< artifice)

- intelligence :

  inter- (between) + *legere* (to gather, choose, read)

  [legend = things to be read]

Timeline : AI – Logical Models

- 1943 McCulloch & Pitts: Boolean circuit model of brain
- 1950 Turing's "Computing Machinery and Intelligence"
- 1956 Dartmouth meeting: "Artificial Intelligence" name
- 1956 Newell & Simon's Logic Theorist,
- 1959 Samuel's checkers program: learned by playing itself

1956 : Logic Theorist

Herbert Simon
&
Alan Newell:

The Logic Theorist 1956

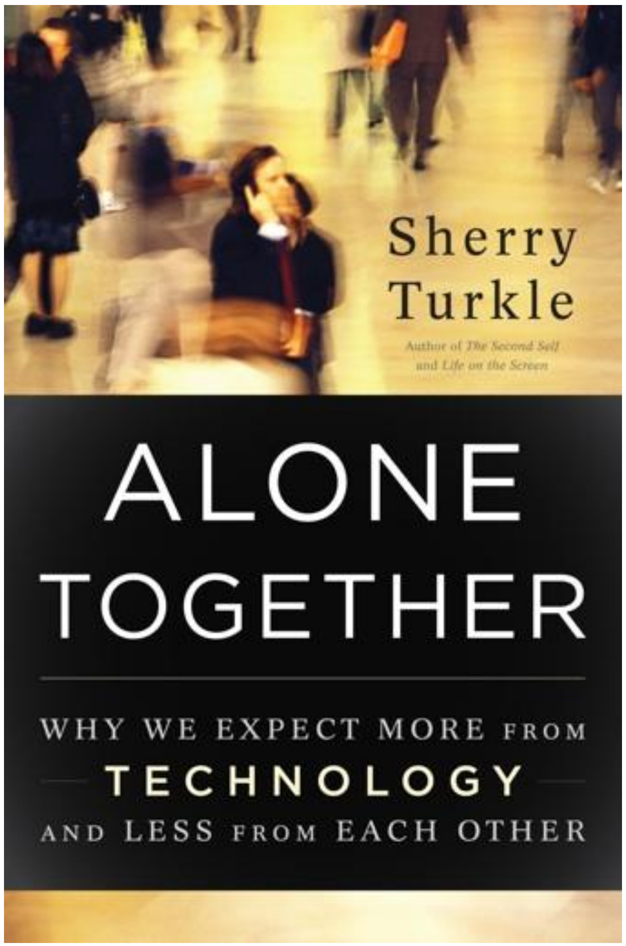
proved 38 of 52 theorems
in ch. 2
Principia Mathematica.
co-author of journal
submission based on a
more elegant proof.
paper was rejected..

Timeline : AI – Logical Models

- 1943 McCulloch & Pitts: Boolean circuit model of brain
- 1950 Turing's "Computing Machinery and Intelligence"
- 1956 Dartmouth meeting: "Artificial Intelligence" name
- 1956 Newell & Simon's Logic Theorist

- 1959 Samuel's checkers program: game search; later version learned by playing itself
- 1964-66 ELIZA (psychotherapist) by Joseph Weizenbaum

1966 : ELIZA (Social)



My first brush with a computer program that offered **companionship** was in the mid-1970s. I was among MIT students using Joseph Weizenbaum's ELIZA, a program that engaged in dialogue in the style of a psychotherapist …

Weizenbaum's students **knew that the program did not understand**; nevertheless, they wanted to chat with it. ... they wanted to be alone with it. They wanted to **tell it their secrets**.

- Sherry Turkle, MIT Sociologist

Timeline : AI – Logical Models

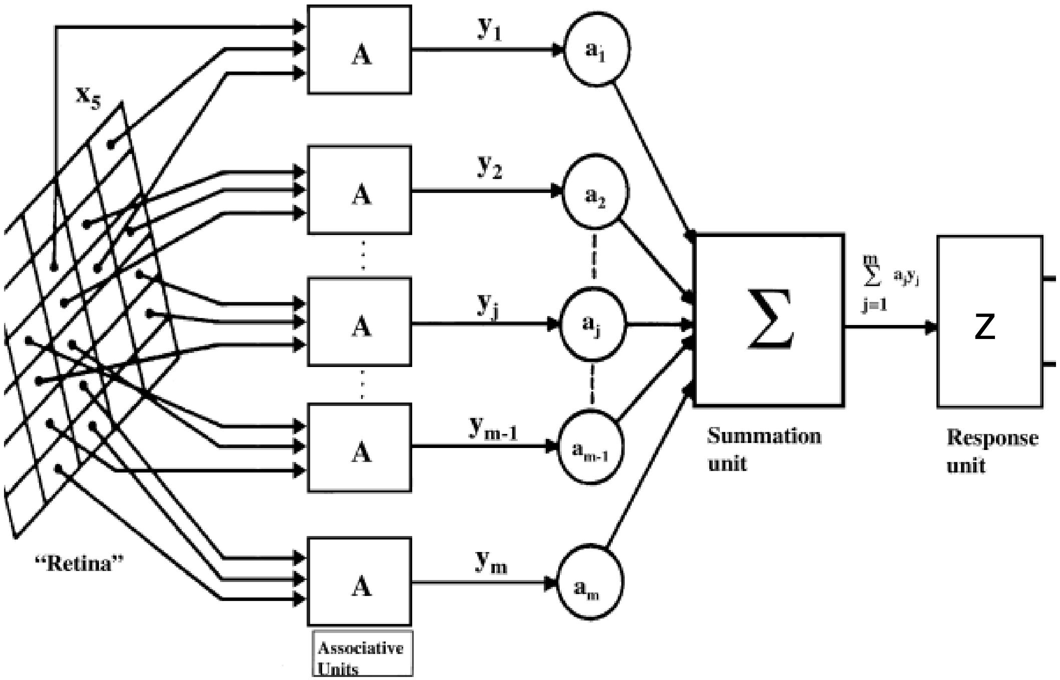
- 1943 McCulloch & Pitts: Boolean circuit model of brain
- 1950 Turing's "Computing Machinery and Intelligence"
- 1956 Dartmouth meeting: "Artificial Intelligence" name
- 1956 Newell & Simon's Logic Theorist

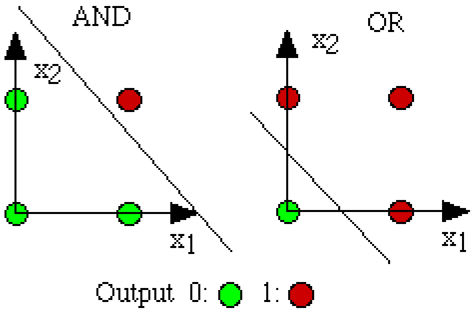
- 1959 Samuel's checkers program: game search; later version learned by playing itself
- 1964-66 ELIZA (psychotherapist) by Joseph Weizenbaum

- 1965 Robinson's resolution algorithm for first order logic

- 1969 Minsky / Papert's *Perceptron*
- 1970-1975 Neural network research almost disappears; [*sociology of science* study]
- 1966-72 Shakey the robot
- 1969-79 Early knowledge-based systems (expert systems)

1958: Rosenblatt - Perceptrons





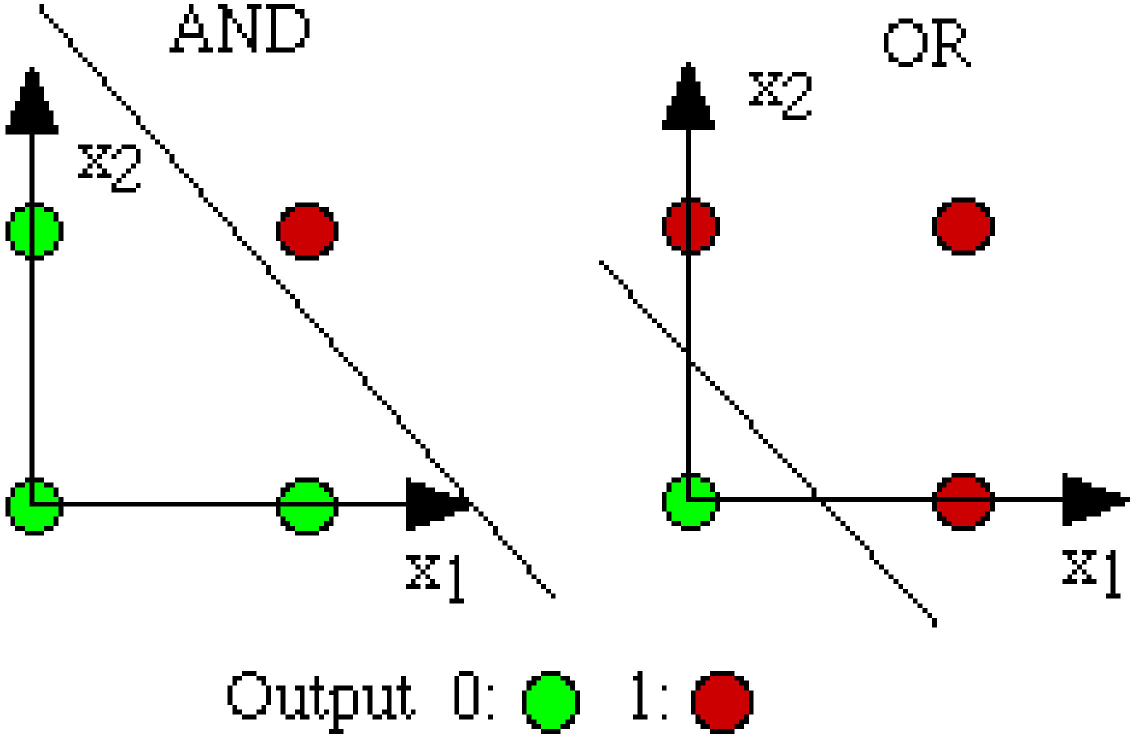
if $\sum > \theta$, *response* $z = 1$, *else zero*

$\Delta\theta = -(t - z)$ [t = correct response]

$\Delta w_i = -(t - z)\, y_i$

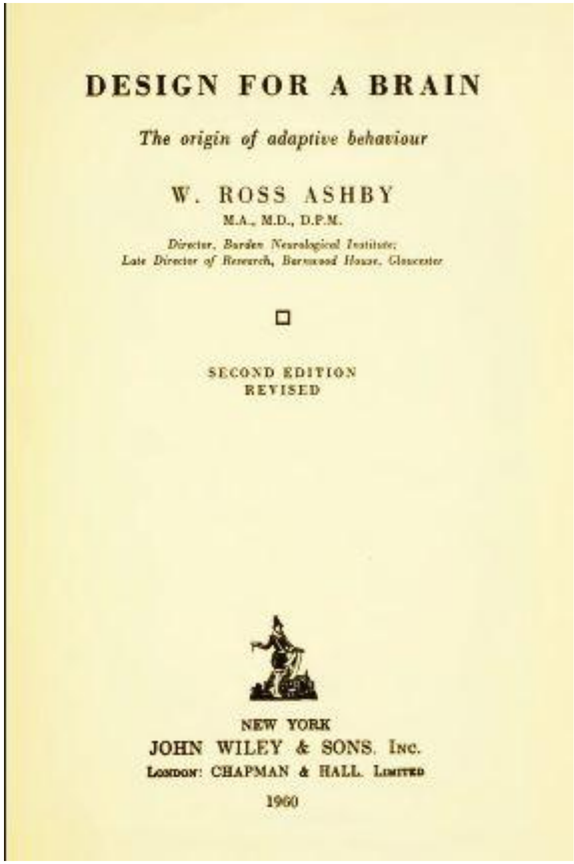
if $z=1$ when $t=0$; then increase θ, and decrease w_i for all positive inputs y_i

1958: Rosenblatt - Perceptrons



Mid 50s: Ashby’s Homeostat

Ross Ashby with Homeostat
Time Magazine 1949:
the closest thing to a synthetic brain so far



DESIGN FOR A BRAIN

The origin of adaptive behaviour

W. ROSS ASHBY

M.A., M.D., D.P.M.

Director, Burden Neurological Institute;
Late Director of Research, Barnwood House, Gloucester

SECOND EDITION
REVISED

NEW YORK
JOHN WILEY & SONS, Inc.
London: CHAPMAN & HALL, Limited
1960

Design for a Brain, 1960

The hype of AI

- Herbert Simon (1957):

  It is not my aim to surprise or shock you—but the simplest way I can summarize is to say that there are now in the world machines that think, that learn and that create.

The hype of AI

Rosenblatt’s press conference 7 July 1958:

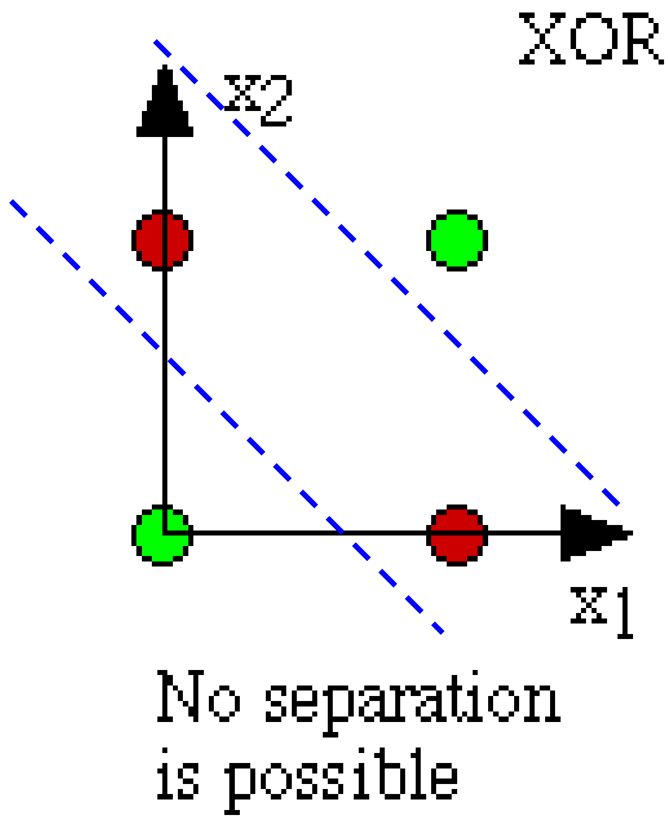
The perceptron, an electronic computer that [was revealed today]

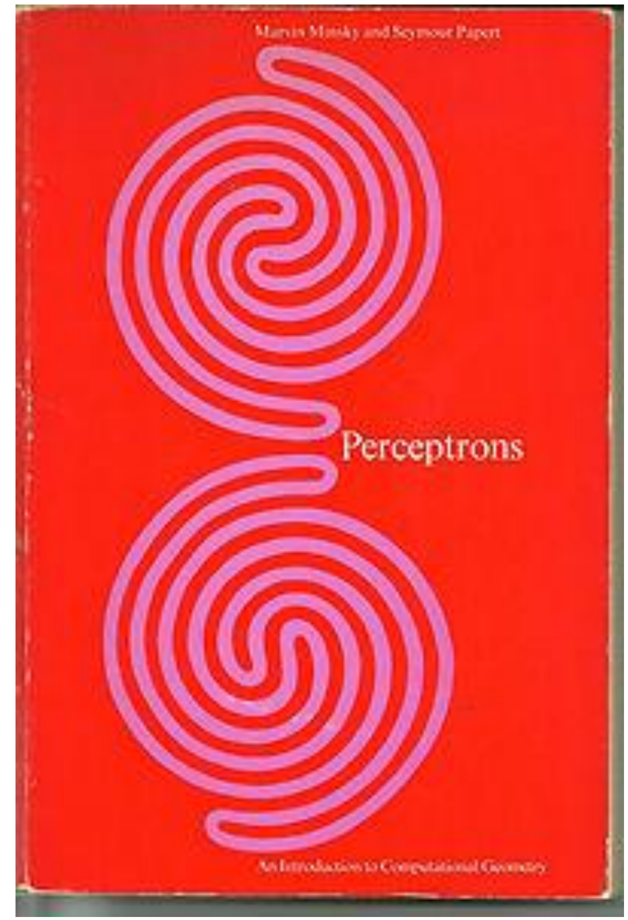
- will be able to walk, talk, see, write, reproduce itself
- be conscious of its existence.

