Chap. 18A

Learning from Examples

Agents can improve their behavior through diligent study of their own experiences.

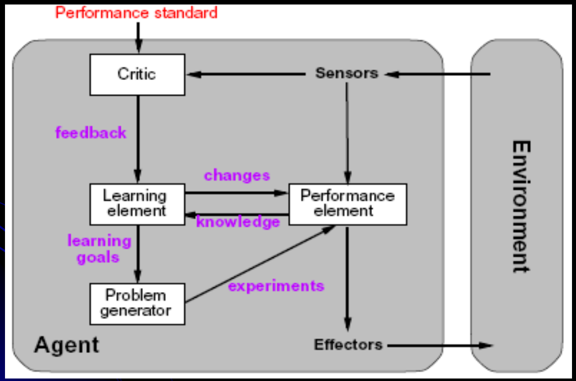
Outline

- Learning Agents
- Inductive Learning

Learning

- Learning is essential for unknown environments,
  i.e., when designer lacks omniscience
- Learning is useful as a system construction method,
  i.e. expose the agent to reality rather than trying to write it down
- Learning modifies the agent's decision mechanisms
  to *improve performance* in the future.

Learning Agents



Learning Element

- Design of learning element for the improvement depends on:
  - what type of *performance element* is used
  - which *functional component* is to be learned
  - what *prior knowledge* the agent already has
  - how that component & data are *represented*
  - what kind of *feedback* is available
- Example scenarios

Performance element	Component	Representation	Feedback
Alpha−beta search	Eval. fn.	Weighted linear function	Win/loss
Logical agent	Transition model	Successor−state axioms	Outcome
Utility−based agent	Transition model	Dynamic Bayes net	Outcome
Simple reflex agent	Percept−action fn	Neural net	Correct action

… continued

- Design of learning element is dictated by
  - Type of performance element
  - *Functional components* to be learned:
    1. Direct mapping from conditions on the current state to actions
    2. A means to infer relevant properties of the world from the percept sequence
    3. Information about the way the world evolves and about the results of possible actions the agent can take.
    4. Utility information indicating the desirability of the world states
    5. Action-value information indicating the desirability of actions
    6. Goals that describe classes of states whose achievement maximizes the agent's utility.

    Each of these components can be learned from appropriate feedback.

.. continued

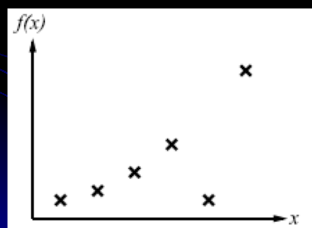
- *Representation and Prior Knowledge*
  - Factored representation (a vector of attribute values) and outputs
  - Linear weighted polynomials for utility function
  - Propositional/first-order logical sentences,
  - probabilistic description: e.g. Bayesian Network
  - Types of learning: inductive learning, deductive learning.
    Generalization/Rules ←/→ Specific Examples/Activities
- *Types of feedback*
  - **Supervised learning**: correct answers for each instance
    - Learning a function from examples of its inputs/outputs
  - **Reinforcement learning**: occasional rewards
    - Learning how the environment works
  - **Unsupervised learning:**
    - Learning patterns in the input with no specific output values
    - E.g.) clustering

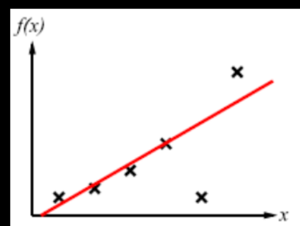
Inductive Learning in Supervised Learning

- Simplest form: learn a function from examples
- f is the target function s.t. $y = f(x)$
- An example is a pair $<x, y>$, e.g.
- Task of pure inductive inference (or induction):
  find a hypothesis h
  such that $h \approx f$, given a training set of examples
- The fundamental problem of Induction: how good $h \approx f$?
  A good hypothesis will generalize well; i.e. predict unseen examples correctly.
- Classification: when the output y is one of a finite set of values
- Regression: when y is a number, i.e. to find a conditional expectation of y given examples.

