CS60020: Foundations of Algorithm Design and Machine Learning

Sourangshu Bhattacharya

Teaching Assistants

- Nikhil Kumar Singh
- Paramita Koley
- Soumi Das

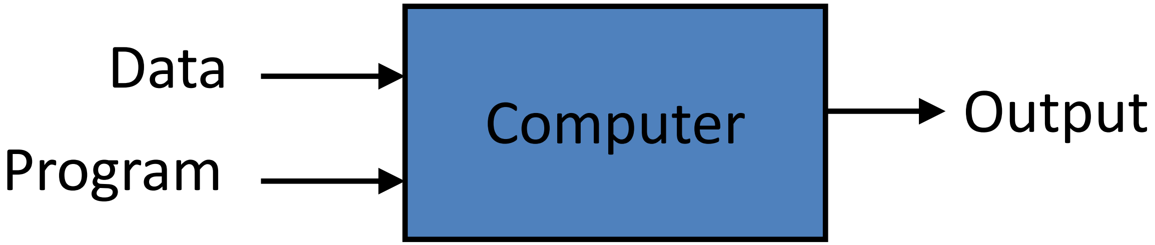
Algorithms

- An algorithm is an unambiguous specification of how to solve a class of problems.
- Example: Euclid’s algorithm for finding the greatest common divisor.
- Important Aspects:
  - Analysis
  - Design

Machine Learning

- Machine learning is a field of computer science that gives computers the ability to learn [from data] without being explicitly programmed.
- Example: Bayesian classifier for automatically filtering email spams.
- Aspects:
  - Modeling
  - Inference and learning

Traditional Programming



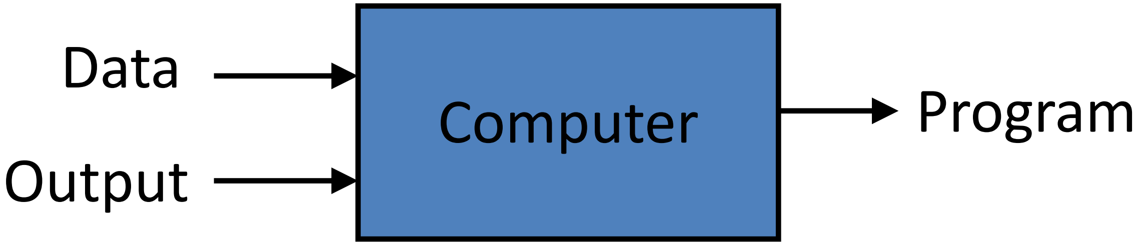
Data

Program

Computer

Output

Machine Learning



Data

Output

Computer

Program

Magic?

No, more like gardening

- **Seeds** = Algorithms
- **Nutrients** = Data
- **Gardener** = You
- **Plants** = Programs

Sample Applications

- Web search
- Computational biology
- Finance
- E-commerce
- Space exploration
- Robotics
- Information extraction
- Social networks
- Debugging
- [Your favorite area]

ORGANISATION

Venue

- Classroom: NC -231
- Slots:
  - Monday (12 - 12:55 pm)
  - Tuesday (10:00 - 11:55 am)
  - Thursday (8:00 - 8:55 am)
- Website:
  http://cse.iitkgp.ac.in/~sourangshu/coursefiles/cs60020_19S.html
- Moodle:
  https://10.5.18.110/moodle/

Pre-requisites

- Basic Programming and Data Structures.
- Mathematics: functions, matrix algebra, optimization.
- Discrete Maths: Graphs, Trees, etc.

- Logical thinking and Hard work !!

Evaluation

- Grades:
  - TA (Assignments and class tests): 40
    - Optional term projects.
  - Mid-sem: 30
  - End-sem: 30

- Assignments: 6 – 10.
- Class test: 2 – 4.

SYLLABUS

Algorithm Design

- Introduction to the design and analysis of efficient algorithms.
- Time and space complexity and order notation.
- Trees. Lists. Hashing.
- Graph and Basic graph algorithms: Definitions and representation, Reachability, Shortest path.
- Divide and Conquer and Dynamic programming
- NP-completeness- basic notions.
- Randomized algorithms and approximation algorithms.
- Sub-linear algorithms, streaming algorithms, sampling

Machine Learning

- Concept learning. Hypothesis space. Inductive Bias. Learnability. Underfitting and overfitting.
- Feature Selection, Dimension Reduction
- SVM and introduction to kernel methods.
- Unsupervised and semi-supervised learning. Expectation maximization. Mixture of Gaussians.
- Active learning, Learning with Imbalanced Data. Anomaly detection.
- Ensemble methods.
- Introduction to graphical models
- Introduction to Deep Learning

LECTURE SCHEDULE

Algorithm Design

- Introduction to the design and analysis of efficient algorithms.
- Time and space complexity and order notation.
- Trees. Lists. Hashing.
- Graph and Basic graph algorithms: Definitions and representation, Reachability, Shortest path.
- Divide and Conquer and Dynamic programming
- NP-completeness- basic notions.
- Randomized algorithms and approximation algorithms.
- Sub-linear algorithms, streaming algorithms, sampling

Machine Learning

- Concept learning. Hypothesis space. Inductive Bias. Learnability. Underfitting and overfitting.
- Feature Selection, Dimension Reduction
- SVM and introduction to kernel methods.
- Unsupervised and semi-supervised learning. Expectation maximization. Mixture of Gaussians.
- Active learning, Learning with Imbalanced Data. Anomaly detection.
- Ensemble methods.
- Introduction to graphical models
- Introduction to Deep Learning