Machine Learning and Data Mining

Benno Stein Theo Lettmann

Contents

I. Introduction

II. Machine Learning Basics

III. Decision Trees

IV. Bayesian Learning

V. Linear Models

VI. Neural Networks

VII. Kernel Methods

VIII. Learning Theory

IX. Evaluation

X. Combined Models and Meta Learning

XI. Latent Variables Analysis

XII. Cluster Analysis

XIII. Other Unsupervised Learning Methods

Objectives

- understand and explain the basic concepts of machine learning
- understand formalized concepts and methods and be able to implement them in the form of algorithms
- sensibly select, adapt, and apply relevant methods
- being able to educate oneself

Related Fields

1. Statistics [paradigms, models]
2. Mathematics

3. Information Retrieval [methods, algorithms]
4. Knowledge Processing
5. Heuristic Search

6. Decision Support Systems [applications]
7. Business Intelligence
8. Web Technology

Literature

Machine Learning:

- Christopher M. Bishop.
  Pattern Recognition and Machine Learning
  2nd edition, Springer 2007.

- Leo Breiman, Jerome H. Friedman, Richard A. Olshen, Charles J. Stone.
  Classification and Regression Trees
  CRC Press reprint, 1998.

- Nello Cristianini, John Shawe-Taylor.
  An Introduction to Support Vector Machines and Other Kernel-based Learning Methods
  Cambridge University Press, 2000.

- Trevor Hastie, Robert Tibshirani, Jerome Friedman.
  The Elements of Statistical Learning
  2nd edition, Springer, 2009.

- Tom Mitchell.
  Machine Learning
  1st edition, McGraw-Hill, 1997. www.cs.cmu.edu/∼tom/mlbook.html

- Vladimir Vapnik.
  The Nature of Statistical Learning Theory
  2nd edition, Springer 2000.

Literature

Data Mining:

- David Hand, Heikki Mannila, Padhraic Smyth.
  Principles of Data Mining
  Bradford, 2001.

- Pang-Ning Tan, Michael Steinbach, Vipin Kumar.
  Introduction to Data Mining
  1st edition, Addison Wesley, 2005.

- Ian H. Witten, Eibe Frank.
  Data Mining: Practical Machine Learning Tools and Techniques
  3rd edition, Morgan Kaufmann, 2011.

Software

Programming environment:

- Borland, IBM, MERANT, QNX, Rational Software, Red Hat, SuSE, TogetherSoft, Webgain, Ericsson, HP, Intel, MontaVista Software, SAP, Serena Software, Actuate, et al.

  Eclipse SDK
  Version 3.2.2 http://www.eclipse.org/downloads/

Machine learning library:

- Eibe Frank, Mark Hall, Geoff Holmes, Mike Mayo, Bernhard Pfahringer, Tony Smith, Ian Witten.

  Weka Machine Learning Project
  Version 3.5.6 http://www.cs.waikato.ac.nz/ml/weka/

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
…	

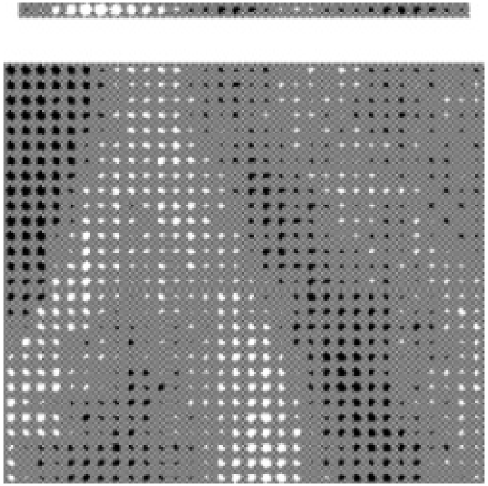
Learned rules:

```
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
```

