

INTRODUCTION TO MACHINE LEARNING

- Background and Motivation?
- What is Learning?
- What is Machine Learning?
- Examples of successful applications
- How can we specify a learning problem?
- An example: learning to play checkers
- What questions should we ask about Machine Learning?

Background: Artificial Intelligence

- Algorithms or computation or information processing provide for study of cognition what calculus provided for physics
- We have a theory of intelligent behavior when we have precise information models (computer programs) that produce such behavior

Scientific Motivation

Information processing models provide useful insight into

- How living things learn
- Information requirements of learning tasks
- The precise conditions under which certain learning goals are achievable
- Inherent difficulty of learning tasks
- How to improve learning: e.g. active vs. passive
- Computational architectures for learning

Practical Motivation

- Intelligent behavior requires knowledge
- Explicitly specifying the knowledge needed for specific tasks is hard, and often infeasible
- If we can program computers to learn from experience, we can
 1. Dramatically enhance the usability of software, e.g. personalized information assistants
 2. Dramatically reduce the cost of software development, e.g. for medical diagnosis
 3. Automate data-driven discovery

Why Should Machine Learn?

- Some tasks are best specified by example

Face recognition

- Some tasks are best shown by demonstration

Landing an airplane

- Buried in large volume of data are useful predictive relationships

Data mining

- The operating environment of certain types of software may not be known at design time

User characteristics

- Environment changes over time

Examples of Applications of Machine Learning

- Data mining

1. Using historical data to improve decisions

Credit risk assessment, diagnosis, electric power usage prediction

Example: medical records → medical knowledge

2. Using scientific data to acquire knowledge

In computational molecular biology

- Software applications we can't program by hand

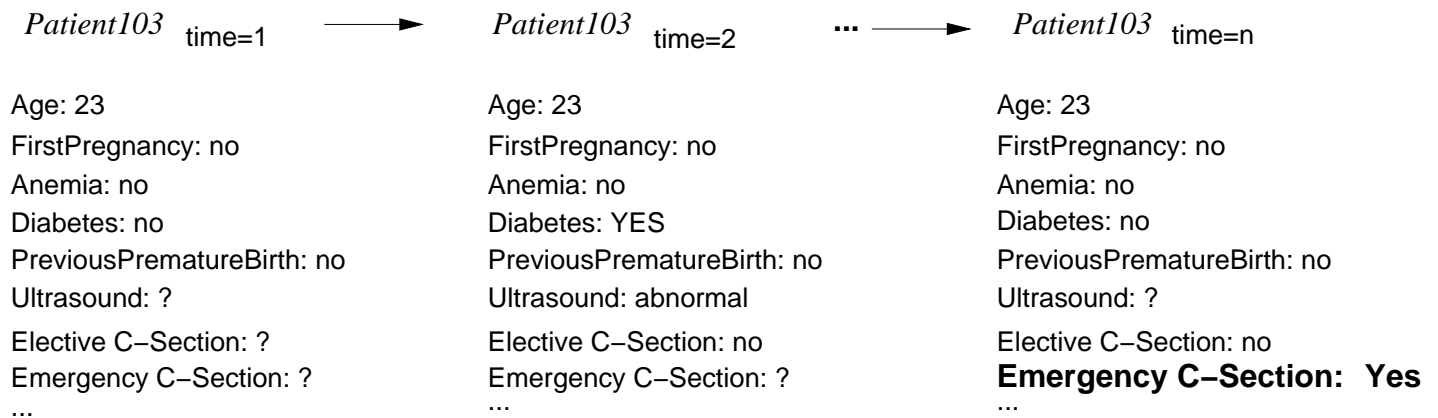
Autonomous driving, face recognition, speech recognition

- Self-customizing programs

Newsreader that learns user interests

Typical Data Mining Task

Data:



Given:

- 9714 patient records, each describing a pregnancy and birth
- Each patient record contains 215 features

Learn to predict:

- Classes of future patients at high risk for Emergency Cesarean Section

Data Mining Result

Data:

<i>Patient103</i> time=1	→	<i>Patient103</i> time=2	...	→	<i>Patient103</i> time=n
Age: 23		Age: 23			Age: 23
FirstPregnancy: no		FirstPregnancy: no			FirstPregnancy: no
Anemia: no		Anemia: no			Anemia: no
Diabetes: no		Diabetes: YES			Diabetes: no
PreviousPrematureBirth: no		PreviousPrematureBirth: no			PreviousPrematureBirth: no
Ultrasound: ?		Ultrasound: abnormal			Ultrasound: ?
Elective C-Section: ?		Elective C-Section: no			Elective C-Section: no
Emergency C-Section: ?		Emergency C-Section: ?			Emergency C-Section: Yes
...	

One of 18 learned rules:

If No previous vaginal delivery, and
 Abnormal 2nd Trimester Ultrasound, and
 Malpresentation at admission
Then Probability of Emergency C-Section is 0.6

Over training data: $26/41 = .63$,

Over test data: $12/20 = .60$

Credit Risk Analysis

Data:

<i>Customer103:</i> (time=t0)	<i>Customer103:</i> (time=t1)	...	<i>Customer103:</i> (time=tn)
Years of credit: 9	Years of credit: 9		Years of credit: 9
Loan balance: \$2,400	Loan balance: \$3,250		Loan balance: \$4,500
Income: \$52k	Income: ?		Income: ?
Own House: Yes	Own House: Yes		Own House: Yes
Other delinquent accts: 2	Other delinquent accts: 2		Other delinquent accts: 3
Max billing cycles late: 3	Max billing cycles late: 4		Max billing cycles late: 6
Profitable customer?: ?	Profitable customer?: ?		Profitable customer?: No
...

Rules learned from synthesized data:

If Other-Delinquent-Accounts > 2, and
Number-Delinquent-Billing-Cycles > 1
Then Profitable-Customer? = No
[Deny Credit Card application]

If Other-Delinquent-Accounts = 0, and
(Income > \$30k) OR (Years-of-Credit > 3)
Then Profitable-Customer? = Yes
[Accept Credit Card application]

Other Prediction Problems

Customer purchase behavior:

<i>Customer103:</i> (time=t0)	<i>Customer103:</i> (time=t1)	...	<i>Customer103:</i> (time=tn)
Sex: M	Sex: M		Sex: M
Age: 53	Age: 53		Age: 53
Income: \$50k	Income: \$50k		Income: \$50k
Own House: Yes	Own House: Yes		Own House: Yes
MS Products: Word	MS Products: Word		MS Products: Word
Computer: 386 PC	Computer: Pentium		Computer: Pentium
Purchase Excel?: ?	Purchase Excel?: ?		Purchase Excel?: Yes
...

Customer retention:

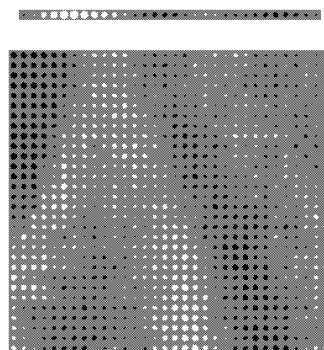
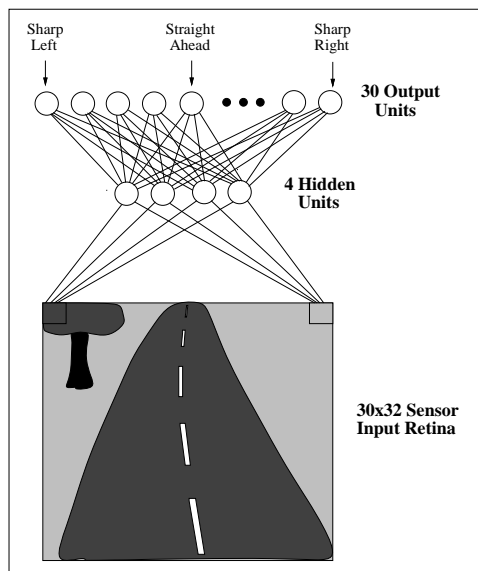
<i>Customer103:</i> (time=t0)	<i>Customer103:</i> (time=t1)	...	<i>Customer103:</i> (time=tn)
Sex: M	Sex: M		Sex: M
Age: 53	Age: 53		Age: 53
Income: \$50k	Income: \$50k		Income: \$50k
Own House: Yes	Own House: Yes		Own House: Yes
Checking: \$5k	Checking: \$20k		Checking: \$0
Savings: \$15k	Savings: \$0		Savings: \$0
Current-customer?: yes	Current-customer?: yes		Current-customer?: No