Later perceptrons will be able to

- recognize people and call out their names
- instantly translate speech in one language to speech and writing in another

1969: Minsky / Papert: Perceptrons





A single-layer perceptron can't learn XOR.

requires

$w_1 > 0$, $w_2 > 0$ but $w_1 + w_2 < 0$

Shakey the Robot : 1972

Stanford SRI 1966-1972

STRIPS: planner

Richard Fikes

Nils Nilsson

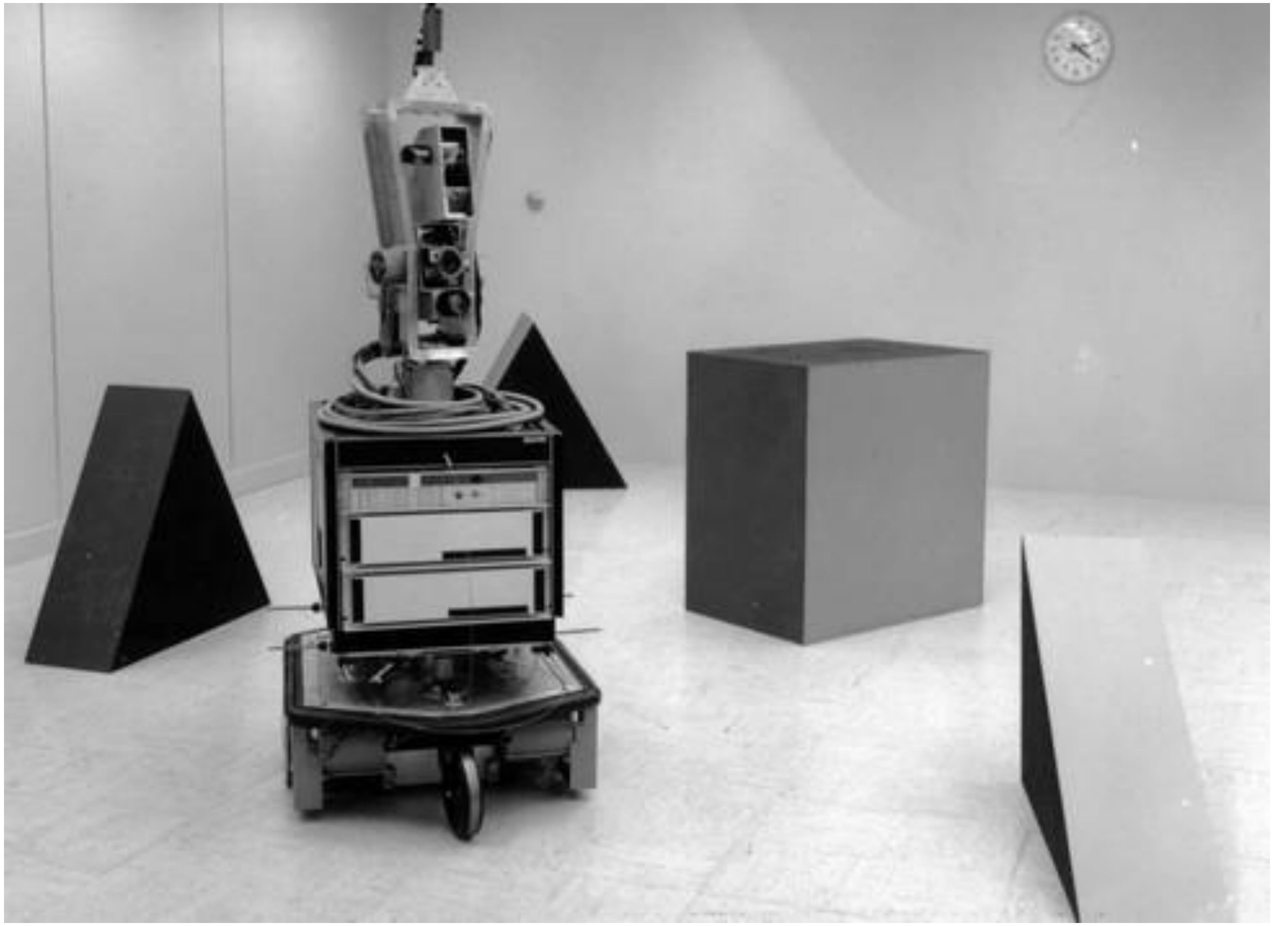
States (propositions)

Actions (pre-condition, post-condition)

Initial / Goal states

Problem w post-conditions: which states are persistent?

→ **Frame Problem**



Shakey the Robot : 1972

Stanford SRI 1966-1972

STRIPS: planner

Richard Fikes

Nils Nilsson

States (propositions)

Actions (pre-condition, post-condition)

Initial / Goal states

Problem w post-conditions: which states are persistent?

→ **Frame Problem**

https://www.youtube.com/watch?v=qXdn6ynwpiI

Timeline : AI – Logical Models

- 1943 McCulloch & Pitts: Boolean circuit model of brain
- 1950 Turing's "Computing Machinery and Intelligence"
- 1956 Dartmouth meeting: "Artificial Intelligence" name
- 1956 Newell & Simon's Logic Theorist
- 1959 Samuel's checkers program: game search; later version learned by playing itself
- 1964-66 ELIZA (psychotherapist) by Joseph Weizenbaum

- 1965 Robinson's resolution algorithm for first order logic
- 1969 Minsky / Papert’s *Perceptron*
- 1970-1975 Neural network research almost disappears; [*sociology of science* study]
- 1966-72 Shakey the robot

- 1964-82 Mathlab / Macsyma : symbolic mathematics
- 1969-79 Early knowledge-based systems (expert systems)

“Expert” systems

DENDRAL 1969:

Expert knowledge for chemical structure

Ed Feigenbaum, Bruce Buchanan Joshua Lederberg

Input:

Chemical formula + ion spectrum from mass spectrometer

Output:

Molecular structure

recognizing ketone (C=O) :

if there are two peaks at x1 and x2 s.t.

(a) x1 + x2 = M +28 (M = molecule mass)

(b) x1−28 is a high peak;

(c) x2−28 is a high peak;

(d) At least one of x1 and x2 is high.

then there is a ketone subgroup

Reduces search by identifying some constituent structures

Timeline : AI – Learning

- 1986 Backpropagation algorithm : Neural networks become popular
- 1990-- Statistical Machine Learning

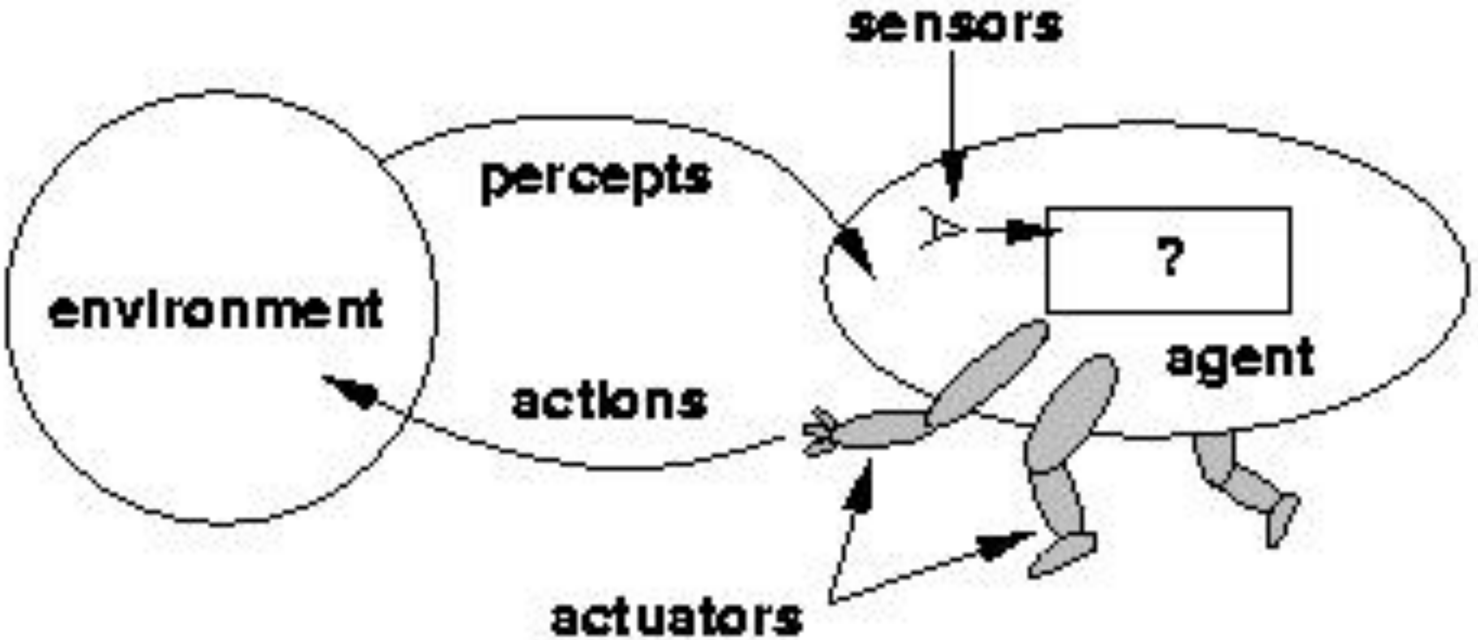
- 1991 *Eigenfaces :* face recognition [Turk and Pentland]
- 1995 [Dickmanns]: 1600km driving, 95% autonomous
  CMU *Navlab*: 5000km 98% autonomous

- 1996 EQP theorem prover finds proof for Robbins' conjecture
- 1997 Deep Blue defeats Kasparov
- 1997 *Dragon Naturally Speaking* speech recognition
- 1999 SIFT local visual feature model
- 2001 [Viola & Jones] : real time face detection
- 2007 DARPA Urban challenge (autonomous driving in traffic)
- 2010 *Siri* speech recognition engine
- 2011 *Watson* wins quiz show *Jeopardy*

xkcd conclusion

Agent Design

Intelligent Agent



Models in Agency

- **Agent** : function from percept histories to actions:

$$[f\colon \mathcal{P} \rightarrow \mathcal{A}]$$

  - Intermediate: Precepts → concept categories
  - Goal : measure of performance [utility]

- Rational agent: one that has best performance

  → utility maximization
  → within computational limitations

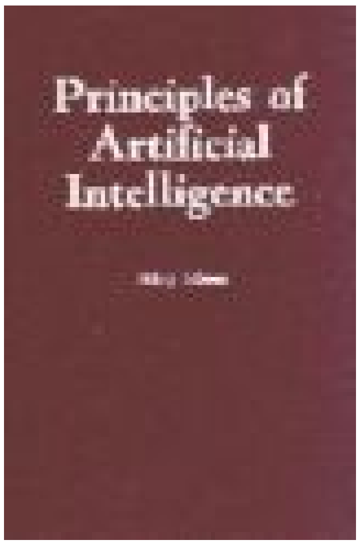
Task / Environment

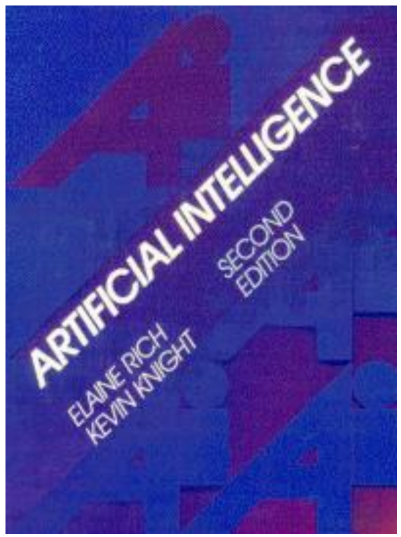
- [f: $\mathcal{P} \rightarrow \mathcal{A}$]

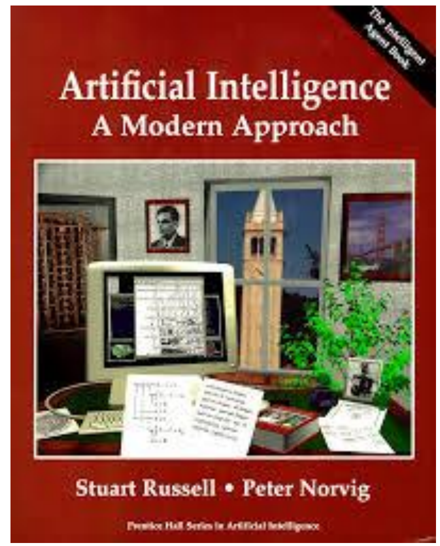
- What are precepts / actions for
  - Bicycle riding
  - Writing notes
    - Language decisions
    - Motor actions
  - Solving a sudoku
  - Drawing a cartoon

