Machine Learning
MODULE-2
(Definition and Designing of Learning Systems)

By
Dr. N. Ramana
Associate Professor
Kakatiya University

Machine Learning

Class Discussion Topic-2

Textbook: Tom Mitchell, First Edition,McGraw-Hill, 1997.

Chapter : 1

Class Discussion Topic(CDT):

- ✓ Definition and Designing of Learning Systems

Lecture Level Learning Outcome

- Student will be able to understand and formulate well proposed problems
- Student will be able to analyse and implement Steps in Designing a Learning System

Well Proposed Problem

Definition:

- 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.

Well Proposed Problem

Example.1:

A computer program that learns to play checkers might improve its performance as measured by its ability to win at the class of tasks involving playing checkers games, through experience obtained by playing games against itself.

Well Proposed Problem

Example 2:

- Learning to drive an autonomous vehicle. Machine learning methods have been used to train computer-controlled vehicles to steer correctly when driving on a variety of road types.
- For example, the ALVINN system (Pomerleau 1989) has used its learned strategies to drive unassisted at 70 miles per hour for 90 miles on public highways among other cars. Similar techniques have possible applications in many sensor-based control problems.

Importance of Machine Learning

- Some tasks cannot be defined well, except by examples (e.g., recognizing people).
- Relationships and correlations can be hidden within large amounts of data. Machine Learning/Data Mining may be able to find these relationships.
- Human designers often produce machines that do not work as well as desired in the environments in which they are used
- The amount of knowledge available about certain tasks might be too large for explicit encoding by humans (e.g., medical diagnostic)
- Environments change over time
- New knowledge about tasks is constantly being discovered by humans. It may be difficult to continuously re-design systems “by hand”

Design a Learning System in Machine Learning Algebra

- Machine Learning enables a Machine to Automatically learn from Data, Improve performance from an Experience and predict things without explicitly programmed.”
- In Simple Words, When we fed the Training Data to Machine Learning Algorithm, this algorithm will produce a mathematical model and with the help of the mathematical model, the machine will make a prediction and take a decision without being explicitly programmed.
- Also, during training data, the more machine will work with it the more it will get experience and the more efficient result is produced

Steps in Designing a Learning System Algebra

1. Problem Description
2. Choosing the Training Experience
3. Choosing the Target Function
4. Choosing a Representation for the Target Function
5. Choosing a Function Approximation Algorithm
6. Final Design

1. Problem Description

A Checker Learning Problem

- **Task T:** Playing Checkers
- Performance Measure P: Percent of games won against opponents
- **Training Experience E:** To be selected ==> Games Played against itself

2. Choosing the Training Experience

- Direct versus Indirect Experience [Indirect Experience gives rise to the credit assignment problem and is thus more difficult]
- Teacher versus Learner Controlled Experience [the teacher might provide training examples; the learner might suggest interesting examples and ask the teacher for their outcome; or the learner can be completely on its own with no access to correct outcomes]
- How Representative is the Experience? [Is the training experience representative of the task the system will actually have to solve? It is best if it is, but such a situation cannot systematically be achieved]

3. Choosing the Target Function

Given a set of legal moves, we want to learn how to choose the best move (since the best move is not necessarily known, this is an *optimization* problem)

- *ChooseMove*: B --> M is called a Target Function

ChooseMove, however, is difficult to learn. An easier and related target function to learn is *V*: B --> R, which assigns a numerical score to each board. The better the board, the higher the score.

- Operational versus Non-Operational Description of a Target Function : An operational description must be given

- Function Approximation :The actual function can often not be learned and must be approximated

4. Choosing a Representation for the Target Function

- **Expressiveness versus Training set size:** The more expressive the representation of the target function, the closer to the "truth" we can get. However, the more expressive the representation, the more training examples are necessary to choose among the large number of "representable" possibilities.
- **Example of a representation:**
  - x1 = black pieces on the board x2 =red pieces on the board
  - x3 =black king on the board x4= red king on the board
  - x5= black pieces threatened by red pieces x6= red pieces threatened by black pieces

• V(b) = w0+w1.x1+w2.x2+w3.x3+w4.x4+w5.x5+w6.x6 (x=coefficient taken randomly)

5. Choosing a Function Approximation Algorithm

➢ Generating Training Examples of the form <b,Vtrain(b)>

- [e.g. <x1=3, x2=0, x3=1, x4=0, x5=0, x6=0, +100 (=blacks won)]
  - Useful and Easy Approach: Vtrain(b) <- V(Successor(b))

➢ Training the System

- Defining a criterion for success [What is the error that needs to be minimized?]
- Choose an algorithm capable of finding weights of a linear function that minimize that error [e.g. the Least Mean Square (LMS) training rule].

6. Final Design for Checkers Learning

- The Performance Module: Takes as input a new board and outputs a trace of the game it played against itself.
- The Critic: Takes as input the trace of a game and outputs a set of training examples of the target function
- The Generalizer: Takes as input training examples and outputs a *hypothesis* which estimates the target function. Good generalization to new cases is crucial.
- The Experiment Generator: Takes as input the current hypothesis (currently learned function) and outputs a new problem (an initial board state) for the performance system to explore

Issues in Machine Learning to analyze before choosing an machine learning algorithm

- What algorithms are available for learning a concept? How well do they perform?
- How much training data is sufficient to learn a concept with high confidence?
- When is it useful to use prior knowledge?
- Are some training examples more useful than others?
- What are best tasks for a system to learn?
- What is the best way for a system to represent its knowledge?

Summary

- 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.

Steps in Designing a Learning System:

1. Problem Description
2. Choosing the Training Experience
3. Choosing the Target Function
4. Choosing a Representation for the Target Function
5. Choosing a Function Approximation Algorithm
6. Final Design

Lecture Level Questions

1. What is the initial stage in creating a learning system?

The first stage is to comprehend the issue area and then choose the best algorithm for the job.

Lecture Level Questions

2. What role do data gathering and preparation play in the creation of a learning system?

Data gathering and preparation are critical components in ensuring that the learning system has access to pertinent and precise data in order to make accurate predictions.

Lecture Level Questions

3. What variables should be considered when choosing a program for a learning system?

The sort and nature of the data, accessible computational tools, and success measures are all factors to consider.

Lecture Level Questions

4. Why is it essential to evaluate the success of a learning system?

Evaluating success enables us to find areas where the model can be enhanced and assists us in making decisions.

References

1. Machine Learning: The Art and Science of Algorithms that make sense of Data, U.K.Cambridge University Press 2012 by Peter Flach.

2. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition, O'Reilly, 2019 by Aurélien Géron.

3., Introduction to Machine Learning with Python: A Guide for Data Scientists (Greyscale Indian Edition), 2016, by Andreas Muller

References

1. Machine Learning: What It is, Tutorial, Definition, Types – Javatpoint
2. Machine learning – Wikipedia
3. https://youtu.be/r4sgKrRL2Ys?si=-8SYorkau9r53u9S
4. https://youtu.be/ukzFI9rgwfU?si=i2s3r_HUWxwsa2pv
5. https://youtu.be/jGwO_UgTS7I?si=jGjlkTqmYSaWiKgV
6. https://youtu.be/Gv9_4yMHFhI?si=o7KYs6_FRZ8YXe3l
7. Machine Learning Full Course 2022 | Learn Machine Learning | Machine Learning Tutorial | Simplilearn (youtube.com)