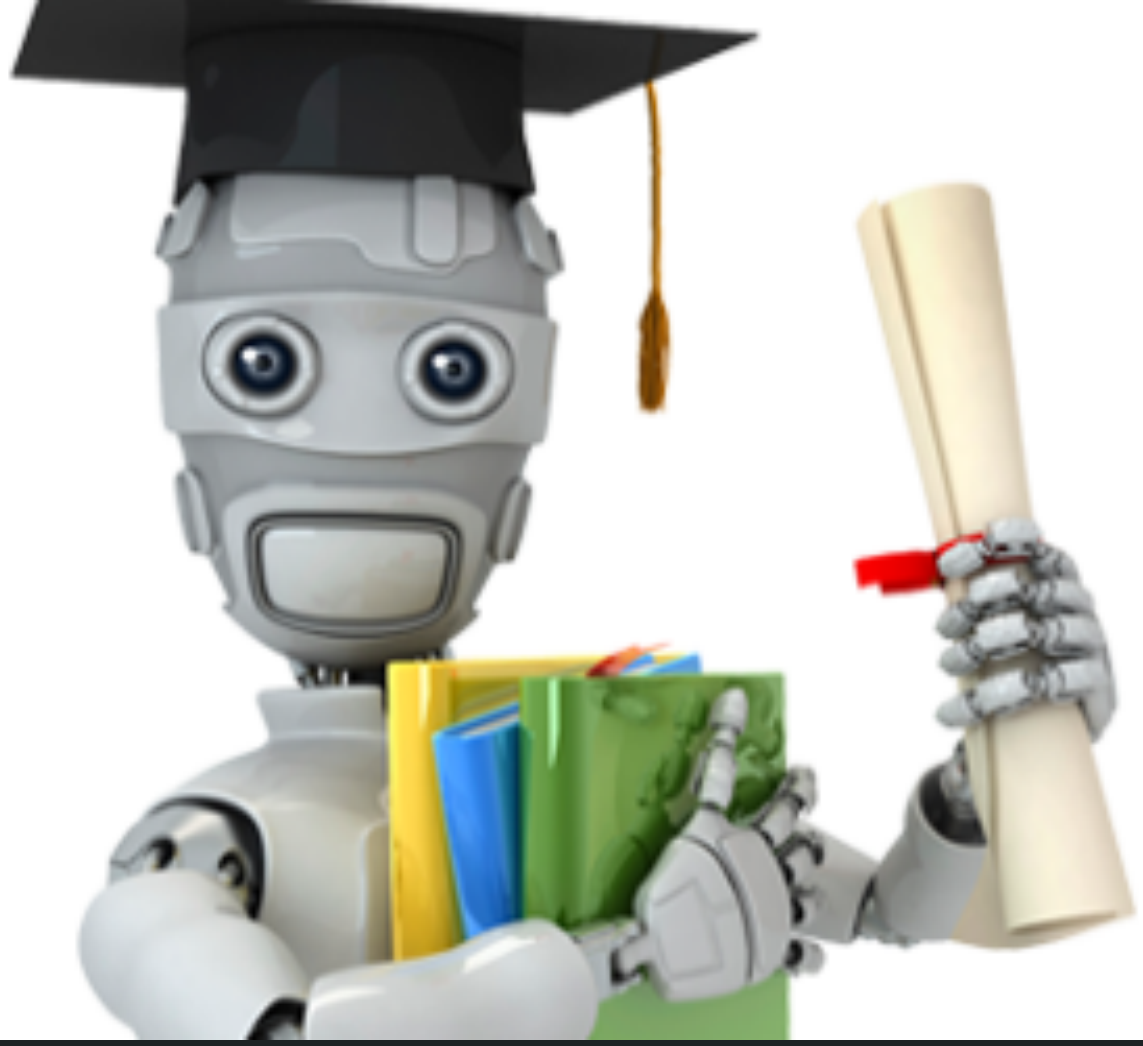


image credit: coursera machine learning class

Machine Learning

teaching the computer to be a bit smarter

CS 4100/5100
Foundations of AI

upcoming deadlines, requirements

FINAL PROJECTS

Upcoming Deadlines

- Mid-project checkpoint: November 19th
- Presentations: November 29th, December 6th
  - We will schedule these today
- Final Turn-In: December 13th

Mid-Project Checkpoint: November 19th

- Short (2-3 paragraphs) written summary
  - What have you done so far?
    - For each group member!
  - How much time have you put into it?
  - Have your goals changed from the proposal?
  - What do you intend to do to finish the project?

Project Presentations: In Class

- 19 groups total
  - 10 groups will present one week, 9 will present the other week
- 15 minute presentation
  - 10-12 minutes talk
  - 3-5 minutes questions **from the audience**
- Your whole group must present
  - Exceptions require prior permission

Project Presentations: In Class

- Requirements
  - What is the problem you are trying to solve/question you are trying to answer?
  - What have other people tried doing to solve it?
  - What method have you used, and why?
  - What are your results so far?

- Evaluation Criteria
  - Peer evaluation
  - Content, organization, preparedness, clarity, visuals

Class Activity: Giving a Good Presentation

- What are some characteristics of great presentations you've seen?
- What do you really hate to see in presentations?
- How does your audience change the way you give talks or put together slides?

Project Reports: December 13th

- Expect 3-4 pages, AAAI format
  - No more than 6, no less than 2
- Structure
  - Introduction
  - Related Work
  - Approach
  - Results and Discussion
  - References

Turning in Final Project

- December 13th:
  - Final report
  - Zip file with your code/data/results **and instructions for running it**

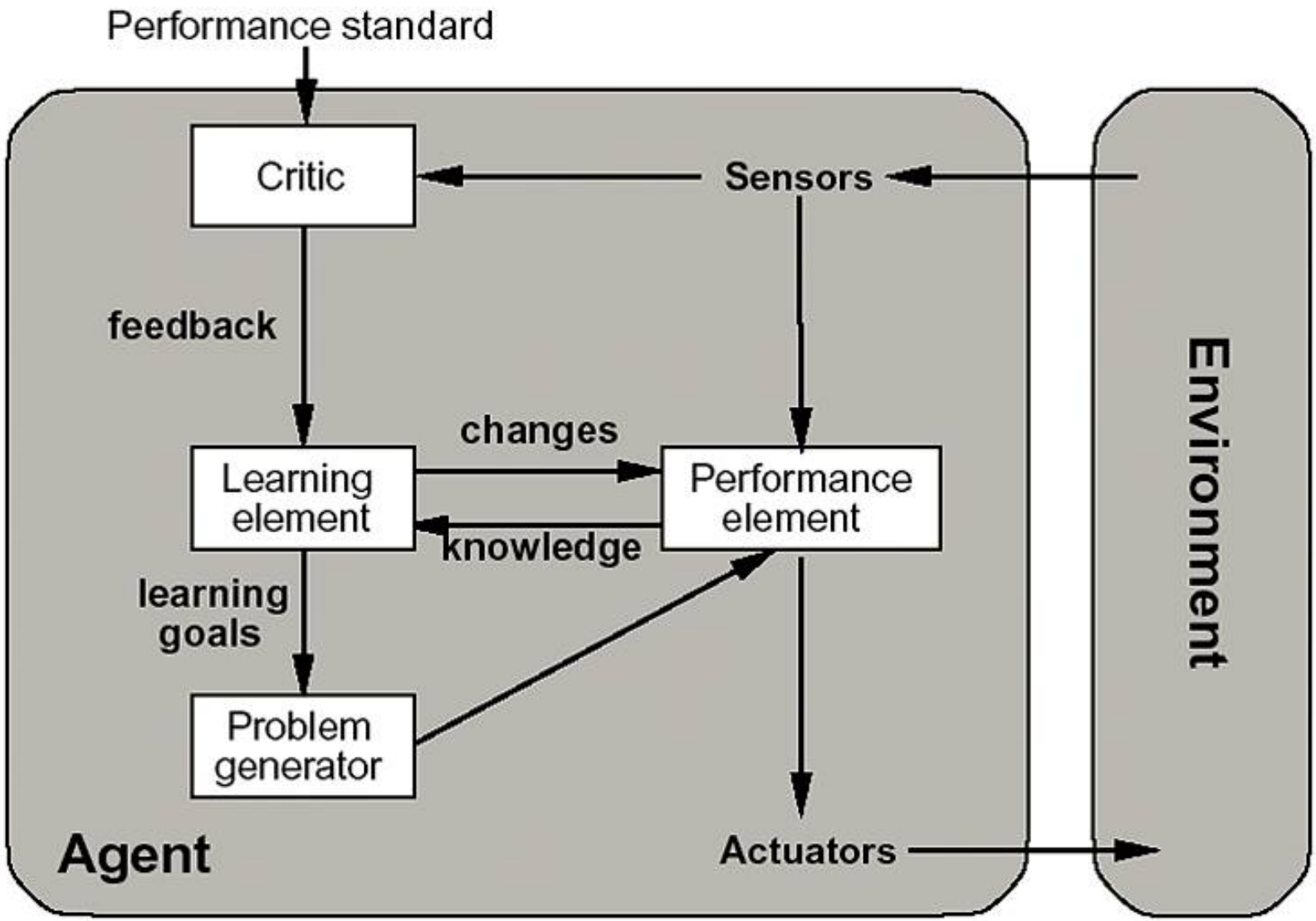
- 10-minute appointment with me to demo your project
  - Optional unless your code is hard to run!
  - During the final exam period (??) or office hours December 11th

WHAT IS MACHINE LEARNING?

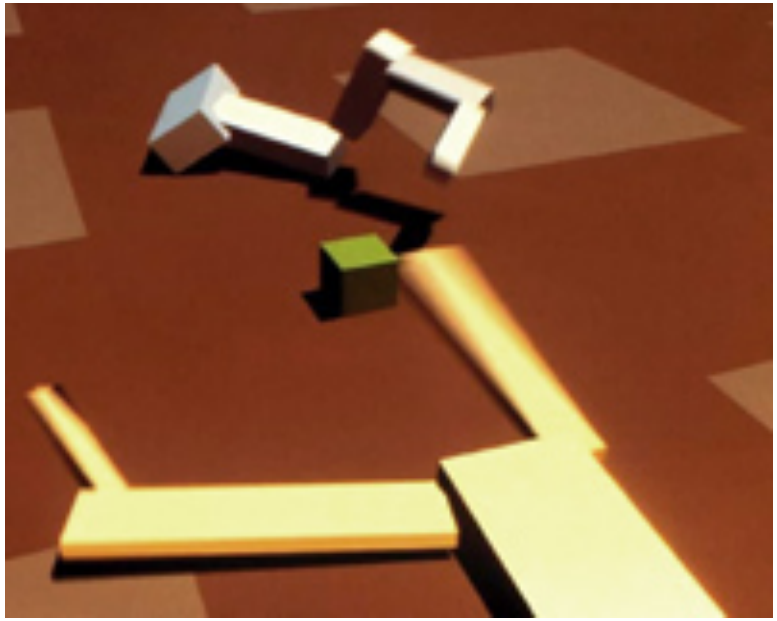
Why Machine Learning?

