Vorbemerkung 1

"Multimedia II" kann im Master entweder als **Multimedia**veranstaltung oder als **Datenbank**veranstaltung („Datenbanken und Informationssysteme“) eingebracht werden

Gilt seit SS 2011

Vorbemerkung 2

Zusatzmaterial

- 2-01c Ergänzung – Ableitungsregeln
- 2-01d Ergänzung - Mathematische Grundlagen
  - 2.2 Matrix-Vektor Multiplikation
  - 2.3 Ableitungsregeln und mehrdimensionale Ableitung

Was ist ein Ingenieur?

1 Introduction

SS 19 – Machine Learning and Computer Vision

Multimedia Computing & Computer Vision, Universität Augsburg
Rainer.Lienhart@informatik.uni-augsburg.de
www.multimedia-computing.{de,org}

Outline

- Why has machine learning become a must to know for computer science students?

- What is a well-defined learning problem?

- An example: *learning to play checkers*

- What questions should we ask about Machine Learning?

Why Machine Learning

- Massive data available
  - Everything has becomes digital, is measured and collected
- Massive compute power available
- Progress in algorithms and theory in the past decades
  - big data analysis or mining

Three examples for machine learning:

- Data mining: using historical data to improve decisions
  - medical records → medical knowledge
- Software applications we can't program by hand
  - autonomous driving, speech recognition
- Customizing programs/services (="reading" the user)
  - Amazon, Facebook, Google learns your preference in order to place profitable ads
  - Sensing homes learn to save energy

Typical Data Mining Task

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:

Patient103 time=1	→ *Patient103* time=2	... → *Patient103* 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**
...	...	...

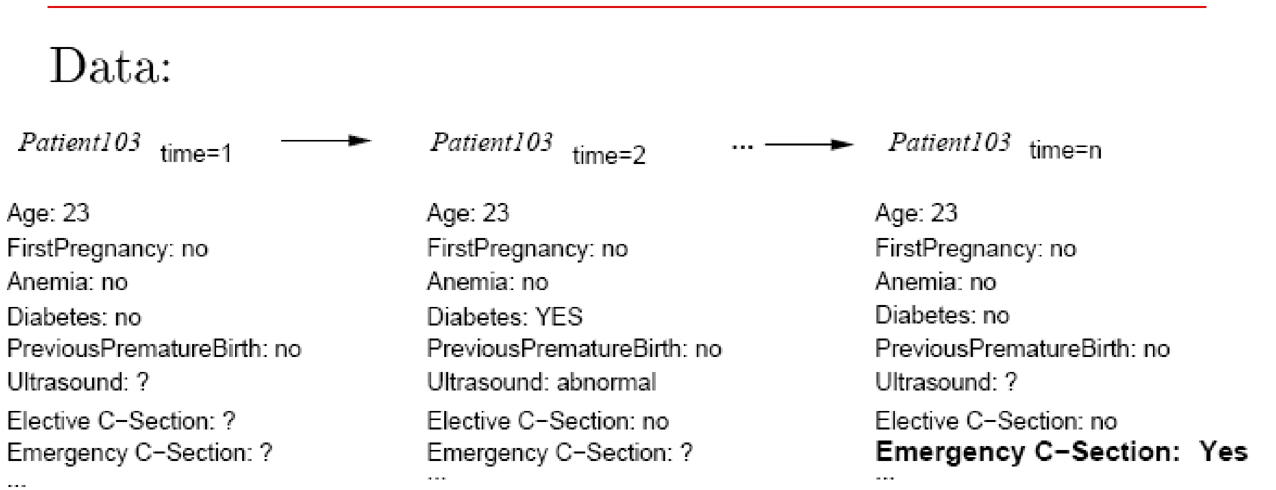
Blutarmut

Data Mining Results

- 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:

Customer103: (time=t0)

Years of credit: 9
Loan balance: $2,400
Income: $52k
Own House: Yes
Other delinquent accts: 2
Max billing cycles late: 3
Profitable customer?: ?
...

Customer103: (time=t1)

Years of credit: 9
Loan balance: $3,250
Income: ?
Own House: Yes
Other delinquent accts: 2
Max billing cycles late: 4
Profitable customer?: ?
...

...

