COMP 4360
Machine Learning

Jacky Baltes
Autonomous Agents Lab
University of Manitoba
Winnipeg, Canada
R3T 2N2

Email: jacky@cs.umanitoba.ca
WWW: http://www.cs.umanitoba.ca/~jacky
http://aalab.cs.umanitoba.ca

ROAS Information

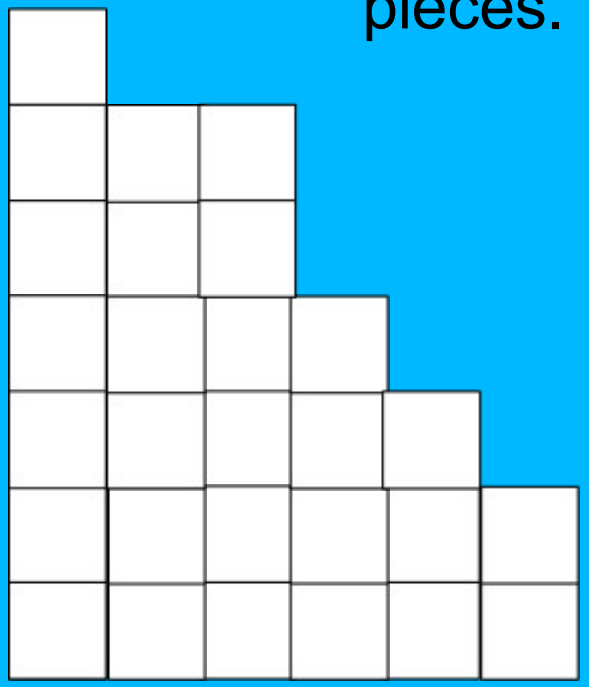
- Timetable: M,W,F: 15:30, EITC E2 304
- Assessment:
  - 30% Assignments (3 Assignments)
  - 20% Midterm test
  - 50% Final exam
- Recommended Reading: Tom Mitchell "Machine Learning," McGraw Hill, 1st Edition, ISBN 0-07-042807-7, 1997.

Academic Dishonesty

- Students are reminded that there are penalties for academic dishonesty. Academic dishonesty includes submitting assignments that are not entirely the student's own work.
- See the UofM Calendar: Academic Dishonesty and Plagiarism and Cheating for more information. A declaration sheet, which states that the work being submitted is completely your own, is available at http://www.cs.umanitoba.ca/honesty.html. This sheet must be printed out, filled in, signed, and attached to every which is submitted. No assignment will be marked unless the declaration is attached.

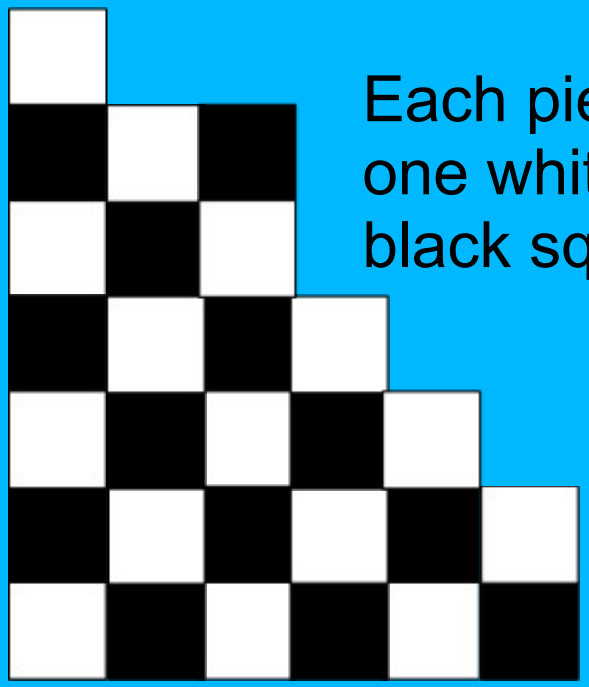
Importance of Representation

Cover the area on the left with domino pieces.



Representation
Mutilated Checkerboard Problem

Think about it as a mutilated checkerboard.