- Uncertain or changing environment
- Don’t know how to program it
- We believe that’s what it means to be intelligent

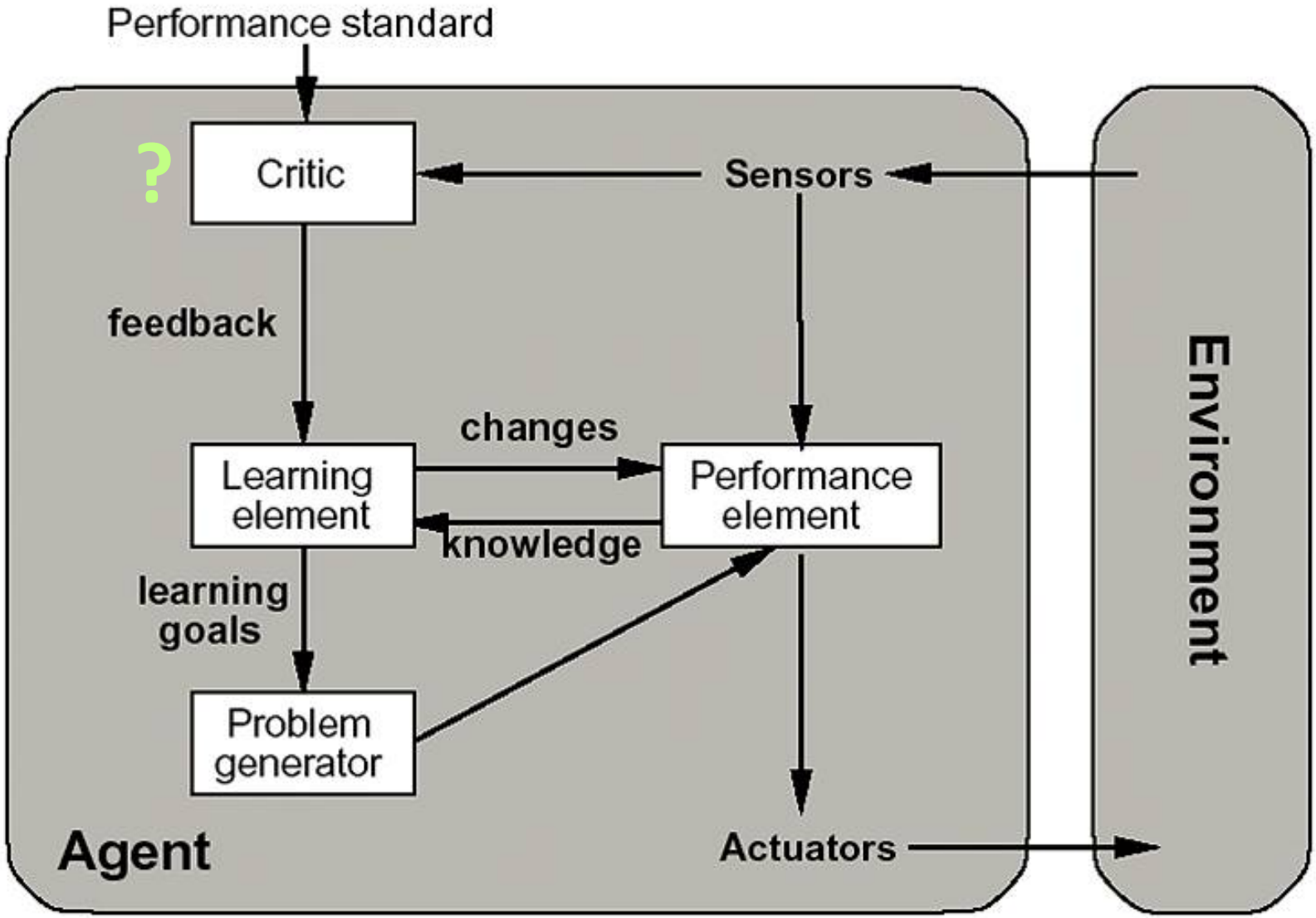
The Learning Agent



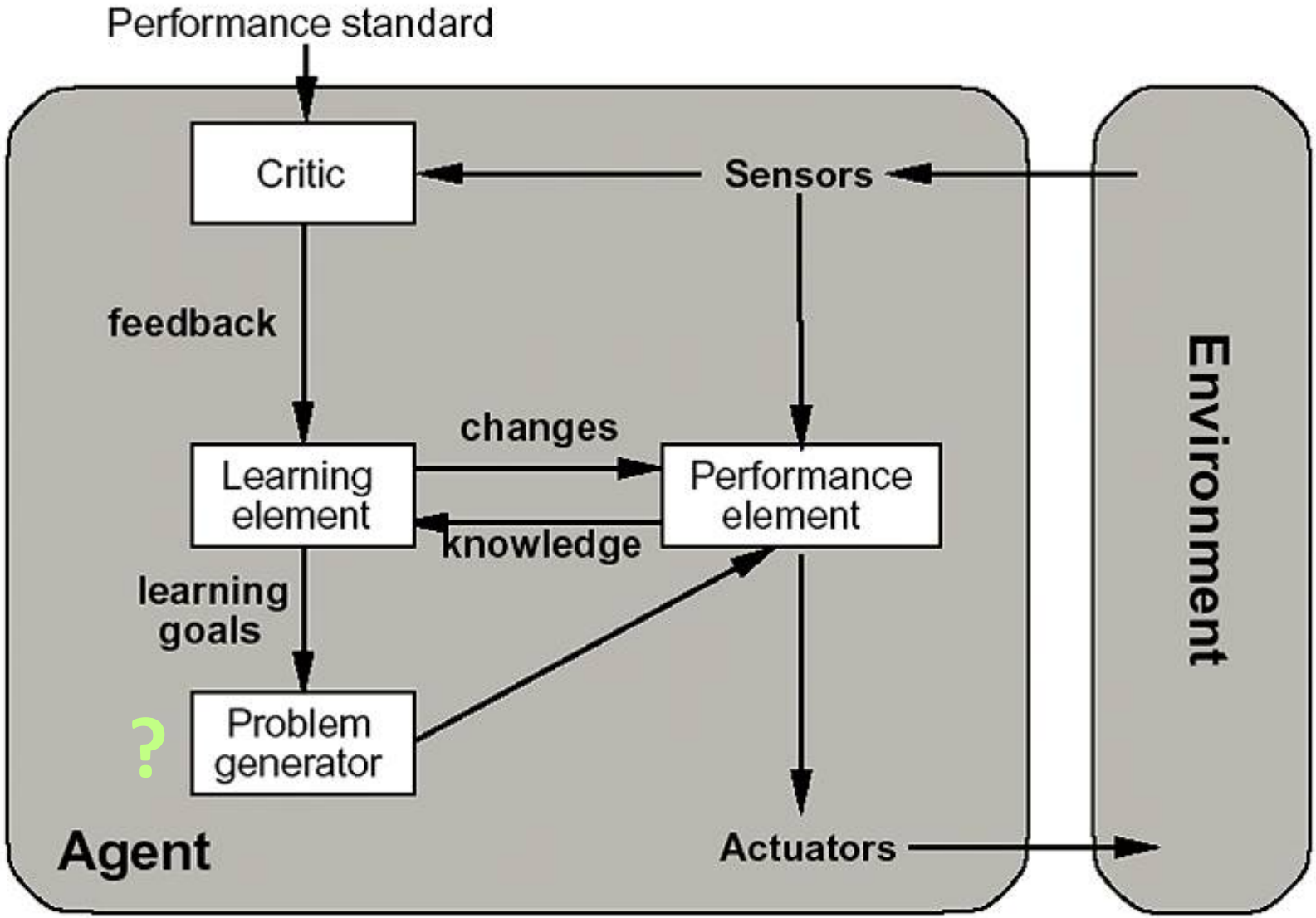
Example: Genetic Algorithms



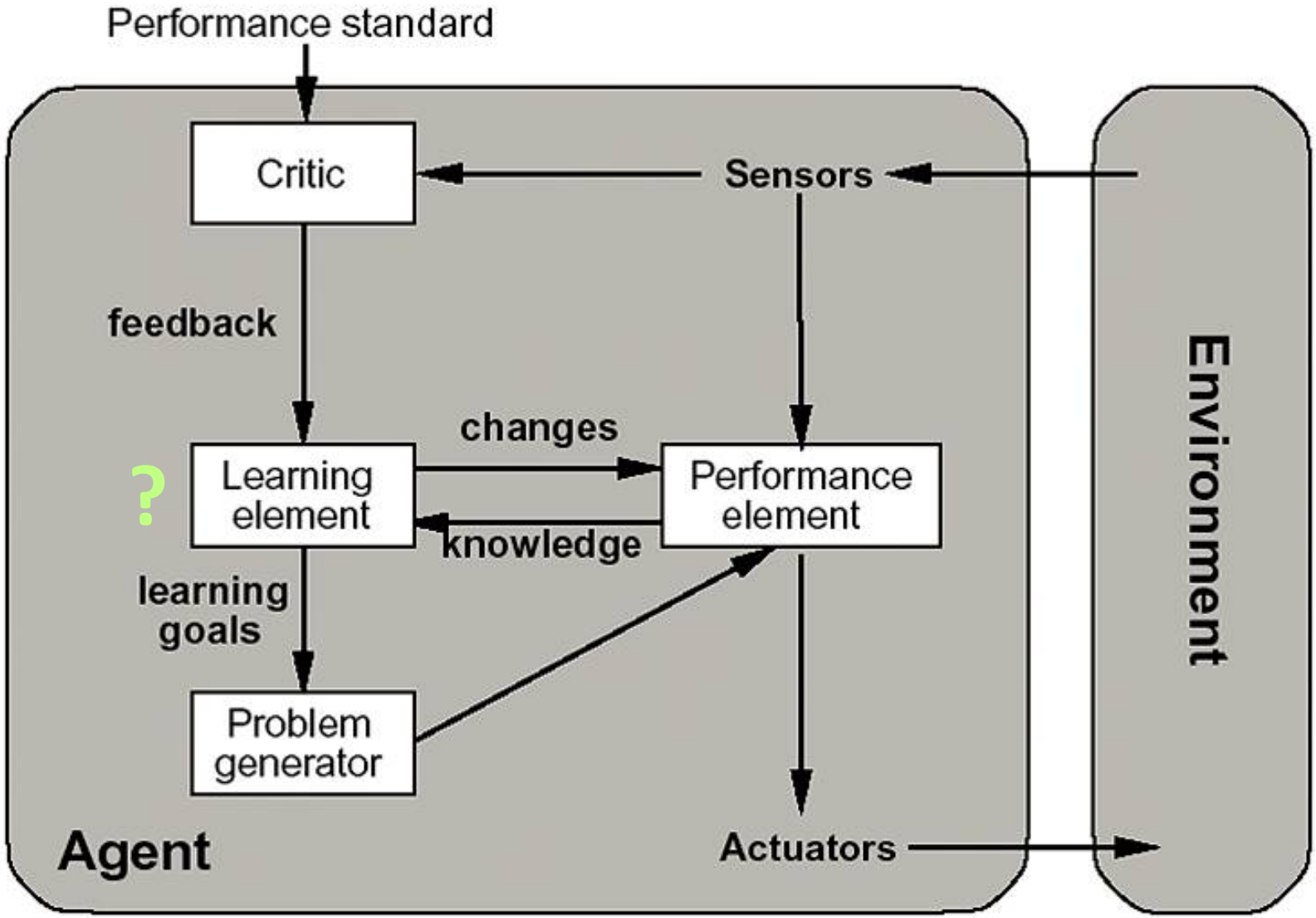
Example: Genetic Algorithms



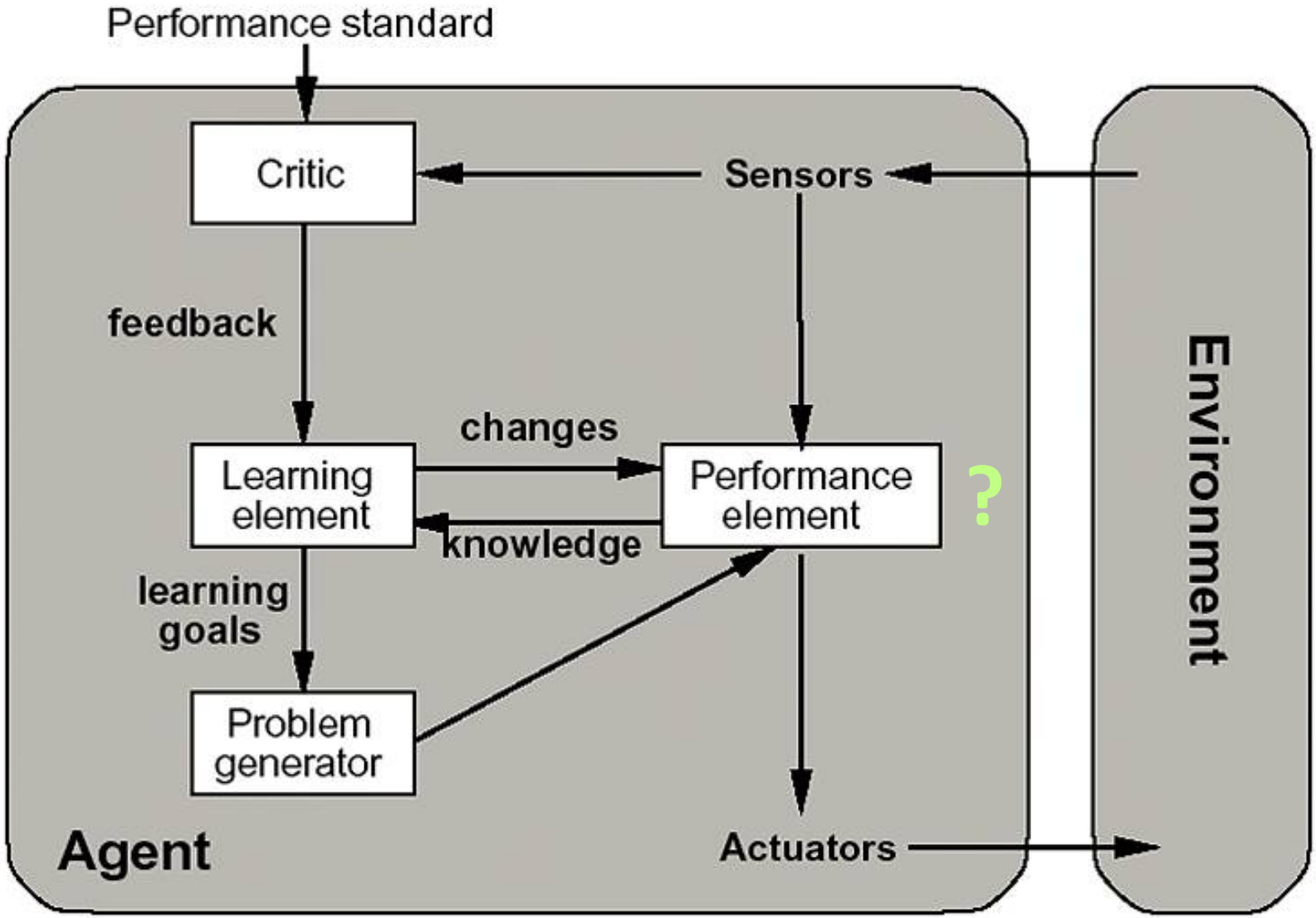
Example: Genetic Algorithms



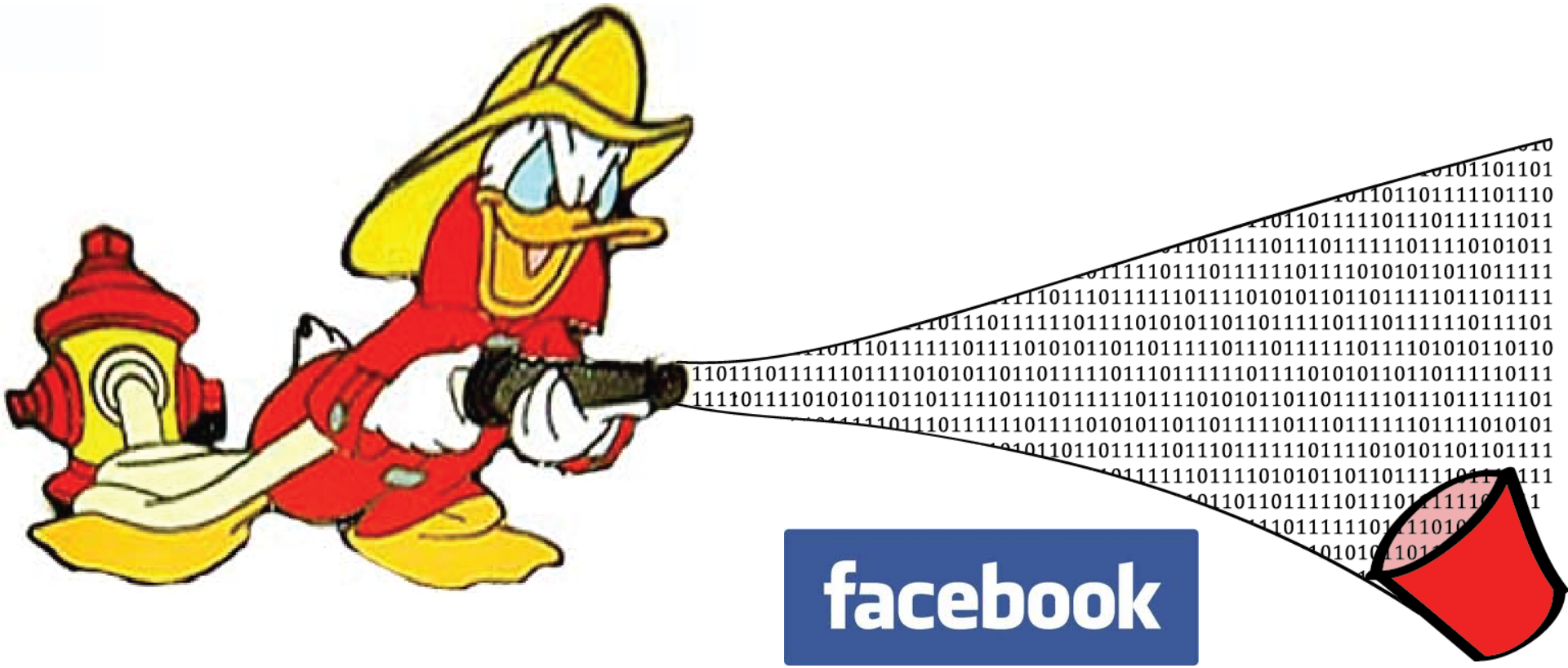
