B. Tech. Sem VII
Machine learning

Lecture 2
Designing A Learning System

Brijesh Bhatt

Outline

- Example Problem
- Traditional approach
- Learning Problem
- Design of Learning Problem

Checkers

- Checkers is a two player game
- Played on a 8x8 board (chess board)
- Each team plays with 12 pieces
- Pieces move only in diagonals (forward)
- Kill opponent piece by jumping over it in diagonal.
- The player who kills/removes all opponent pieces is the winner.

Traditional Approach

Search problem

- Define moves
- Define rules of the game
- Define a strategy to make best possible move

Checkers : A Learning Problem

RECALL : “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.”

T : ?

P : ?

E : ?

A checkers learning problem

- Task (T): playing checkers
- Performance measure (P): percent of games won against opponents
- Training experience (E): playing practice games against itself

Questions

How to write/develop an algorithm which can model the learning problem defined in the previous slide?

How to define procedures for T, P, E?

How (and when) does the algorithm terminate?

Does it always give correct output?

DESIGNING A LEARNING SYSTEM

- Choosing the training experience :
  - choose the training which provides best learning.
- Choosing the Target Function :
  - determine exactly what type of knowledge will be learned and how this will be used by the performance program
- Choosing representation for the target function
  - a represfentation that the learning program will use to describe the function that it will learn.
- Choosing a function approximation algorithm
  - A procedure to learn target function from the given data

Choosing the Training Experience (Direct or indirect feedback)

- The type of training experience available can have a significant impact on success or failure of the learner.
- provides direct or indirect feedback regarding the choices made by the performance system.
  - the system might learn from direct training examples consisting of individual checkers board states and the correct move for each
  - indirect information consisting of the move sequences and final outcomes of various games played

Choosing the Training Experience (Control of training example)

- The learner might rely on the teacher to select informative board states and to provide the correct move for each.
- The learner might itself propose board states that it finds particularly confusing and ask the teacher for the correct move.
- The learner may have complete control over both the board states and (indirect) training classifications, as it does when it learns by playing against itself with no teacher

Choosing the Training Experience (Quality of distribution)

- how well it represents the distribution of examples over which the final system performance P must be measured.
- In general, learning is most reliable when the training examples follow a distribution similar to that of future test examples
- In our checkers learning scenario, the performance metric P is the percent of games the system wins in the world tournament.
- If its training experience E consists only of games played against itself, there is an obvious danger that this training experience might not be fully representative of the distribution of situations over which it will later be tested. For example, the learner might never encounter certain crucial board states that are very likely to be played by the human checkers champion
- Training Experience: Games played against itself

Choosing the Target Function

- The next design choice is to determine exactly what type of knowledge will be learned and how this will be used by the performance program.
- Let us begin with a checkers-playing program that can generate the legal moves from any board state.
- The program needs only to learn how to choose the best move from among these legal moves.
- This learning task is representative of a large class of tasks for which the legal moves that define some large search space are known a priori, but for which the best search strategy is not known.
- Many optimization problems fall into this class.... Examples???

Choosing the Target Function

Learning function : problem of improving performance P at task T

ChooseMove1 : B → M, accepts as input any board from the set of legal board states B and produces as output some move from the set of legal moves M.

ChooseMove2 : B → R, maps any legal board state from the set B to some real value R. We intend for this target function V to assign higher scores to better board states.

we have reduced the learning task to the problem of discovering an operational description of the ideal target function

Choosing representation for the target function

The choice of representation involves a crucial tradeoff.

On one hand, we wish to pick a very expressive representation to allow representing as close an approximation as possible to the ideal target function.

On the other hand, the more expressive the representation, the more training data the program will require in order to choose among the alternative hypotheses it can represent.b

Define a good representation function!!

A representation function

- x1: the number of black pieces on the board
- x2: the number of white pieces on the board
- x3: the number of black pieces threatened by white pieces
- X4: the number of white pieces threatened by black pieces

$$f(B) = w1x1 + w2x2 + w3x3 + w4x4$$

Parameters = { w1, w2, we, w4 }

The Formulation

Task T: playing checkers Performance measure

P: percent of games won in the world tournament Training experience

E: games played against itself

Targetfunction: f : Board → R

Targetfunction representation : w1x1 + w2x2 + w3x3 + w4x4

Choosing the function approximation algorithm

The algorithm that learns function f by observing the training data E.