Each piece covers one white and one black square

Machine Learning

- Growing flood of data in our society today
  - 20th century to create data
  - 21st century to analyze data
- Computational power and storage capacity are increasing
- New industry (E-commerce, sales, marketing)

Machine Learning

- Data mining: Using historical data to look for patterns
- Machine learning: technical basis for data mining
- Applications that are too hard to program by hand
  - Autonomous driving, robotics
  - Speech recognition
- Customizable software and interfaces
  - OS queuing
  - Newsreader that learns your interest
  - Spam filter

Machine Learning

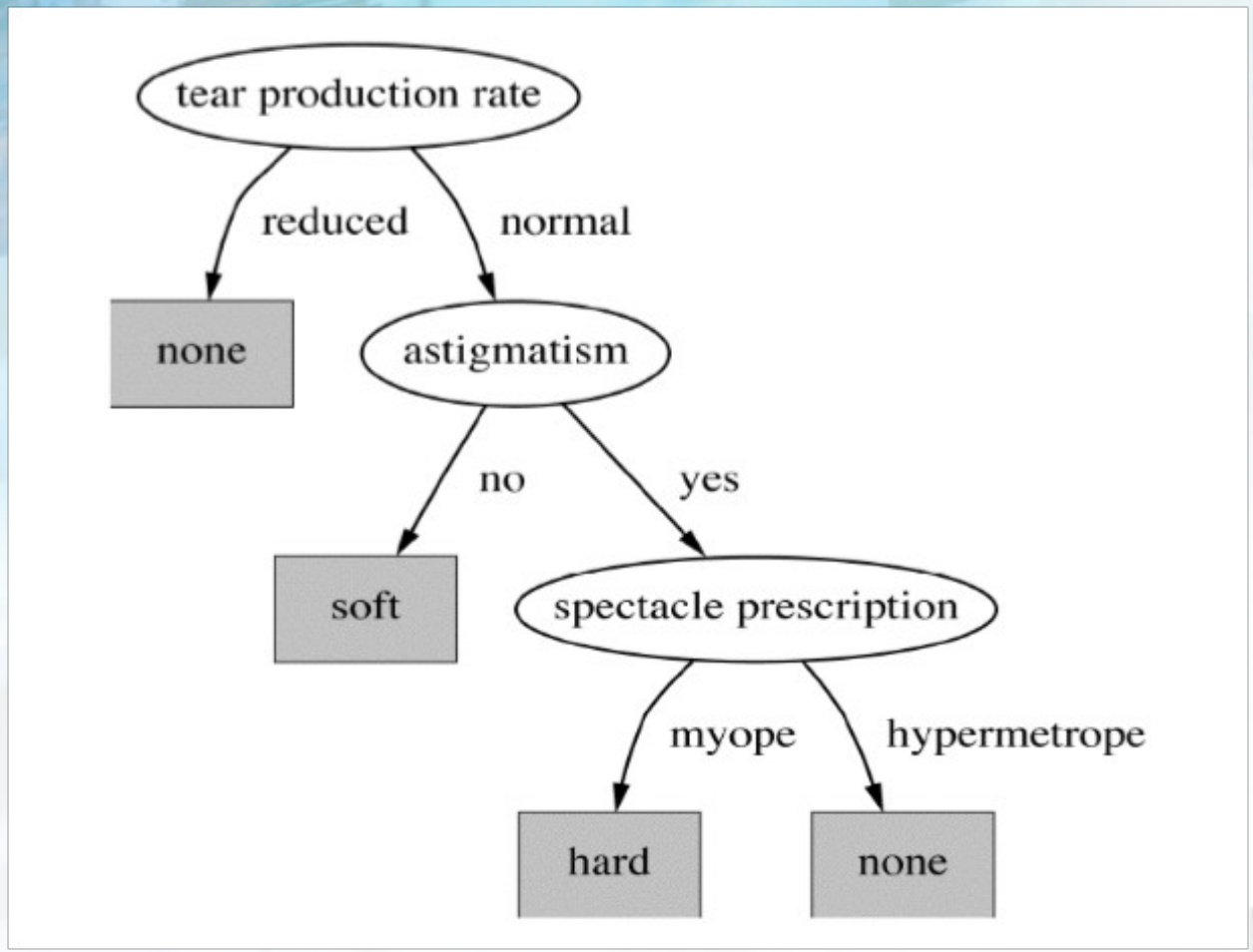
- Structural description of the learned patterns
  - Can be used to make predictions
  - Understand and explain why a prediction was made
- Contact lens database
  - If..then.. rules

```
If tear production rate = reduced then recommendation = none
Otherwise, if age = young and astigmatic = no
  then recommendation = soft
```

Age	Spectacle prescription	Astigmatism	Tear production rate	Recommended lenses
Young	Myope	No	Reduced	None
Young	Hypermetrope	No	Normal	Soft
Pre-presbyopic	Hypermetrope	No	Reduced	None
Presbyopic	Myope	Yes	Normal	Hard
...	...	...	...	...