Example: Genetic Algorithms



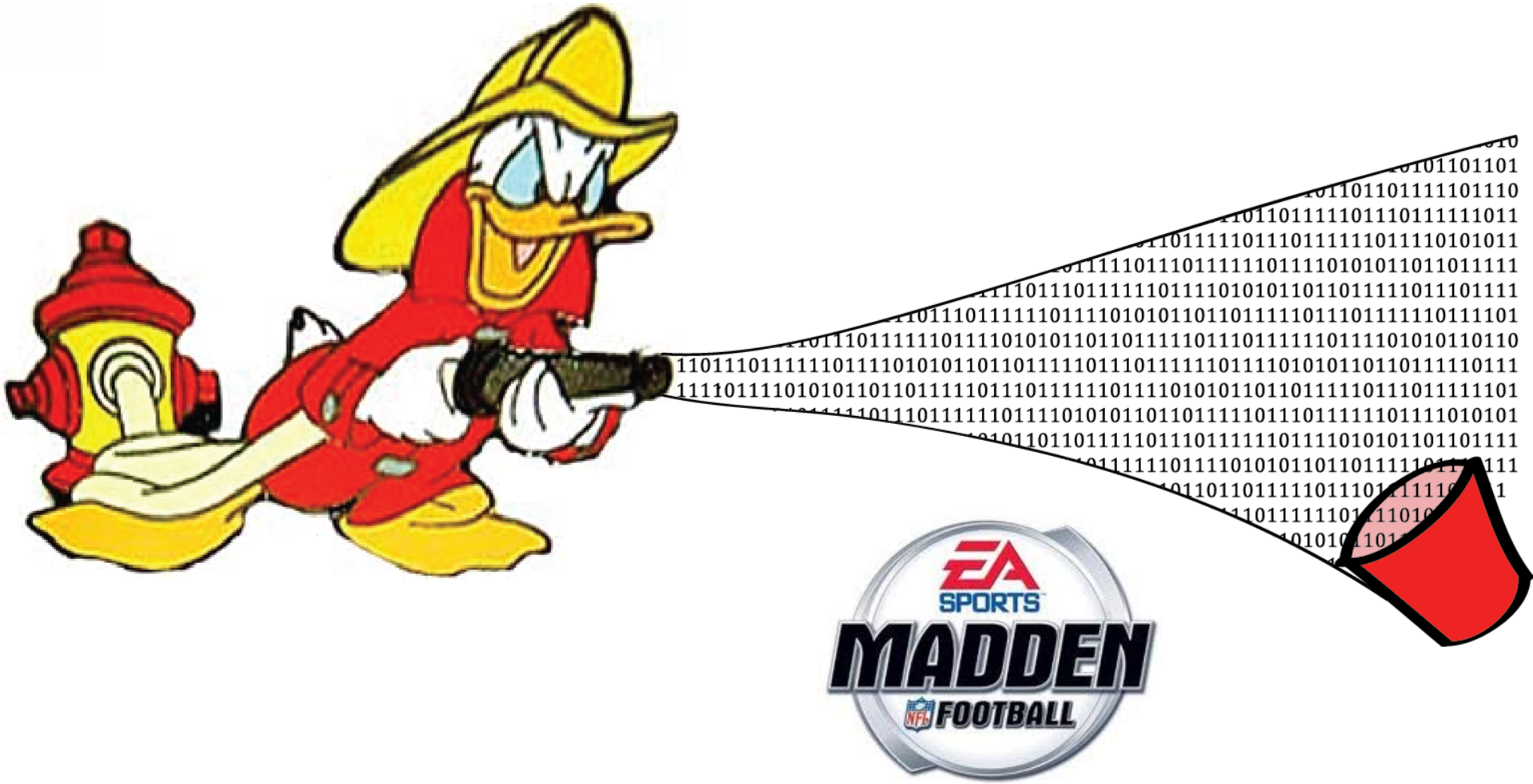
Example: Genetic Algorithms



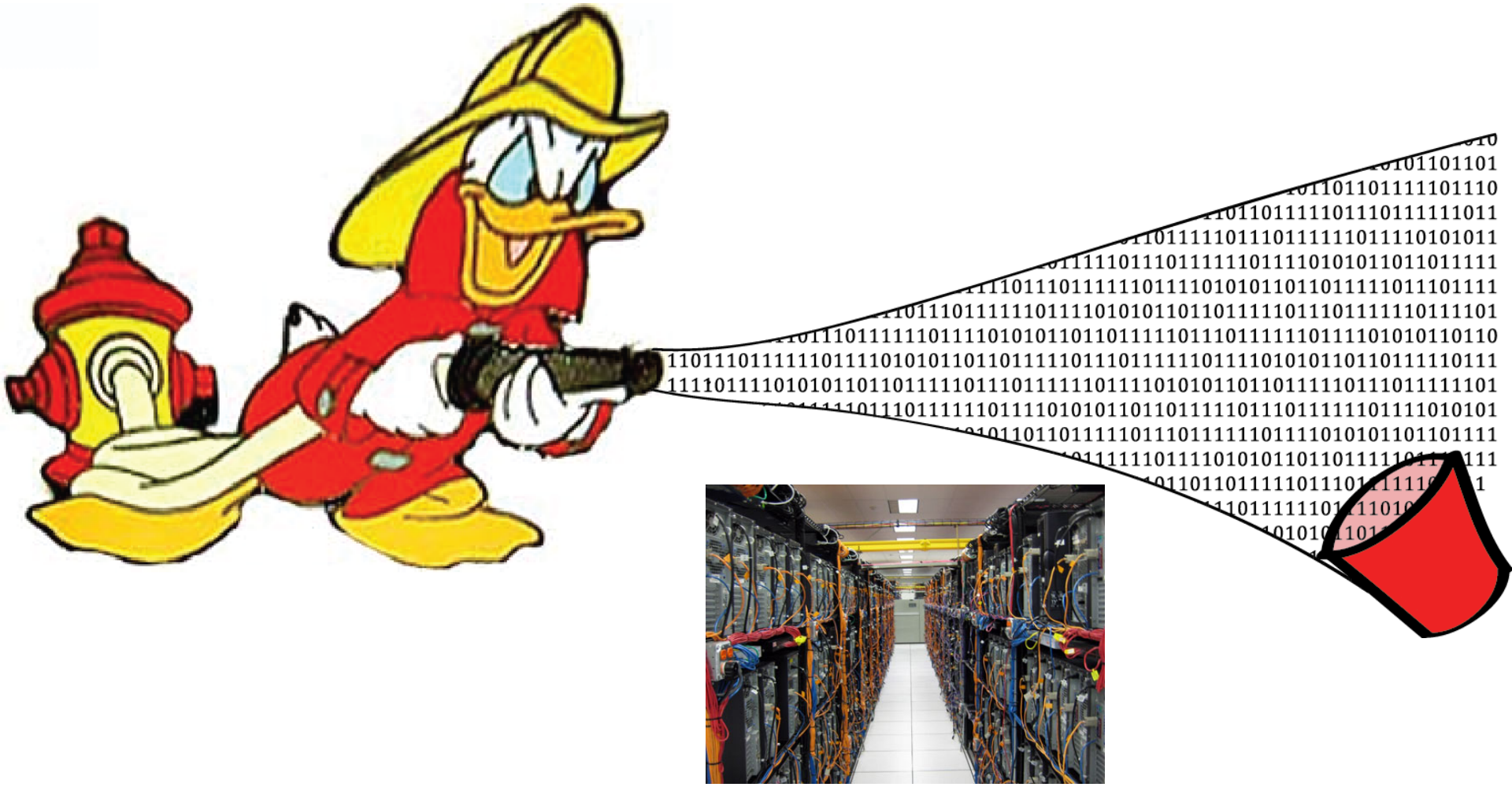
Big Data



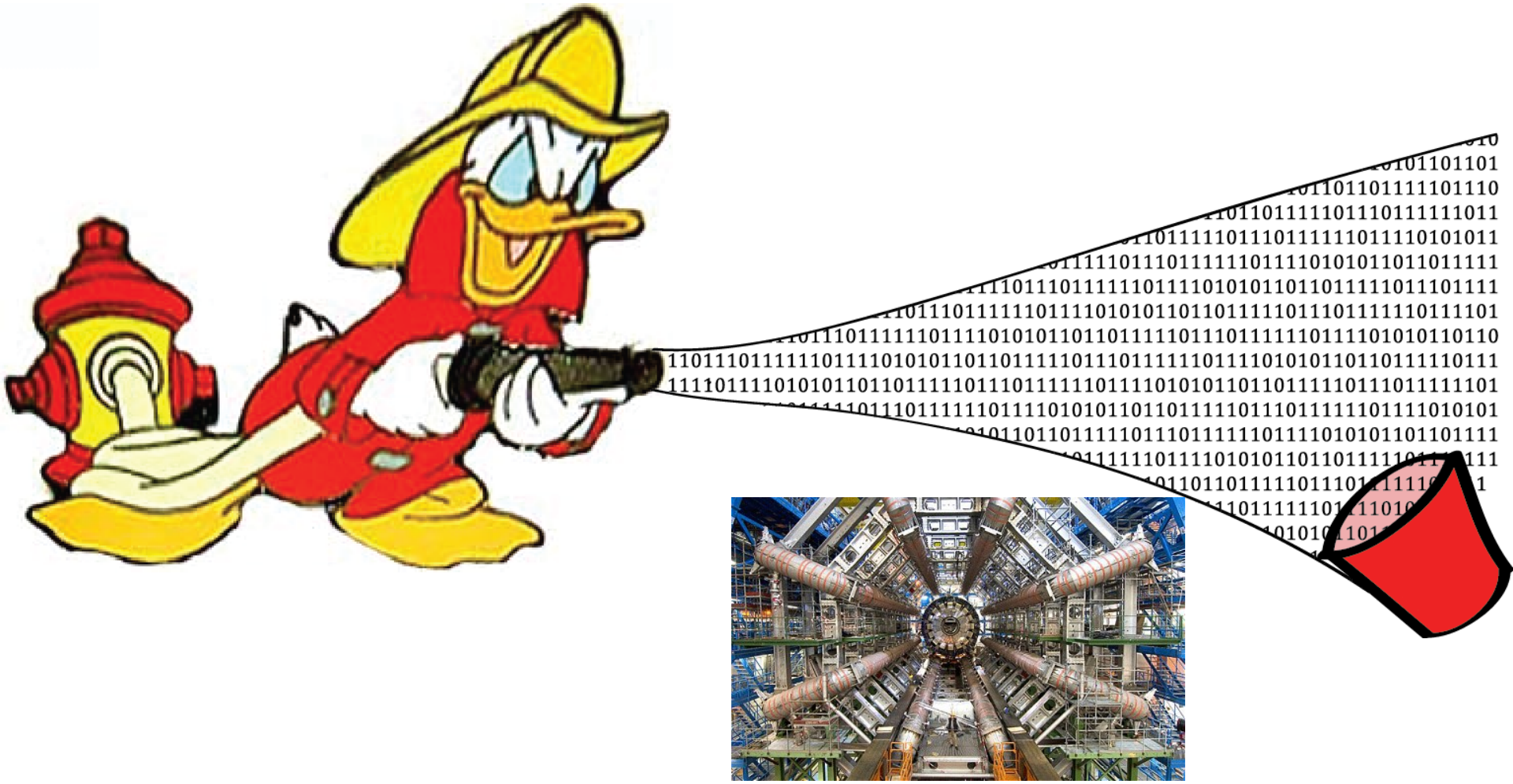
Big Data



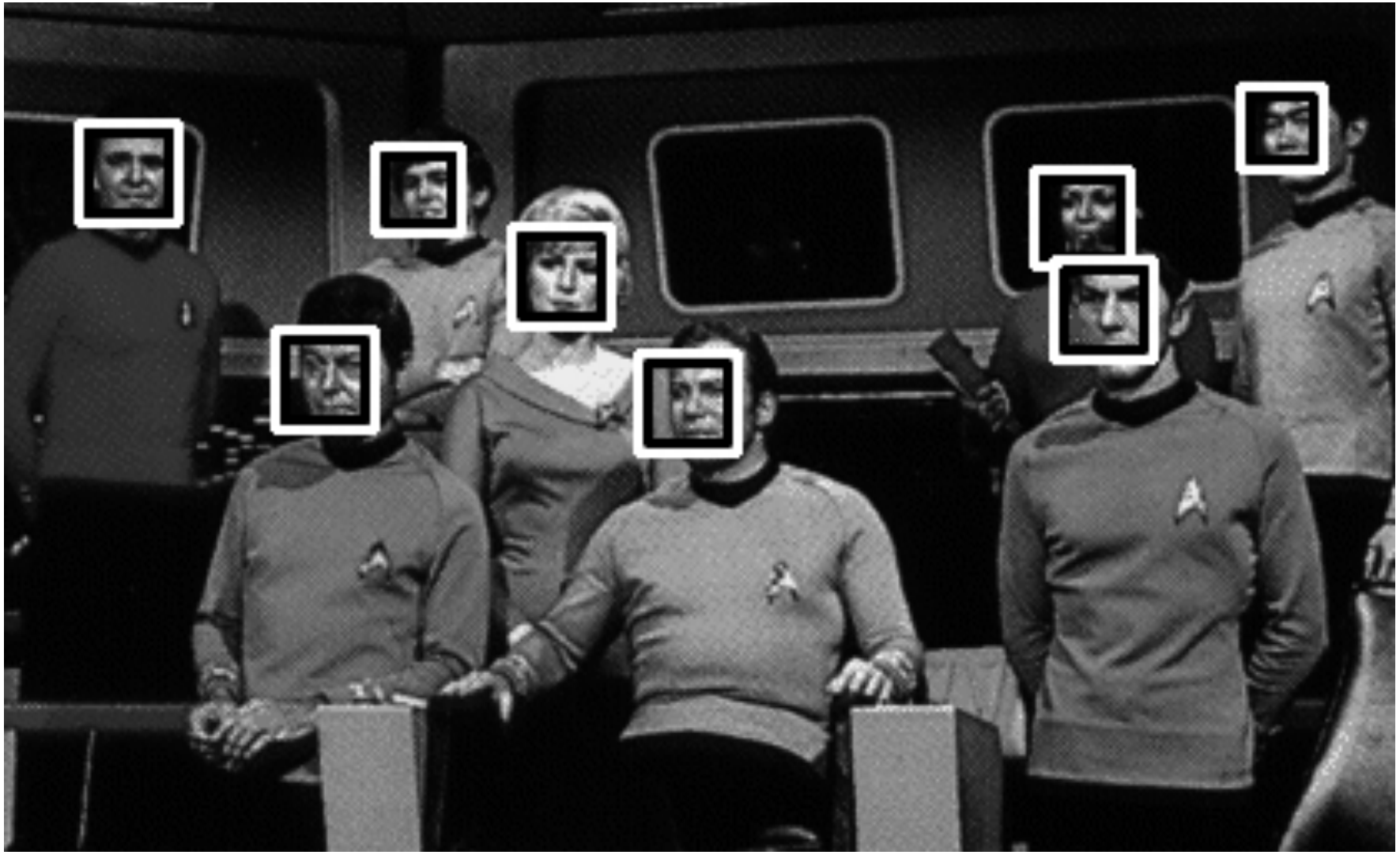
Big Data



Big Data



Applications: Computer Vision



Applications: Cyber Security

Applications: Analytics

Qualities of Data

- Factored into attributes