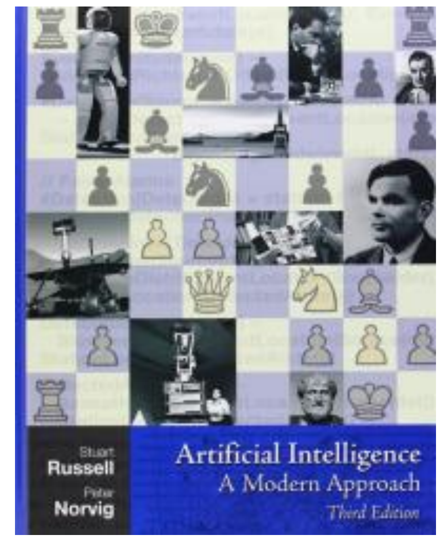
AI: the rise of Learning

AI textbooks : pages dealing with learning



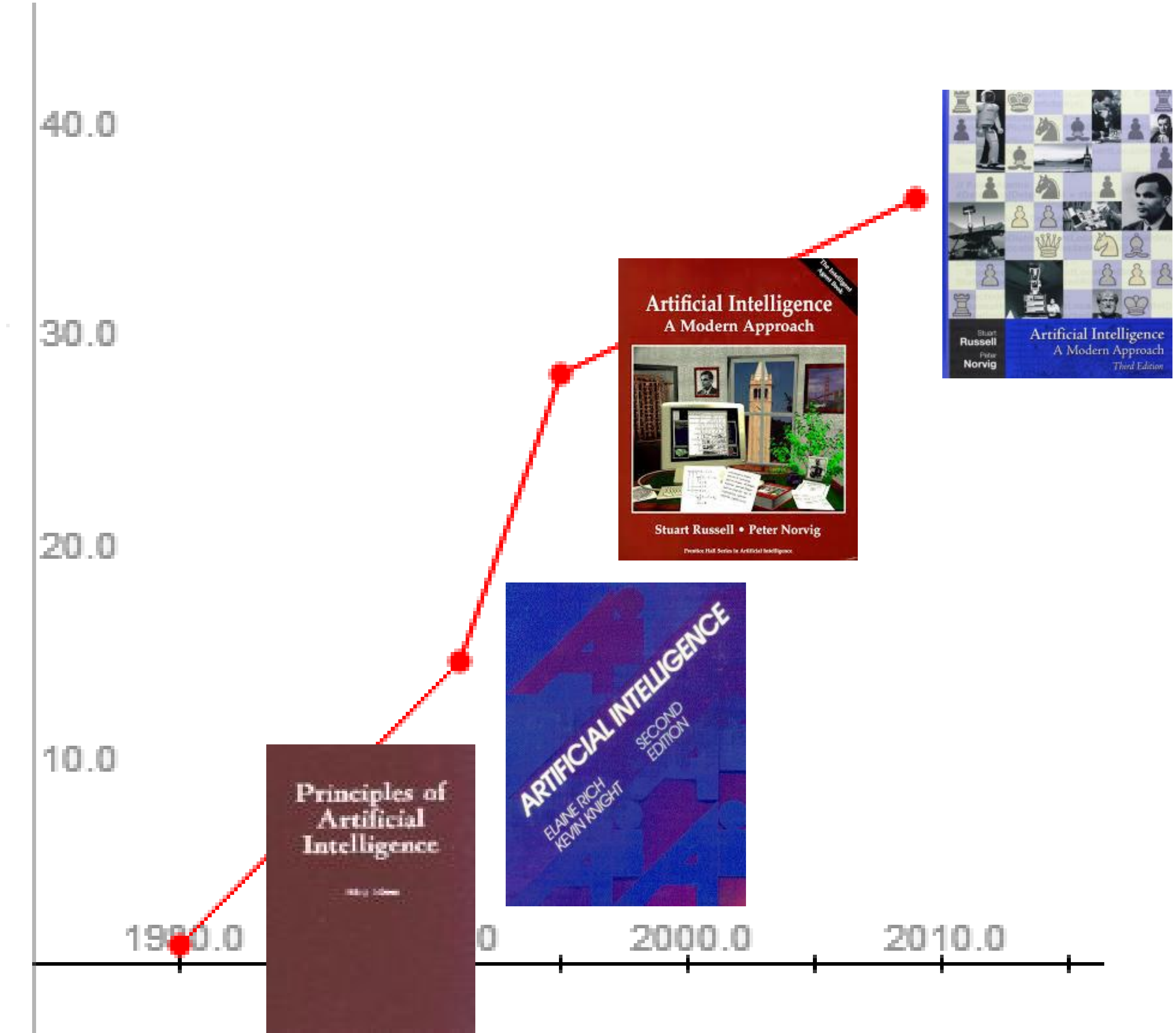




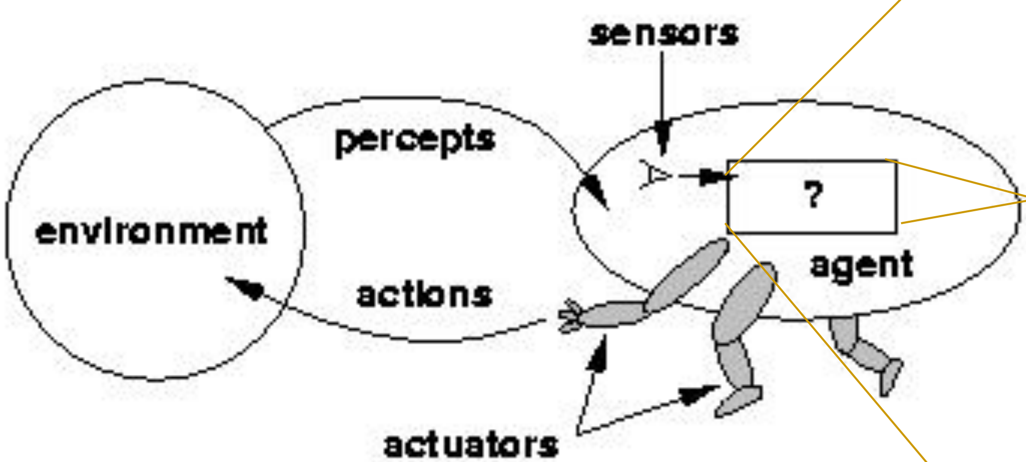


Nilsson	Rich & Knight	R & N	R & N
PoAI	AI 2nd ed	AIMA	AIMA 3d ed
1980	1991	1995	2009
3 / 427 p	82 /582 p	236 /849 p	380 /1052

AI: the rise of Learning



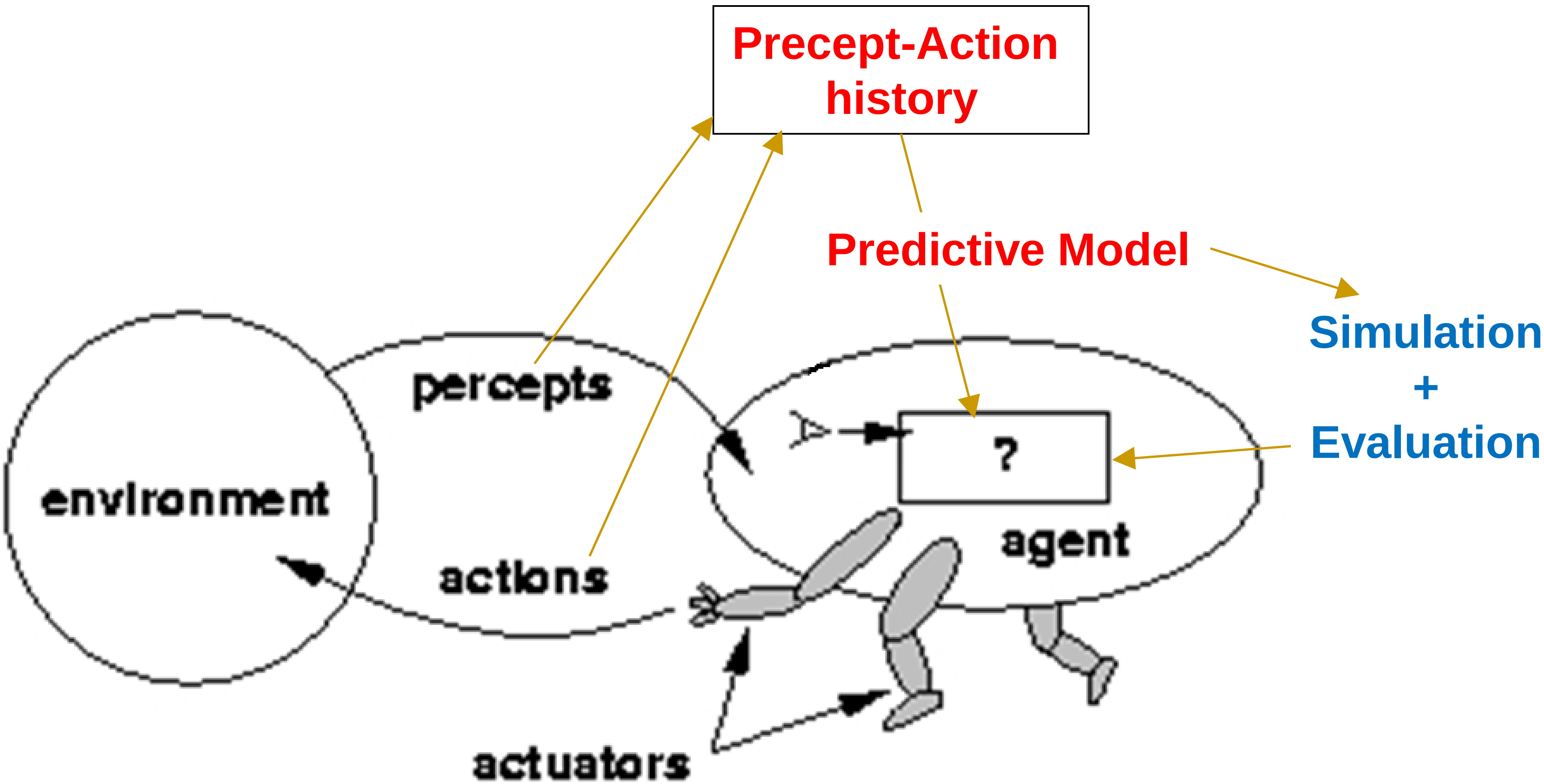
Intelligent Agent



f(): estimated input-output relation, is pre-programmed, e.g. using **logic**

Use precept-action-goal history (experience) to **learn** input-output relation f()

Learning Agent



“Carry a ‘small-scale model’ of external reality and of possible actions within its head “ – Kenneth Craik 1943

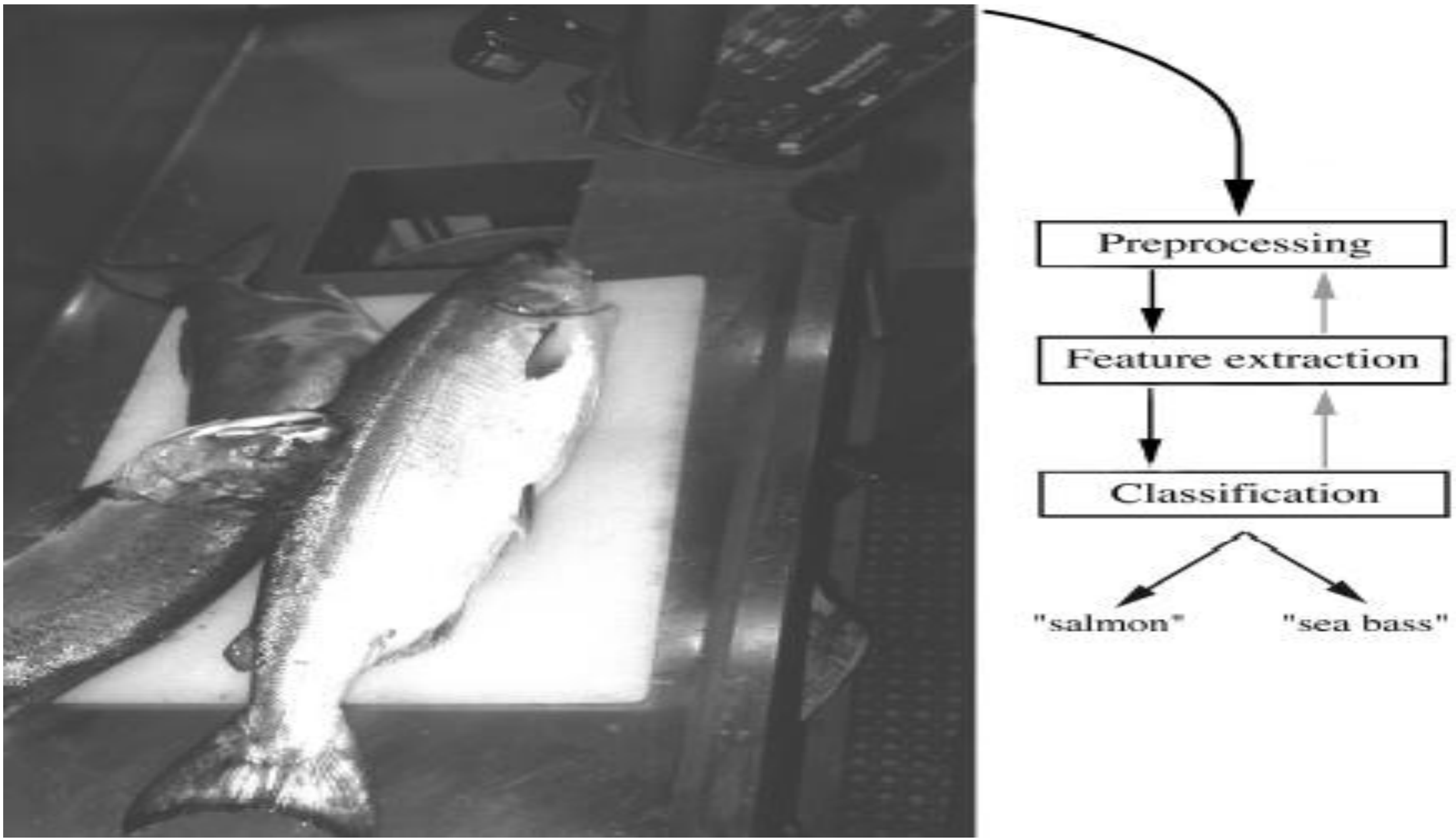
Learning vs Hand-coding

- Predictive model : [f: $\mathcal{P} \rightarrow \mathcal{A}$]