Machine Learning

- Decision tree: Test an attribute at each level of the tree

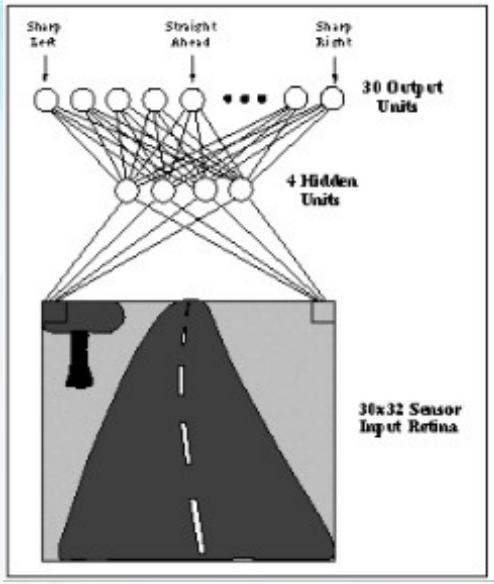


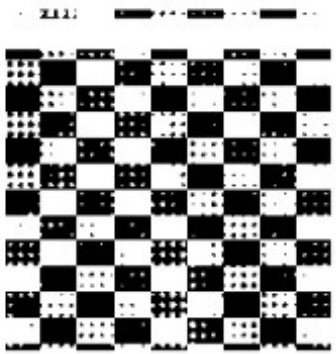
Typical Machine Learning Tasks

- Patient records (all information)
  - Learn to predict when a patient is likely to require emergency medical attention
- A user executes a series of commands: Induce the intention of the user and complete the task
  - Or, customize the software to the user's perceived abilities
- Automatically organize new bookmark entries/mail messages into existing folders
- Credit Risk Analysis

Problems to Difficult to Program by Hand

ALVINN [Pomerleau] drives 70 mph on highways





Autonomous Driving

- A lot of interest and improvements in recent years
- Video: Autonomous Sliding Parking
  - http://www.youtube.com/watch?v=gzI54rm9m1Q
  - Stanford University Thrun, Ng, et. al.

Relevant Disciplines "Adaptation"

- Artificial Intelligence
- Bayesian Methods
- Computational Complexity Theory
- Control Theory
- Information Theory
- Philosophy
- Psychology and Neurobiology
- Statistics

Definition of Learning

- Learning = Improving with experience at some task
- Improve over task T with performance measure P based on experience E
- Checkers
  - Task T - Play according to the rules
  - Performance P – Percentage of games won
  - Experience E - Play against yourself

Practical Applications
Loan Applications

- Processing loan applications
  - Should we lend money?
  - 90% covered by statistical tests (correlations)
  - Borderline send to case officer
  - 50% of accepted borderline cases default on the loan
- Rule induction system
  - Predicted correct outcome 66% of the time
  - Rules can be explained to customers and new staff

Practical Applications
Image screening

- A lot of image information is available. Requires highly trained personnel to be identified
  - Stars
  - Breast cancer
  - Oil slicks
  - Forest fires
  - Cell components to build models

Practical Applications
Load Forecasting

- Electricity company need to forecast minimum and maximum load
- Statistical model for yearly variations
- Variations due to weather are ignored

Practical Applications
Machine Fault

- Diagnosis: expert domain
- Vibrations measured at different points of the machine
- What fault is present?
- Information very noisy
- Learned rules outperformed hand-crafted rules after a number of iterations

Practical Applications Marketing and Sales

- Customer loyalty: Identify customers that are likely to start purchasing from another company
- Special offers: Identify highly profitable customers. Extra cash for the holidays on your credit card
- Market basket analysis: Which items occur together in a shopping basket. Put them closer together in the store
- Prospective customers: focused and targeted advertising

Practical Applications
Call Center Scheduling for Electrical Company

- Estimate the number of staff needed in the call center of an utility company
- Days and events
  - Day of the week, billing date
  - Weather: temperature, rain, ...
  - Plant operation: Electrical load, ...
  - Customer relations: new services, ...

Practical Applications
Call Center Scheduling for Electrical Company

- Estimate the number of staff needed in the call center of an utility company
- Days and events
  - Day of the week, billing date
  - Weather: temperature, rain, ...
  - Plant operation: Electrical load, ...
  - Customer relations: new services, ...
- One rule
  - Directly proportional to number of terminations notices send last week
  - Duh!!! If you shut off people's power, they are likely to call

Learning to Play Checkers

- T – Play checkers
- P – Percentage of games won
- E- Experience
  - What experience?
  - What should be learned?
  - What representation do we use?
  - What algorithms should be used?

