Administrative Trivia

• Instructor:
  – Dr. Xiaoli Fern
  – web.engr.oregonstate.edu/~xfern
  – Office hour: one hour before class or by appointment

• Course webpage
  
  classes.engr.oregonstate.edu/eecs/fall2009/cs434

• Please check course webpage frequently
  – Learning objectives
  – Syllabus
  – Course policy
  – Course related announcements
Briefly

• Grading (tentative):
  – Homeworks – 40%
  – Exam – 25%
  – Final project – 35%

• Homeworks
  – due at the beginning of the class (first 5 minutes of the class)
  – Late submission will be accepted if it’s no more than 24 hours late, but only gets 80%

• Collaborations policy (for solo assignments)
  – Verbal discussion about general approaches and strategies allowed
  – Can talk about examples not in the assignments
  – Anything you turn in has to be created by you and you alone
  – Violation
  For team assignments, the above policies apply between teams.
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)
  - Pattern recognition and machine learning by Chris Bishop (Bishop)
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
Machine learning

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
- Sometimes humans can do it but can’t describe how they do it
  - Recognize visual objects such as hand-written digits
- Sometimes the things we need to learn change frequently
  - Stock market, weather forecasting, computer network routing
- Sometimes the thing we need to learn needs customization
  - Spam filters
Sub-fields of Interest

• Supervised learning – learn to predict
• Unsupervised learning – learn to understand and describe the data
• Reinforcement learning – learn to act

Data mining

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

To obtain useful knowledge from large volume of data
Supervised Learning: example

• Learn to predict output from input
  – 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 is absent/present in a certain environment …)

Medical diagnosis …
Unsupervised learning

- Find patterns and structure in data
Example Applications

• Market Segmentation: divide a market into distinct subsets of customers
  – Collect different attributes of customers based on their geographical and lifestyle
  – Find clusters of similar customers, where each cluster may conceivably be selected as a market target to be reached with a distinct marketing strategy

• Document clustering
  – Generate a categorized view of a collection of documents
  – For organizing search results etc.

• Bioinformatics
  – Clustering the genes based on their expression profile
  – Find clusters of similarly regulated genes – functional groups
Reinforcement learning

internal state

reward

environment

action

observation

learning rate $\alpha$
inverse temperature $\beta$
discount rate $\gamma$
Example Applications

• Robot controls
• Elevator scheduling
• AI game agents
  – backgammon, chess, go …
Course Learning Objectives

• Students are able to apply supervised learning algorithms to prediction problems and evaluate the results.
• Students are able to apply unsupervised learning algorithms to data analysis problems and evaluate results.
• Students are able to apply reinforcement learning algorithms to control problem and evaluate results.
• 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

- **Task:** play checkers
- **Performance:** percent of games won in world tournament

To design a learning system for this task, we need to consider:

- What experience? (Training data)
- What should we exactly learn? (Target function)
- How should we represent *this thing* that we are learning? (Representation of the target function)
- What specific algorithm to use? (Learning algorithm)
Type of training experience

• Direct training (like watching a master play)
  – For a given board state, we observe a best move for that position
  – Observe many states and many moves (that will be our training data)
  – Try to learn a formula of some sort that tells us what is the best move for any arbitrary state

• Indirect training (like learning by playing)
  – Just observe a sequence of plays and the end result
  – More difficult, because
    • which of the moves are the bad (good) ones for a bad (good) game?
    • This is the credit assignment problem, very difficult to solve
Choose the Target Function (what should we learn)

- Choose move: board state -> move?
- V: Board state -> Reward (value of the state)?
  - 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?
  \[ w_0 + w_1 f_1(b) + w_2 f_2(b) + w_3 f_1^2(b) + w_4 f_2^2(b) + w_5 f_1(b)f_2(b) + \cdots \]
- Neural network?
- …
A diagram of design choices

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!