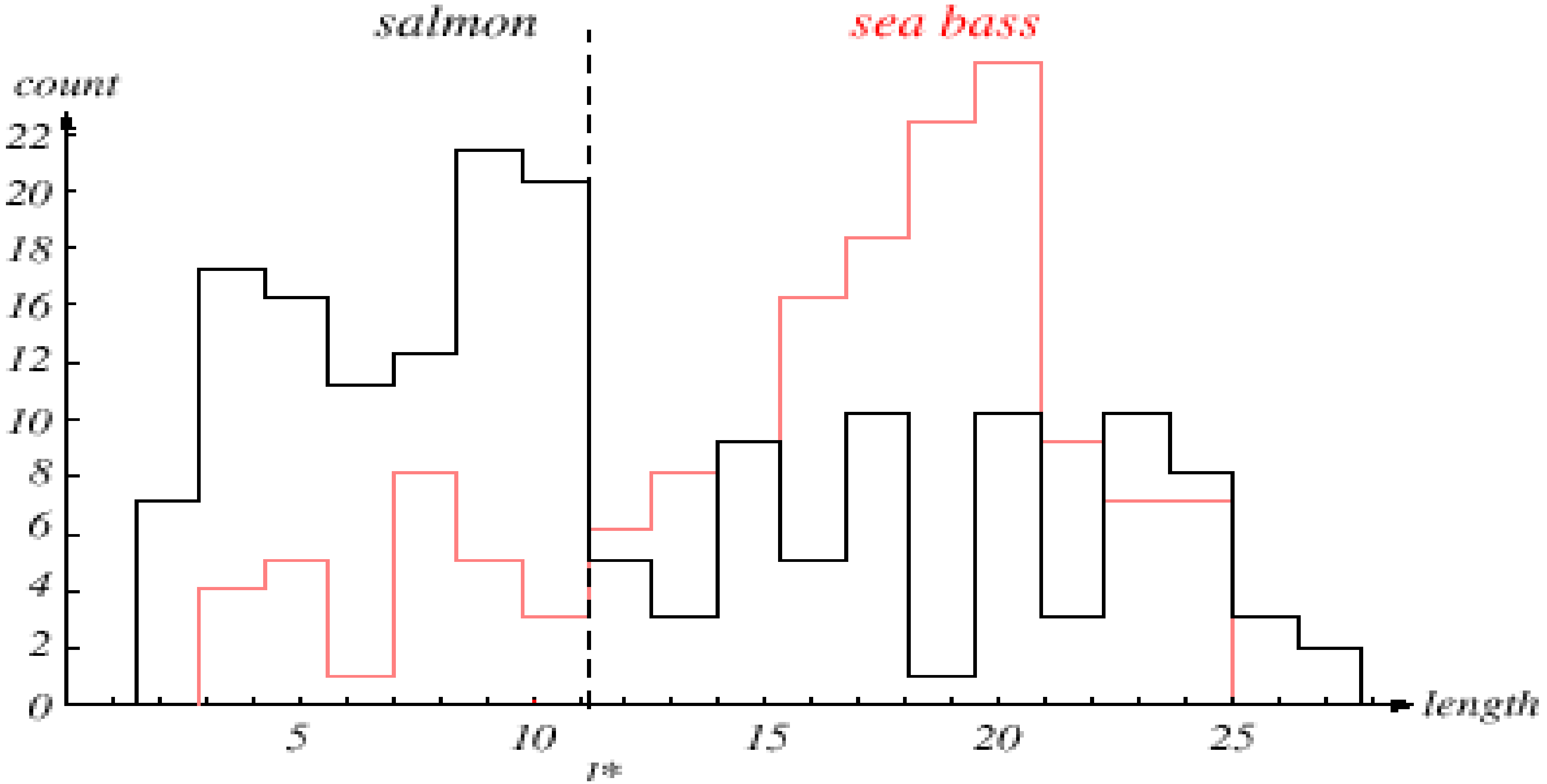
- Should we try to learn the function *f*, or try to use our own ideas about it (hand-code)?
  - Guessing / Hand-coding may be **quicker** in the short run
  - Learning : more robust and stable, but may require **lots of data**

Features, Models and Dimensionality

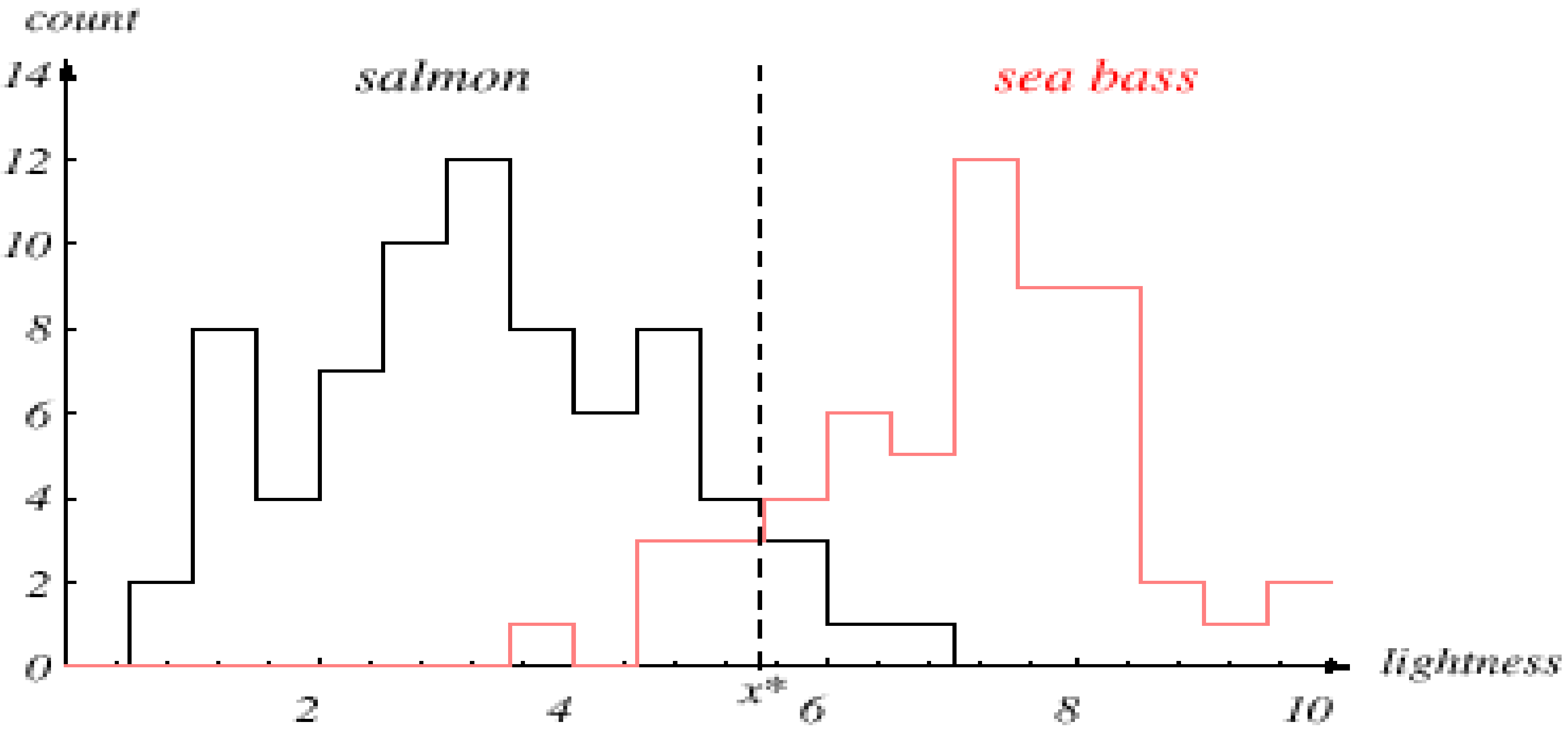
Binary Classification



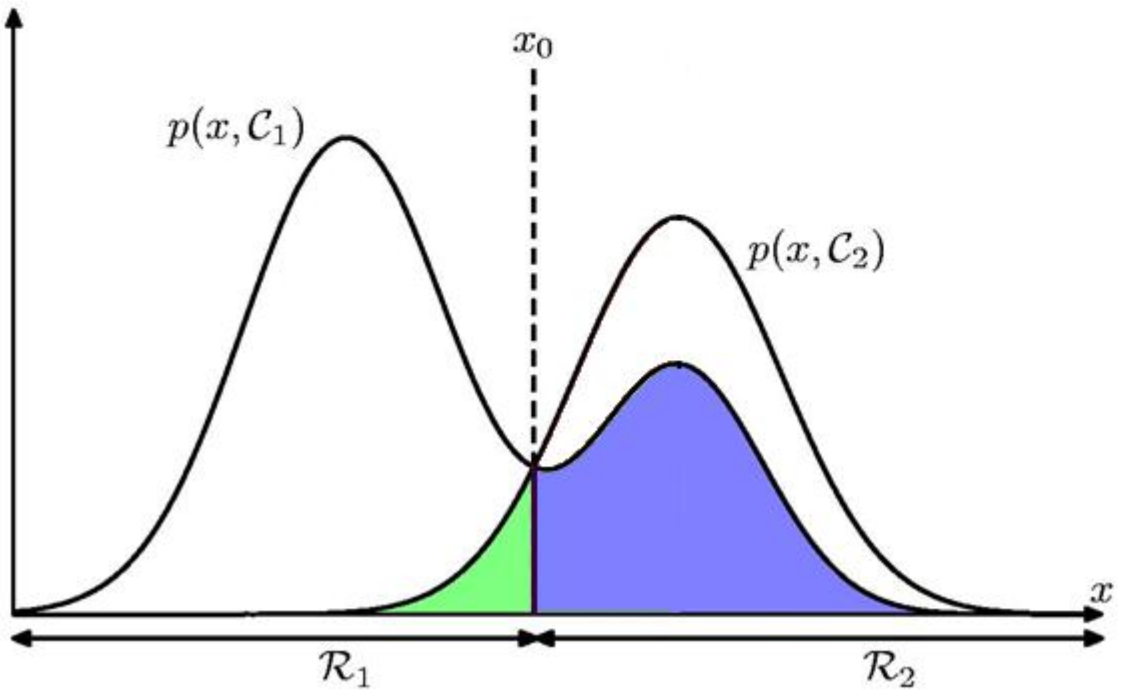
Feature : Length



Feature : Lightness



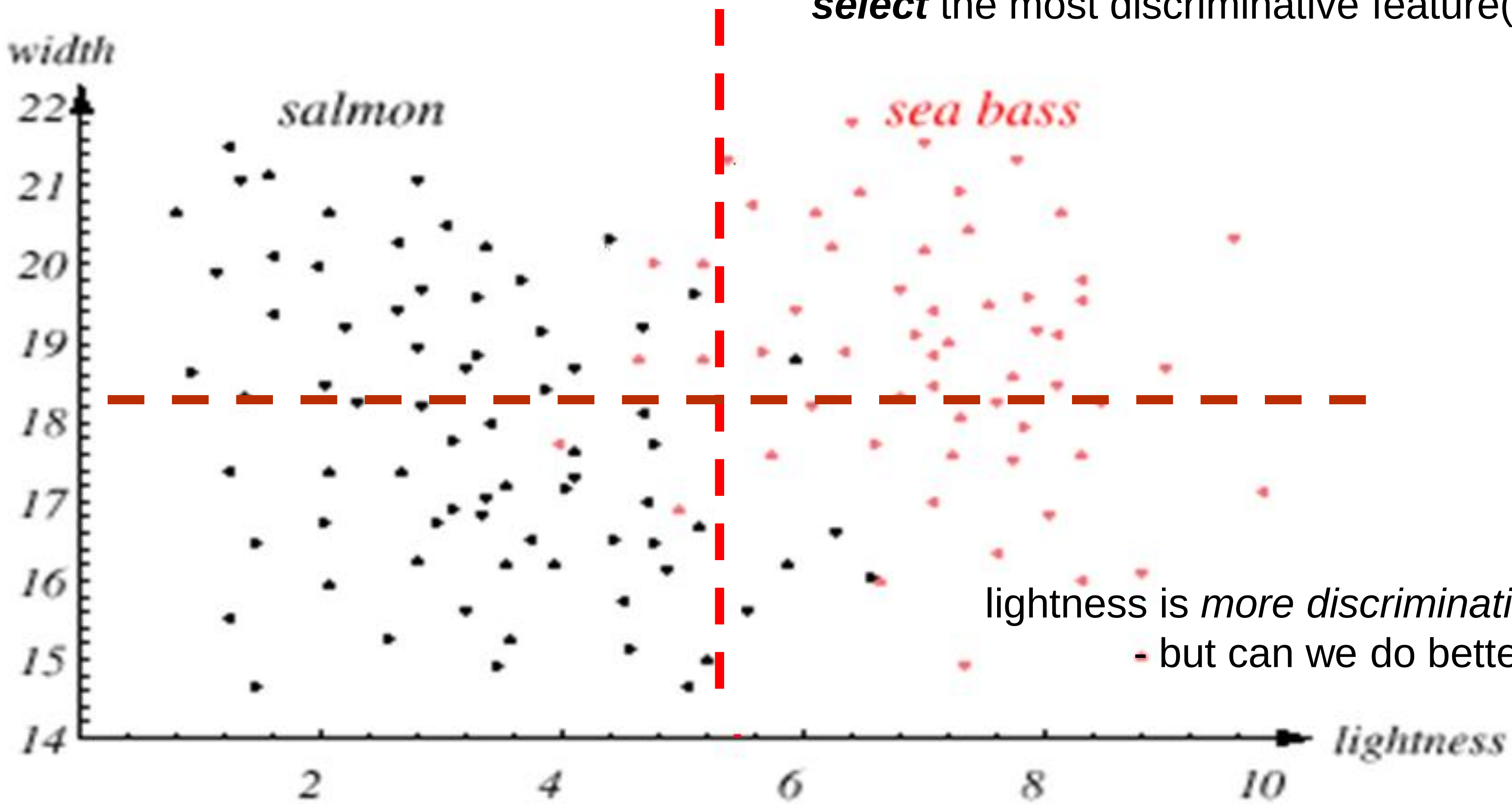
Minimize Misclassification



$$
\begin{aligned}
p(\text{mistake}) &= p(\mathbf{x} \in \mathcal{R}_1, \mathcal{C}_2) + p(\mathbf{x} \in \mathcal{R}_2, \mathcal{C}_1) \\
&= \int_{\mathcal{R}_1} p(\mathbf{x}, \mathcal{C}_2)\,\mathrm{d}\mathbf{x} + \int_{\mathcal{R}_2} p(\mathbf{x}, \mathcal{C}_1)\,\mathrm{d}\mathbf{x}.
\end{aligned}
$$

Feature Selection: *width / lightness*

select the most discriminative feature(s)