Type of Training Experience

- Direct or indirect
  - Direct: presentation of moves and outcome
  - Indirect: other types of feedback, e.g. randomly selected moves in a game and the overall outcome
- Teacher or not?
  - A teacher can select experiences that promote fast and correct learning; it might take a great deal of time to cover many of these over the course of everyday play
- Training experience must be representative of performance goal

Target Function

- What are we trying to learn
  - ChooseMove(Board) -> Move [Direct mapping to best next move]
  - V(Board) -> R [target function: what's the outcome of this board? Can choose moves indirectly from this]
- V(b) can be defined recursively
  - V(Lost) = -100
  - V(Won) = +100
  - V(b) = V(b') where b' is the best board state that can be reached from b by playing optimally, e.g. Minimax
- State representation = One V(b) for each state
  - Need to represent entire state space
  - 10^120 states for chess
  - Not tractable (need for V^(b) – best estimate of target function)
- Must visit each state (possibly more than once) to learn correct value

Representation of the Target Function

- Many similar states have similar values V(b)
- So instead we need to come up with tractable/practical representations
  - Group many similar states together
  - Find a practical representation to group them together
- Popular approaches
  - Collection of rules
    - If knight on the edge of the board and no rook then V(b) = V(b) - 5
    - If queen in the center, then V(b) = V(b) + 5
    - Sum up all values to estimate value of current position
  - Neural networks
    - Simplified model of processing in the human brain

A Learned Function
Linear combination of six board features

- Polynomial function of predetermined board features are also very popular. E.g., Samuels Checker player.
- During learning, the system learns the weights w0,...w5 of each feature
- Many states map to the same value function

$$w_0+w_1\cdot bp(b)+w_2\cdot rp(b)+w_3\cdot bk(b)+w_4\cdot rk(b)+w_5\cdot bt(b)$$

- $bp(b)$: 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

What to learn from?

- V(b): correct target function
  - What is the correct value for a position/state in chess
  - This is most often only known for some states (checkmate, draw)
- V^(b): learned function
  - This is currently our best estimate of the correct value of V(b)
- $V_{train}(b)$: the training value
  - The value as indicated by new training data. V^(b) = 90 indicated that a state is very good. Then we got checkmated two moves later. Now we know that that state is bad $V_{train}(b)=-99$.
- Error minimization
  - Try to reduce the error between V^(b) and $V_{train}(b)$

Choose weight training rule

Many different ways in which we can change the weights to reduce this error. Change weights such that $\hat{V}(b) = V_{train}(b) = -99$.

The Least Mean Squared (LMS) error update rule is very popular. Much more on this later in the course.

To compensate for noise we usually only change our weights a little bit each time by using a learning rate

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:

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

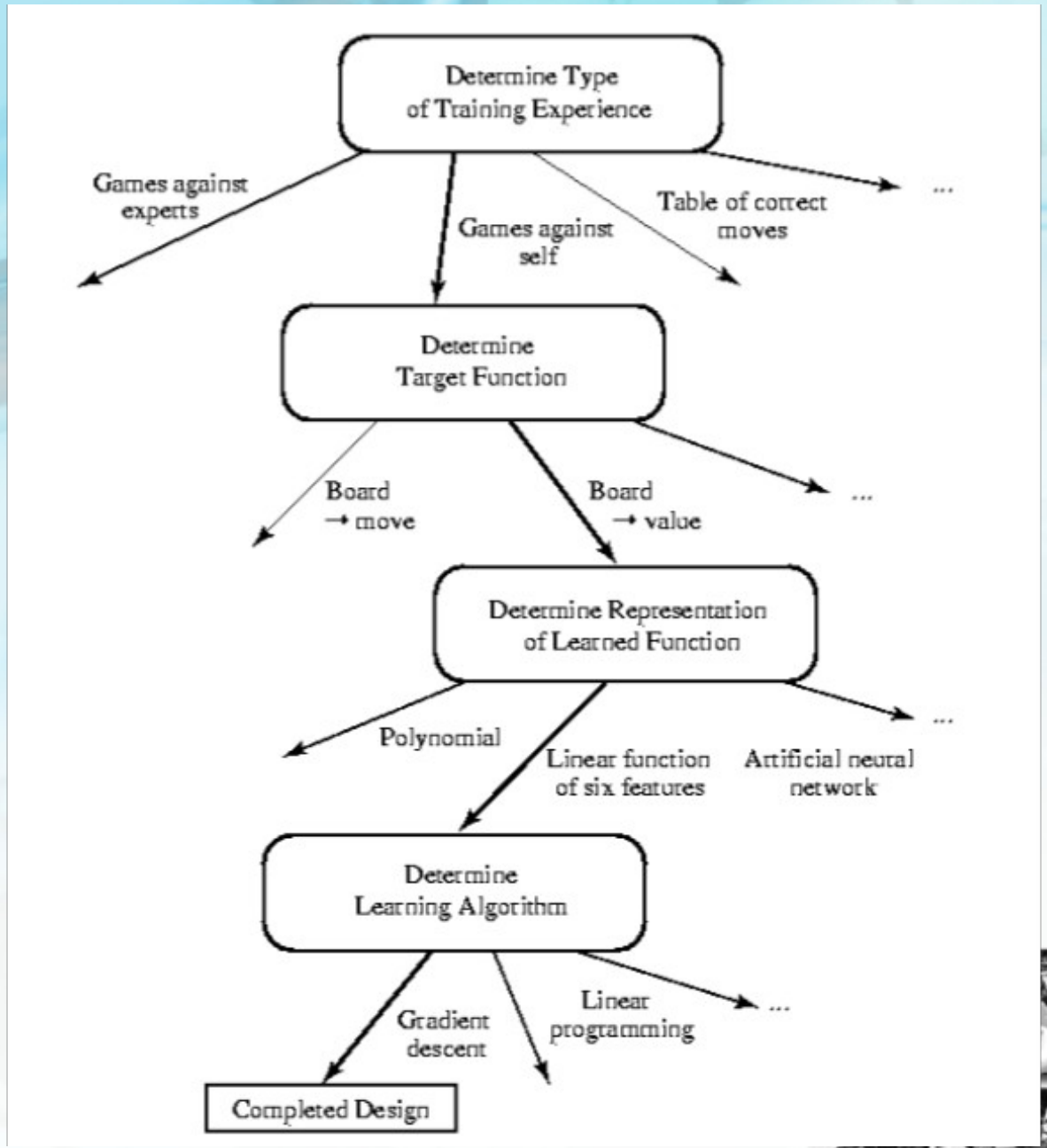
c is some small constant, say 0.1, to moderate the rate of learning

Design Choices

Most ML researchers often passionately argue about their favorite ML algorithm

But as a ML practitioner, you must decide on training experience, target function, and representation first

In fact, if you choose those well, then the choice of ML algorithm is often secondary. Good choices to the first three problems will make it so that any suitable ML algorithm will be able to learn it



Generalization as Search

- Inductive learning of classifiers:
  - Find a concept description that fits the data
- Example:
  - Rule set as description language (hypothesis space)
- Simple algorithm:
  - Search through the hypothesis space (all possible concepts that can be learned)
  - Eliminate the ones that do not fit
- Surviving descriptions match the training data

Machine Learning as Search

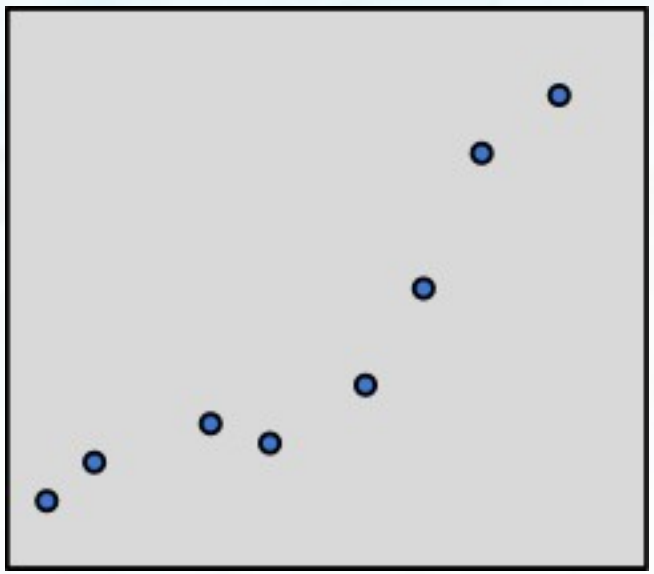
- In introductory AI, you learned about (avoiding) search as the underlying problem in AI.
  - In path planning, we search through the space of possible routes from the initial state to the goal state
  - In problem solving, we search through the space of possible action sequences to achieve a given goal
  - In computer vision, we search to find the most likely state of the world given the information in the image
  - In natural language processing, we search for the most likely sentence given an utterance by a human speaker