- Structured vs. unstructured

Kinds of Machine Learning

- Unsupervised Learning
- Supervised Learning
- Reinforcement Learning

Kinds of Machine Learning

- Unsupervised Learning
- Supervised Learning

semi-supervised learning

- Reinforcement Learning

clustering

UNSUPERVISED LEARNING

Unsupervised Learning

- Unlabeled data
- Find patterns or anomalies

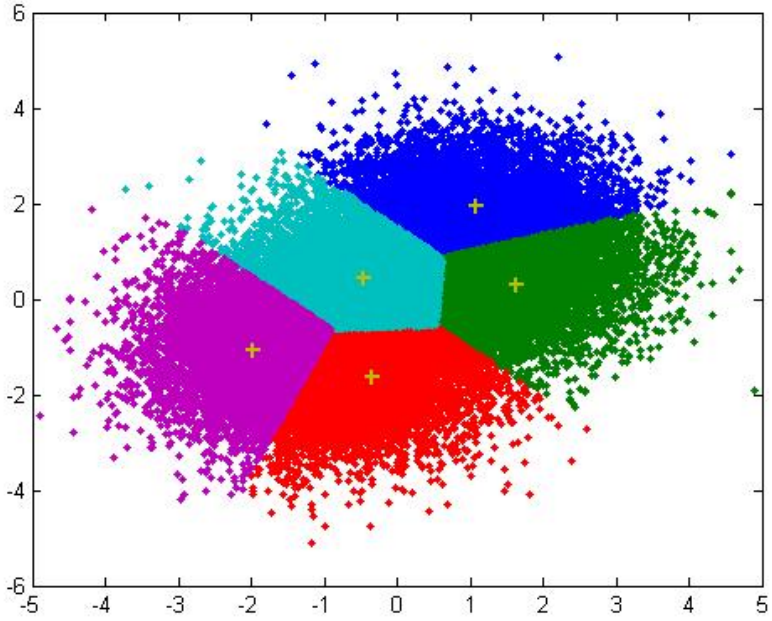
Example: Clustering for Image Segmentation