Customer103: (time=tn)

Years of credit: 9
Loan balance: $4,500
Income: ?
Own House: Yes
Other delinquent accts: 3
Max billing cycles late: 6
Profitable customer?: No
...

Rules learned from 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]
```

Customer purchase behavior:

Customer103: (time=t0)	*Customer103:* (time=t1)	...	*Customer103:* (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**
...	...		...

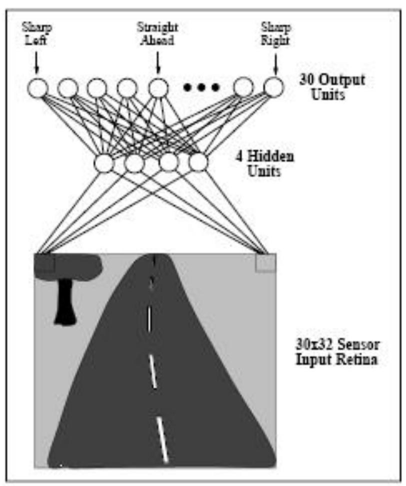
Today: Your browser and your cell phone apps collect the necessary data

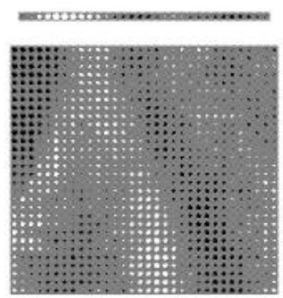
Too Difficult to Program by Hand

ALVINN [Pomerleau] drives 70 mph on highways

1989

Camera resolution:
30 x 32





Ouput:
1 of 30 steering commands

Software that Customizes to User

Query:

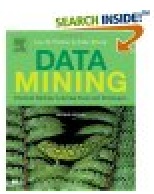
Result:

Machine Learning. (Mcgraw-Hill International Edit) (Taschenbuch)
von Tom M. Mitchell (Autor) "Ever since computers were invented, we have wondered whether they might be made to learn ..." (mehr)
(5 Kundenrezensionen)

Statt: ~~EUR 67,95~~
Jetzt: **EUR 67,45** Kostenlose Lieferung. Siehe Details.
Sie sparen: EUR 0,50 (1%)

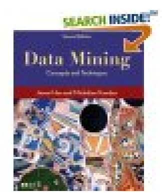
Source: www.amazon.de

Kunden, die diesen Artikel gekauft haben, kauften auch:

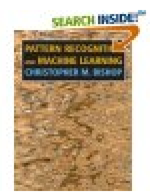


Data Mining. Practical Machine Learning Tools and Techniques (Morgan Kaufmann Series in

Foundations of Statistical Natural Language Processing von Christopher D. Manning



Data Mining. Concepts and Techniques (Morgan Kaufmann Series in Data Management



Pattern Recognition and Machine Learning (Information Science and Statistics) von Christopher M. Bishop

An Introduction to Support Vector Machines. And Other Kernel-based Learning Methods

