CS434
Machine Learning and Data Mining

Administrative Trivia

- Instructor:
  - Dr. Xiaoli Fern
  - web.engr.oregonstate.edu/~xfern
  - Office hour (tentative): Thur after class till 4:30
- TA:
  - Evgenia Chunikhina (chunikhe@oregonstate.edu)
  - Office hour: TBA
- Course webpage on canvas
  - Read the syllabus to find out how the class will be run!

Course materials

- No text book required, slides and reading materials will be provided on course webpage
- There are a few recommended books that are good references
  - Machine learning by Tom Mitchell (TM) – slightly out of date but good intro to some topics
  - Pattern recognition and machine learning by Chris Bishop (Bishop) – denser but more up to date material

What is learning?

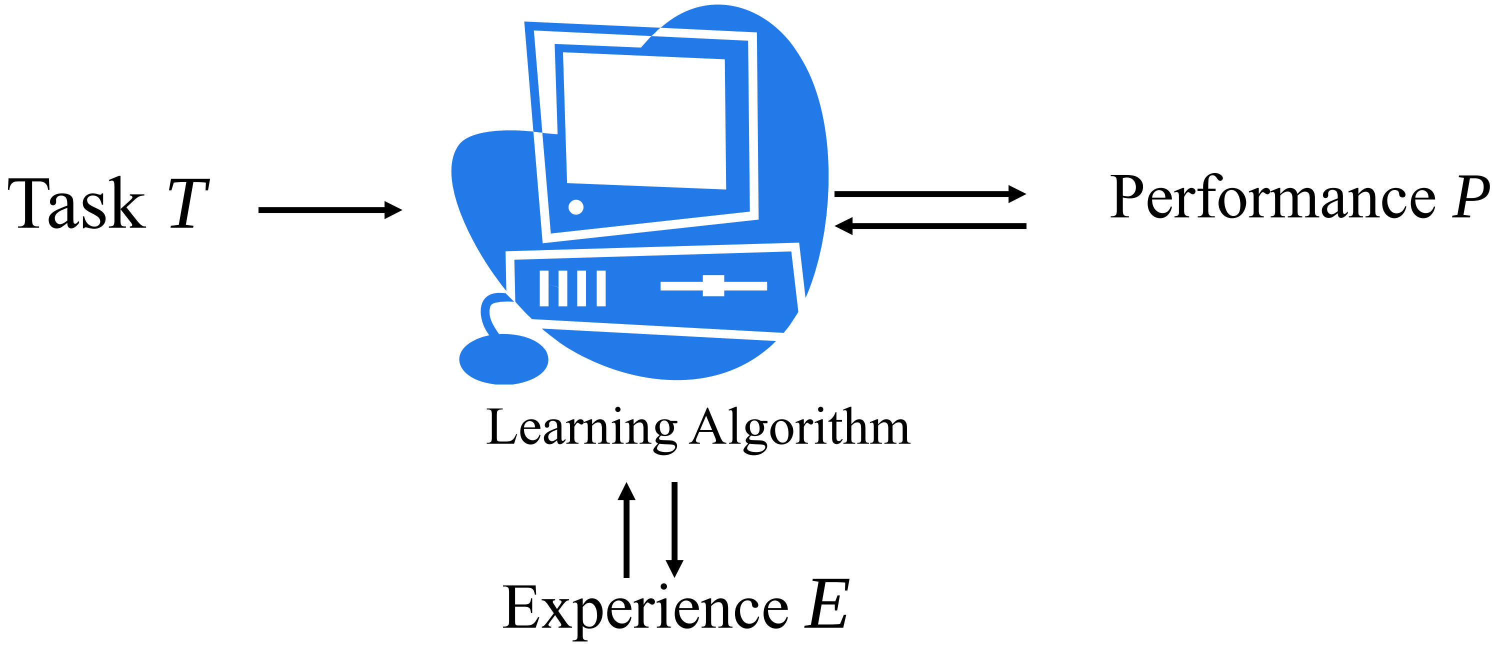
Generally speaking

“any change in a system that allows it to perform better the second time on repetition of the same task or on another task drawn from the same distribution”

--- Herbert Simon*

* One of the founding fathers of AI, Turing award winner

Machine learning



Task *T*

Performance *P*

Learning Algorithm

Experience *E*

Learning = Improving with experience at some task

- Improve over task *T*
- with respect to *P*
- based on experience *E*

When do we need computer to learn?