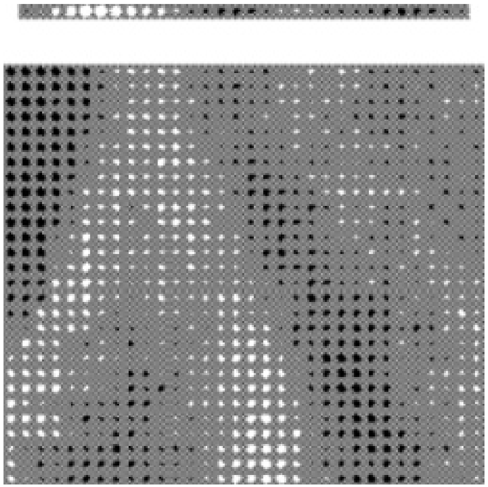
Examples of Learning Tasks

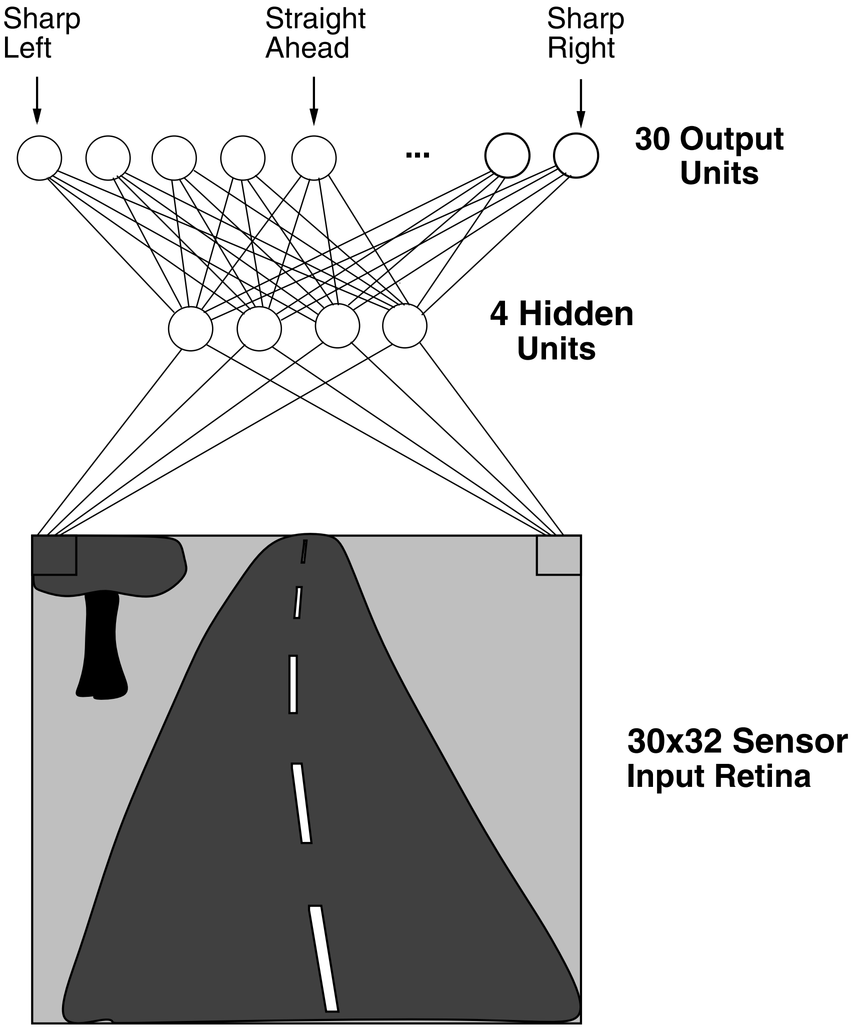
Image Analysis [Mitchell 1997]



Examples of Learning Tasks

Image Analysis [Mitchell 1997]





Sharp Left

Straight Ahead

Sharp Right

30 Output Units

...