Process optimization:

<i>Product72:</i> (time=t0)	<i>Product72:</i> (time=t1)	...	<i>Product72:</i> (time=tn)
Stage: mix	Stage: cook		Stage: cool
Mixing-speed: 60rpm	Temperature: 325		Fan-speed: medium
Viscosity: 1.3	Viscosity: 3.2		Viscosity: 1.3
Fat content: 15%	Fat content: 12%		Fat content: 12%
Density: 2.8	Density: 1.1		Density: 1.2
Spectral peak: 2800	Spectral peak: 3200		Spectral peak: 3100
Product underweight?: ??	Product underweight?: ??		Product underweight?: Yes
...

Problems Too Difficult to Program by Hand

ALVINN [Pomerleau] drives 70 mph on highways



Software that Customizes to User

File Edit View Go Bookmarks Options Directory Window Help

Back Forward Home Reload Images Open Print Find Stop

Netsite: 

What's New? What's Cool? Destinations Net Search People Software

WiseWire.com Home Index Help

How chained are you?
Click here to take the dependency quiz

Community rating: 9.1 out of 10  About: URL: <http://www.winespectator.com/Wine/Spectator/homepage>

Tell your Agent what you think. (below)

Like it?
 Yes
 Slightly
 No

RIEDEL *The world's finest wine glasses*

Wine Spectator

THE MOST COMPREHENSIVE WINE WEB SITE IN THE WORLD

Welcome to the Windy City

Wine Spectator editors bring you Chicago in the [current issue](#). Even the fabled Midwestern understatement can't conceal that this is a city on the move. According to our first annual [Readers' Choice Awards](#), Chicago chef Charlie Trotter is the best chef currently working in the United States, and his eponymous restaurant is the best restaurant. But the rest of the city is also delivering exciting dining experiences. You can pick up a list of award-winning [restaurants](#) on this site, though for full details, plus stories on hotels and wine bars, saloons and cigar bars, architecture, auctions, shopping and five of the city's premier residents, you'll have to check the October 15 issue.

[In the Current Issue](#)
Chicago
[Subscribe To](#)

SELECTIONS

- [Sign In](#)
- [Daily Report](#)
- [Wine Search](#)
- [Wine Menu](#)
- [Travel](#)
- [Dining](#)
- [Wine Forum](#)
- [Wine Library](#)
- [Weekly Poll](#)
- [Stock Quotes](#)
- [Events](#)

88K read (stalled)

<http://www.wisewire.com>

Where Is this Headed?

- First-generation algorithms: neural nets, decision trees, regression ...
- Applied to well-formatted database
- Budding industry
- Opportunity for tomorrow: enormous impact
 - Learn across full mixed-media data
 - Learn across multiple internal databases, plus the web and newsfeeds
 - Learn by active experimentation
 - Learn decisions rather than predictions
 - Cumulative, lifelong learning
 - Programming languages with learning embedded?