What is not learning?

- × A program that does tax return
- × A program that looks up phone numbers in phone directory
- × …

When do we need learning?

- Sometimes there is no human expert knowledge
  - Predict whether a new compound will be effective for treating some disease
  - Predict whether two profiles on match.com would be a good match (or does this belong the next category?)
- Sometimes humans can do it but can’t describe how they do it
  - Recognize visual objects
  - Speech recognition
  - Anything involving human perception
- Sometimes the things we need to learn change frequently
  - Stock market analysis, weather forecasting, computer network routing
- Sometimes the thing we need to learn needs customization
  - Spam filters, movie/product recommendation

Sub-fields of Interest

- Supervised learning – learn to predict (regression and classification)
- Unsupervised learning – learn to understand and describe the data (clustering, frequent pattern mining)
- Reinforcement learning – learn to act

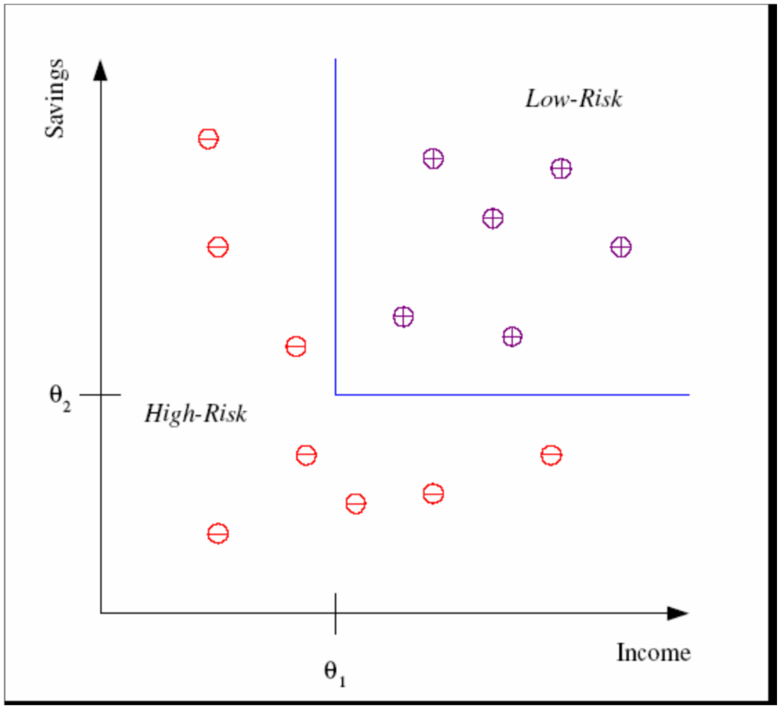
Data mining

A highly overlapping concept, but heavier focus on large volume of data:

To obtain useful knowledge from large volume of data

Supervised Learning: example

- Learn to predict output from input
  - Output can be continuous (regression) or discrete (classification)
  - E.g. predict the risk level (high vs.low) of a loan applicant based on income and savings



MANY successful applications!