4 Hidden Units

30x32 Sensor Input Retina

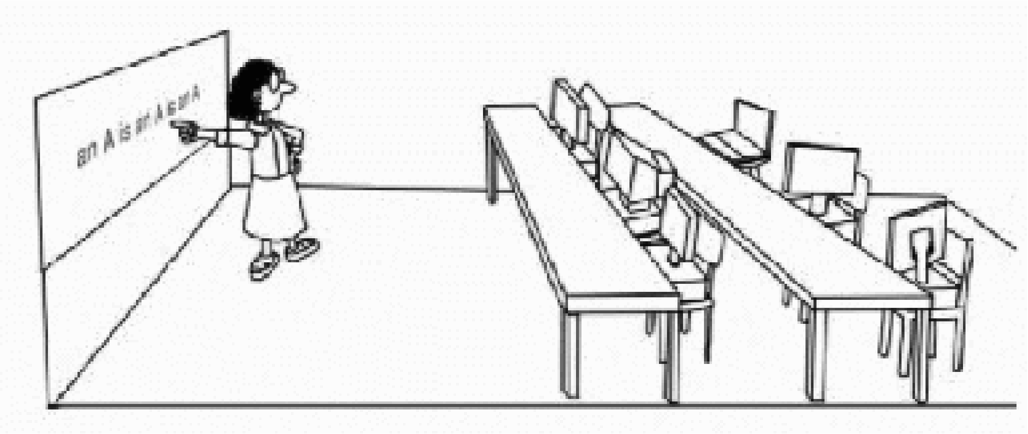
Specification of Learning Problems

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”: Different Kinds 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”: Different Kinds 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 student 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": Different Kinds 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 student 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) $\textit{chooseMove} : \textit{Boards} \rightarrow \textit{Moves}$

(b) $\gamma : \textit{Boards} \rightarrow \mathbf{R}$

Recursive definition of γ, implementing 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 won.
3. $\gamma(o) = 0$, if o represents a final board state that is drawn.
4. $\gamma(o) = \gamma(o^*)$ otherwise.

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

[cf. Reading "Search", Stein 1998-2012]

Specification of Learning Problems

Example “Chess”: Approximation y of the Ideal Target Function γ

$$\gamma(o) \;\Leftrightarrow\; y(\alpha(o)) \equiv y(\mathbf{x}) := w_0 + w_1 \cdot x_1 + w_2 \cdot x_2 + w_3 \cdot x_3 + w_4 \cdot x_4 + w_5 \cdot x_5 + w_6 \cdot x_6$$

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 following terms are used synonymously: target concept, target function, classifier.
- Usually, the ideal target function γ is unknown, or it cannot be learned, or it cannot be formalized. The function y, however, gives a possible mathematical formulation of γ.
- The key difference between the ideal target function γ and its mathematical formulation y lies in the size and the representation of the respective input spaces:

  The ideal target function γ interprets the real-world, say, the real-world object o, to "compute" $\gamma(o)$. The function y, however, is restricted to particular—typically easily measurable—features $\mathbf{x}$ that are derived from o, with $\mathbf{x} = \alpha(o)$. 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 $\mathbf{x}$, $\mathbf{x} = \alpha(o)$.
- 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 (school grades for example) are to be considered, 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 also differ with regard to the evaluation of the achieved effectiveness or fidelity. For regression problems the sum of the squared distances may be a sensible criterion; for classification problems the number of misclassified examples is often relevant.

Specification of Learning Problems

Specification of Classification Problems

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

Specification of Classification Problems

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.

Classification:

- Determination of the class $\gamma(o) \in C$ for the given $o \in O$.

Approach to automation:

1. Compilation of a set of examples of the form $(o, \gamma(o))$.
2. Abstraction of the $o \in O$ towards feature vectors $\mathbf{x}$, $\mathbf{x} = \alpha(o)$.
3. Computation of examples $(\mathbf{x}, c(\mathbf{x}))$, with $\mathbf{x} = \alpha(o)$ and $c(\mathbf{x})$ defined as $\gamma(o)$.
4. Mathematical formulation of a relation between $\mathbf{x}$ and $c(\mathbf{x})$.

Specification of Learning Problems

Specification of Classification Problems

Characterization of the model (model world):

- X is the space of instances (feature space) over a finite set of features.
- C is a set of classes.
- $c : X \rightarrow C$ 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

Specification of Classification Problems

Characterization of the model (model world):

- X is the space of instances (feature space) over a finite set of features.
- 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:

- Construct a classifier $y : X \to C$ based on the information in D, i.e., approximate the ideal classifier c by the classifier y.

