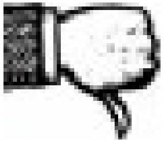
Chapter ML:I

I. Introduction

- Examples of Learning Tasks
- Specification of Learning Problems

Examples of Learning Tasks

Car Shopping Guide



?

Which criteria form the basis of a decision?

Examples of Learning Tasks

Risk Analysis for Credit Approval

Customer 1	
house owner	yes
income (p.a.)	51 000 EUR
repayment (p.m.)	1 000 EUR
credit period	7 years
SCHUFA entry	no
age	37
married	yes
...	

...

Customer n	
house owner	no
income (p.a.)	55 000 EUR
repayment (p.m.)	1 200 EUR
credit period	8 years
SCHUFA entry	no
age	?
married	yes
...	

Examples of Learning Tasks

Risk Analysis for Credit Approval

Customer 1	
house owner	yes
income (p.a.)	51 000 EUR
repayment (p.m.)	1 000 EUR
credit period	7 years
SCHUFA entry	no
age	37
married	yes
...	

...

Customer n	
house owner	no
income (p.a.)	55 000 EUR
repayment (p.m.)	1 200 EUR
credit period	8 years
SCHUFA entry	no
age	?
married	yes
...	

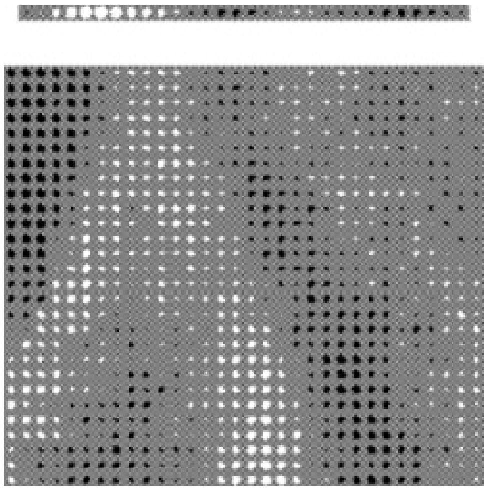
Learned rules:

```
IF    (income>40 000 AND credit_period<3) OR
      house_owner=yes
THEN  credit_approval=yes

IF    SCHUFA_entry=yes OR
      (income<20 000 AND repayment>800)
THEN  credit_approval=no
```