Spam filters

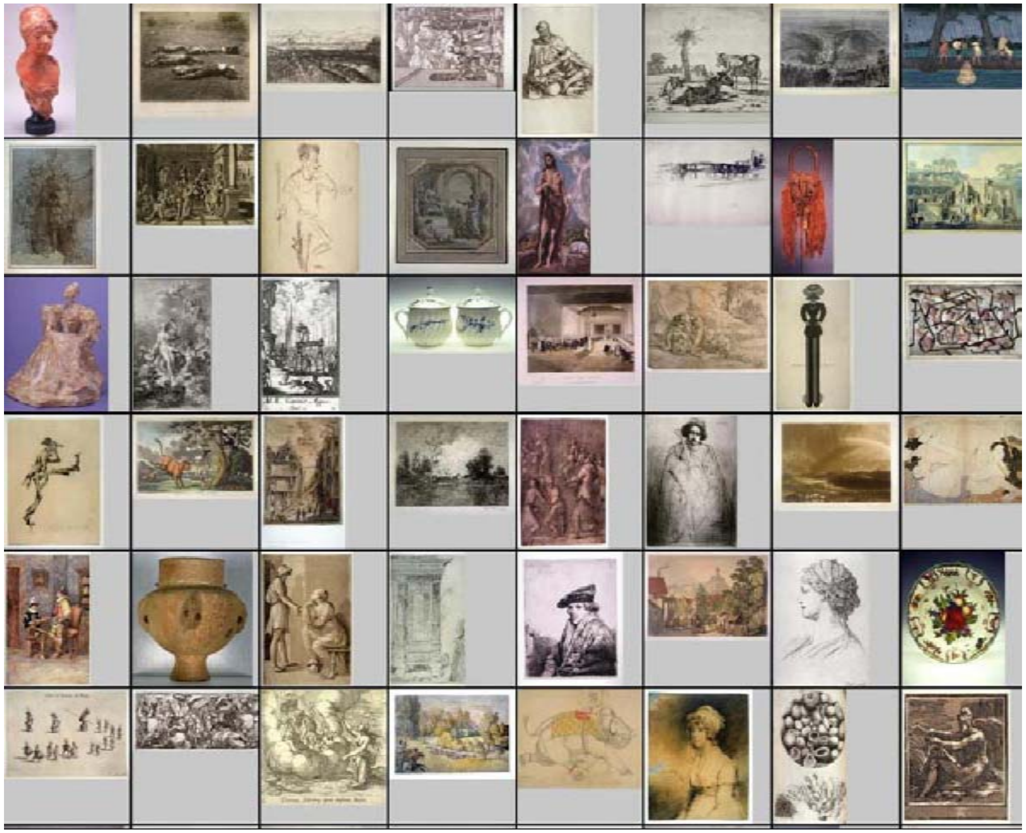
Collaborative filtering (predicting if a customer will be interested in an advertisement …)

Ecological (predicting if a species absence/presence in a certain environment …)