Where is this headed?

Data collection & machine learning is everywhere

- Cars, cell phones, browser, apps
- Data mining, big data analysis, AI
- Robotics
- …. What else?

Opportunity for tomorrow

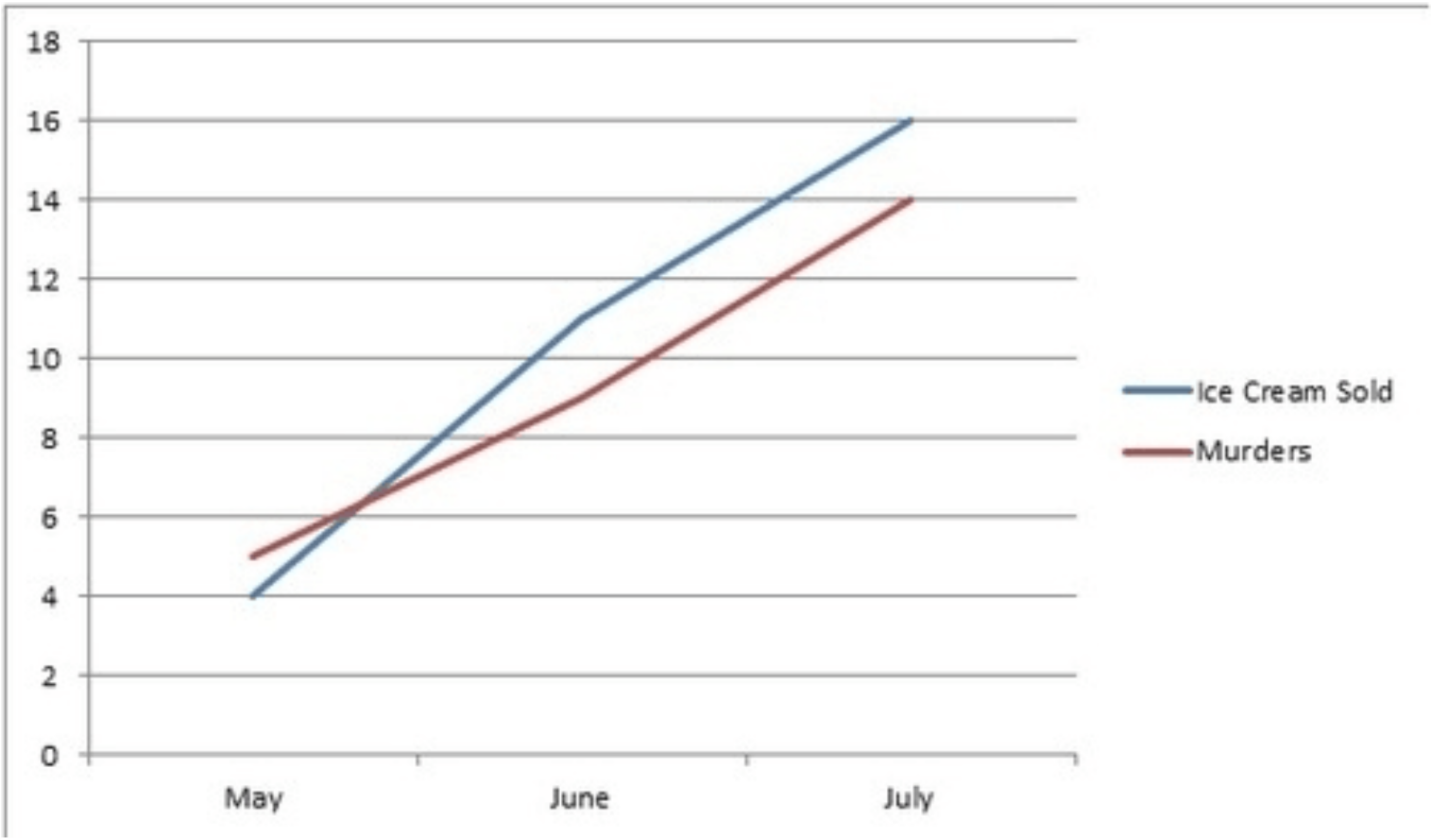
- Learn across mixed-media
- Learn across multiple internal databases, plus the web and newsfeeds
- Learn by active experimentation (Reinforcement Learning)
- Cumulative, lifelong learning
- Programming languages with learning as integral part?
- Explainable AI

Challenges

- Free will
- Privacy
- "Skynet" (Terminator)
- Absurd correlations

Absurd Correlation – Example 1

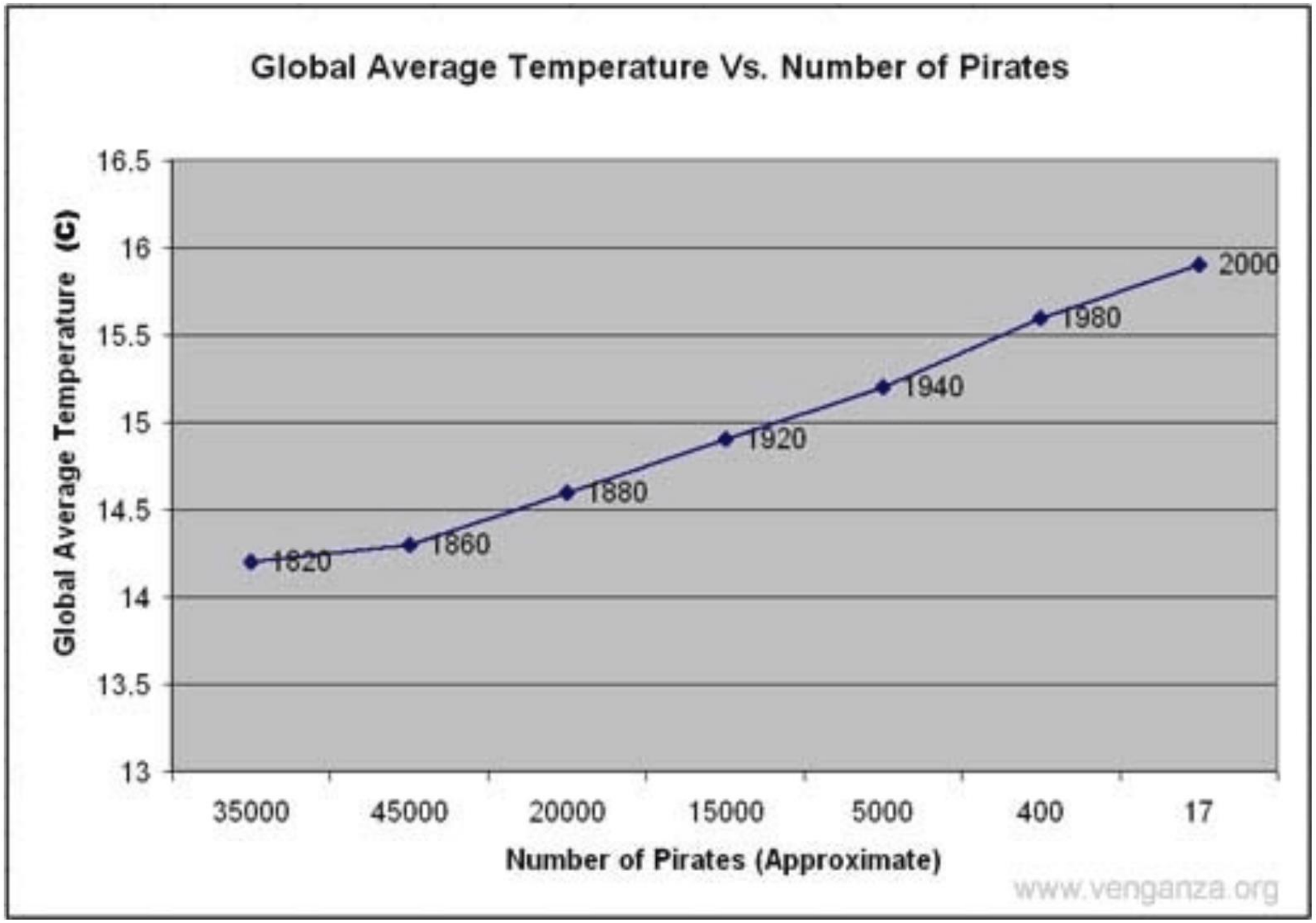
1. Ice cream consumption leads to murder.



badpsychologyblog.org

Absurd Correlation – Example 2

2. A pirate shortage caused global warming.



Relevant Disciplines

- Artificial intelligence
- Bayesian methods
- Computational complexity theory
- Control theory
- Information theory
- Philosophy
- Psychology and neurobiology
- Statistics
- ...

What is the Learning Problem?

What is the Learning Problem?

Learning = Improving with experience at some task:

- Improve over task *T*,
- with respect to performance measure *P*,
- based on experience *E*.

E.g., Learn to play checkers:

- *T*: Play checkers,
- *P*: % of games won in world tournament,
- *E*: opportunity to play against oneself.

Examples of Task T (1)

- Classification
  - e.g., an instance is described by feature $\boldsymbol{x} \in \mathbb{R}^n$, then
    a) $f(\boldsymbol{x}): \mathbb{R}^n \to \{1, \dots, k\}$ or
    b) $P(\boldsymbol{x}): \mathbb{R}^n \to \{p_1, \dots, p_k\}$ mit $\sum_{i=1}^{k} p_i = 1$, $p_i \geq 0$ und $\boldsymbol{p} \in \mathbb{R}^k$.

- Classification with missing inputs
  - Must learn a set of functions