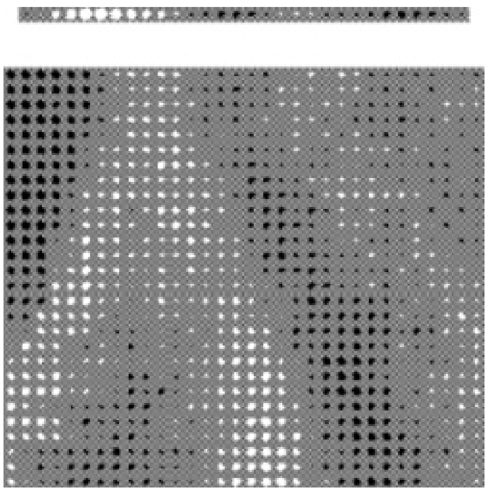
Examples of Learning Tasks

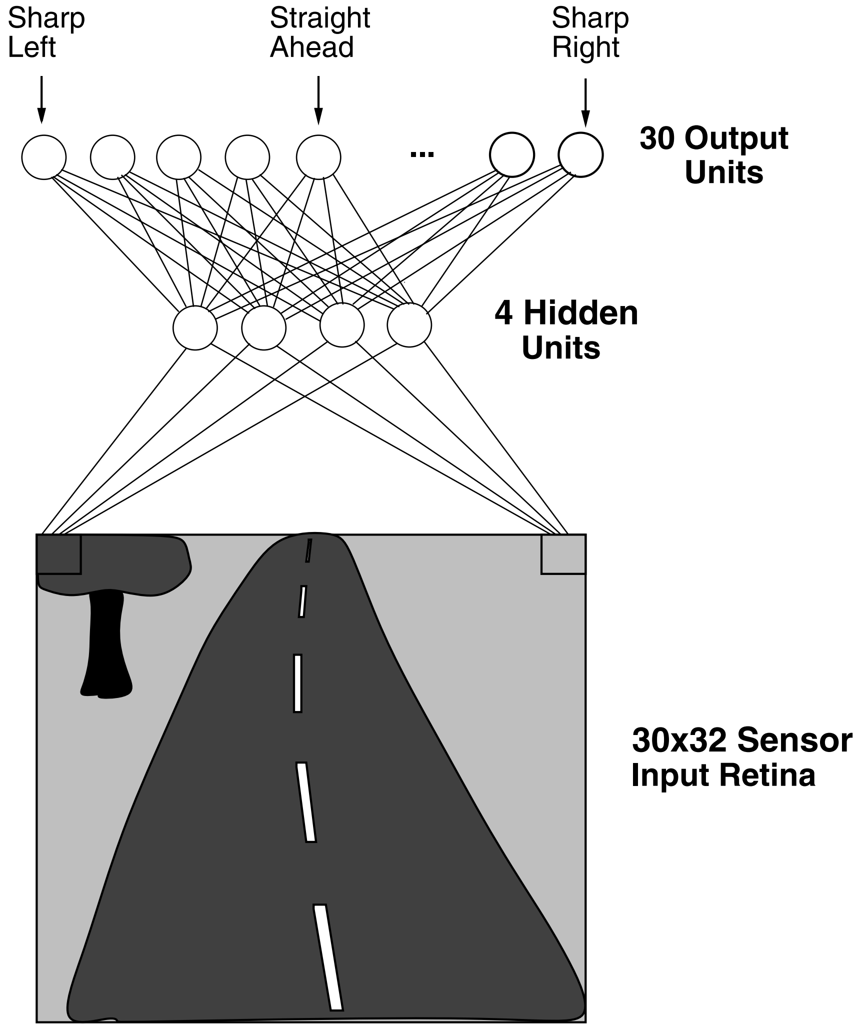
Image Analysis [Mitchell 1997]



Examples of Learning Tasks

Image Analysis [Mitchell 1997]





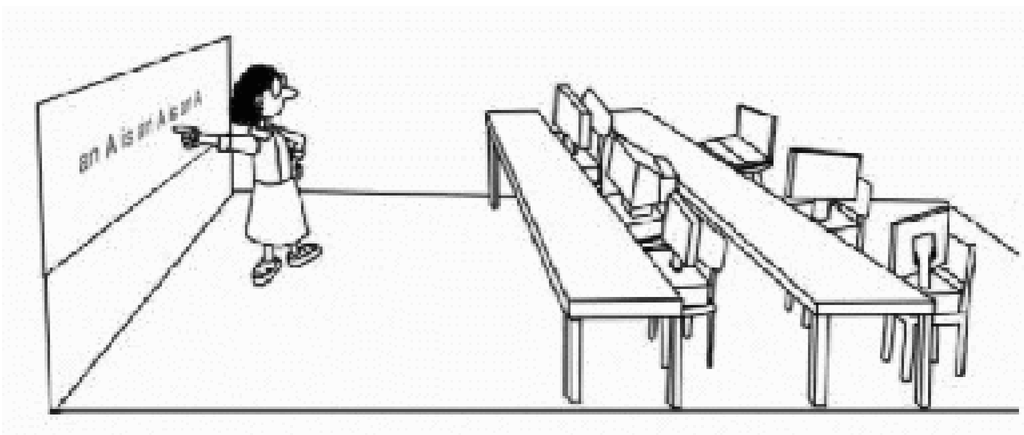
Specification of Learning Problems

Definition 1 (Machine Learning [Mitchell 1997]**)**

A computer program is said to learn

- from experience
- with respect to some class of tasks and
- a performance measure,

if its performance at the tasks improves with the experience.



Remarks:

- Example chess
  - task = playing chess
  - performance measure = number of games won during a world championship
  - experience = possibility to play against itself
- Example optical character recognition
  - task = isolation and classification of handwritten words in bitmaps
  - performance measure = percentage of correctly classified words
  - experience = collection of correctly classified, handwritten words
- A corpus with labeled examples forms a kind of “compiled experience”.
- Consider the different corpora that are exploited for different learning tasks in the webis group. [www.webis.de/research/corpora]

Specification of Learning Problems

Learning Paradigms

1. Supervised learning

2. Unsupervised learning

3. Reinforcement learning

Specification of Learning Problems

Learning Paradigms

1. Supervised learning

   Learn a function from a set of input-output-pairs. An important branch of supervised learning is automated classification. Example: optical character recognition

2. Unsupervised learning

   Identify structures in data. Important subareas of unsupervised learning include automated categorization (e.g. via cluster analysis), parameter optimization (e.g. via expectation maximization), and feature extraction (e.g. via factor analysis).