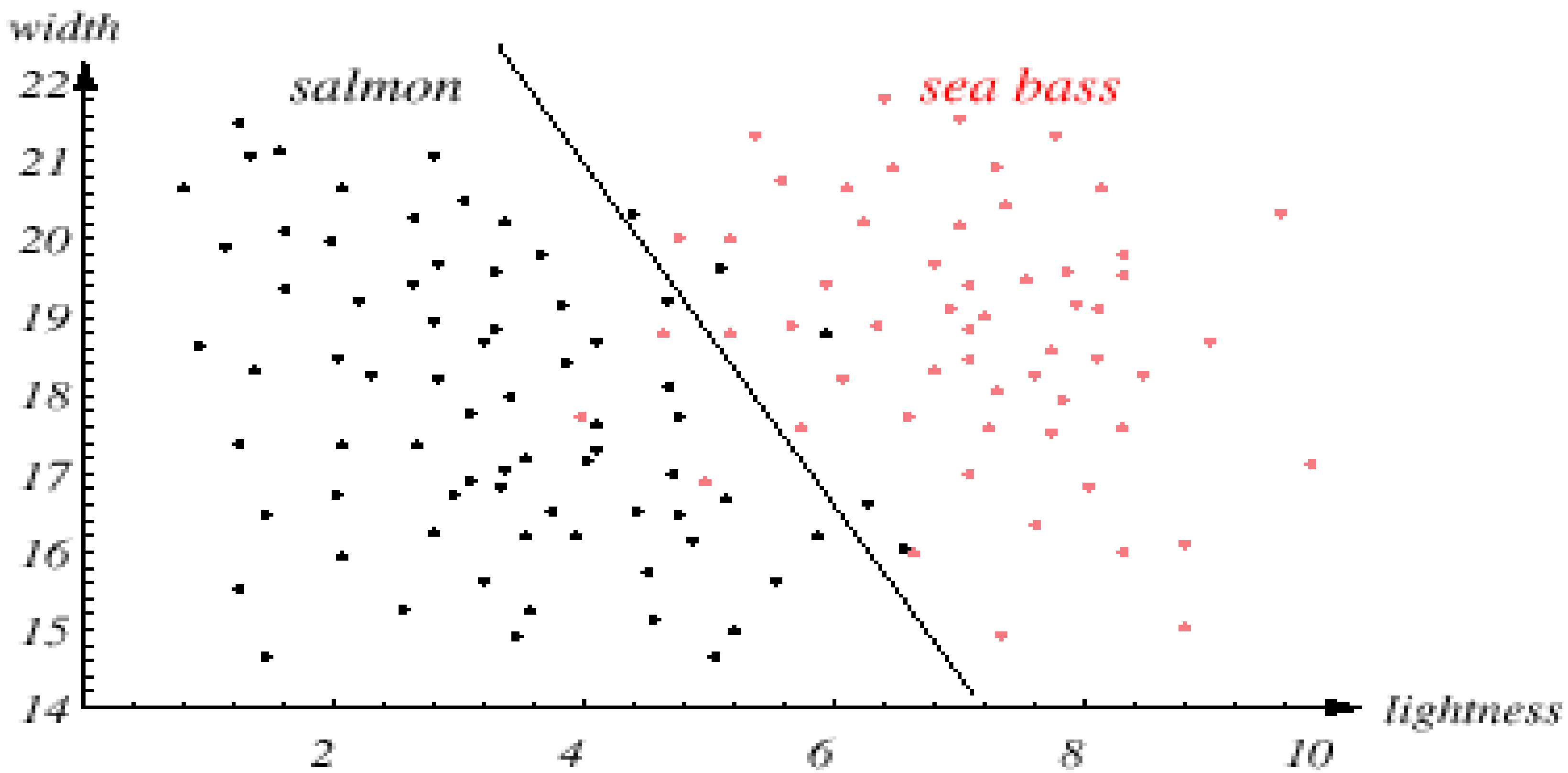
lightness is *more discriminative*
- but can we do better?

Feature Selection

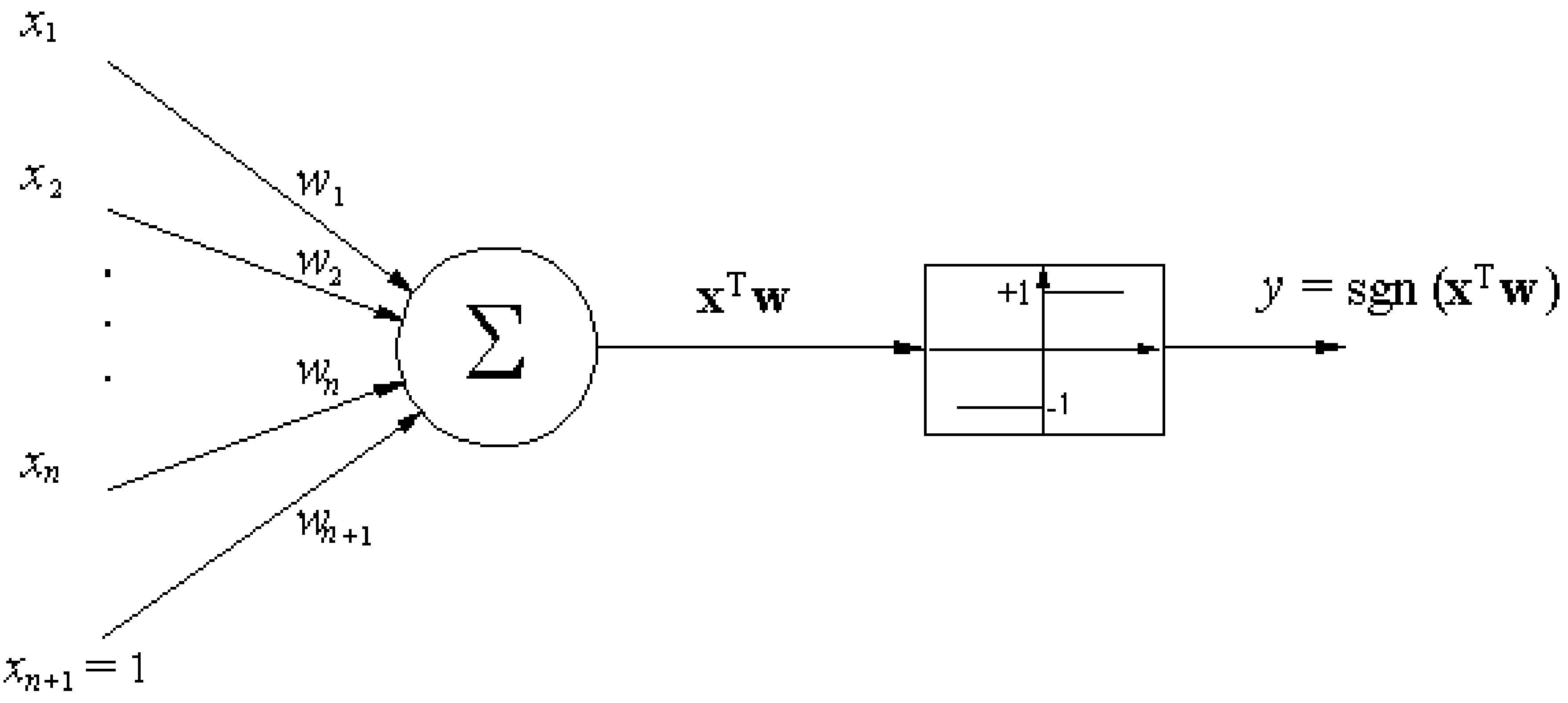
- Feature selection : which feature is maximally discriminative?
  - Axis-oriented decision boundaries in feature space
  - Length – or – Width – or Lightness?

- Feature Discovery: discover discriminative function on feature space : g()
  - combine aspects of length, width, lightness