- Regression

  Approximate some function $f^*(\boldsymbol{x}): \mathbb{R}^n \to \mathbb{R}$
  - e.g., by $f(\boldsymbol{x}; \boldsymbol{\theta}): \mathbb{R}^n \to \mathbb{R}$.
    $\boldsymbol{\theta}$ are the parameters that are learned

Examples of Task T (2)

- Transcription
  - Transcribe the information in some relatively unstructured representation into a discrete textual form
  - E.g., OCR, speech recognition

- Machine Translation
  - Convert a sequence of symbols in one language into a sequence of symbols in another language.
  - E.g., translate from French to English

- Anomaly detection

Examples of Task T (3)

- Structured output
  - Any task where the output data is some data structure with important relationships between the different output elements.
  - E.g. image captioning, parse tree, bounding boxes, transcription and translation

- Synthesis and sampling
- Imputation of missing values
- Denoising
- Density estimation

Learning to Play Checkers

- *T*: Play checkers
- *P*: Percent of games won in world tournament

Questions we will address in the following:

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

Type of Training Experience

- Direct or indirect?

  → Problem of credit assignment

  - Direct: E.g., set of board states with correct moves
  - Indirect: E.g., set of actual games played (all moves) with final outcome

- Teacher or not?

  → Everything given, oracle, or play against oneself? (= All given, can ask questions, or autonomous exploration)

A problem: is training experience representative of performance goal?

→ How representative is your training data?

- Common assumption: Training and test data are i.i.d

Choose the Target Function

➔What type of knowledge should be learned?

Input: set of legal board states

