Course Overview

- Syllabus
- Goals
- Evaluation
- Deadlines

Course Objectives

- to introduce current machine learning and data mining methods
- to provide enough background to allow students to apply machine learning and data mining techniques to learning problems in a variety of application areas
- to provide those skills necessary to understand, critique, and possibly even extend existing research in machine learning

Evaluation

Homework (30%): Homework assignments (≈ 4) will consist of a combination of written problems and programming assignments. Your lowest score on the assignments will be dropped.

Paper critiques (10%): During the semester, you'll be asked to prepare one-page critiques of selected research papers. (≈ 6)

Final project and paper (25%): You will also complete an ML project of your choosing and write-up the results in a short paper (7 pages). Due during the last week of classes.

Class participation / Other (10%): You'll be expected to participate in class discussion or otherwise demonstrate an interest in the material studied in the course.

Exam (15%): There will be a final exam. (Monday, May 15?)

Machine Learning

- If we could make computers learn.....
  - Database mining: discover emerging trends in the spread of diseases
  - Self-customizing programs: on-line news service that learns interests of user
  - Applications we can't program by hand: autonomous driving, speech recognition
  - Understand human learning and teaching

Where we are today

- learn to recognize spoken words
- predict fraudulent use of credit cards
- drive autonomous vehicles on public highways
- play games such as Backgammon at human champion levels

Theoretical results: characterize the fundamental relationships among the number of training examples, observed the number of hypotheses under consideration and the expected error in the learned hypotheses

Slide CS478 – Introduction 5

Relevant Disciplines

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

Slide CS478 – Introduction 6

Tom Mitchell's definition of learning

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

Example: Learn to play checkers

T: play checkers (and win)

P: % of games won against opponents

E: play practice games against itself

Slide CS478 – Introduction 7

Learning to Play Checkers

T: Play checkers

P: Percent of games won in a world tournament

- What experience?
- What type of knowledge should be learned?
- How shall it be represented?
- What specific algorithm should be applied to learn it?

Slide CS478 – Introduction 8

Choosing a training experience

- Direct or indirect feedback provided to the performance system?
- Teacher? (does the learner have control of the sequence of training samples)
- Is the training experience representative of the performance goal?

Choosing the Target Function

Learn a function:

$$ChooseMove : Board \rightarrow Move$$

Learn an evaluation function:

$$V : Board \rightarrow \Re$$

Possible Definition for the 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 not a final board state, 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.

Choosing a Representation for the Target Function

- Table with one entry per board?
- Collection of rules?
- Polynomial function of board features?
- Neural network?

Our choice

$\hat{V}(b) = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + w_5x_5 + w_6x_6$, where

x_1: the number of black pieces on board b

x_2: the number of red pieces on board b

x_3: the number of black kings on board b

x_4: the number of red kings on board b

x_5: the number of black pieces threatened by red

x_6: the number of red pieces threatened by black

Obtaining Training Examples

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

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

$<< x_1 = 3, x_2 = 0, x_3 = 1, x_4 = 0, x_5 = 0, x_6 = 0 >, +100 >$

A rule for estimating training values:

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

Specifying the Learning Algorithm

- Need to find w_i that *best* fits the training examples.
- A definition of *best*: minimize the squared error E between the training values and the values predicted by the hypothesis $\hat{V}$:

$$\sum_{<b,V_{train}(b)> \in training\, examples} (V_{train}(b) - \hat{V}(b))^2$$

Specifying the Learning Algorithm

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 x_i, update weight w_i:

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

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

Would this program have beat the world champion?

Probably not.

Why?

- the linear function is too simple a representation to capture the nuances of the game
- but, given a more sophisticated target function, this approach is very successful.

Some Issues in ML

- What algorithms can approximate functions well (and in what situations)?
- How does the number of training examples influence accuracy?
- How does the complexity of hypothesis representation impact it?
- What effect does noise in the data have on performance?
- What are the theoretical limits of learnability?
- How can prior knowledge help?
- What clues can we use from knowledge of biological learning systems?
- How can systems alter their own representations?