3. Reinforcement learning

   Learn, adapt, or optimize a behavior strategy in order to maximize the own benefit by interpreting feedback that is provided by the environment. Example: development of behavior strategies for agents in a hostile environment.

Specification of Learning Problems

Example Chess: Kind of Experience [Mitchell 1997]

1. Feedback
   – direct: for each board configuration the best move is given.
   – indirect: only the final result is given after a series of moves.

Specification of Learning Problems

Example Chess: Kind of Experience [Mitchell 1997]

1. Feedback
   - direct: for each board configuration the best move is given.
   - indirect: only the final result is given after a series of moves.

2. Sequence and distribution of examples
   - A teacher presents important example problems along with a solution.
   - The learner chooses from the examples; e.g. pick a board for which the best move is unknown.
   - The selection of examples to learn from should follow the (expected) distribution of future problems.

Specification of Learning Problems

Example Chess: Kind of Experience [Mitchell 1997]

1. Feedback
   - direct: for each board configuration the best move is given.
   - indirect: only the final result is given after a series of moves.

2. Sequence and distribution of examples
   - A teacher presents important example problems along with a solution.
   - The learner chooses from the examples; e.g. pick a board for which the best move is unknown.
   - The selection of examples to learn from should follow the (expected) distribution of future problems.

3. Relevance under a performance measure
   - How far can we get with experience?
   - Can we master situations in the wild?
     (playing against itself will be not enough to become world class)

Specification of Learning Problems

Example Chess: Ideal Target Function γ [Mitchell 1997]

(a) γ : *Boards* $\rightarrow$ *Moves*

(b) γ : *Boards* $\rightarrow \mathbf{R}$

Specification of Learning Problems

Example Chess: Ideal Target Function γ [Mitchell 1997]

(a) γ : *Boards* $\rightarrow$ *Moves*

(b) γ : *Boards* $\rightarrow \mathbf{R}$

A recursive definition of γ, following a kind of *means-ends analysis*:

Let be $o \in$ *Boards*.

1. $\gamma(o) = 100$, if o represents a final board state that is won.
2. $\gamma(o) = -100$, if o represents a final board state that is lost.
3. $\gamma(o) = 0$, if o represents a final board state that is drawn.
4. $\gamma(o) = \gamma(o^*)$ otherwise.

o^* denotes the best final state that can be reached if both sides play optimally.
Related: game playing, minimax strategy, α-β pruning.

[Study course on Search Theory, Stein 1998-2014]

Specification of Learning Problems

Example Chess: From the Real World γ to a Model World y

$$\gamma(o) \;\rightsquigarrow\; y(\alpha(o)) \equiv y(\mathbf{x})$$

$$y(\mathbf{x}) = w_0 + w_1 \cdot x_1 + w_2 \cdot x_2 + w_3 \cdot x_3 + w_4 \cdot x_4 + w_5 \cdot x_5 + w_6 \cdot x_6$$

where

x_1 = number of black pawns on board o
x_2 = number of white pawns on board o
x_3 = number of black pieces on board o
x_4 = number of white pieces on board o
x_5 = number of black pieces threatened on board o
x_6 = number of white pieces threatened on board o

Other approaches to formulate y:

- case base
- set of rules
- neural network
- polynomial function of board features

Remarks:

- The ideal target function γ interprets the real world, say, a real-world object o, to "compute" $\gamma(o)$. This "computation" may be operationalized by a human or by some other (even arcane) mechanism of the real world.
- To simulate the interesting aspects of the real world by means of a computer, we define a model world. This model world is restricted to particular—typically easily measurable—features $\mathbf{x}$ that are derived from o, with $\mathbf{x} = \alpha(o)$. In the model world, $y(\mathbf{x})$ is the formalized counterpart of $\gamma(o)$.
- y is called model function or model, α is called model formation function.
- The key difference between an ideal target function γ and a model function y lies in the size and the representation of their respective domains. Examples:
  - A chess grand master assesses a board o in its entirety, both intuitively and analytically; a chess program is restricted to particular features $\mathbf{x}$, $\mathbf{x} = \alpha(o)$.
  - A human mushroom picker assesses a mushroom o with all her skills (intuitively, analytically, by tickled senses); a classification program is restricted to a few surface features $\mathbf{x}$, $\mathbf{x} = \alpha(o)$.