- ***ChooseMove***: *Board* → *Move* ??

  →Starting point: Learn to choose best move among legal moves

- ***V***: *Board* → $\Re$??

  →Generate the successor states produced by every legal move. Then use *V* to choose the best successor state and therefore best legal move.

Possible Definition for Target Function *V*

- If *b* is a final board state that is won, then *V*(*b*) = 100
- If *b* is a final board state that is lost, then *V*(*b*) = -100
- If *b* is a final board state that is drawn, then *V*(*b*) = 0
- If *b* is a 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.

This gives correct values, but is not operational.

– What does ‘not operational’ mean?
– Why?
– Should V(b) only take on three values?

Choose Representation for Target Function

Many options:

- Collection of rules?
- Neural network?
- Polynomial function of board features?
- ...
- Linear regression

A Representation for the 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

More Compact Formulation

$$\hat{V}(b) = \sum_{i=0}^{6} w_i \cdot f_i\,(b) \quad \text{with} \quad f_0 = 1$$

- $f_1(b)$:= *bp*(*b*): number of black pieces on board *b*
- $f_2(b)$:= rp(*b*): number of red pieces on *b*
- $f_3(b)$:= *bk*(*b*): number of black kings on *b*
- $f_4(b)$:= *rk*(*b*): number of red kings on *b*
- $f_5(b)$:= *bt*(*b*): number of red pieces threatened by black (i.e., which can be taken on black's next turn)
- $f_6(b)$:= *rt*(*b*): number of black pieces threatened by red

Obtaining Training Examples

Training examples:

D = {<b,$V_{train}(b)$>}

- $V(b)$: the true target function
- $\hat{V}(b)$: the learned function
- $V_{train}(b)$: the training value

One rule for estimating training values:

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