source: http://www.ai.mit.edu/courses/6.891/lectnotes/lect14/lect14-slides.pdf

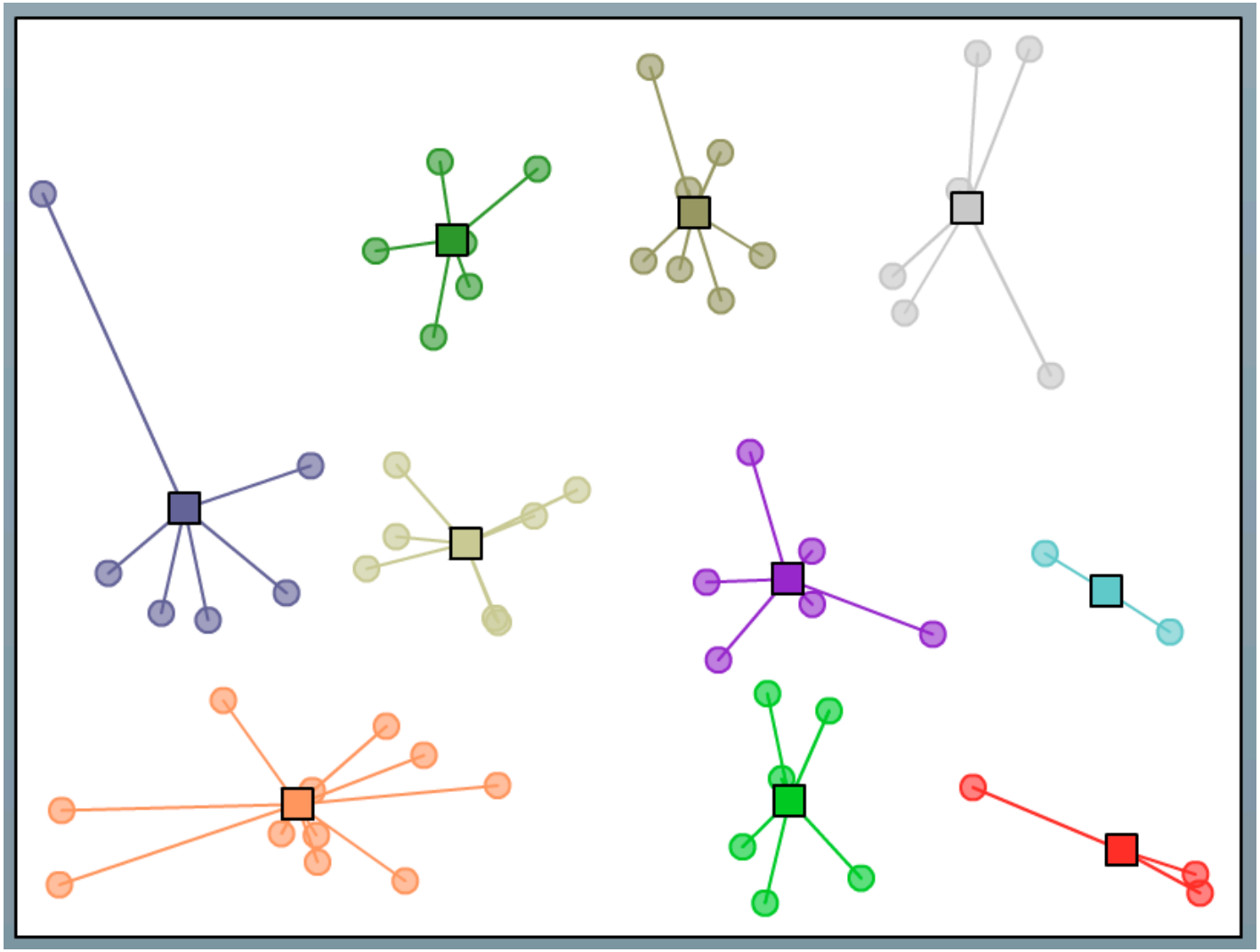
K-means Clustering

- How many clusters do you want? ***K***

- Pick a random K points in space to be the center of your clusters
- Until cluster centers do not change
  - Assign every data point to closest cluster center
  - Update cluster center to be centroid of newly formed cluster



Interactive Demo

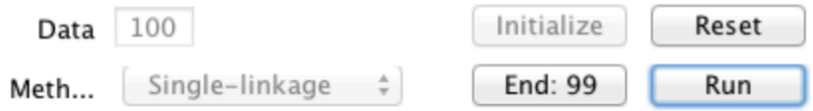


K-means clustering

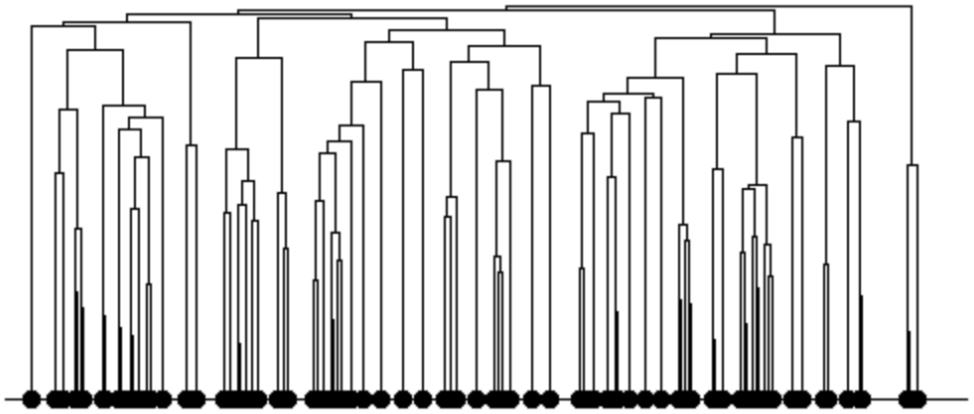
- The Good
  - Simple
  - Fast
  - Does a reasonable job for simple clusters
- The Bad and/or Ugly
  - What is k?
  - Non-overlapping clusters
  - Sensitive to outliers

Hierarchical clustering

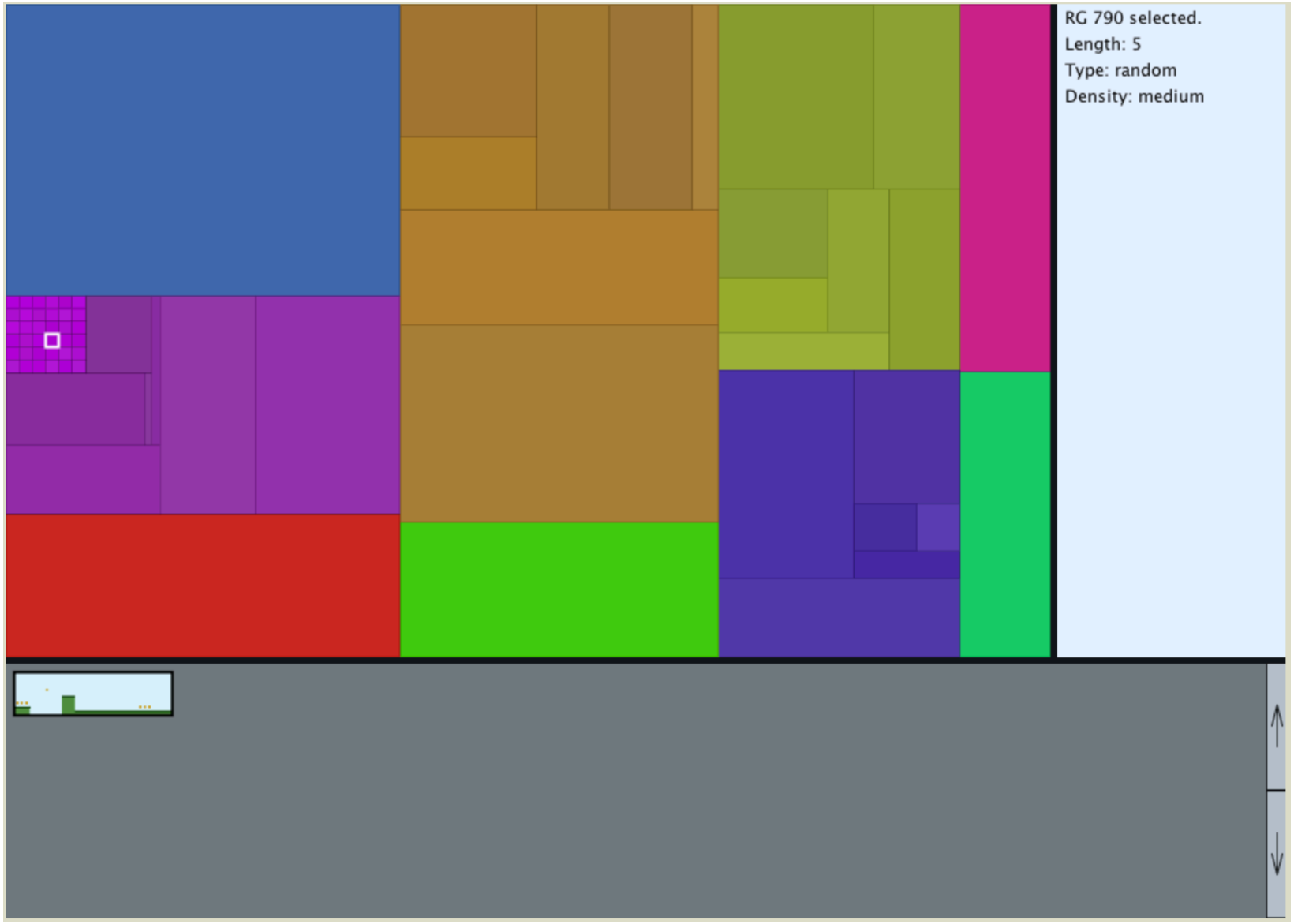
- Construct a hierarchy of how data points are related to each other



- Start with each datapoint as a "cluster"
- Iteratively merge closest clusters together



Example: Classifying Generated Levels



SUPERVISED LEARNING

What is Supervised Learning?