Means:

- statistics, theory and algorithms from the field of machine learning.

Specification of Learning Problems

Specification of Classification Problems

$$\begin{tikzcd}
\overset{\text{Objects}}{O} \arrow[r, "\gamma"] \arrow[d, "\alpha"'] & \overset{\text{Classes}}{C} \\
\underset{\text{Feature space}}{X} \arrow[ur, "c \approx y"'] &
\end{tikzcd}$$

Mappings:

γ. Ideal classifier for real-world objects.

α. Model formation function.

c. Ideal classifier for the feature space.

y. Approximation function for c.

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 world.
- The model formation function α determines the level of abstraction of $\mathbf{x}$, $\mathbf{x} = \alpha(o)$.
  Similarly: α determines the representation fidelity, exactness, quality, simplification of $\mathbf{x}$.
- Though the function α models an object $o \in O$ only imperfectly, as $\mathbf{x} = \alpha(o)$, the function $c(\mathbf{x})$ 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 a different input space each, but they return the same images.

- Decision problems are classification problems with two classes.
- The halting problem for Turing machines is an undecidable classification problem.

Specification of Learning Problems

LMS Algorithm for the Computation of y

Algorithm: *LMS* Least Mean Square

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 for the elements $\mathbf{x}$ in D.

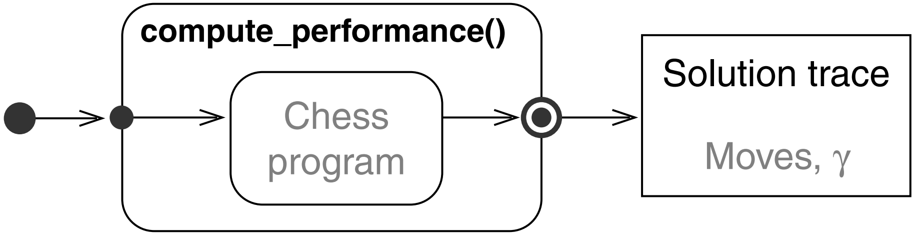
Output: $\mathbf{w}$ Weight vector.