Contributing Disciplines

- Computer Science

Artificial Intelligence, Algorithms and Complexity, Databases, Data Mining

- Statistics

Statistical Inference, Experiment Design, Exploratory Data Analysis

- Mathematics

Abstract Algebra, Logic, Information Theory, Probability Theory

- Psychology and Neuroscience

Behavior, Perception, Learning, Memory, Problem Solving

- Philosophy


Ontology, Epistemology, Philosophy of Mind, Philosophy of Science


Application Areas


- Bioinformatics and Computational Biology
- Human Computer Interaction and Pervasive Computing
- Economics and Commerce
- Computer Assisted Collaborative Learning and Discovery
- Intelligent Information Infrastructure
- Digital Government Cognitive Modelling
- Robotics
- Engineering, . . .

What is Learning?

- Learning is a process by which the learner improves its performance on a task or a set of tasks as a result of experience within some environment
- Learning = Inference + Memorization
- Inference

Deduction 
$$\frac{\forall x \text{ At}(\text{Smoke}, x) \Rightarrow \text{At}(\text{Fire}, x) \quad \text{At}(\text{Smoke}, \text{Room1})}{\text{At}(\text{Fire}, \text{Room1})}$$

Induction 
$$\frac{\text{At}(\text{Smoke}, \text{Room2}) \wedge \text{At}(\text{Fire}, \text{Room2}) \quad \text{At}(\text{Smoke}, \text{Room1}) \wedge \text{At}(\text{Fire}, \text{Room1}) \quad \text{At}(\text{Ice}, \text{Room3}) \wedge \neg \text{At}(\text{Fire}, \text{Room3})}{\forall x \text{ At}(\text{Smoke}, x) \Rightarrow \text{At}(\text{Fire}, x)?}$$

Abduction 
$$\frac{\forall x \text{ At}(\text{Smoke}, x) \Rightarrow \text{At}(\text{Fire}, x) \quad \text{At}(\text{Fire}, \text{Room1})}{\text{At}(\text{Smoke}, \text{Room1})?}$$

What is Machine Learning?

- A computer program M is said to learn from experience E with respect to some class of tasks T and performance P , if its *performance* as measured by P on *tasks* in T in an environment Z improves with *experience* E .

Example 1:

T : Cancer diagnosis

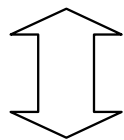
E : A set of diagnosed cases

P : Accuracy of diagnosis on new cases

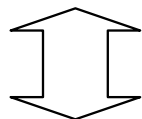
Z : Noisy measurements, occasionally misdiagnosed training cases

M : A program that runs on a general purpose computer

Data



Learning = Inference + Memorization



Knowledge

What is Machine Learning?

(Continued)