- Learning from example data
- Labeled data with outcomes
- Data consists of attribute-value pairs

Training Sets

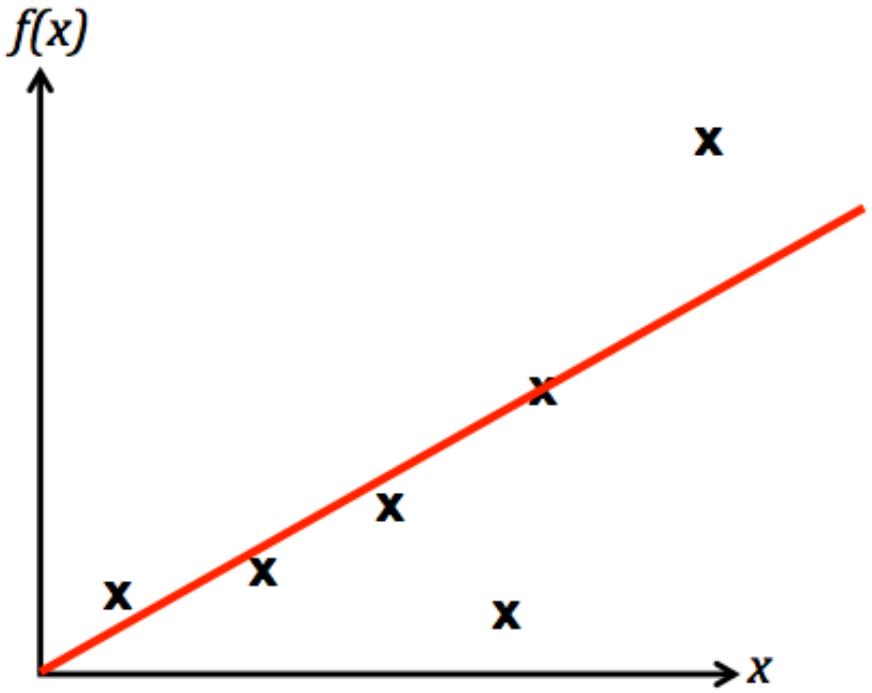
- The data that the computer will learn from
- Needs to be different from test set!
  - But be a nice representative sample

Inductive Learning (aka Science)

- *f*: target function that actually explains data
- *h:* the hypothesis given a training set of examples
- Simplifies real learning
  - Ignores prior knowledge
  - Assumes a fully observable environment
  - Assumes (good) examples are given

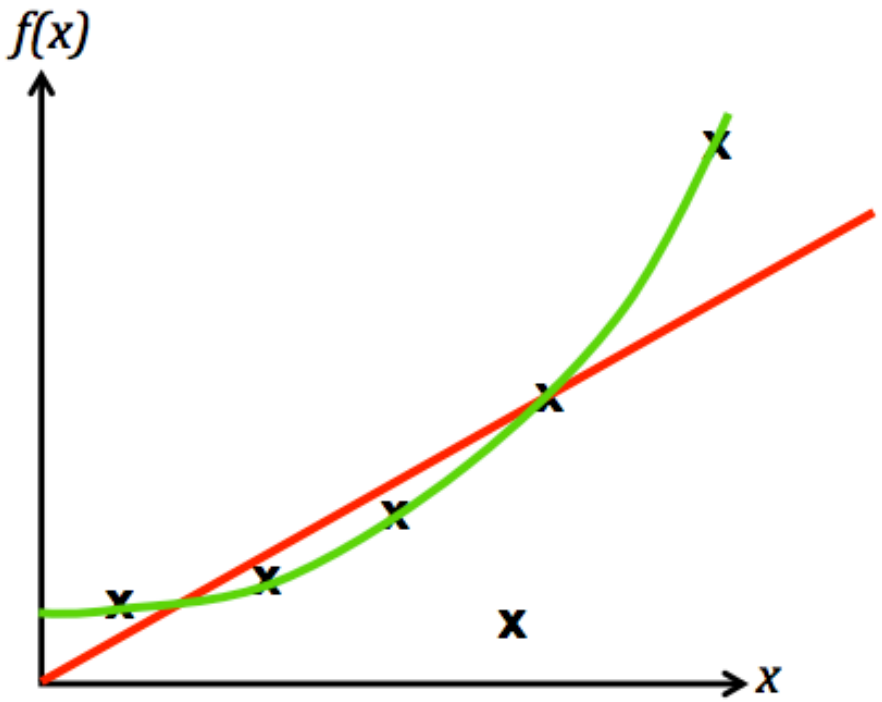
Inductive Learning Method

- Construct/adjust *h* to agree with *f* on the training data



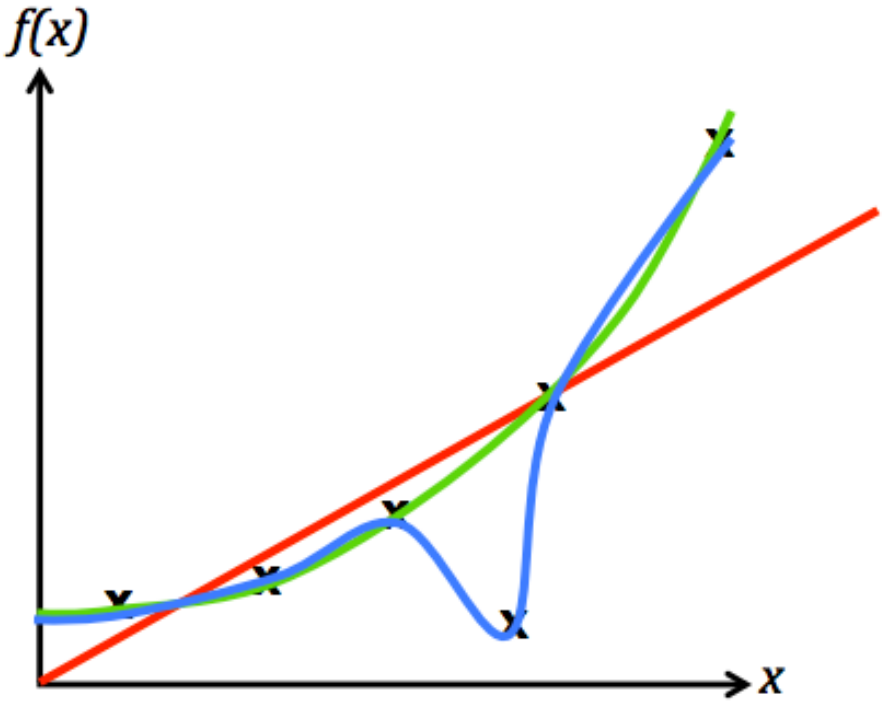
Inductive Learning Method

- Construct/adjust *h* to agree with *f* on the training data



Inductive Learning Method

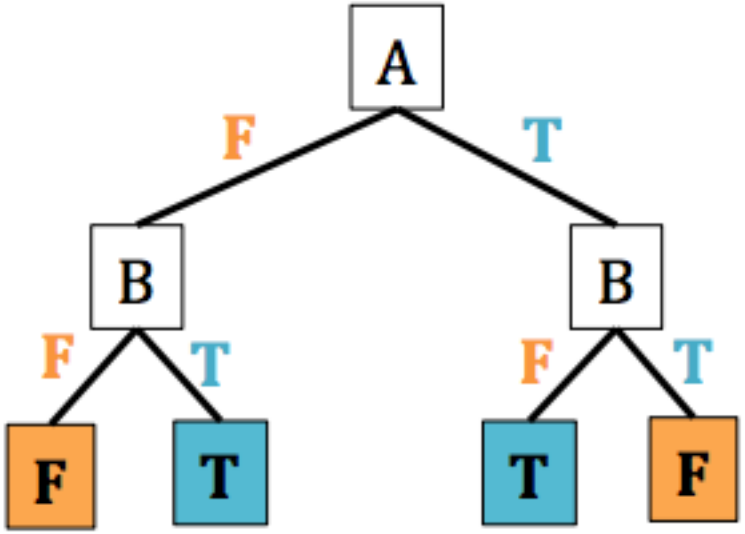
- Construct/adjust *h* to agree with *f* on the training data



Decision Trees

- Tree representation that can express any function of input attributes

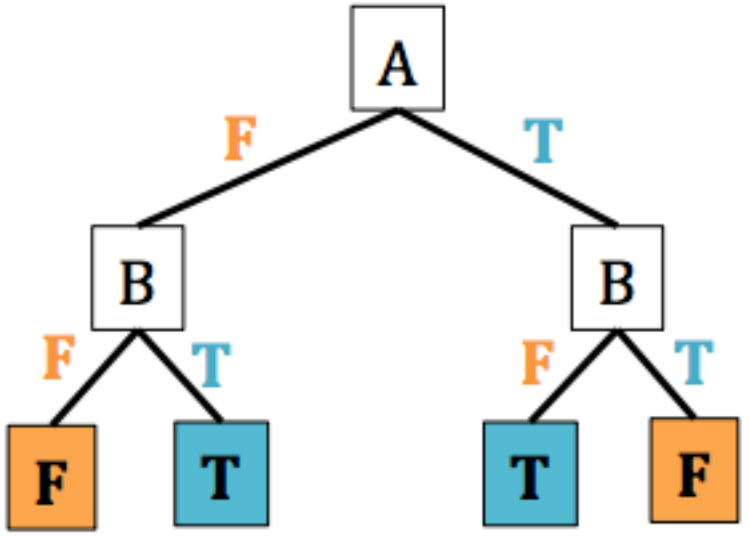
A	B	A xor B
F	F	F
F	T	T
T	F	T
T	T	F



Hypothesis Space

- How many distinct decision trees are there for n boolean attributes?
  - Number of Boolean functions = 2^n
  - Number of distinct truth tables with 2^n rows = 2^{2^n}

A	B
F	F
F	T
T	F
T	T

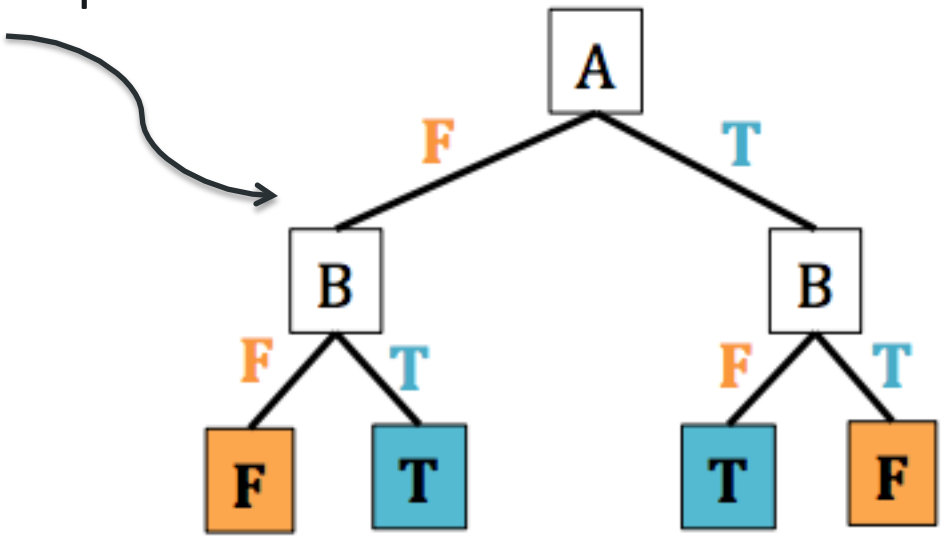


Hypothesis Space

- How many distinct decision trees are there for n boolean attributes?
  - Number of Boolean functions = 2^n
  - Number of distinct truth tables with 2^n rows = 2^{2^n}

A	B
F	F
F	T
T	F
T	T

I'm 1 of 16 potential trees!