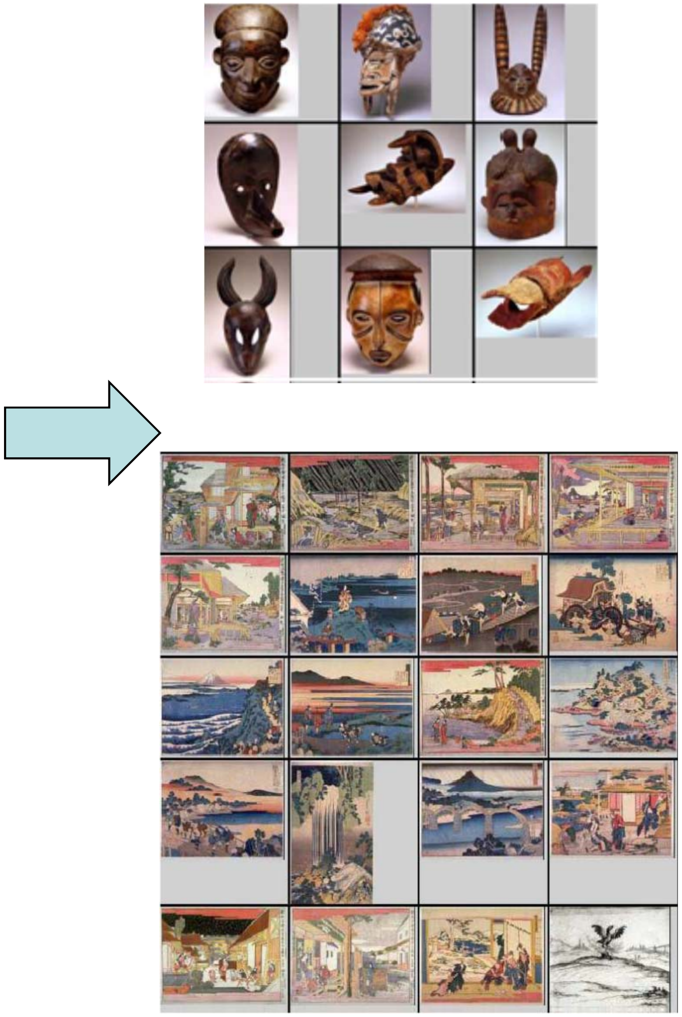
Medical diagnosis …

Unsupervised learning

- Find patterns and structure in data



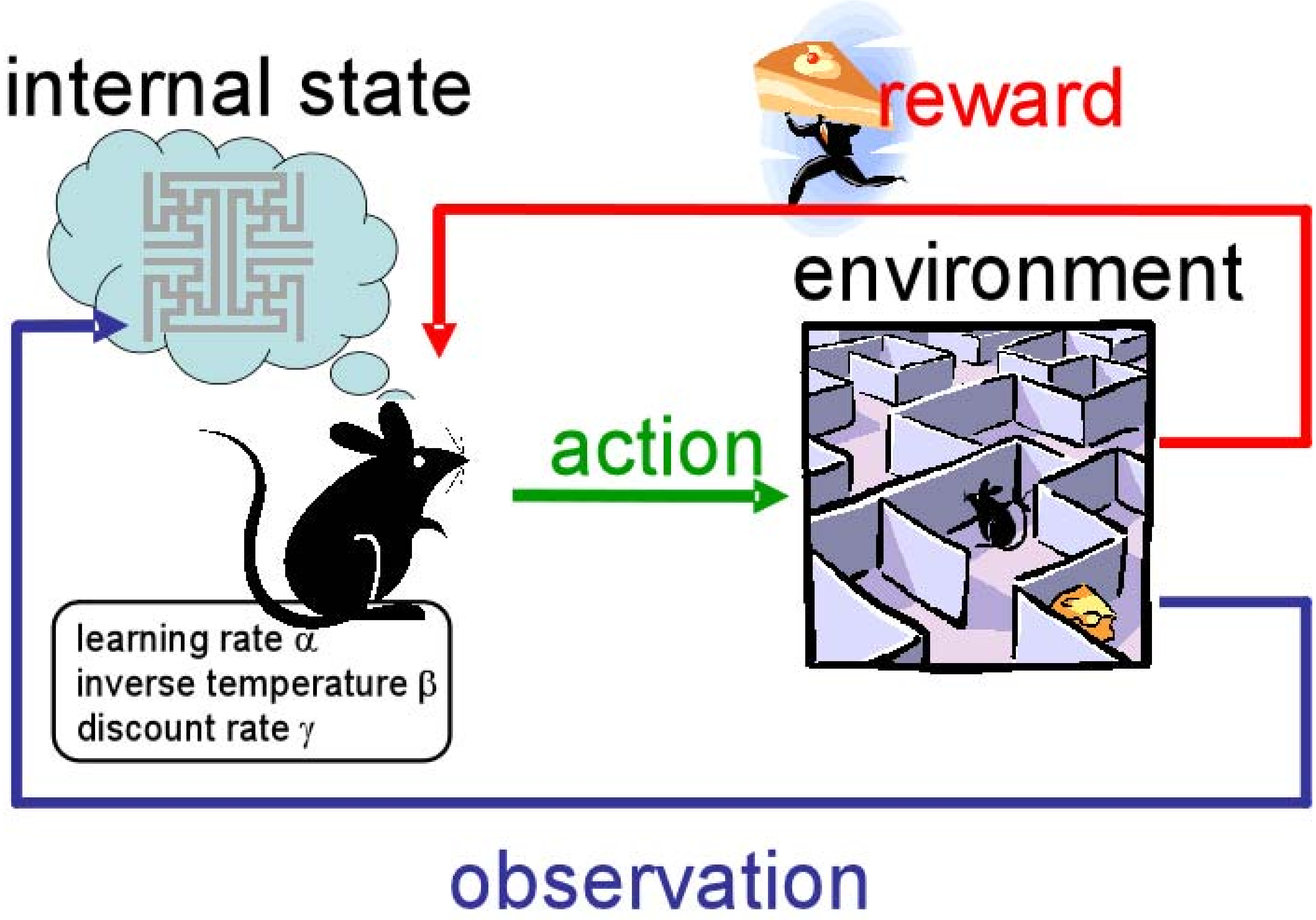
Clustering art



Example Applications

- Market partition: divide a market into distinct subsets of customers
  - Find clusters of similar customers, where each cluster may conceivably be selected as a market target to be reached with a distinct marketing strategy
- Automatic organization of information
  - Automatic organization of images
  - Generate a categorized view of a collection of documents
  - Organize search results to diversify results
- Scientific applications:
  - Bioinformatics: clustering the genes based on their expression profile to find clusters of similarly regulated genes – functional groups
  - Atmospheric science: clustering temporal signals (e.g., temperature, wind, pressure) for finding different weather regimes