Remarks (continued) :

- For automated chess playing a real-valued assessment function is needed; such kind of problems form regression problems. If only a small number of values are to be considered (e.g. school grades), one is given a classification problem. A regression problem can be transformed into a classification problem by means of domain discretization.
- Regression problems and classification problems often differ with regard to assessing the achieved accuracy or goodness of fit. For regression problems the sum of the squared residuals may be a sensible criterion; for classification problems the number of misclassified examples may be more relevant.
- For classification problems, the ideal target function γ is also called ideal *classifier*; similarly, the model function y is called classifier.
- Decision problems are classification problems with two classes.
- The halting problem for Turing machines is an undecidable classification problem.

Specification of Learning Problems

How to Construct a Classifier y

Characterization of the real world:

- O is a set of objects.
- C is a set of classes.
- $\gamma : O \rightarrow C$ is the ideal classifier for O.

Specification of Learning Problems

How to Construct a Classifier y

Characterization of the real world:

- O is a set of objects.
- C is a set of classes.
- $\gamma : O \to C$ is the ideal classifier for O.

Classification problem:

- Given some $o \in O$, determine its class $\gamma(o) \in C$.

Acquisition of classification knowledge:

1. Build a database of examples of the form $(o, \gamma(o))$, $o \in O_D$, $O_D \subseteq O$.
2. Abstract the objects $o \in O_D$ towards feature vectors $\mathbf{x} \in X$, with $\mathbf{x} = \alpha(o)$.
3. Compute $(\mathbf{x}, c(\mathbf{x}))$, with $\mathbf{x} = \alpha(o)$ and $c(\mathbf{x})$ defined as $\gamma(o)$, $o \in O_D$.

Specification of Learning Problems

How to Construct a Classifier y (continued)

Characterization of the model world:

- X is a set of feature vectors, also called feature space.
- C is a set of classes.
- $c : X \to C$ is the ideal classifier for X.
- $D = \{(\mathbf{x}_1, c(\mathbf{x}_1)), \ldots, (\mathbf{x}_n, c(\mathbf{x}_n))\} \subseteq X \times C$ is a set of examples.

Specification of Learning Problems

How to Construct a Classifier y (continued)

Characterization of the model world:

- X is a set of feature vectors, also called feature space.
- C is a set of classes.
- $c : X \to C$ is the ideal classifier for X.
- $D = \{(\mathbf{x}_1, c(\mathbf{x}_1)), \ldots, (\mathbf{x}_n, c(\mathbf{x}_n))\} \subseteq X \times C$ is a set of examples.

Machine learning problem:

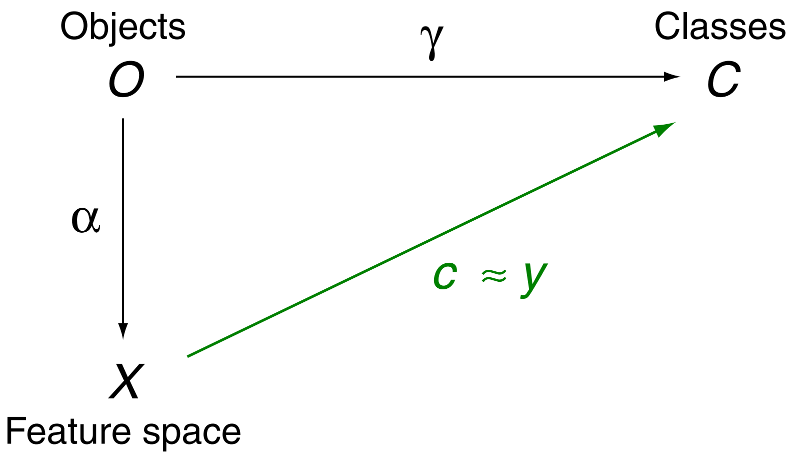
- Based on D, approximate the ideal classifier c by a classifier y.

➜ Formulate a model function $y : X \to C, \;\; \mathbf{x} \mapsto y(\mathbf{x})$

➜ Apply statistics, as well as theory and algorithms from the field of machine learning to maximize the goodness of fit between c and y.

Specification of Learning Problems

How to Construct a Classifier y (continued)



Semantics:

- γ Ideal classifier for real-world objects.
- α Model formation function.
- c Ideal classifier for vectors from the feature space.
- y Classifier.
- $c \approx y$ c is approximated by y.