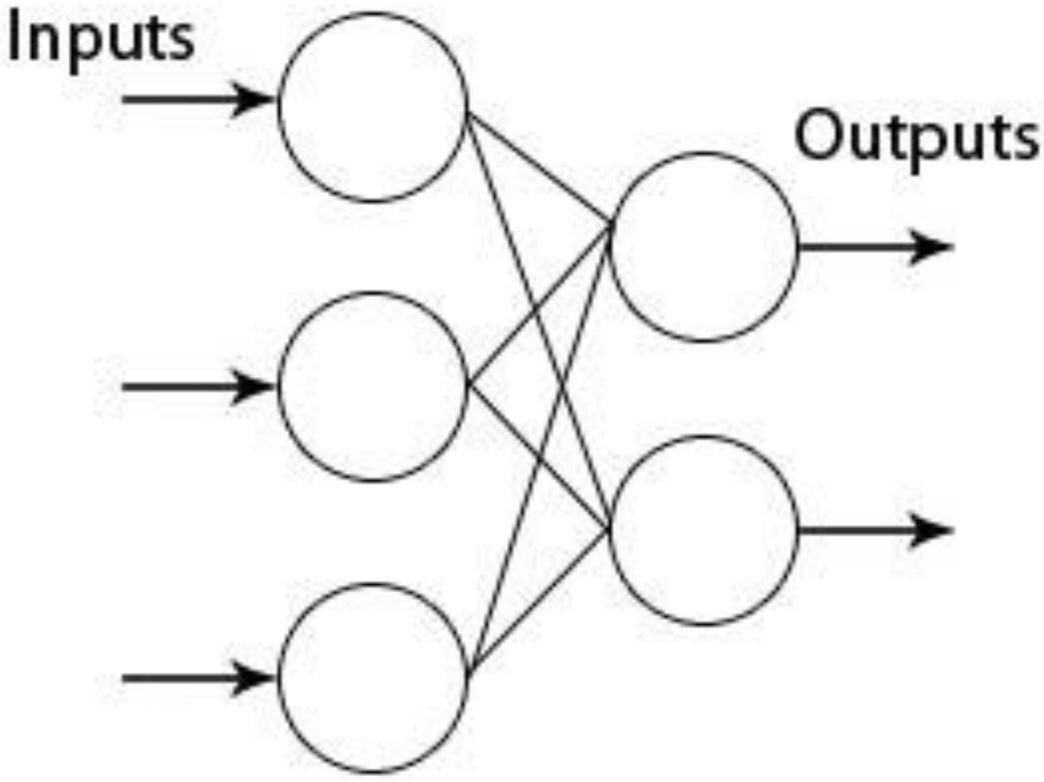
Feature Discovery : Linear



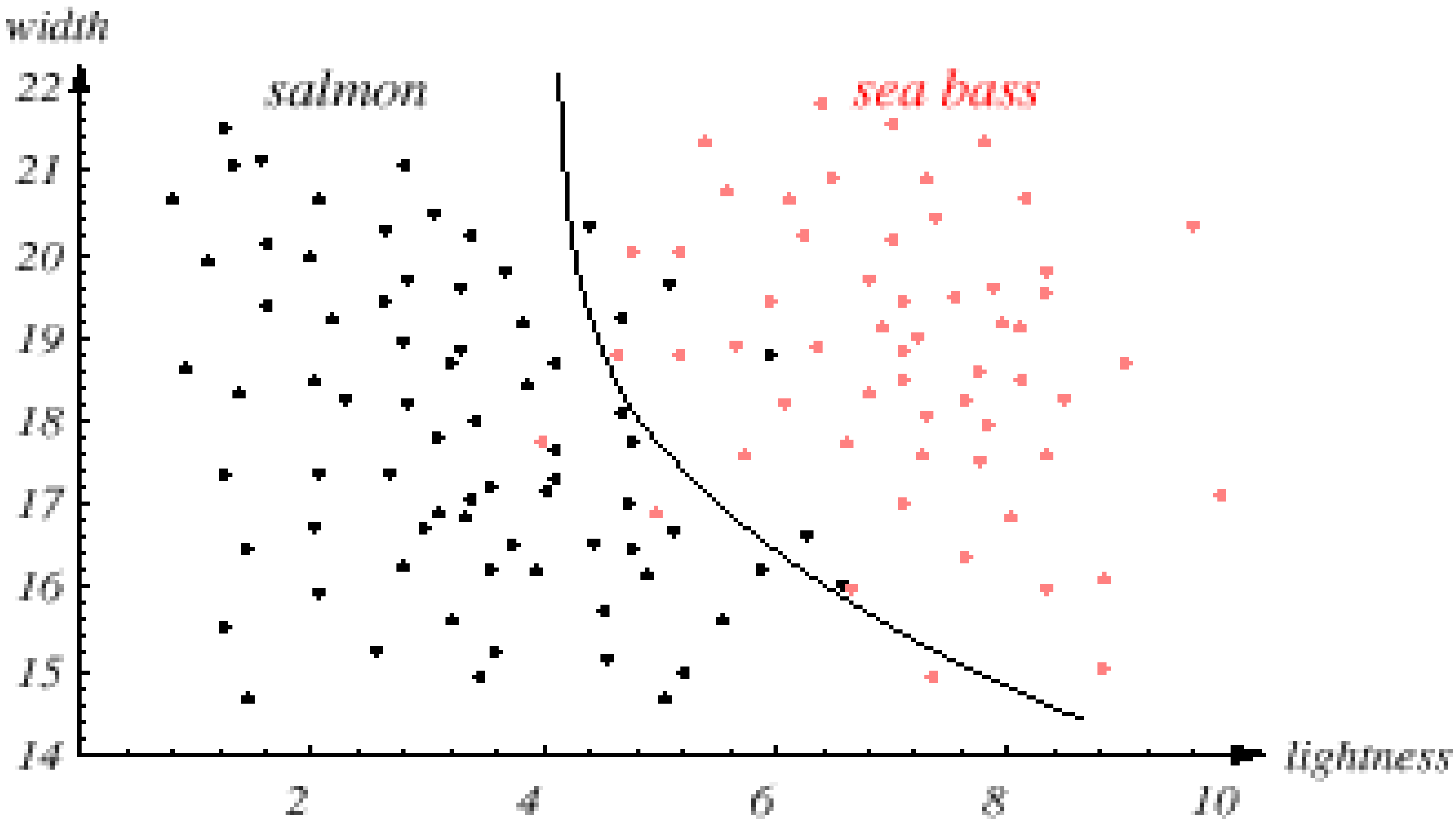
Linear Perceptron Unit



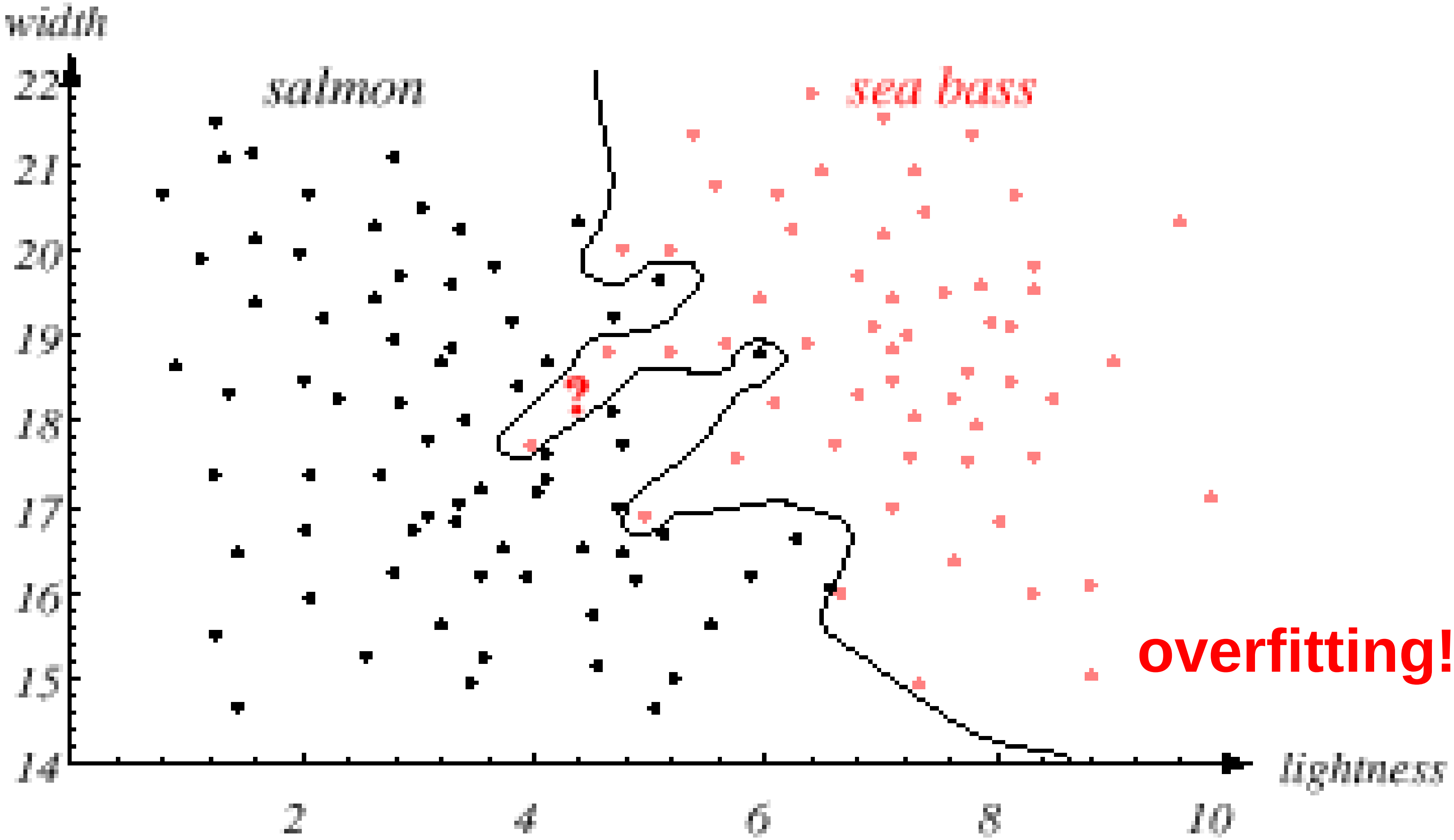
Multi-layer Perceptron



Feature Discovery : non-linear



Decision Surface : non-linear



Learning process

- Feature set : representative? complete?

- Sample size :
  - Training set : bigger the better?
  - Test set: unseen real data
  - Validation set : tune parameters of learning

- Model selection:
  - Unseen data → overfitting?
  - Quality vs Complexity
  - Computation vs Performance

Agent Models

Models of Agency

- **Agent** : function from percept histories to actions:

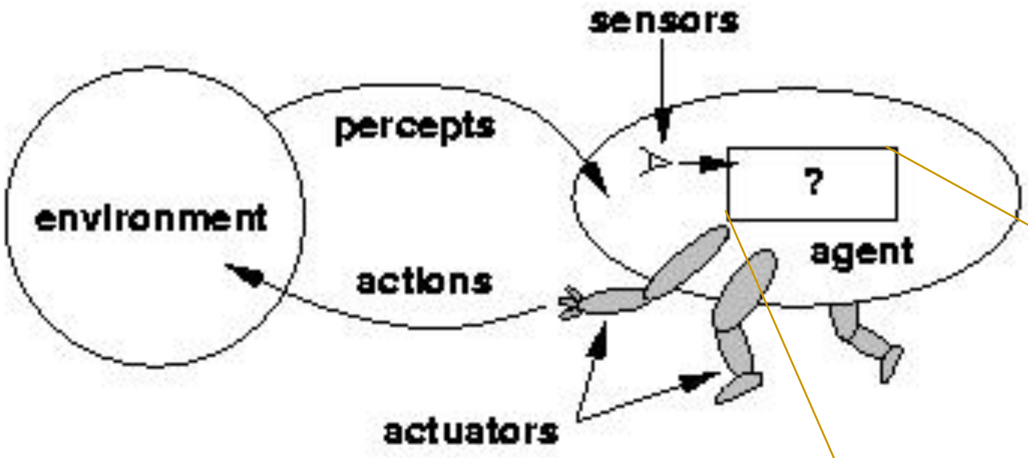
  $$[f: \mathcal{P} \rightarrow \mathcal{A}]$$

  - Intermediate: Precepts → concept categories
  - Goal : measure of performance [utility]

- Rational agent: one that has best performance

  → utility maximization
  → within computational limitations

Intelligent Agent



Predictive model

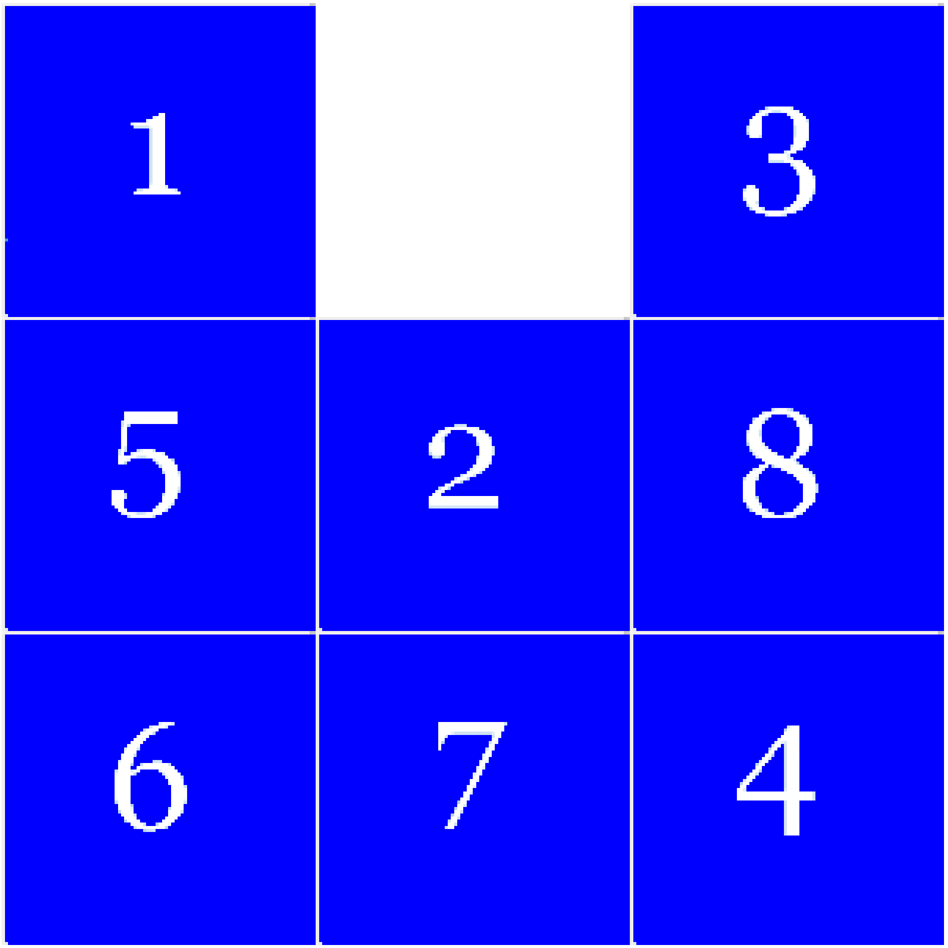
[f: $\mathcal{P} \rightarrow \mathcal{A}$]

Task / Environment

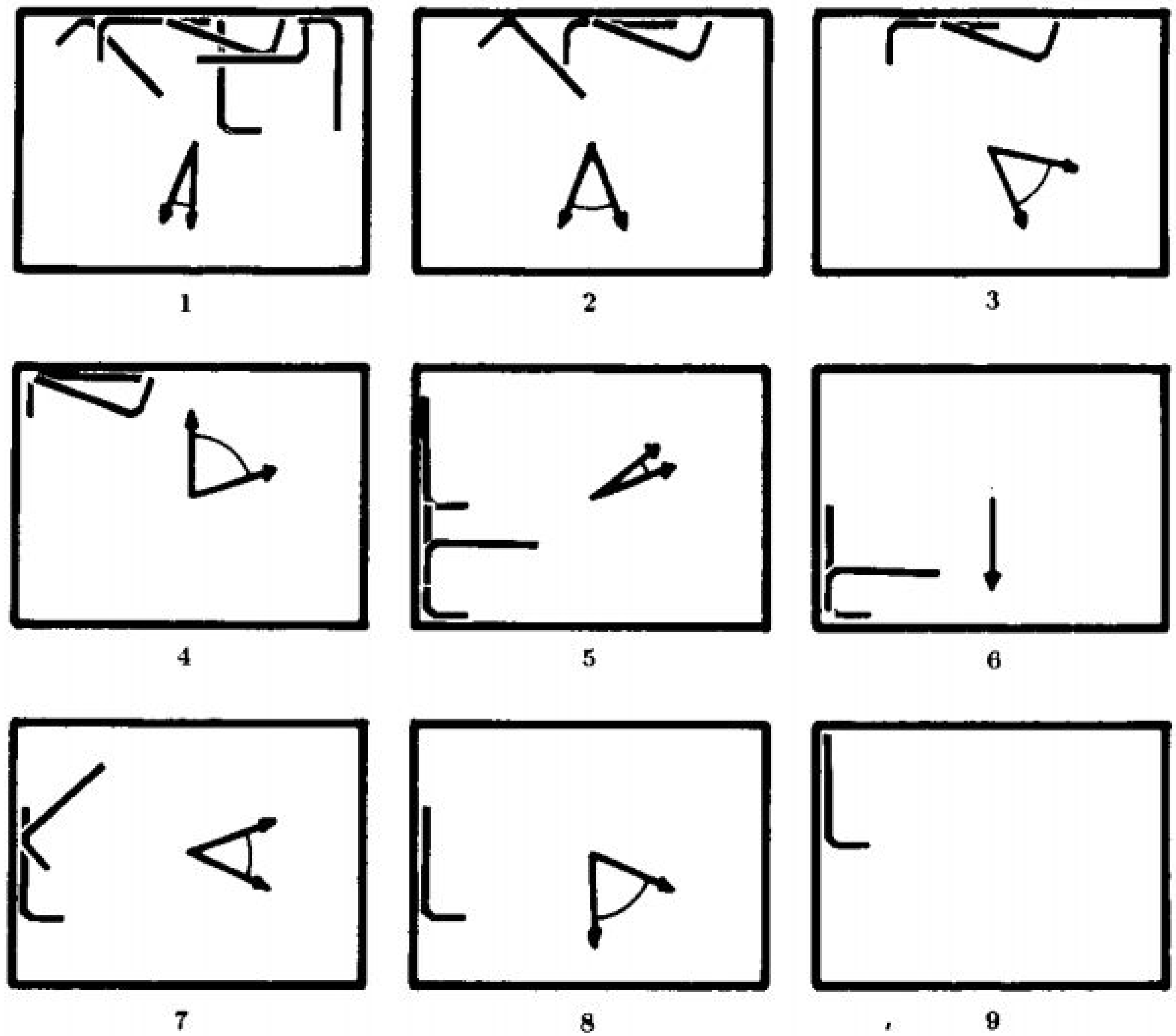
- $[f: \mathcal{P} \rightarrow \mathcal{A}]$

- What are precepts / actions for
  - Bicycle riding
  - Writing notes
    - Language decisions
    - Motor actions
  - Solving a sudoku
  - Drawing a cartoon

8-puzzle



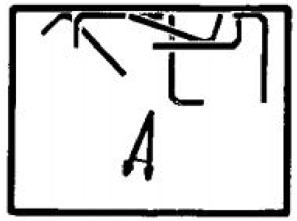
Unobservable Problems



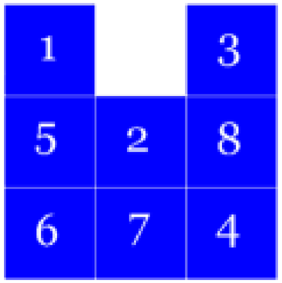
[erdmann / mason 1987]

Nature of Task

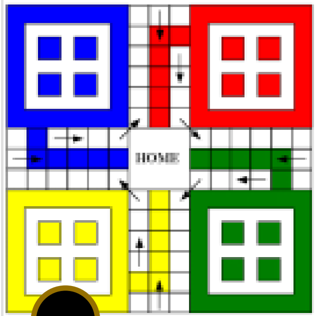
soccer
driving
face recognition



continuous



chess
puzzles
& games



ludo
backgammon

discrete

deterministic

stochastic

Nature of Environment

- static
- dynamic
  - other agents?

- fully observable
- partly observable
- unobservable

Environment types

- Static (vs. dynamic): Environment is as presented by sensor – it does not change while agent is deliberating.

- Discrete (vs. continuous): A limited number of distinct, clearly defined percepts and actions.

- Single agent (vs. multiagent): An agent operating by itself in an environment.

Environment types

- Fully observable (vs. partially observable): Sensors give tell the complete (relevant) state of the environment
- Deterministic (vs. stochastic): Given action in a given state completely determines the next state.
  - Strategic : Deterministic, but with other agents
- Episodic (vs. sequential): Experience composed of atomic "episodes" (percept-action pairs); action in an episode is independent of other episodes.

Agent-Environment-Goal (PEAS)

- E.g. Task = design an automated taxi driver:
  - **P: Performance measure**: Safe, fast, legal, comfortable trip, maximize profits
  - **E: Environment**: Roads, other traffic, pedestrians, customers
  - **A: Actuators:** Steering wheel, accelerator, brake, signal, horn
  - **S: Sensors:** Cameras, sonar, speedometer, GPS, odometer, engine sensors, keyboard

Learning

- [f: $\mathcal{P} \rightarrow \mathcal{A}$]

- Nature of $\mathcal{P}$ / $\mathcal{A}$:
  - continuous : regression
  - discrete : categorization

- Performance evaluation function?
- Intermediate “features”?