Generalization and Overfitting

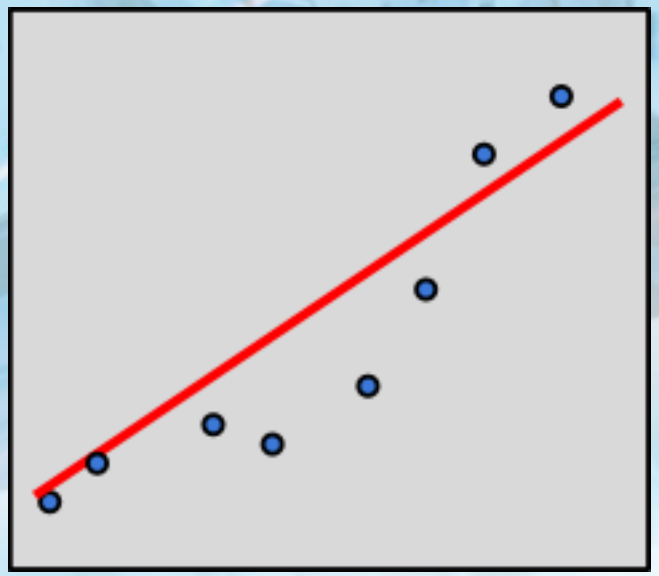
- Generalization is the most important ability of a ML system
  - Correctly indentifying board positions that you have seen before is easy.
  - Just store them in a database and retrieve
  - However, we want to learn to play positions that we have never seen before correctly as well – we have to be able to generalize as oppose to only respond to what we have seen precisely
- Problem is to avoid *overfitting*

Overfitting

- We are only shown a small subset of all the data
- Induction: infer general rule from samples
- Example: What is the best function to describe the following dataset?