Example 2

T – solving calculus problems

E – practice problems + rules of calculus

P – score on a test

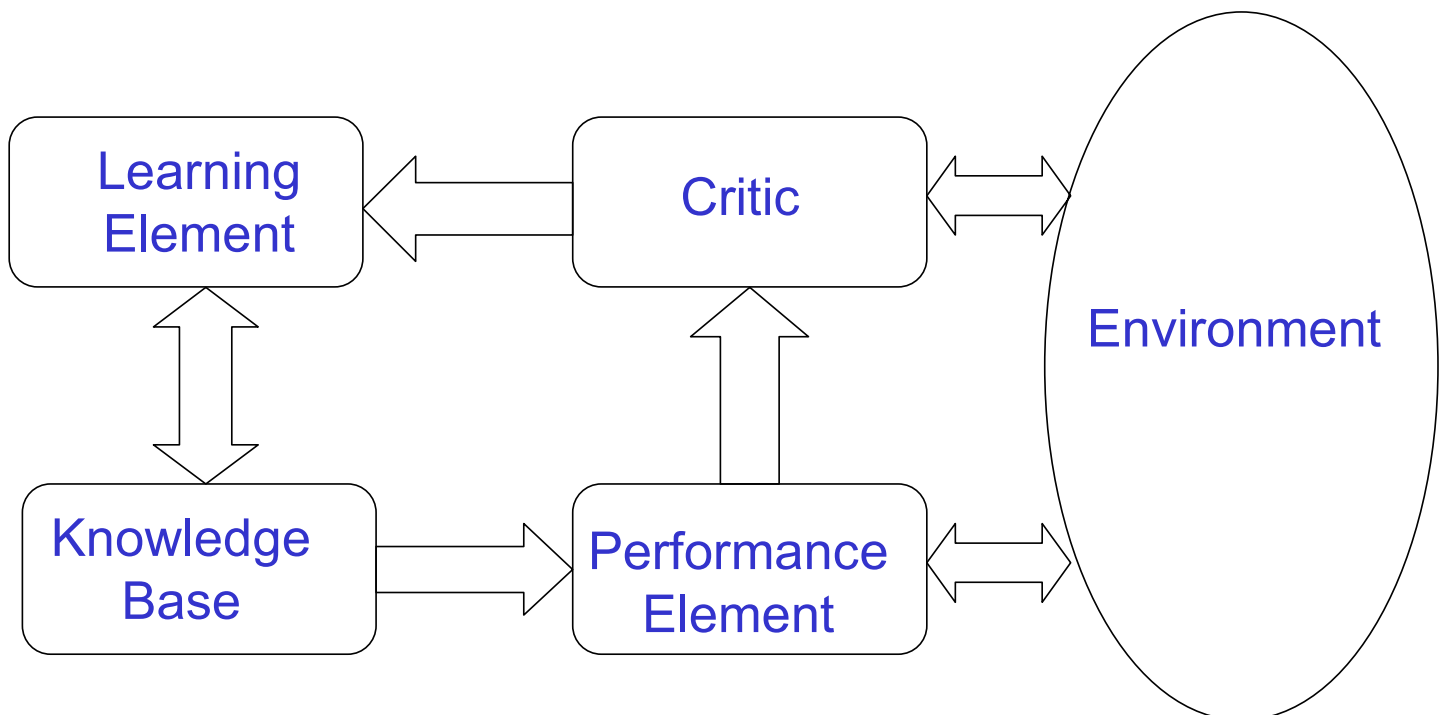
Example 3

T – driving on the interstate

E – a sequence of sensor measurements and driving actions recorded while observing an expert driver

P – mean distance traveled before an error as judged by a human expert

A General Model of Learning



Types of Learning

- Rote Learning: Useful when it is less expensive to store and retrieve some information than to compute it
- Learning from Instructions: Transform instructions into operationally useful knowledge
- Learning from Examples (and counter-examples): Extract predictive or descriptive regularities from data
- Learning from Deduction (and explanation): Generalize instances of deductive problem-solving
- Learning from Exploration: Learn to choose actions that maximize reward

Designing a Learning Program for a Task

- Experience: What experiences are available?

Data: in medical diagnosis, expert diagnosis cases, feedback

How representative is the experience?

- Critic: Can the learner ask questions?

What types of question?

When to ask questions?

How am I doing? — performance query

How would you diagnose X ? — example based query

Why was I wrong? — explanation

- Environment

Deterministic or stochastic?

Noisy or noise-free?

Designing a Learning Program for a Task

(Continued)

- Performance Element:

1. How is the learned knowledge encoded?

Rules, probabilities, programs, . . .

2. How is the learned knowledge used?

e.g. Matching rules

3. What is the performance measure?

4. How is the performance measured?

Online? Batch?

- Learning Element: What is the learning algorithm?

Search for a set of classification rules that are likely to perform well on novel cases (how?)

Estimate a class conditional probability distribution (how?)

Machine Learning

- Learning involves synthesis or adaptation of computational structures

Functions

Logic programs

Rules

Grammars

Probability distributions

Action policies

Behaviors

- Machine Learning =

(Statistical) Inference + Data Structures + Algorithms

Learning to Play Checkers

- What experience?
- What exactly should be learned?
- How shall it be represented?
- What specific algorithm to learn it?
- Playing checkers

T: Play checkers

P: Percent of games won in world tournament

E: opportunity to play against self or ...