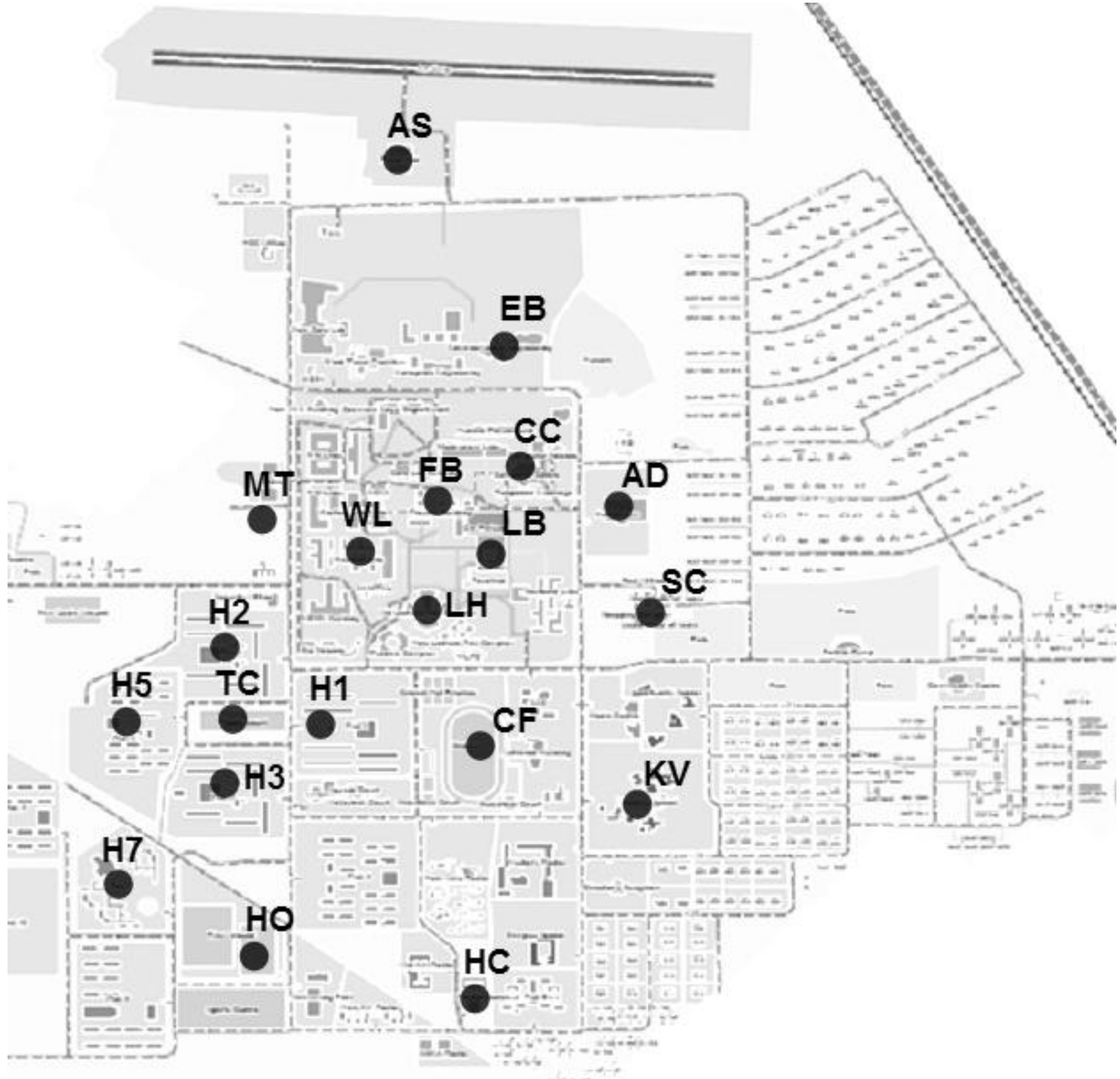
Nature of Representation

- Explicit : Intermediate states are known

- Implicit : Not aware of intermediate states

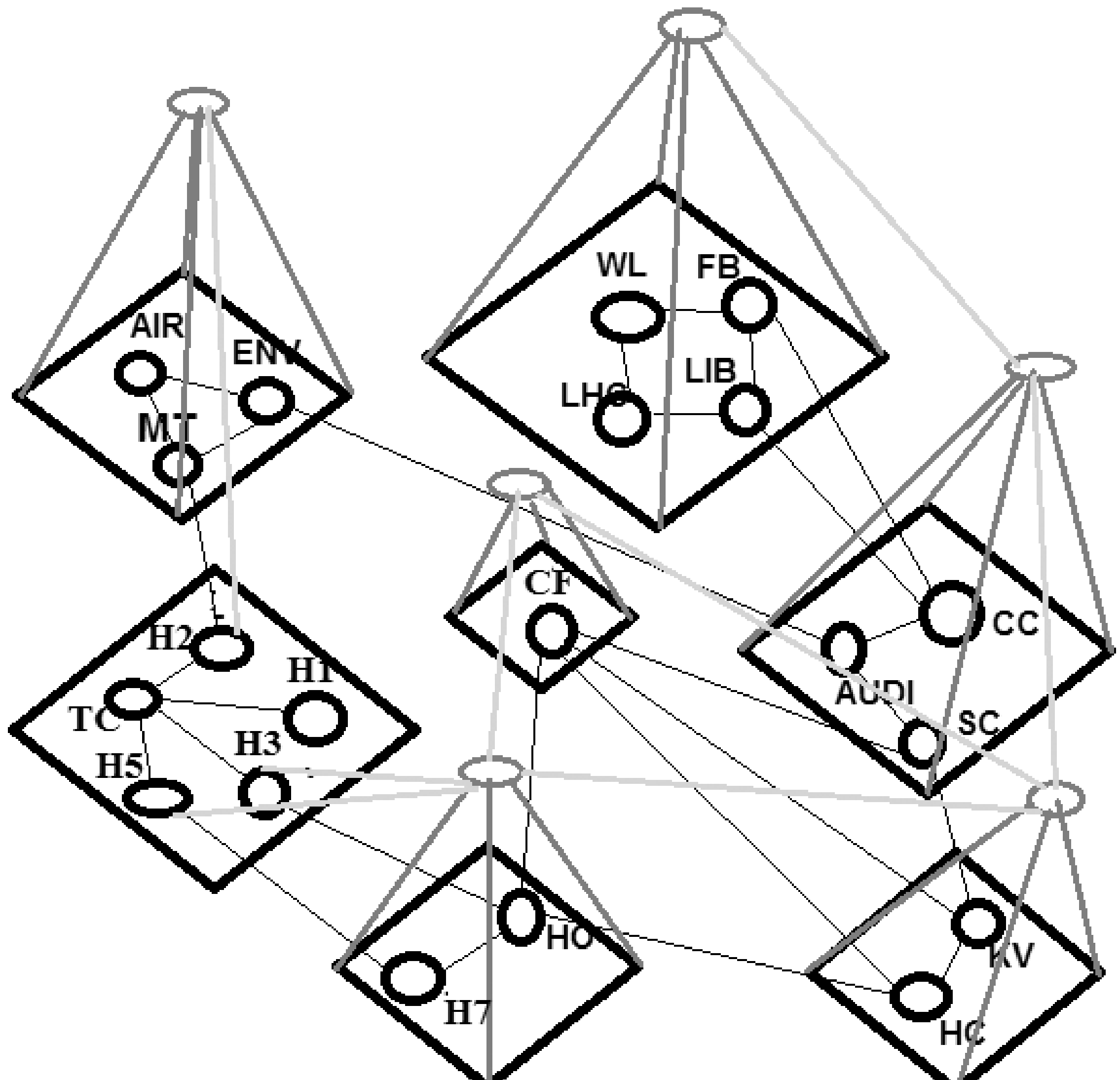
  e.g. Driving

Learning : Explicit → Implicit

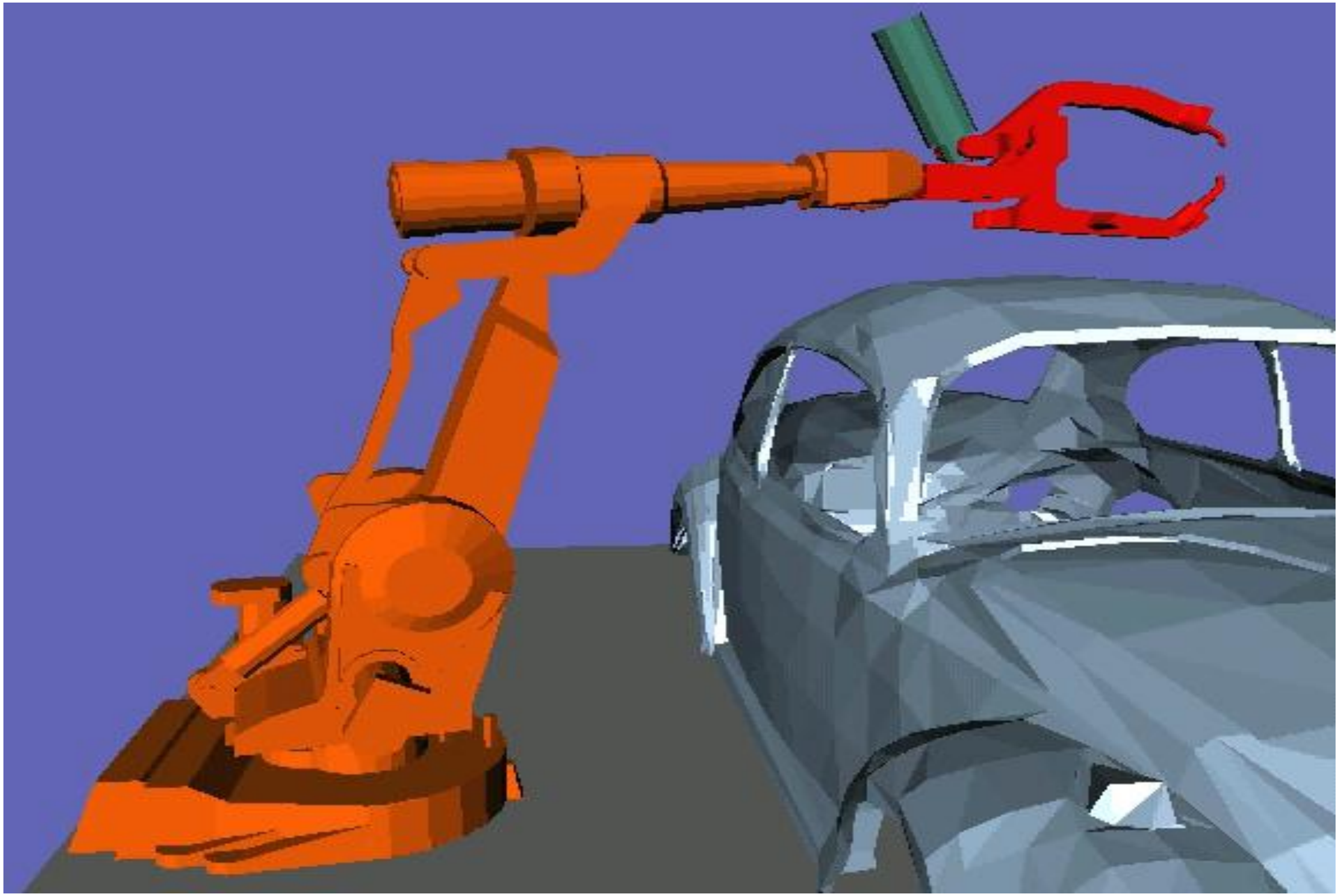


AS
EB
CC
FB
AD
MT
WL
LB
SC
LH
H2
H5
TC
H1
CF
KV
H3
H7
HO
HC

Hierarchical graph



PEAS : Welding Robot



PEAS : Welding Robot

- **Performance measure**: spot weld strengths
- **Environment**: Cars on conveyor belts, other robots
- **Actuators:** Jointed arm and hand
- **Sensors**: Camera, joint angle sensors, arc current

PEAS : Medical Diagnosis

- **Performance measure**: Healthy patient, minimize costs, lawsuits
- **Environment:** Patient, hospital, staff
- **Actuators:** Screen display (questions, tests, diagnoses, treatments, referrals)
- **Sensors:** data fields and text - (list of symptoms, findings, patient's answers)

Learning Agents

Motivation for Learning Agents

- **Implicit knowledge**:
  Experts often can’t explain why they favour some decisions

- **Unknown domains:**
  System works in a finite environment, but may fail for new problems

- **Model structures**:
  Learning reveal properties (regularities) of the system
  - Modifies agent's decision models to **reduce complexity** and improve performance

Feedback in Learning

- Type of feedback:
  - Supervised learning: correct answers for each example
    - Discrete (categories) : classification
    - Continuous : regression
  - Unsupervised learning: correct answers not given
  - Reinforcement learning: occasional rewards

Inductive learning

- Simplest form: learn a function from examples

An example is a pair (x, y) : x = data, y = outcome

assume: y drawn from function f(x) : y = f(x) + noise

f = target function

Problem: find a hypothesis h

such that $h \approx f$

given a training set of examples

Note: highly simplified model :

- Ignores prior knowledge : some h may be more likely
- Assumes lots of examples are available
- Objective: maximize prediction for unseen data – Q. How?

