Introduction - Logistics and Background

Govind Gopakumar

IIT Kanpur

Introduction

Rules and Administration

- Please keep your ID cards handy, only registered participants allowed
- Feel free to use the KD Ground floor lab
- Treat the department nicely and follow rules posted all around

Logistics

- Classes held 1900 - 2030 weekdays, KD101
- Instructor - Govind Gopakumar (Govind, no sir / bhaiyya)
- Contact - govindg@cse.iitk.ac.in
- Webpage - govg.github.io/acass

Course Outline

- 10 lectures * 1.5 hours each
- Online quizzes (After every 3 lectures)
- Projects for students with prior exposure (totally optional)
- Tentative curriculum has been put up on the website
- Open to suggestions about additional material / substitutes

Course goals and outcomes

- Understand basic tools in Machine Learning
- Recognize the need and suitability of models
- Be able to apply basic techniques and get reasonable results
- Have enough background to explore advanced methods, maybe research
- Read articles about the dangers of AI in newspapers and understand why they're (probably) wrong.

Origins of Machine Learning / AI

- Game playing bots - '52 A bot could play checkers
- Model of a neuron - '59 A toy model of a neuron was genrated
- Statistical models - '43 German Tank problem

All of these come together and now, we know them by the general terms Artificial Intelligence, Machine Learning, Data Science

Machine Learning : the present

Teaching Computers to think

"When we write programs that 'learn', it turns out that we do and they don't." - Alan Perils

- YouTube tells us what video to watch next
- Google gives us relevant search queries
- Amazon tells us what to buy next

All of these are examples of Machine learning in action!

Common element - Some form of data, expectation of some trend

- Machine Learning is modelling this trend using data

What constitutes trend and data?

- Genre of video, your past watch history, general trends in region
- Search input query, related terms
- Your recent purchases, demographic, history

How would you model this information? How would you make predictions?

Detailed system - autonomous drones

- Drone needs to fly on its own
- Avoid obstacles, birds of prey
- Automatically account for wind changes, speeds

Involved systems :

- Vision : Recognize sky, earth, birds, other flying objects
- Control : Strategy to avoid these things, maintain speed
- Path planning : How to get from point A to point B fast

All these involve some form or the other of Machine Learning!

Formal description of Machine Learning

- Data (Denoted by D, N)
- Model (Denoted generally by w)
- Training (Learning your model from data)
- Testing (Using your model on new data)

Questions that we need to ask :

1. How do you get this data? (Feature engineering)
2. How do you choose an appropriate model? (Model selection)
3. How do you train this efficiently? (Optimization and Learning)
4. How do you use this model? (Inference)
5. Why should this work? (Learning theory!)

Supervised Learning

Predict an outcome, a value, or a class. What do we have?

- Data, in form of numbers, words
- Attribute of data that needs to be predicted / learnt (label, value)

What is our goal?

- Learn a mapping from data -> attributes
- Figure out how this could have been generated

Supervised Learning : Examples

Classification

- Is this orange good or bad?
- Will Real Madrid win the Champions League or not?

Regression

- How much will this house sell for?
- How many points will Chelsea obtain next season?

Labelling

- Decide relevant tags for a Wikipedia article
- Decide relevant topics for a book

Unsupervised Learning

Find out patterns, modify the data automatically. What do we have?

- Data, in raw form

What is our goal?

- Learn some structure within the data
- Maybe as an intermediate step for later usage?

Unsupervised Learning : Examples

Clustering

- What styles of football teams exist?
- What kinds of fruit are available?

Density estimation

- What probability distribution does this data belong to?

Dimensionality reduction

- How do I extract meaningful data from a large set?
- How can we know what the important or "useful" subsets of data are?

The Mathematics

For ease of use, we'll always refer consider two phases :

- **Training** the model : Collect the data, initialize the model, and hopefully "learn" something meaningful
- **Testing** the model : Obtain the trained model, get new data in some form, make new predictions
- **Features** : These describe what data we have
- **Parameters** and hyperparameters : These describe our models
- **Loss** : Define how good / bad our model is

Training a model

What steps do we take to train a model?

- Clean up the data : Digestible by our model
- Set up some sort of measure of training (loss function, objective)
- Start training !

This is where words like “gradient descent” and “optimization” come in.

We need to find a right setting of parameters that do well by our measure.

Toy example - 2D Classification

Input Data : points in 2D space, with labels

Model selected : Straight line (unknown slope, offset)

Train time - Find proper splitting

Test time - Assign label to a new point

Machine Learning : The Future

Current applications

- Machine Translation in real time - Google Keyboard
- Autonomous Driving - Google, Uber
- Healthcare Monitoring - Clinical systems

Lecture takeaways

- What is Machine Learning? (General idea)
- What is the *data* in data science?
- How can we predict next year's football games?

Code

Policy

No code for this lecture, but further lectures will have associated code. Will be provided in the form of an IPython notebook, for ease of use and reproducibility.

References

Optional reading

- Lecture 1, CS771A - IIT Kanpur
- Review paper in Science Journal

Next class

Overview of mathematics - Linear Algebra, Probability

Next class Overview

Linear Algebra

What you must be comfortable with :

- Matrices
- Vectors
- Dot products and norms

Probability

- $P(a) = \frac{n(a)}{n(a)+n(a')}$
- Bayes Theorem
- Random variables and expectations

Functions - Shapes, Optima

Shape of a function

- Convex : Unique minima, easy to find the best point
- Non-convex : Weird structure, hard to optimize
- Smooth vs non-Smooth : Theoretical guarantees on how fast we can find a best point

Optima finding

- Gradient descent : Follow the slope
- For simple functions, we can derive the gradients easily by hand
- Iteratively find the optima using gradients!