Reinforcement learning



Example Applications

- Robotics
  - Gait control for robotic legs
  - Routing of the robot in a complex environment
- Controls
  - Helicopter control, automatous vehicle
- Operation research
  - Automatic pricing of internet advertisements
- AI game agents
  - Real time strategy game agent
  - GO, Chess ..

Course Learning Objectives

1. Students are able to ***apply supervised learning algorithms to prediction problems and evaluate the results.***
2. Students are able to ***apply unsupervised learning algorithms to data analysis problems and evaluate results.***
3. Students are able to ***apply reinforcement learning algorithms to control problem and evaluate results.***
4. Students are able to ***take a description*** of a new problem and ***decide what kind of problem*** (supervised, unsupervised, or reinforcement) it is.

Example: Learning to play checkers

- ***T**ask:* play checkers
- ***P**erformace: percent of games won in the world tournament
- To design a learning system for this task, we need to consider:
  - What experience to learn from? (the ***training data***)
  - What should we exactly learn? (the ***target function***)
  - How should we represent *this thing* that we are learning? (***Representation of the target function***)
  - What type of learning is it – supervised, unsupervised, or reinforcement learning, and what specific algorithm to use?

Type of training experience

- Direct training (like watching a master play)
  - Given an input, i.e., a given board state, we observe the desired output for that input (aka a good move for that position)
  - Observe many states and many moves (that will be our training data, in the form of input and output pairs)
  - Try to learn a formula of some sort that tells us what is the best move for any arbitrary state
  - This fits in ***supervised learning***
- Indirect training (like learning by playing)
  - Just observe or try out a sequence of plays and observe the end result (win or loss)
  - More difficult, because
    - which of the moves are the bad (good) ones for a bad (good) game?
    - This is the credit assignment problem, challenging to solve
  - This is more like **reinforcement learning**

Choose the Target Function (what should we learn)

- Choosemove: board state -> move?
  - Supervised learning
- V: Board state -> Reward (value of the state)?
  - Reinforcement learning
  - If you know the value of all possible states, at any state you can choose a move that leads to the best next state
  - This is more similar to how people understands the game

Possible definition for target function V

- If b is a final board state that won, V(b)=100
- If b is a final board state that is lost,   V(b)= -100
- If b is a final board state that is drawn, the V(b)=0
- If b is not a final board state, then V(b)=V(b'), where b' is the best possible final state reachable from b.

This gives correct values, but is not operational

A more practical approach is to compute a set of features describing the board state and the value of the board state is a function of these features

  - Features can be: # of black pieces, # of red pieces, # of black king pieces, ….

Choose representation for target function

- Linear function of the board features?