Remarks:

- The feature space X comprises vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots$, which can be considered as abstractions of real-world objects $o_1, o_2, \ldots$, and which have been computed according to our view of the real world.
- The model formation function α determines the level of abstraction between o and $\mathbf{x}$, $\mathbf{x} = \alpha(o)$. I.e., α determines the representation fidelity, exactness, quality, or simplification.
- Though α models an object $o \in O$ only imperfectly as $\mathbf{x} = \alpha(o)$, $c(\mathbf{x})$ must be considered as *ideal* classifier, since $c(\mathbf{x})$ is defined as $\gamma(o)$ and hence mimics the real-world classes. I.e., c and γ have different domains each, but they return the same images.
- $c(\mathbf{x})$ is often termed "ground truth" (for $\mathbf{x}$ and the underlying classification problem). Observe that this term is justified by the fact that $c(\mathbf{x}) \equiv \gamma(o)$.

Specification of Learning Problems

LMS Algorithm for Fitting y [*PT Algorithm*]

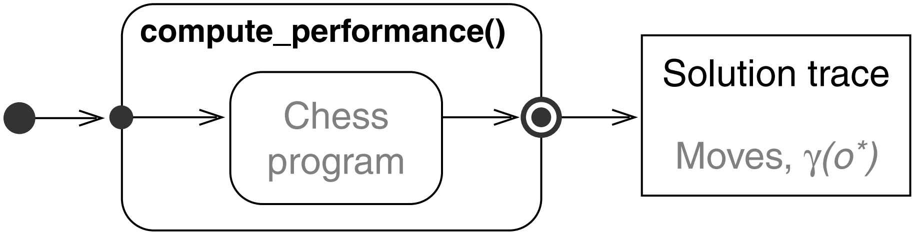
Algorithm:	*LMS*	Least Mean Squares.
Input:	D	Training examples of the form $(\mathbf{x}, c(\mathbf{x}))$ with target function value $c(\mathbf{x})$ for $\mathbf{x}$.
	η	Learning rate, a small positive constant.
Internal:	$y(D)$	Set of $y(\mathbf{x})$-values computed from the elements $\mathbf{x}$ in D given some $\mathbf{w}$.
Output:	$\mathbf{w}$	Weight vector.

LMS(D, η)

initialize_random_weights$((w_0, w_1, \ldots, w_p))$
REPEAT
  $(\mathbf{x}, c(\mathbf{x}))$ = *random_select*(D)
  $y(\mathbf{x}) = w_0 + w_1 \cdot x_1 + \ldots + w_p \cdot x_p$
  error $= c(\mathbf{x}) - y(\mathbf{x})$
  FOR $j = 0$ **TO** p **DO**
    $\Delta w_j = \eta \cdot$ *error* $\cdot x_j$ // $\forall_{\mathbf{x} \in D} : \mathbf{x}|_{x_0} \equiv 1$
    $w_j = w_j + \Delta w_j$
  ENDDO
UNTIL(*convergence*$(D, y(D))$)
return$((w_0, w_1, \ldots, w_p))$