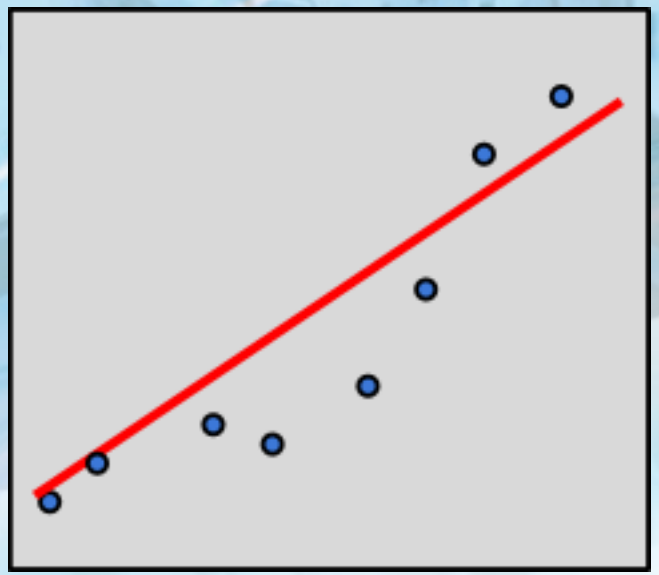


Overfitting

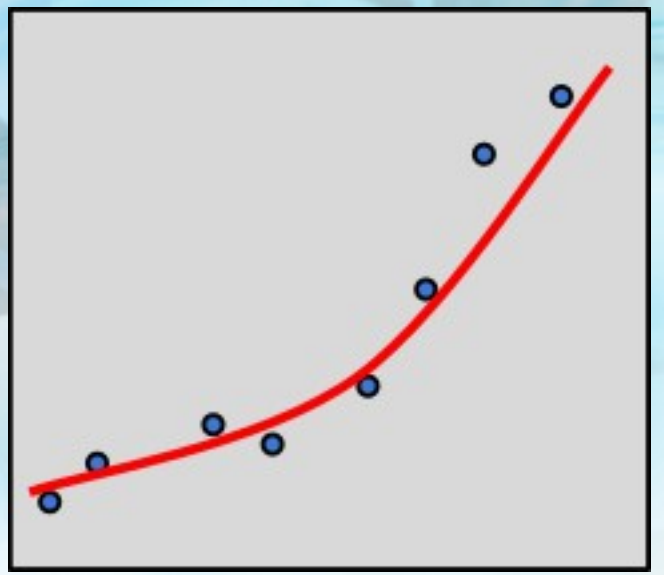


Straight line

Overfitting

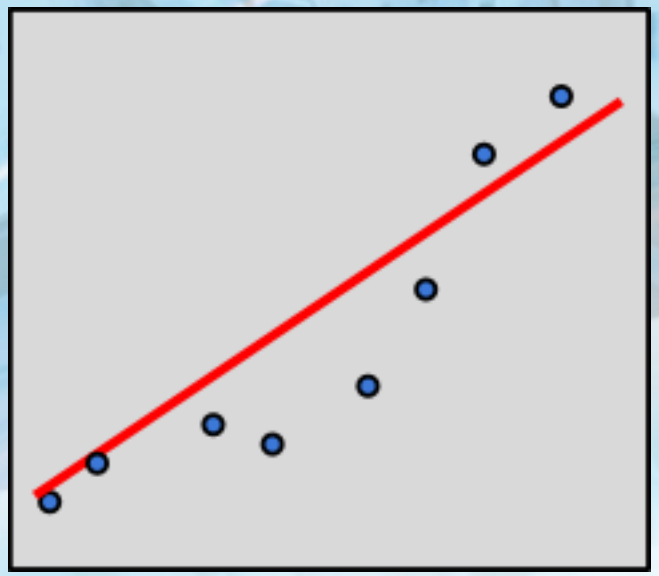


Straight line

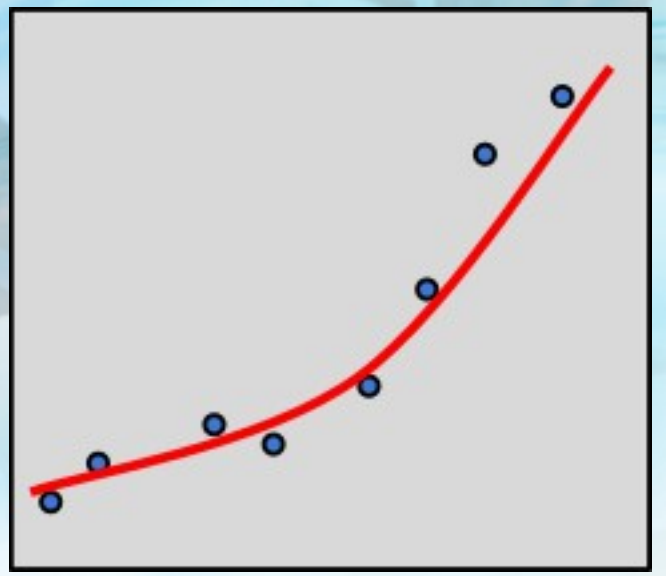


Parabola

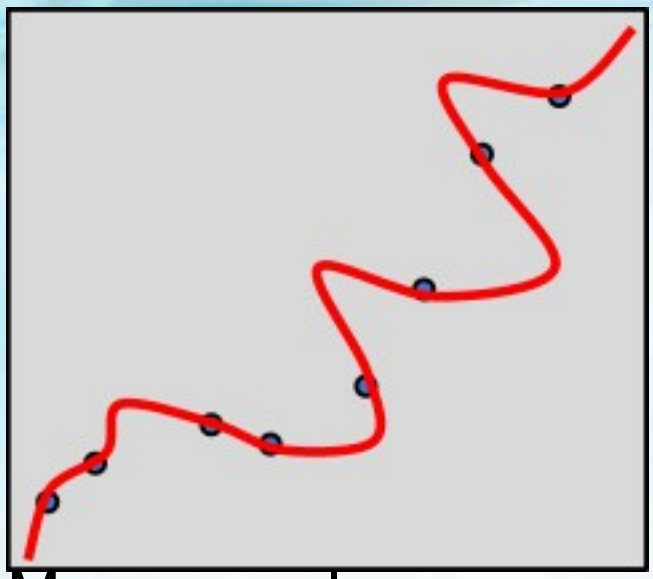
Overfitting



Straight line



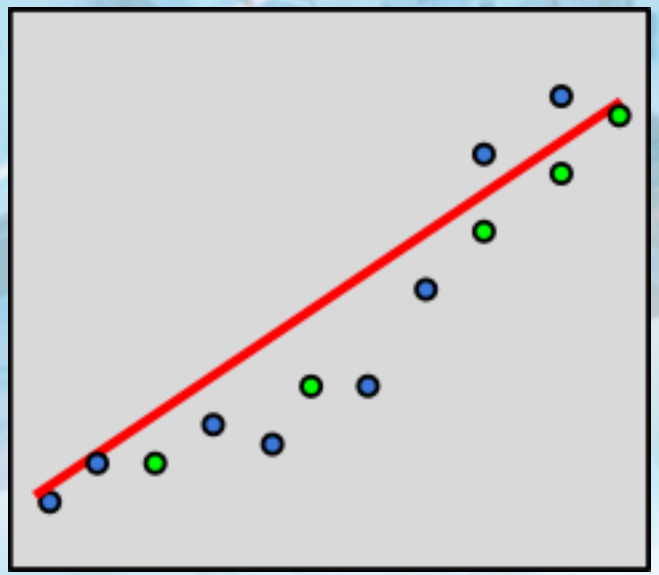
Parabola



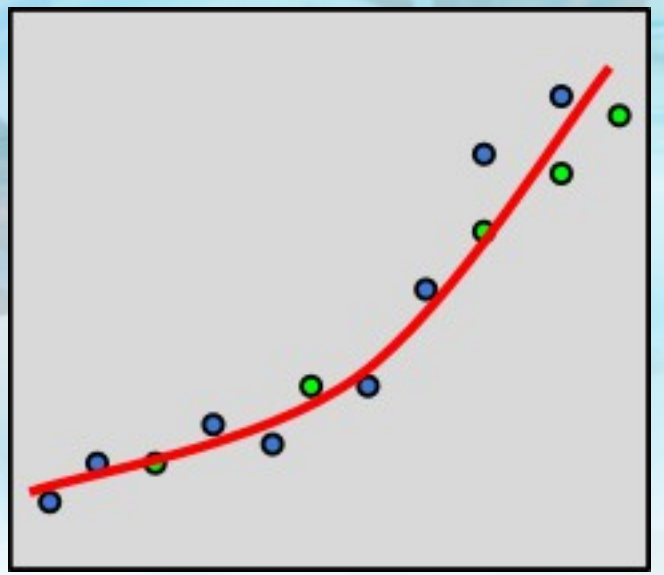
More complex

Smallest error on training data

Overfitting



Straight line

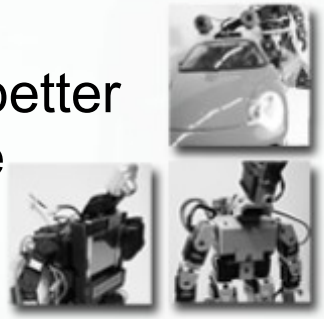


Parabola

More complex

Worst error on unknown data

Overfitting Problem: Performance on the training data keeps getting better
But, performance on unknown data (generalization ability) gets worse



Generalization as Search

- Generalization can be seen as search through the space of possible descriptions of what a good board position is
  - Some caveats
  - More than one description may survive
    - More than one plausible explanation of the data seen so far
  - No description may survive
    - No element of our hypothesis space matches the data
    - For example, if data is corrupt (noisy)

Generalization as Search

- Even simple domains lead to an explosion
- Example: Contact lens database
  - The domain has 5 Attributes with (3/2/2/2/2) values per attribute
  - So the number of possible preconditions for each rule is
    - 3 * 2 * 2 * 2 * 2 = 48 possible rules
    - That does not sound so bad
  - But usually, a single rule is not enough to describe the target function correctly
  - Find a set of 14 rules to describe the target function – suitable for contact lenses
    - 48 choose 14 = 4.8 * 10^11 possible rule sets
- Intractable

Bias

- Looking at ML as a search problem
- We have to make the following important decisions