Checkers: Type of Training Experience

- Direct or indirect?

Direct: Board states + Correct moves

Indirect: Sequence of moves + final outcome

- Teacher or not?

Supervised: Teacher provides examples

Semi-supervised: Learner proposes then ask teacher for advice

Un-supervised: Learner has control over everything

- Is training experience representative of performance goal?

Reliable learning: when distribution of training examples is similar to the distribution of the total set of examples

Checkers: Type of to be Learned

- Learn to choose *best* moves among *legal* moves

Known (a priori): Set of legal moves

Unknown: Best search strategy

- Knowledge to be learned = Target function

- Choose the target function

1. $ChooseMove : B \rightarrow M$??

B = Set of legal board states

M = Set of legal moves

hard to learn

2. $V : B \rightarrow \mathfrak{R}$??

\mathfrak{R} = Set of real values

Should assign higher scores to better board states

Use V to choose best *successor* state

Checkers: Definition for Target Function

1. If b is a final board state that is won, then $V(b) = 100$
2. If b is a final board state that is lost, then $V(b) = -100$
3. If b is a final board state that is drawn, then $V(b) = 0$
4. If b is not a final state in the game, then $V(b) = V(b')$, where b' is the best final board state that can be achieved starting from b and playing optimally until the end of the game.
 - Requires searching ahead
 - This gives correct values, but is not operational
 - Goal of learning: Discover ideal function V within realistic time bounds
 - Reasonable goal of learning: Discover an approximation $\hat{V}(b)$ of V within reasonable time

Checkers: Representation for Target Function

- Possible choices:
 1. Collection of rules?
 2. Neural network?
 3. Polynomial function of board features?
 4. Population of chromosomes?
 5. Decision tree?
 6. Relation?
 7. Table?
 8. Set of cases?
 9. ...
- Degree of expressivity of representation

Checkers: Representation for Learned Function

$$\hat{V}(b) = w_0 + w_1 \cdot bp(b) + w_2 \cdot rp(b) + w_3 \cdot bk(b) + w_4 \cdot rk(b) + w_5 \cdot bt(b) + w_6 \cdot rt(b)$$

- $bp(b)$: number of black pieces on board b
- $rp(b)$: number of red pieces on b
- $bk(b)$: number of black kings on b
- $rk(b)$: number of red kings on b
- $bt(b)$: number of red pieces threatened by black (i.e., which can be taken on black's next turn)
- $rt(b)$: number of black pieces threatened by red

Checkers: Estimating Training Values

- Recall: Indirect training experience
- $V(b)$: the true target function
- $\hat{V}(b)$: the learned function
- $V_{train}(b)$: the training value
- Rule for estimating training values:

$$V_{train}(b) \leftarrow \hat{V}(Successor(b))$$

We use estimates of the value of $Successor(b)$ to estimate the board state b

\hat{V} tends to be more accurate for board states closer to game's end

Checkers: Weight Tuning Rule

Find the weights w_i that *best fit* the training data.
That is: which set of weights *minimizes* the (*mean squared*) error E between training values $V_{train}(b)$ and predicted values $\hat{V}(b)$?

We seek the weights (or equivalently, the \hat{V}) that minimize E for the observed training set

LMS weight update rule: Do repeatedly:

- Select a training example b at random

1. Compute $error(b)$:

$$error(b) = V_{train}(b) - \hat{V}(b)$$

2. For each board feature f_i , update weight w_i :

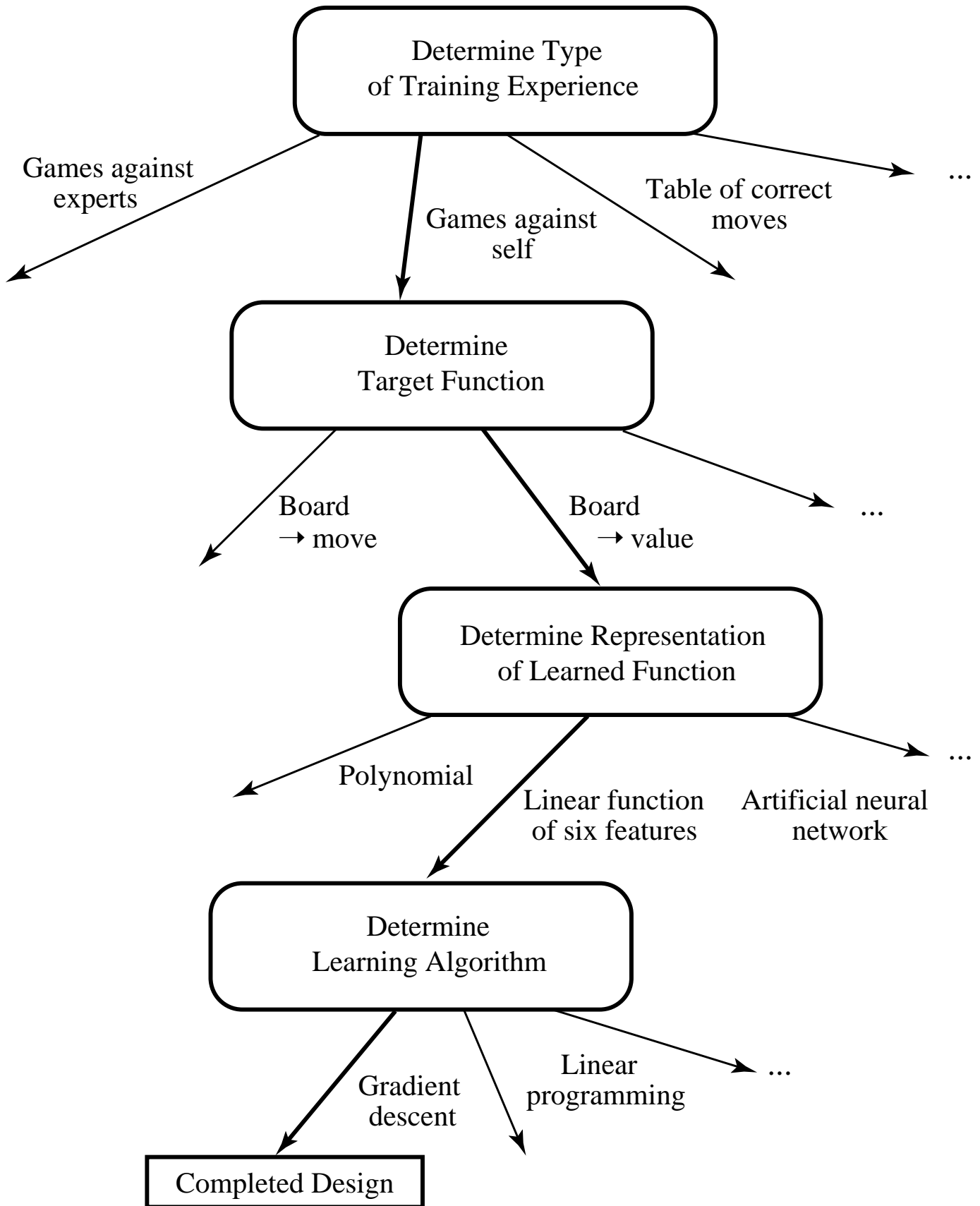
$$w_i \leftarrow w_i + c \cdot f_i \cdot error(b)$$

c is some small constant, say 0.1, to moderate the rate of learning

LMS adjusts the weights in the direction that reduces the error

No weight change if error is 0

Design Choices



Some Issues in Machine Learning

- What algorithms can approximate functions well (and when)?
- How does number of training examples influence accuracy?
- How does complexity of hypothesis representation impact it?
- How does noisy data influence accuracy?
- What are the theoretical limits of learnability?
- How can prior knowledge of learner help?
- What clues can we get from biological learning systems?
- How can systems alter their own representations?