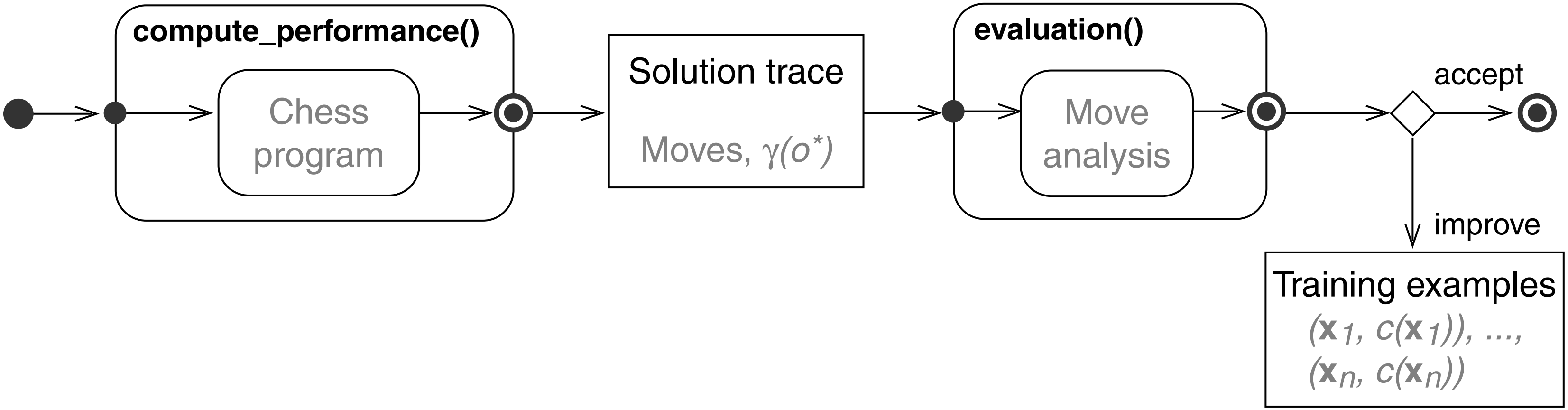
Specification of Learning Problems

Design of Learning Systems [p.12, Mitchell 1997]



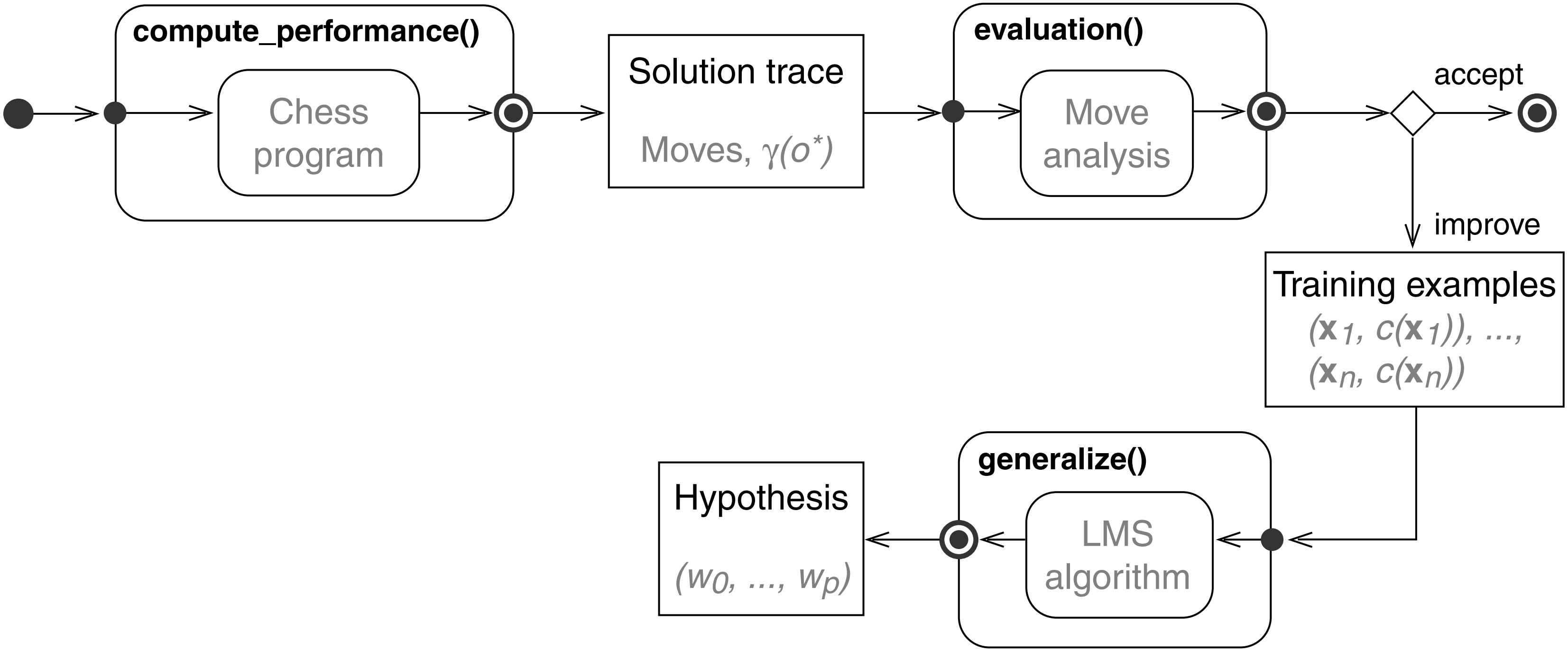
Specification of Learning Problems

Design of Learning Systems [p.12, Mitchell 1997]



