Machine Learning: Introduction

Jayanta Mukhopadhyay
Dept. of Computer Science and Engg.

Two paradigms of computing

- Computing by algorithms
  - Traditional
  - Knowledge driven
    - Deductive
- Computing by learning
  - Non-traditional
  - Data driven
    - Inductive

Computing by algorithms

- An algorithm: a sequence of instructions to be carried out to transform the input to output.
  - Needed for solving a problem on a computer
    - Example: Sorting of a set of numbers.
      - Many algorithms may exist for the same task.
- For some problems no algorithm may exist!
  - To detect spam mails.

Computing by learning

- No knowledge of an algorithm, but many examples of input and output.
  - Two sets of examples
    - Positive examples: Spam mails.
    - Negative examples: Ordinary mails.
  - To ‘learn’ what characterizes a mail spam.
    - The task is to extract the algorithm from examples and apply on any arbitrary input to get the output.
    - Automatic learning by a machine

Machine learning: underlying assumptions

- Existence of a process generating the data.
  - Details are not known.
- Existence of certain patterns in the data.
  - Such patterns help in characterizing the process.
  - Not going to change in near future.
- Approximate construction of a model of the process possible from examples.

Machine learning: various contexts

- Data Mining: Application of machine learning on a large database.
  - A large volume of data processed to construct a simple model with valuable use.
    - Predicting consumer behavior.
    - Credit rating, fraud detection, etc. in financing.
    - Medical diagnosis.
    - Traffic analysis in telecommunication.

Machine learning: various contexts

- Intelligent System:
  - Adaptive to change ‘algorithm’ (learn) in a dynamic environment.
  - A part of AI
    - Robotics, Computer Vision, Speech Recognition.
    - Pattern Recognition, Biometry.
    - Autonomous driving car.

What is a Learning Problem?

- Learning involves performance improving
  - at some task *T*
  - with experience *E*
  - evaluated in terms of performance measure *P*
- Example: learn to play chess.
  - Task *T*: playing chess
  - Experience *E*: playing against a person
  - Performance *P*: percent of games won

Another Example

- Learn to recognise objects from a visual scene or an image
  - T: identify all objects
  - P: accuracy (e.g. a number of objects correctly recognized)
  - E: a database of objects recorded

What, how, when ….

- What is to be learnt?
  - How might this be represented?
    - A target function, a set of rules
- How it is to be learnt?
  - What specific algorithm to be used?
    - Applying experiences, instances, …
- When it is accomplished?
  - Performance evaluation
    - Whether the performance improved at a given task over time, without reprogramming?

Learning and computing

- Programming computers to optimize a performance criterion using example data or past experience.
  - Build a model described by parameters.
  - Execute a program to learn parameters.
    - Optimization algorithms on data.
  - Use model for performing tasks.
    - Prediction: Of output.
    - Description: Gain knowledge from data.

A few examples

Learning Association

- An example:
  - Finding associations between products bought by customers (basket analysis).
    - People buying X most likely would buy Y
      - X → Y
    - People buying X, probably would NOT buy Y.
      - X → ~Y

Classification and detection

- Examples:
    - ‘low risk’ vs. ‘high risk’ creditors.
        - Information about customers and credit history.
    - Multiple classes
        - Character recognition, Biometry, Face Recognition
    - Outlier detection

Web applications

- A lot of data for machine learning
  - Required various tasks to perform on them.
    - especially if the data is noisy or non-stationary.
- Spam filtering, fraud detection:
  - The enemy adapts so we must adapt too.
- Recommendation systems:
  - Lots of noisy data.
- Information retrieval:
  - Find documents or images with similar content.

Different types of learning

Supervised learning

- Learning with labeled data.
  - To learn a mapping from the input to an output
    - labels provided by a supervisor.
- Classification
    - Classify digits from hand written numerals.
- Regression
    - Predict the price of a car given a set of its attributes (brand, year, mileage, engine capacity, etc.).

Unsupervised learning

- Learning from only input data.
  - No labels of instances available.
  - no supervisor to provide mapping between input and output.
- The aim is to find the regularities / structures / patterns in the input.
  - Clustering

Reinforcement learning

- Learns a sequence of actions.
  - No emphasis on a single action
  - Emphasis on learning the policy
    - sequence of correct actions to reach the goal.
    - no such thing as the best action in any intermediate state.
    - an action is good if it is part of a good policy.
  - Examples:
    - Game playing, robot navigation.

Objectives of this course

- A brief exposure to the theory of computational learning
- Formulation and analysis of various learning problems
- Various approaches and methodologies to solve these problems.
- Performing a few case studies through implementation as programming assignments.

Syllabus

- Concept learning
- Decision Tree
- Introduction to probability
- Evaluation of hypotheses
- Bayesian learning
- Parametric method
- Dimension Reduction
- Instance based learning
- Unsupervised learning
- Linear discriminant functions
- Support Vector Machines
- Artificial Neural Networks
- Ensemble learning
- Reinforcement learning

Background required

- 1st level courses on
  - Linear Algebra
  - Probability Theory
  - Discrete Mathematics
  - Algorithms
- Strong in programming
  - Programming knowledge in Python a must.

Books

- “Machine learning” by Tom M. Mitchel
- “Introduction to Machine Learning” by Ethem Alpaydin.
- “Pattern Classification” by Duda, Hart and Stork.

Weekly classes

- Wed: 11 AM – 11:55 AM
- Thu: 12 Noon – 12:55 PM
- Fri: 8 AM – 8:55 AM

- All the classes to be held online through MS Team.
  - No separate email on video conference link
  - links to be made available in MST class page.

Evaluation

- Online quizzes : 40
- Assignments: 50
- Weekly summary of lectures (1 page per week): 10

All submissions and evaluation through CSE Moodle Server.

Slides and other resources to be uploaded in the CSE Moodle Server

Grading Policy

- Relative grading.
- The scoring intervals for grades to be decided as follows.
- Consider mean and s.d. of the marks as μ and σ.
  - F: $< \mu - 4\sigma$
  - P: $[\mu - 4\sigma, \mu - 3\sigma)$
  - D: $[\mu - 3\sigma, \mu - 2\sigma)$
  - C: $[\mu - 2\sigma, \mu - \sigma)$
  - B: $[\mu - \sigma, \mu + \sigma)$
  - A: $[\mu + \sigma, \mu + 2\sigma)$
  - EX: $\geq \mu + 2\sigma$

Bonus marks

- If all the assignments and weekly reports are submitted and a score of 80% obtained in both of these components, a bonus of 5 to be given after computing the thresholds of grades.

Rules of participation in online classes

- Keep your microphone muted and video turned off.
  - If asked turn them on.
- Raise your hand (through MS Team application).
  - We will take your question.
- Do not chat among yourselves.
  - Send your query to our TAs.
- Any violation may cause removal from the class team.
- Attendances to be recorded from the MST.
  - Missing classes without prior notice may cause deregistration.

Assignment submission

- By groups of two.
  - Pair by yourselves, and notify to TA coordinator by Friday (11th Aug.) 5 PM, otherwise we will do it randomly.
- All programming assignments to be in Python.
- No copy case (sharing and copying to be treated in the same manner)
  - Deduction of 10 marks from the total for each copy case.
- Acknowledge if you have taken help from someone or from other resources during submission.
  - Marks would be adjusted, but not penalized to -10.
- Strictly follow the deadline of submission.

Best wishes!