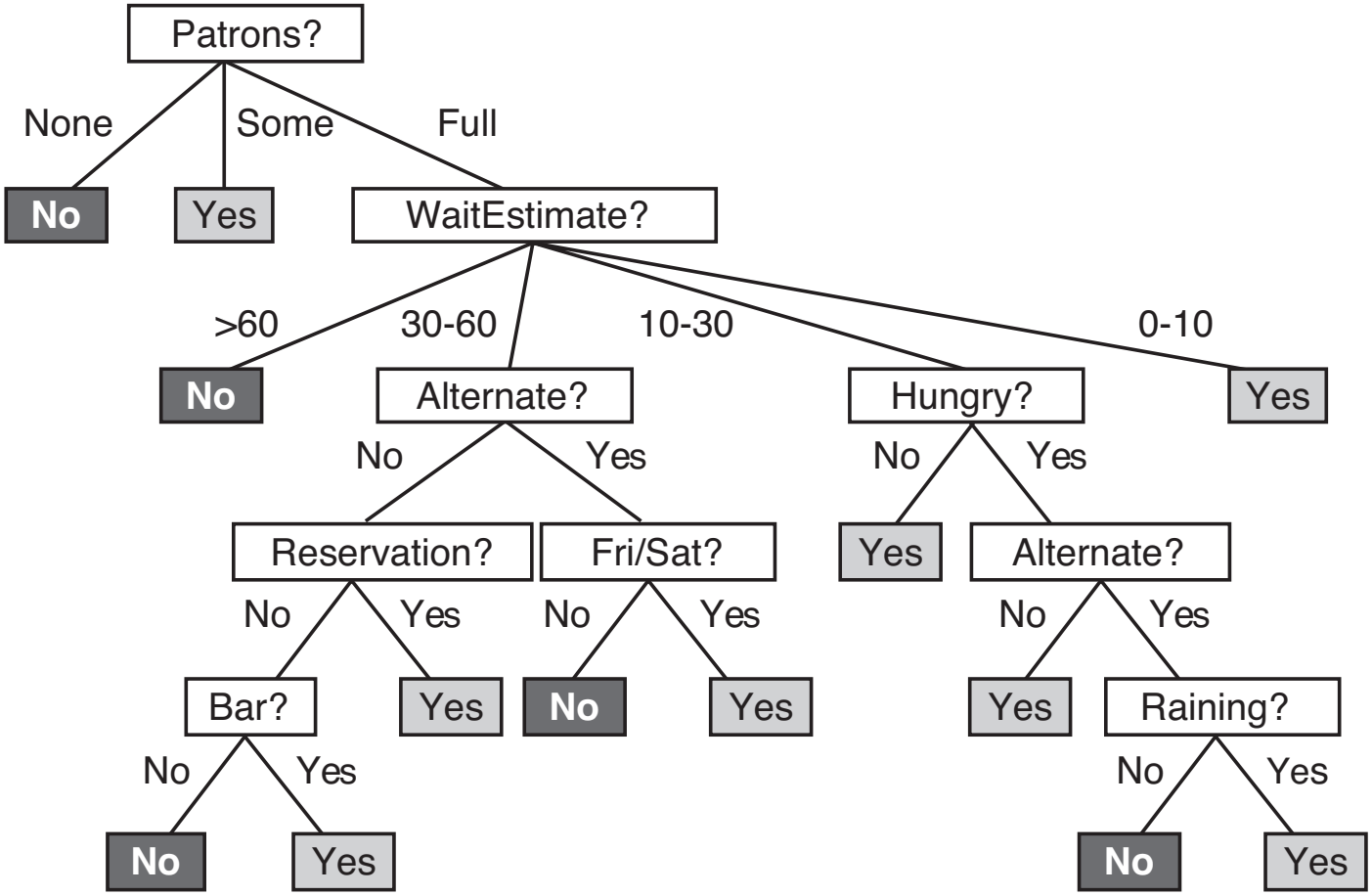


Hypothesis Space

- How many distinct decision trees are there for n boolean attributes?
  - Number of Boolean functions = 2^n
  - Number of distinct truth tables with 2^n rows = 2^{2^n}

- For a table with 6 boolean attributes?
  - 2^{2^n} = **18,466,744,073,709,551,616** trees

Decision Trees



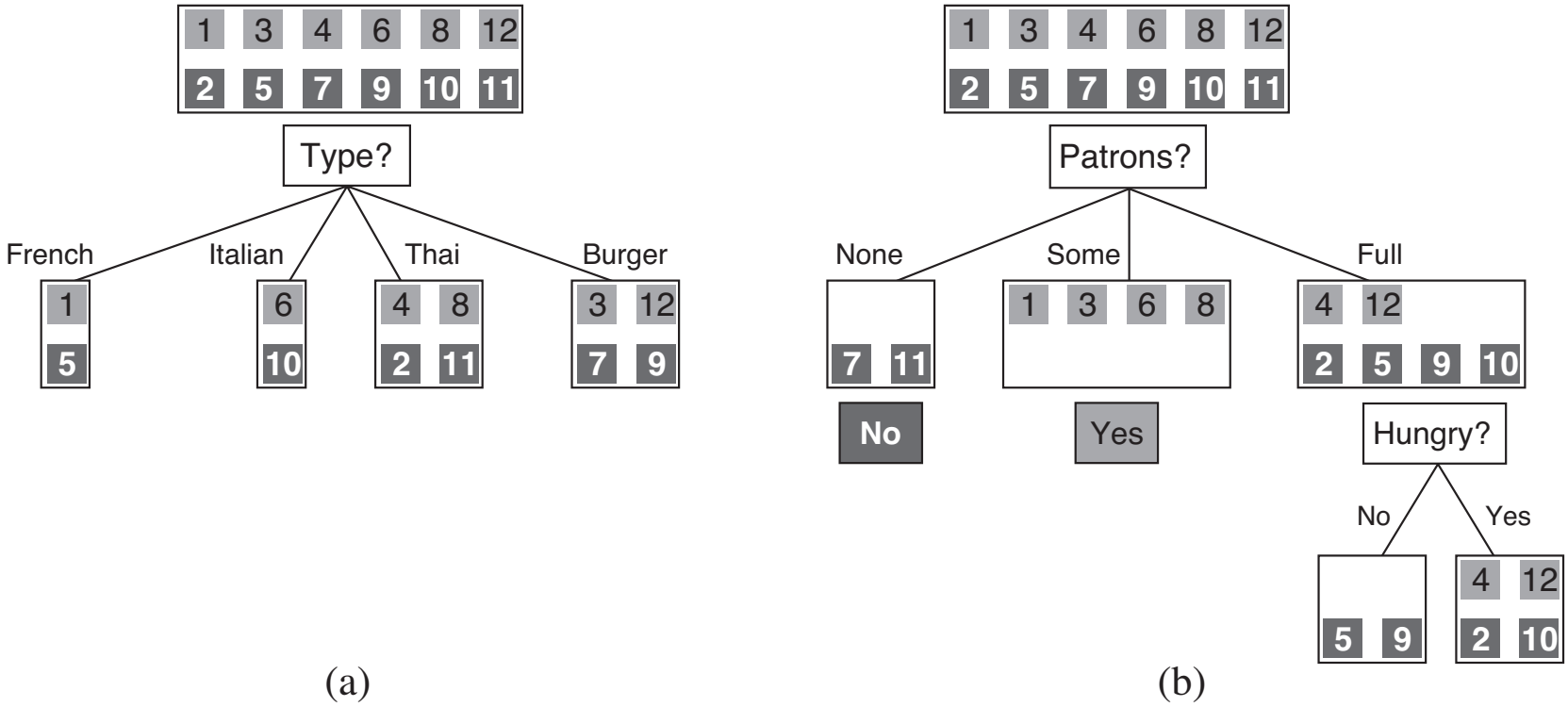
Decision Tree Learning

- Goal: find a small tree consistent with training
- Intuition: choose **most significant** attribute as root of (sub)tree

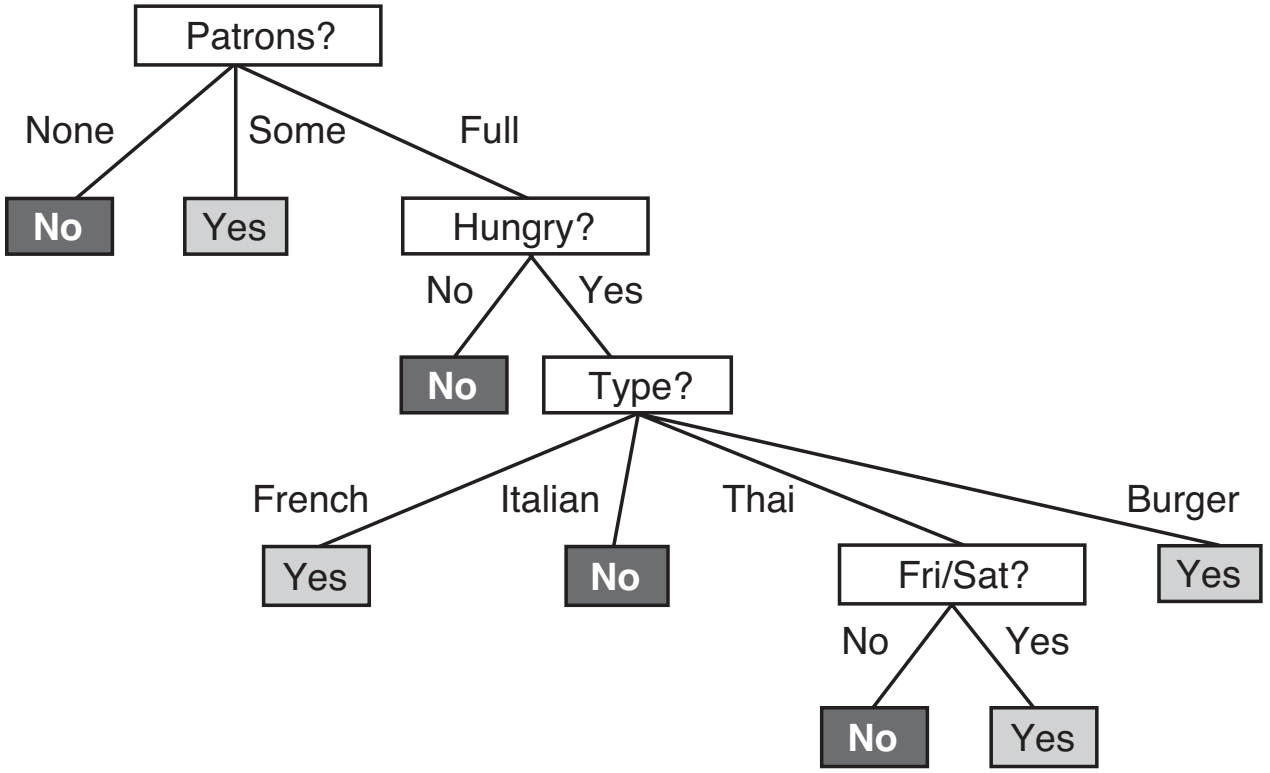
```
function DECISION-TREE-LEARNING(examples, attributes, parent_examples) returns
tree

  if examples is empty then return PLURALITY-VALUE(parent_examples)
  else if all examples have the same classification then return the classification
  else if attributes is empty then return PLURALITY-VALUE(examples)
  else
      A ← argmax_{a ∈ attributes} IMPORTANCE(a, examples)
      tree ← a new decision tree with root test A
      for each value v_k of A do
          exs ← {e : e ∈ examples and e.A = v_k}
          subtree ← DECISION-TREE-LEARNING(exs, attributes − A, examples)
          add a branch to tree with label (A = v_k) and subtree subtree
      return tree
```