Inductive Learning Method

- Construct/adjust *h* to agree with *f* on training set
  (*h* is consistent if it agrees with f on all examples)
- Hypothesis space H: the set of hypotheses
- E.g.) curve fitting:





9

.. continued

- Construct/adjust *h* to agree with *f* on training set
  (*h* is consistent if it agrees with *f* on all examples)
- Hypothesis space H: the set of hypotheses
  - *How to choose from among multiple consistent hypotheses?*
- E.g.) curve fitting:

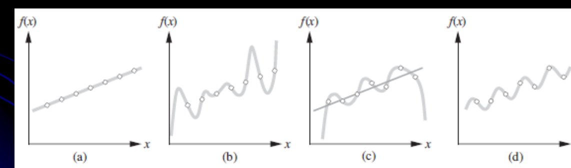
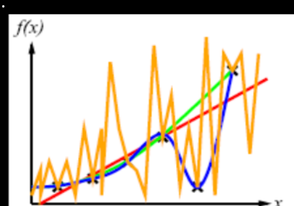


Figure 18.1 **FILES: figures/xy-plot.eps (Tue Nov 3 16:24:13 2009).** (a) Example $(x, f(x))$ pairs and a consistent, linear hypothesis. (b) A consistent, degree-7 polynomial hypothesis for the same data set. (c) A different data set, which admits an exact degree-6 polynomial fit or an approximate linear fit. (d) A simple, exact sinusoidal fit to the same data set.

10

.. continued

- Construct/adjust *h* to agree with *f* on training set
  (*h* is consistent if it agrees with f on all examples)
- E.g.) curve fitting:



- Hypothesis space H: the set of hypotheses
  - *How to choose from among multiple consistent hypotheses?*
  - **Ockham's razor:** maximize a combination of consistency and *simplicity*
- Tradeoff between the complex hypotheses that fit the training data well and simpler hypotheses that may generalize better.

11

.. continued

- Possibility/Impossibility of finding a simple, consistent hypothesis depends on the *hypothesis space chosen.*
  - E.g.) a polynomial function vs. a sinusoidal function.
  - A learning is *realizable* if the hypothesis space contains the *true function*; otherwise, *unrealizable*.
    - To use *prior knowledge* to derive a hypothesis space in which we know the true function must lie.
    - To use the largest possible hypothesis space => *computational complexity of learning.*
    - How probable a hypothesis is: Choose *h** that is most probable given the data. $h^* = \underset{h\in\mathcal{H}}{\text{argmax}}\, P(h|data) \,.= \underset{h\in\mathcal{H}}{\text{argmax}}\, P(data|h)\, P(h)$
  - A tradeoff b/t the expressiveness of a hypothesis space and the complexity of finding a simple, consistent hypothesis within that space.

12

Decision Tree Representations

- Examples described by the attribute values (Boolean, discrete, continuous, etc.) -- Factored representation, *Attribute-based representation*.
  - Input: an object/situation described by a vector of attributes
  - Output: a decision – the predicted output value for the input
- E.g.) situations where I will/won't wait for a table:

Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	*T*	*F*	*F*	*T*	*Some*	*$$$*	*F*	*T*	*French*	*0–10*	*T*
X_2	*T*	*F*	*F*	*T*	*Full*	*$*	*F*	*F*	*Thai*	*30–60*	*F*
X_3	*F*	*T*	*F*	*F*	*Some*	*$*	*F*	*F*	*Burger*	*0–10*	*T*
X_4	*T*	*F*	*T*	*T*	*Full*	*$*	*F*	*F*	*Thai*	*10–30*	*T*
X_5	*T*	*F*	*T*	*F*	*Full*	*$$$*	*F*	*T*	*French*	*>60*	*F*
X_6	*F*	*T*	*F*	*T*	*Some*	*$$*	*T*	*T*	*Italian*	*0–10*	*T*
X_7	*F*	*T*	*F*	*F*	*None*	*$*	*T*	*F*	*Burger*	*0–10*	*F*
X_8	*F*	*F*	*F*	*T*	*Some*	*$$*	*T*	*T*	*Thai*	*0–10*	*T*
X_9	*F*	*T*	*T*	*F*	*Full*	*$*	*T*	*F*	*Burger*	*>60*	*F*
X_{10}	*T*	*T*	*T*	*T*	*Full*	*$$$*	*F*	*T*	*Italian*	*10–30*	*F*
X_{11}	*F*	*F*	*F*	*F*	*None*	*$*	*F*	*F*	*Thai*	*0–10*	*F*
X_{12}	*T*	*T*	*T*	*T*	*Full*	*$*	*F*	*F*	*Burger*	*30–60*	*T*

- **Classification** of examples is positive (T) or negative (F)
  - Boolean classification