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 $i = 1$ TO $p$ DO
    $\Delta w_i = \eta \cdot$ error $\cdot x_i$
    $w_i = w_i + \Delta w_i$
  ENDDO
UNTIL(convergence($D, y(D)$))
return($(w_0, w_1, \ldots, w_p)$)
```

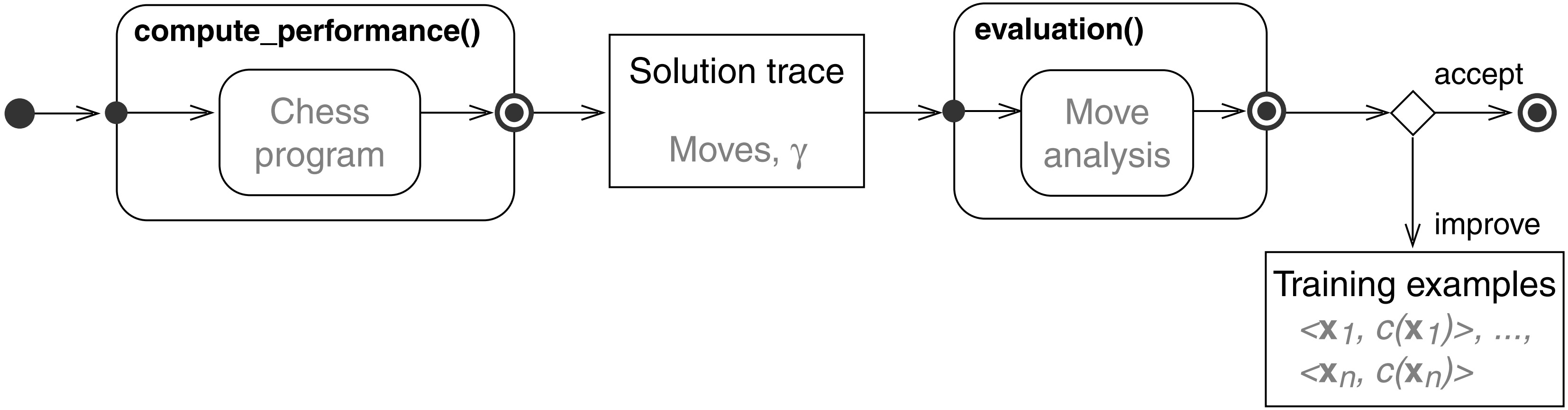
Specification of Learning Problems

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



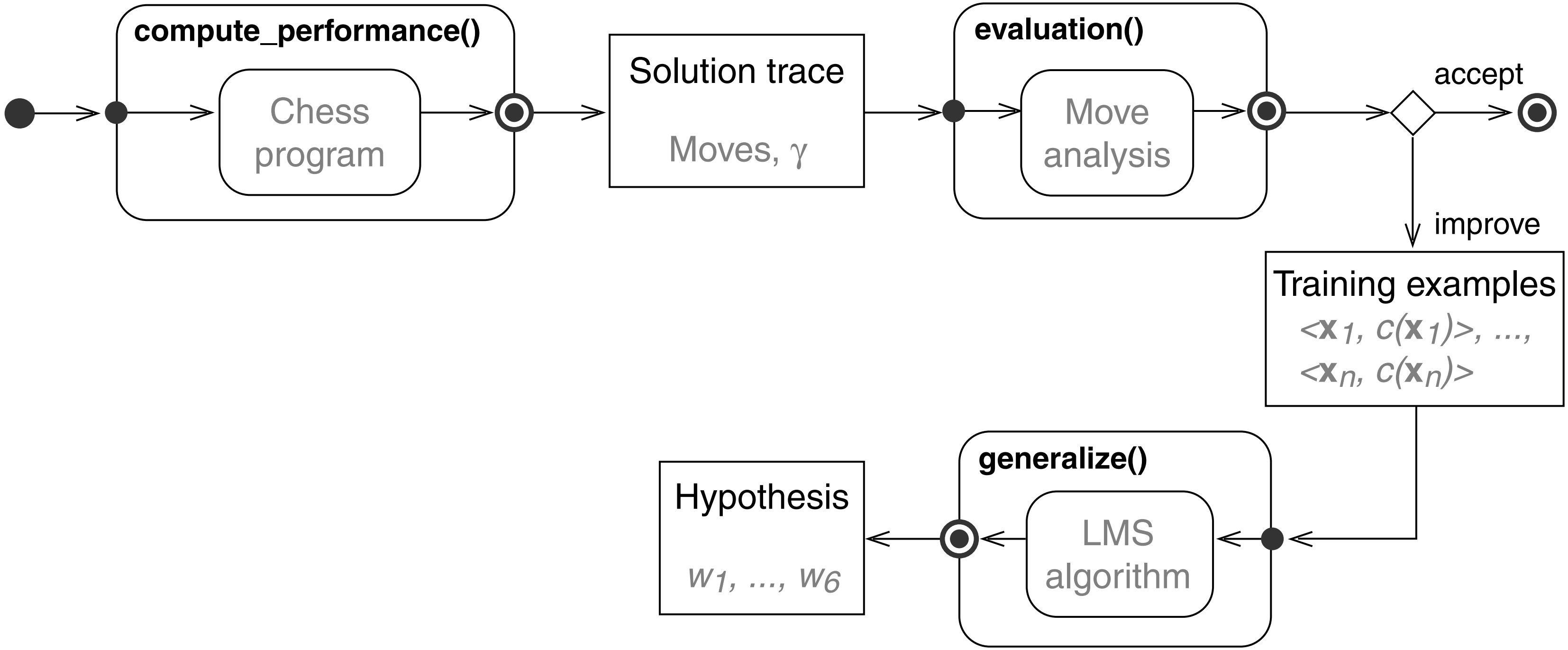
Specification of Learning Problems

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



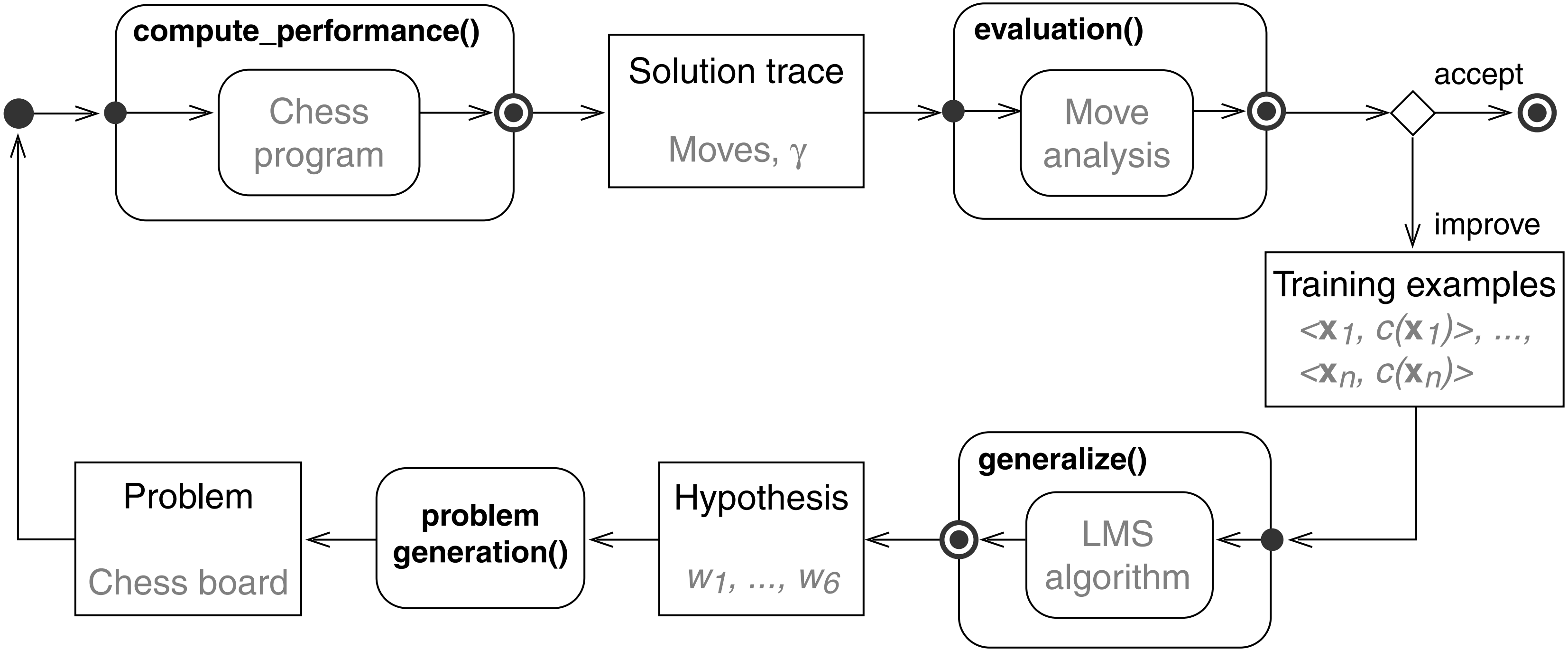
Specification of Learning Problems

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



Specification of Learning Problems

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



Important design decisions:

1. kind of learning experience
2. structure of the ideal target function γ
3. fidelity of the model formation function $\alpha : O \rightarrow X$
4. learning method for the construction of a classifier y

Specification of Learning Problems

Related Questions

- How does noise affect the effectiveness?
- What are methods to learn approximation functions?
- How does the number of examples affect the effectiveness?
- What are measures to assess the effectiveness of approximation functions?
- What are the theoretical limits of learnability?
- How can we use nature as a model for learning?
- How to integrate background knowledge and human expertise into the learning process?