Choosing an attribute



Final Learned Tree

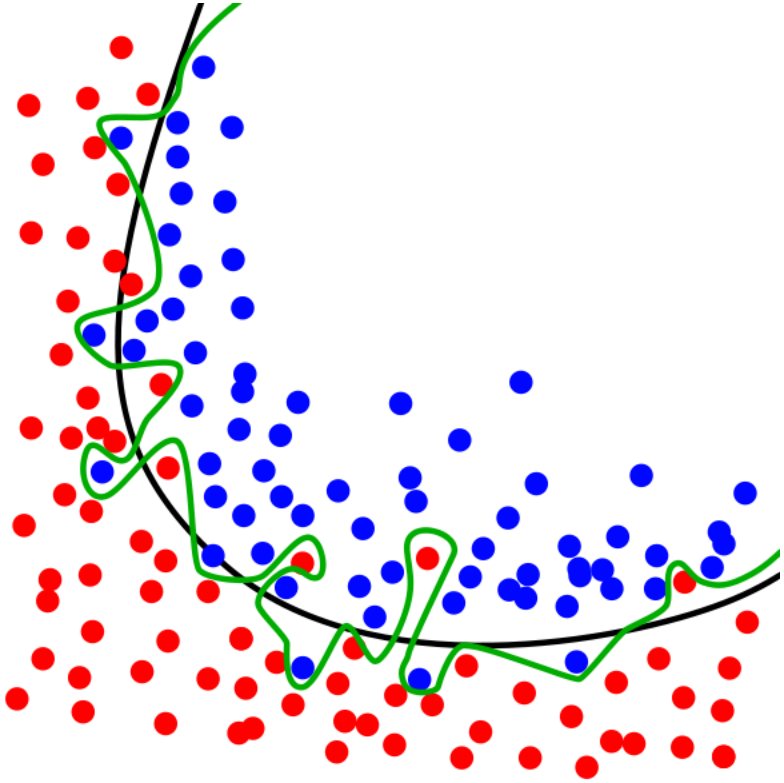


Decision Tree for a Good Talk?

- Given some of the criteria we talked about earlier, what does a decision tree for a good talk look like?

Beware Overfitting

- Overfitting: learning a tree that is too good on the example data and will not generalize to test data



Beware Overfitting

- Overfitting: learning a tree that is too good on the example data and will not generalize to test data

- Accidentally learning the wrong things!
  - When I roll the blue, marbled dice with my left hand after 3pm on Sundays, it will be a 6.

COURSE RECAP

What is AI?

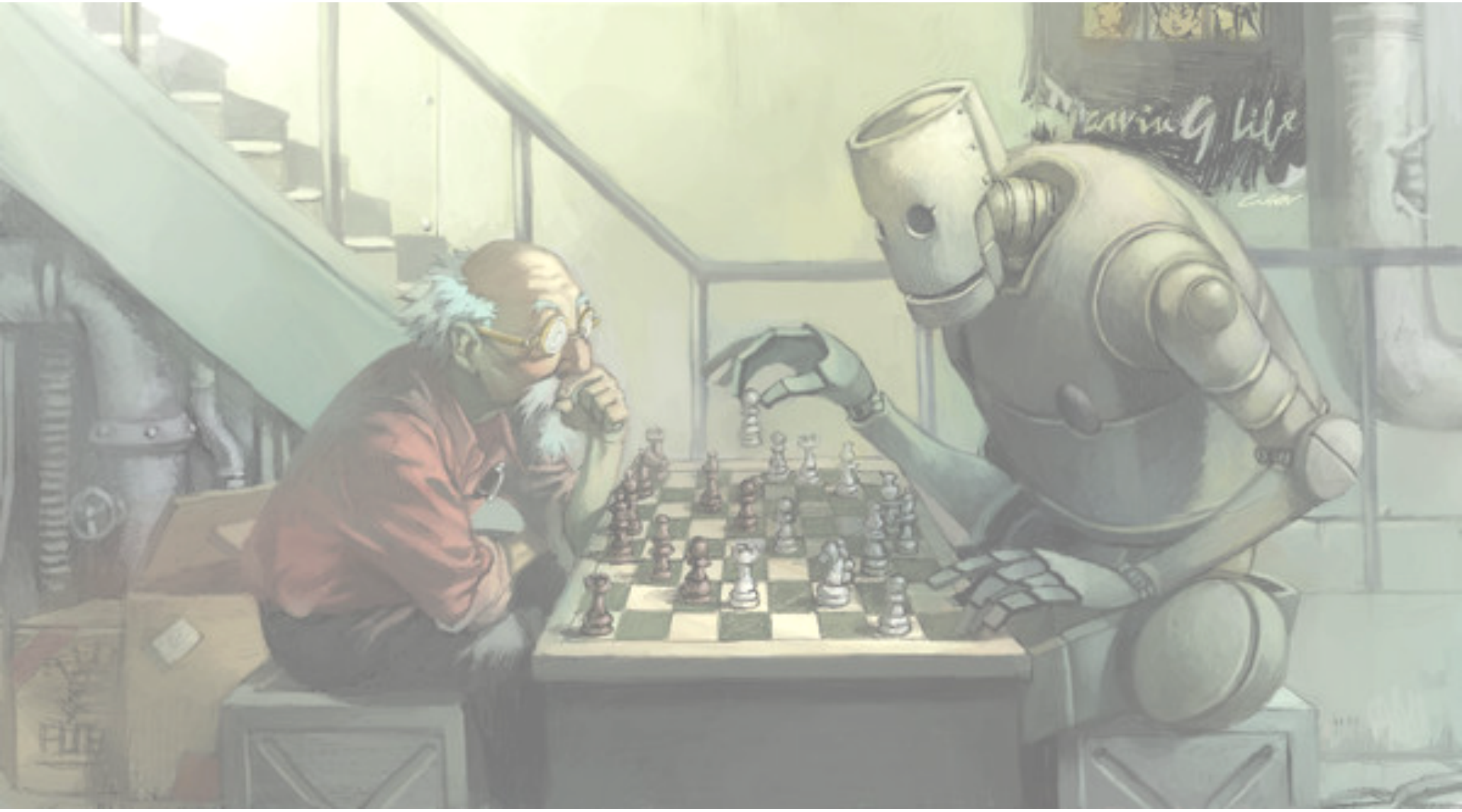


image credit: cuson (deviantart)
http://cuson.deviantart.com/art/play-chess-with-robot-61467927

What is AI?

- **R**etrieval
- **I**nference
- **K**nowledge Representation
- **L**earning
- **S**earch

What is AI?

- **R**etrieval
  - Finding *relevant* information
- **I**nference
- **K**nowledge Representation
- **L**earning
- **S**earch

What is AI?

- **R**etrieval
- **I**nference
  - Reasoning, finding evidence, drawing conclusions
- **K**nowledge Representation
- **L**earning
- **S**earch

What is AI?

- **R**etrieval
- **I**nference
- **K**nowledge Representation
  - Structuring knowledge to computer
- **L**earning
- **S**earch

What is AI?

- **R**etrieval
- **I**nference
- **K**nowledge Representation
- **L**earning
  - Computer improves itself
- **S**earch

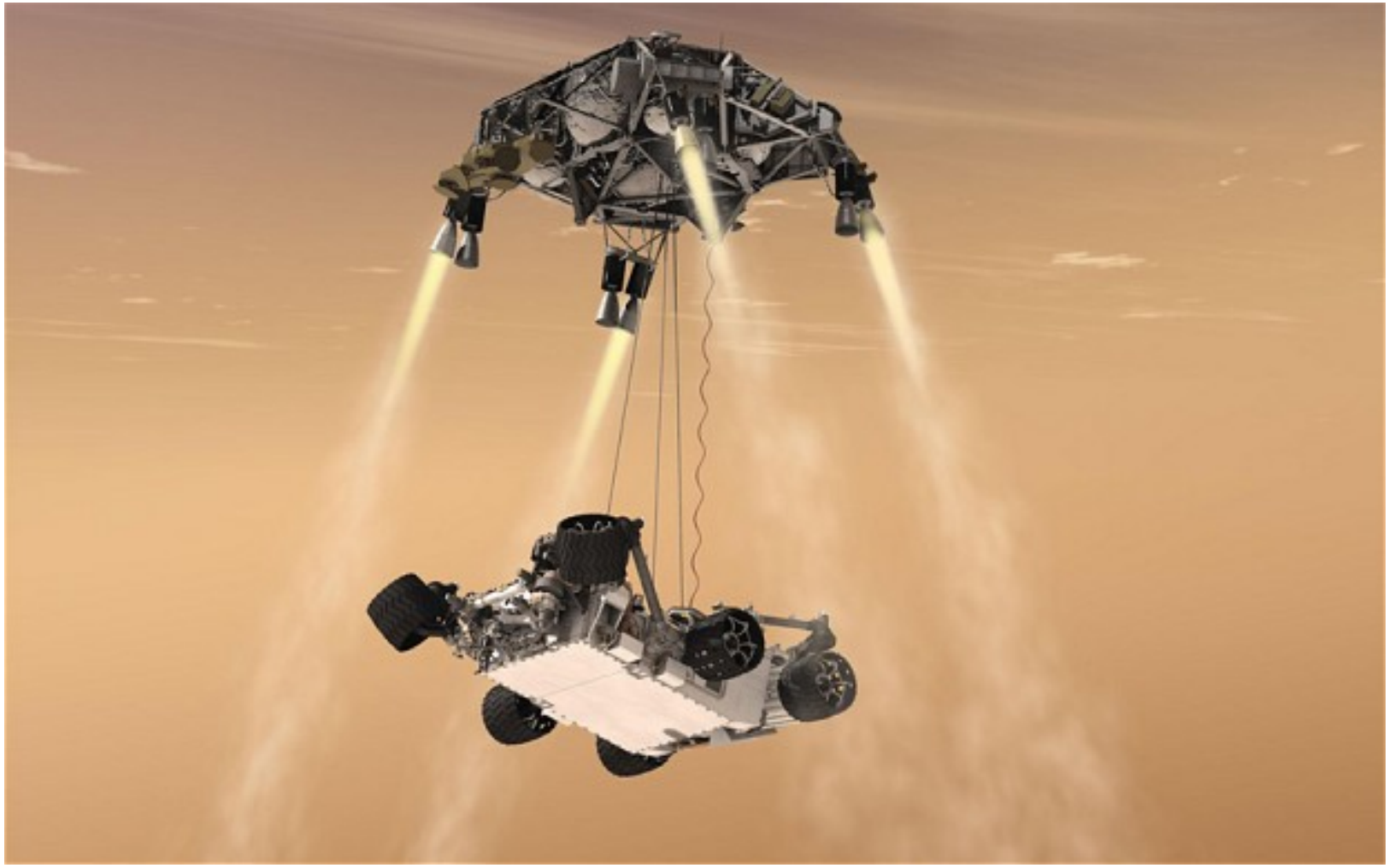
What is AI?

- **R**etrieval
- **I**nference
- **K**nowledge Representation
- **L**earning
- **S**earch
  - Hunting for solutions to problems

Games


2804
Terran Siege Tank
Master Sergeant
Kills: 3
MENU
Tank Mode

Robotics



Computer Vision

Scheduling

Creativity

Where do I go from here?

- CS 6140 – Machine Learning
- CS 6120 – Natural Language Processing
- CS 6200 – Information Retrieval
- CS 6220 – Data Mining Techniques
- CS 5330 – Pattern Recognition & Computer Vision
- CS 5335 – Robotic Science and Systems

Discussion