Use current value function to refine data

Note

- Only the results of complete games are available (won, lost, draw)
- Need to assign values to the intermediate board states
- Above is just one simple approach that has proven to be successful
- *Successor*(b) := next board state following b for which it is again the program‘s turn to move (i.e., the board state following the program‘s move and the opponent‘s response.

What Means ’*Best Fit*’?

Common Error Function:

- MSE = Mean Squared Error (value)
- LMS = Least Mean Squares (algorithm)
- LSE = Least Squared Error (result)

$$E = \cancel{\frac{1}{|\text{training examples}|}} \sum_{\langle b, V_{train}(b)\rangle \in \mathrm{D}} \left(V_{train}(b) - \hat{V}(b)\right)^2$$

$$E = \sum_{\langle b, V_{train(b)}\rangle \in \mathbf{D}} \left(V_{train}(b) - \sum_{i=0}^{6} w_i f_i(b)\right)^2 \rightarrow min$$

LMS Weight update rule:

Do repeatedly:

- Select a training example <b,$V_{train}(b)$> at random from **D**
  1. Compute *error*(b):

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

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

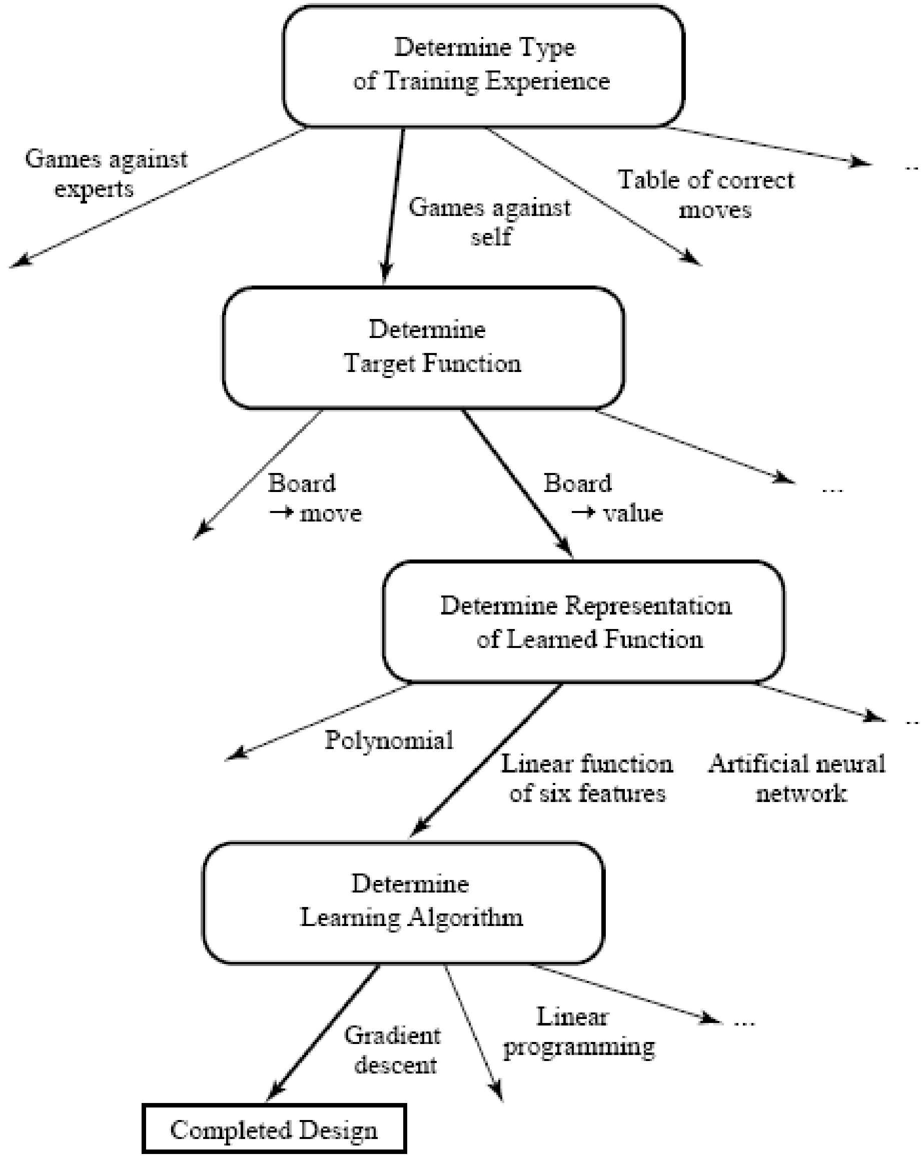
     $$w_i \leftarrow w_i + c \cdot f_i \cdot error(b)$$

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

Consider the following cases:
Case 1: $error(b) == 0$
Case 2: $error(b) > 0$
Case 3: $error(b) < 0$

Why LMS? → Exercise

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 the complexity of the hypothesis representation impact it?
- How does noisy data influence accuracy?
- What are the theoretical limits of learnability?
- How can prior knowledge of the learner help?
- What clues can we get from biological learning systems?
- How can systems alter their own representations?