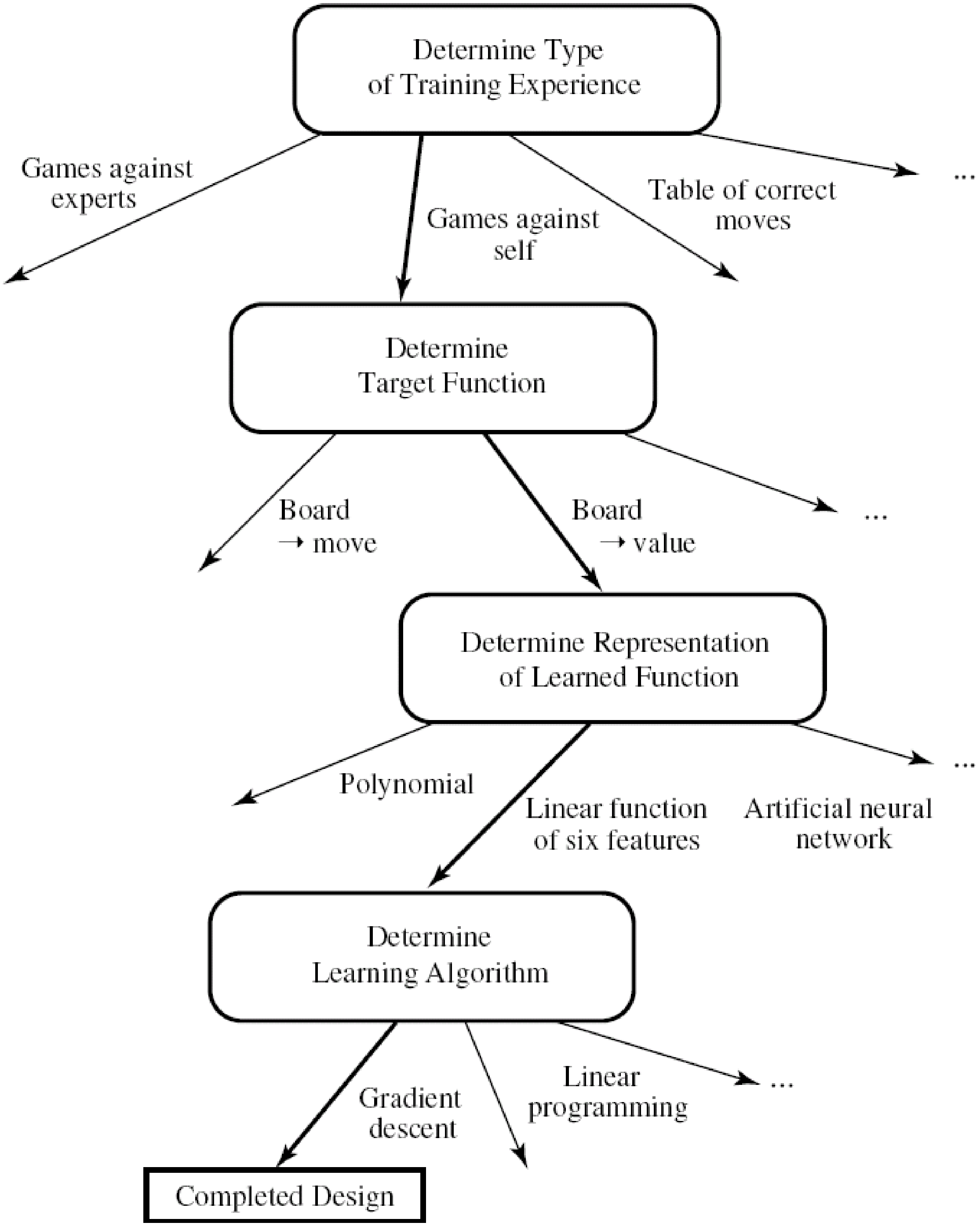
$$w_0 + w_1 f_1(b) + w_2 f_2(b) + \cdots + w_n f_n(b)$$

- Polynomial functions of board features?

$$w_0 + w_1 f_1(b) + w_2 f_2(b) + w_3 f_1^{\,2}(b) + w_4 f_2^{\,2}(b) + w_5 f_1(b)\, f_2(b) + \cdots$$

- …

A diagram of design choices


Determine Type of Training Experience
Games against experts
Games against self
Table of correct moves
...
Determine Target Function
Board → move
Board → value
...
Determine Representation of Learned Function
Polynomial
Linear function of six features
Artificial neural network
...
Determine Learning Algorithm
Gradient descent
Linear programming
...
Completed Design

In this class, you will become familiar with many of these choices, and even try them in practice.

We would like to prepare you so that you can make good design choices when facing a new learning problem!

Next few weeks: Supervised learning

- We will study a variety of algorithms
  - Linear Regression
  - Perceptron, Logistic regression
  - Naïve Bayes
  - Decision trees
  - Support Vector Machines
- Different algorithms consider different objectives
  - Minimizing different loss functions
  - Maximizing likelihood of the data assuming a probabilistic model
- Some key issues in supervised learning
  - How to avoid overfitting: regularization
  - How to select models: cross-validation