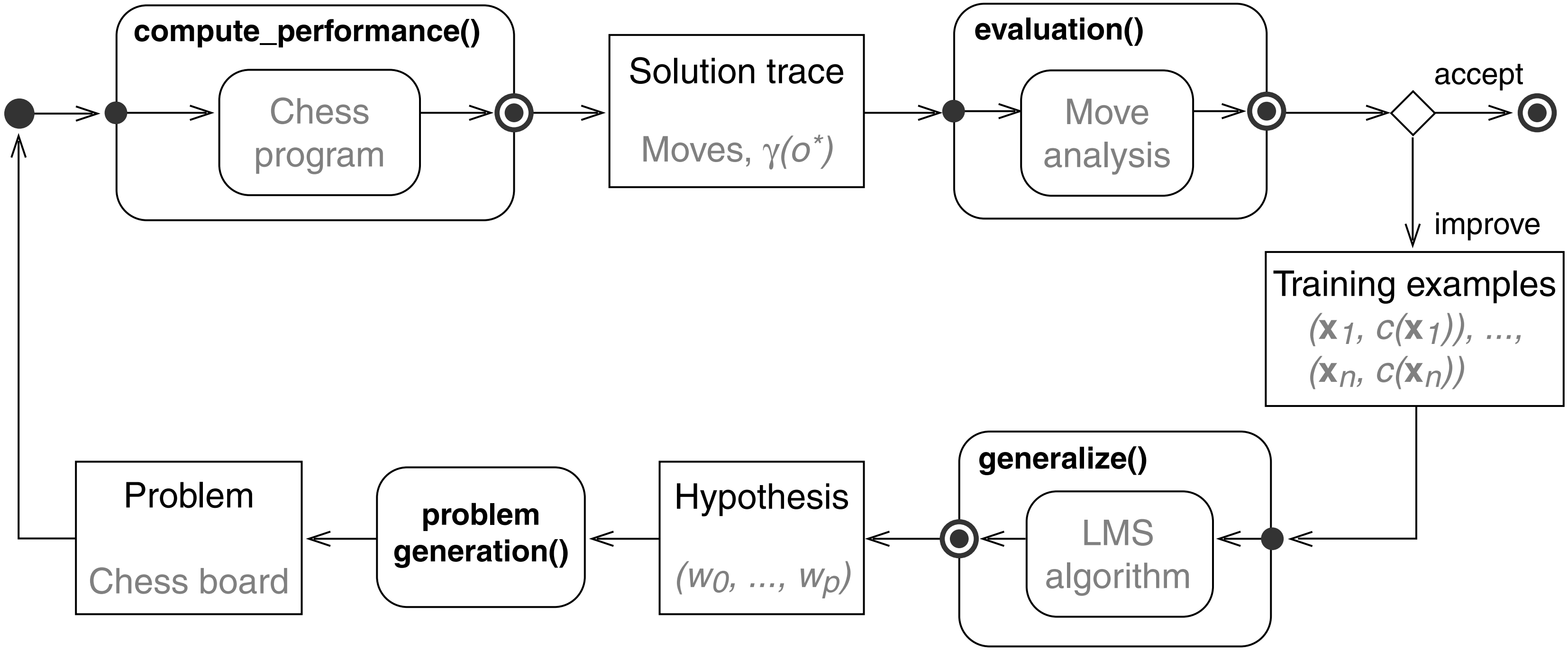
Specification of Learning Problems

Design of Learning Systems [p.12, Mitchell 1997]



Specification of Learning Problems

Design of Learning Systems [p.12, Mitchell 1997]



Important design aspects:

1. kind of experience
2. fidelity of the model formation function $\alpha : O \rightarrow X$
3. class of the model function y
4. learning method for fitting y

Specification of Learning Problems

Related Questions

Model functions y:

- What are important classes of model functions?
- What are methods to fit (= learn) model functions?
- What are measures to assess the goodness of fit?
- How does the example number affect the learning process?
- How does noise affect the learning process?

Specification of Learning Problems

Related Questions (continued)

Generic learnability:

- What are the theoretical limits of learnability?
- How can we use nature as a model for learning?

Knowledge acquisition:

- How can we integrate background knowledge into the learning process?
- How can we integrate human expertise into the learning process?