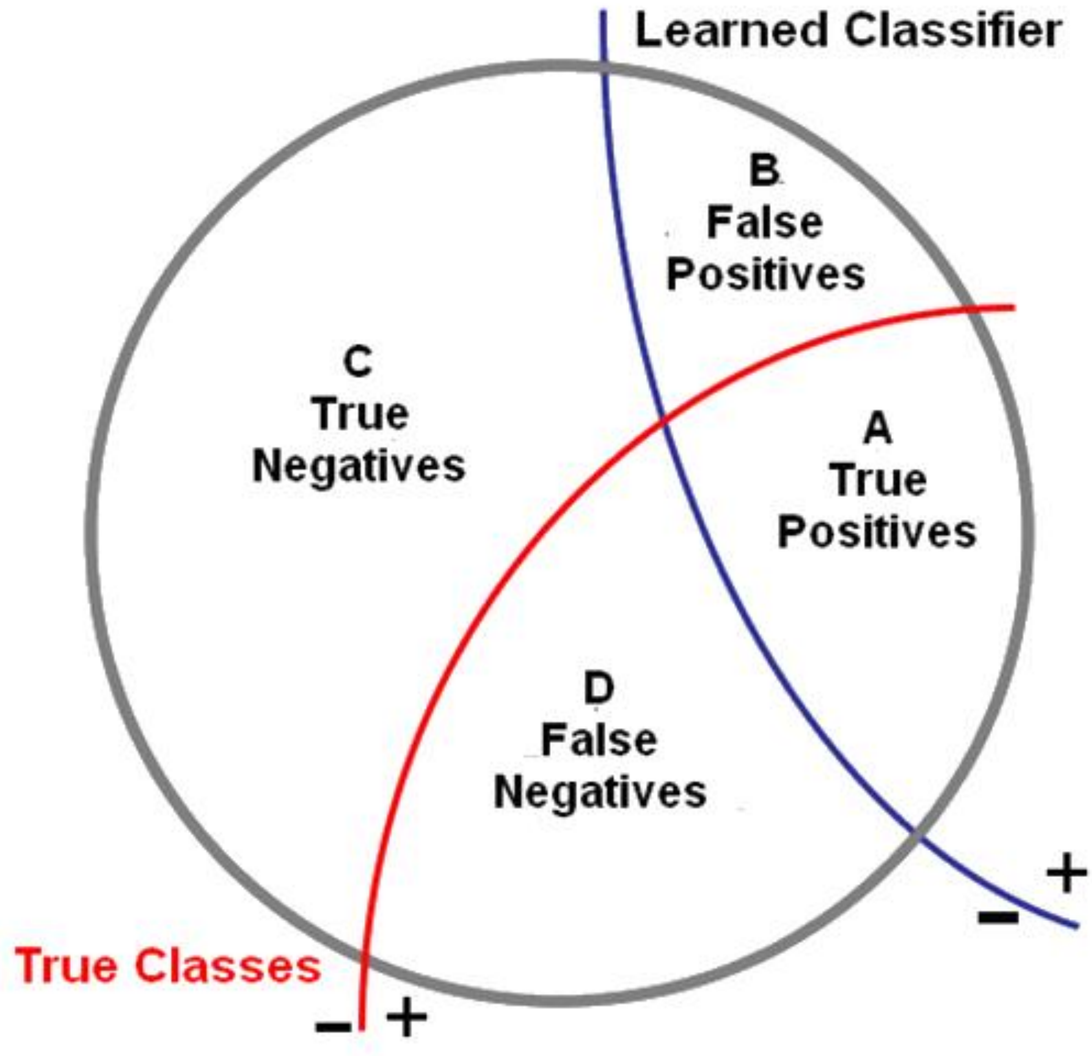
Precision vs Recall

Precision:

A / Retrieved Positives

Recall:

A / Actual Positives



Discrete-Deterministic Spaces:

Search

Problem types

- Deterministic, fully observable → single-state problem
  - Agent knows exactly which state it will be in; solution is a sequence
  -
- Non-observable → sensorless problem (conformant problem)
  - Agent may have no idea where it is; solution is a sequence
  -
- Nondeterministic and/or partially observable → contingency problem
  - percepts provide new information about current state
  - often interleave search, execution
  -
- Unknown state space → exploration problem

State-Space formulation

State description. Plus four items:

1. initial state e.g., "at Arad“

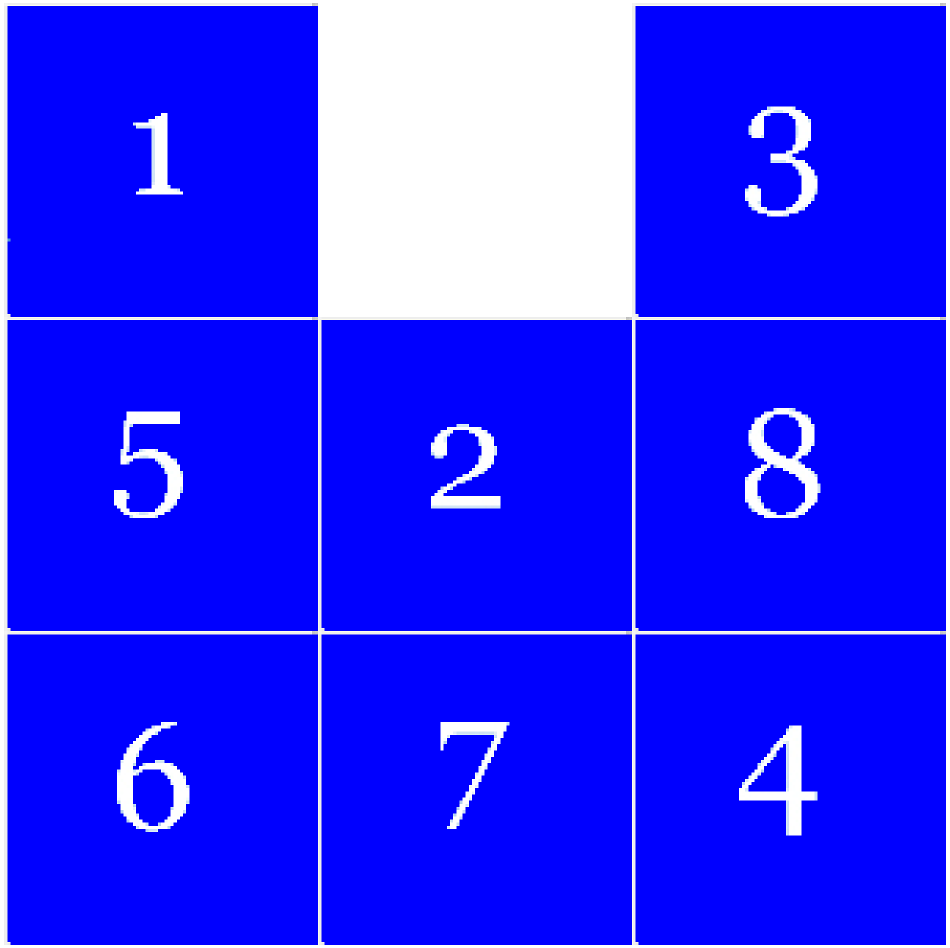
2. actions or successor function *S(x)* = action / result state pairs
   - e.g., *S(Arad)* = {<*Arad* → *Zerind, Zerind*>, … }

3. goal test, can be
   - explicit, e.g., *x* = "at Bucharest"
   - implicit, e.g., *Checkmate(x)*

4. path cost (additive)
   - e.g., sum of distances, number of actions executed, etc.
   - *c(x,a,y)* is the step cost, assumed to be ≥ 0

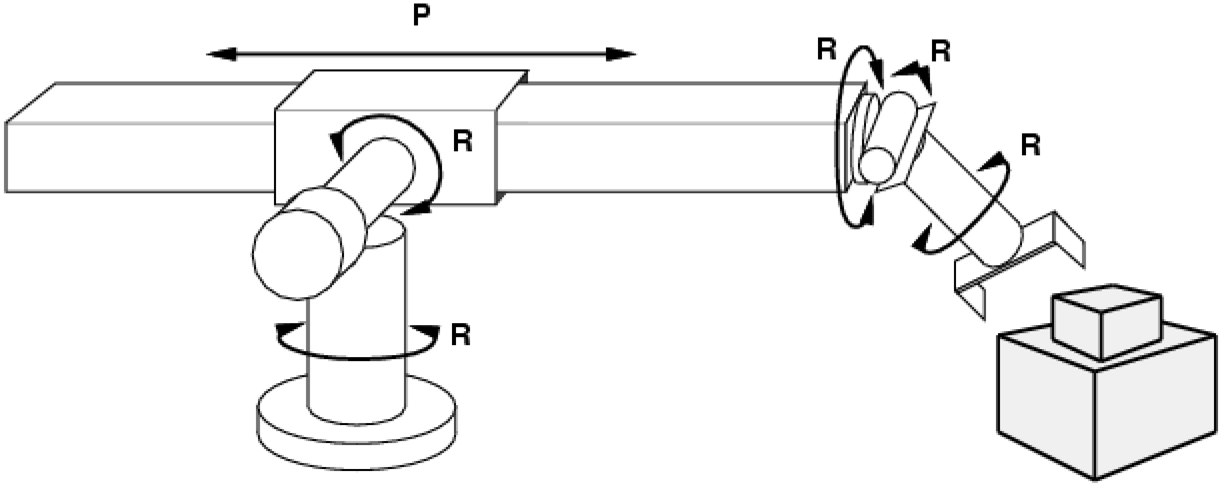
- solution = sequence of actions leading to goal state

Choosing a state space



1. States:
2. Actions :
3. Goal test:
4. Cost:

Example: robotic assembly



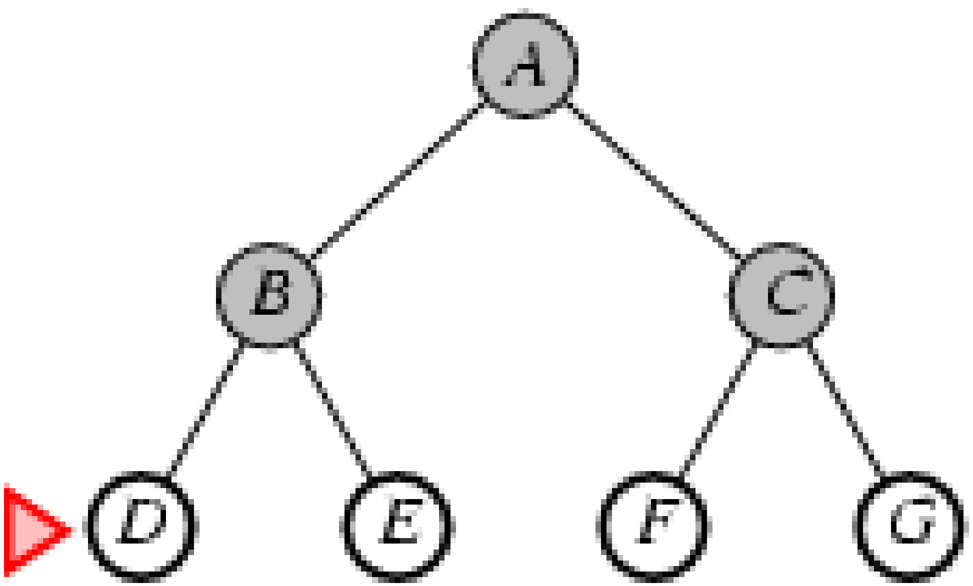
- states?: real-valued joint coordinates + poses (6-DOF) of parts
- actions?: continuous motions of robot joints
- goal test?: is assembly complete?
- path cost?: time / safety / energy / path length / success probability /

Uninformed search strategies

- Uninformed search strategies use only the information available in the problem definitio
- Breadth-first search
- Uniform-cost search
- Depth-first search
- Depth-limited search
- Iterative deepening search

Breadth-first search

- Expand shallowest unexpanded node
- Fringe: FIFO queue new successors go at end

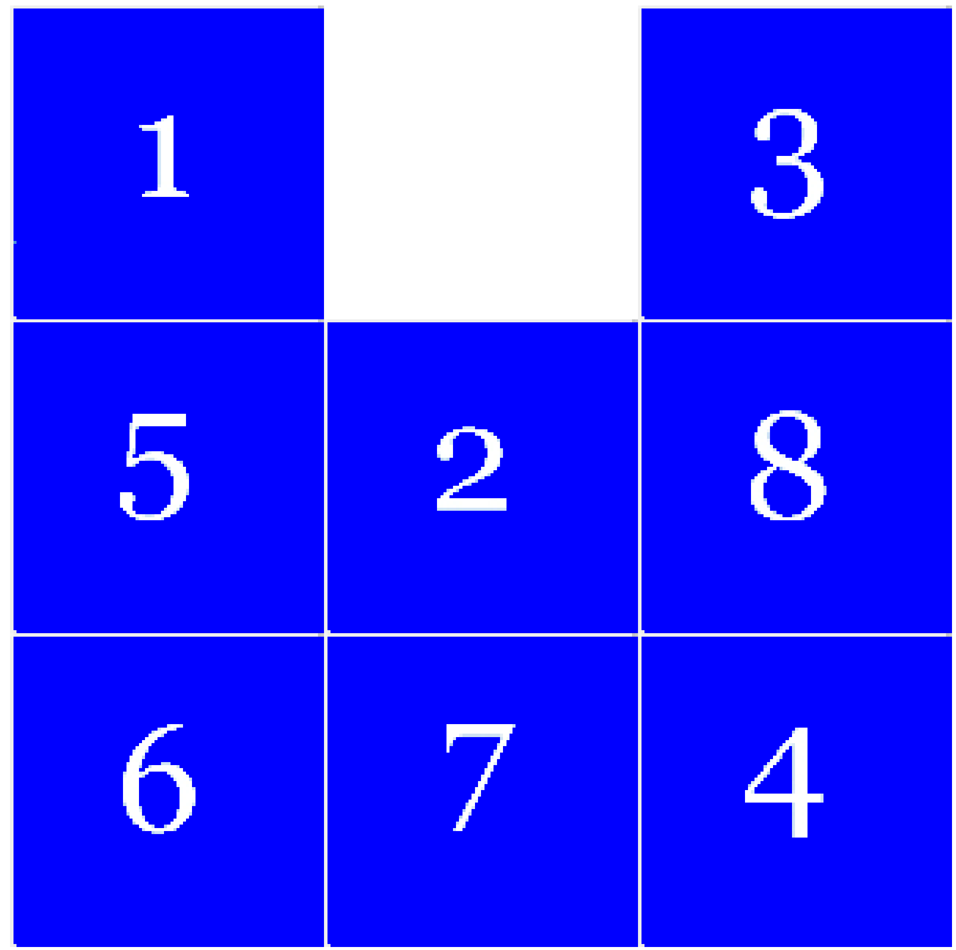


Properties of breadth-first search

- Complete? Yes (if *b* is finite)
- Time? $1+b+b^2+b^3+\ldots +b^d + b(b^d-1) = O(b^{d+1})$
- Space? $O(b^{d+1})$ (keeps every node in memory)
- Optimal? Yes (if cost = 1 per step)

Choosing a state space

1. States:
2. Actions :
3. Goal test:
4. Cost:



8-puzzle heuristics

Admissible:

- h1 : Number of misplaced tiles

  = 6

- h2: Sum of Manhattan distances of the tiles from their goal positions

  = 0+0+1+1+2+3+1+3=11

1		3
5	2	8
6	7	4

goal:

1	2	3
4	5	6
7	8	

8-puzzle heuristics

Nilsson's Sequence

$$Score(n) = P(n) + 3\ S(n)$$

P(n) : Sum of Manhattan distances of each tile from its proper position

S(n), sequence score : check around the non-central squares, +2 for every tile not followed by its proper successor and 0 for every other tile. piece in center = +1