  - Concept description language (representation) – this represents the space that we will search. Because of the size of the space, it is usually defined implicitly (e.g., rules to create new states from other states)
  - Order in which this space is searched
  - Method chosen to prevent overfitting to the data
- To make it tractable, we have to reduce the size of the search space
- We are making a selection. Technically this is called a bias. In ML we have to deal with three biases specific to the decisions above
  - Language bias
  - Search bias
  - Overfitting bias

Language Bias

- What type of concepts are allowed
- Conjunctions (And) only
- Disjunctions (Or) allowed
- Examples
  - Generalization hierarchies
    - Version spaces and conjunctive concept description language
  - Rule sets (Disjunctions)
  - Weights in neural nets

Search Bias

- Many different search algorithms
  - Greedy search
  - Beam search
  - ...
- Direction of search
  - General to specific
    - Specializing a rule by adding constraints
  - Specific to general
    - Dropping conditions from the antecedents of rules

Overfitting Bias

- Fits data to tightly because of random circumstances
- Seen as a form of search bias
- Modify evaluation criteria to include “simplicity of description”
- Occam's razor: prefer simpler hypothesis
- Example:
  - Pruning nodes from a decision tree
  - Remove overly specific rules from the rule base

Issues in Machine Learning

- What algorithms can approximate functions well and under what conditions?
- How does the number of training examples impact on the system's performance?
- How does the complexity of the hypothesis affect its correctness
- Robustness to noise
- How to include prior knowledge
- Biological inspirations
- Learning Bias