Week 7: Tutorial introduction to AI and ML

M P Gururajan, Hina Gokhale and N N Viswanathan

Department of Metallurgical Engineering & Materials Science
Indian Institute of Technology Bombay

September, 2024

Outline

1. ML as hypothesis testing + optimization

2. Data preprocessing

ML as hypothesis testing + optimization

Mathematical view of ML

- First half of the course: probability and statistics
- Probability and statistics: a way of quantifying uncertainty and a way of putting numbers on our beliefs (recall Bayes' priors)
- Machine learning: a combination of statistics and optimization
- Machine learning: emphasis is on large scale data (and, hence python in this course)
- Machine learning: a subset of artificial intelligence

What is learning?

- Observe data; (how stones fall, how tides rise, how moon goes around the earth, ...)
- Identify the common features; (gravitational pull ...)
- Apply the understanding in a new scenario (can I shoot a rocket to the moon?)
- Let us look at a much simpler problem!

Chairs!

Figure: A large number of chairs (labelled) or a class of objects (unlabelled). From https://www.cabinfield.com/blog/different-types-of-dining-chairs/

Is this a chair? Or, does this belong to the same class?

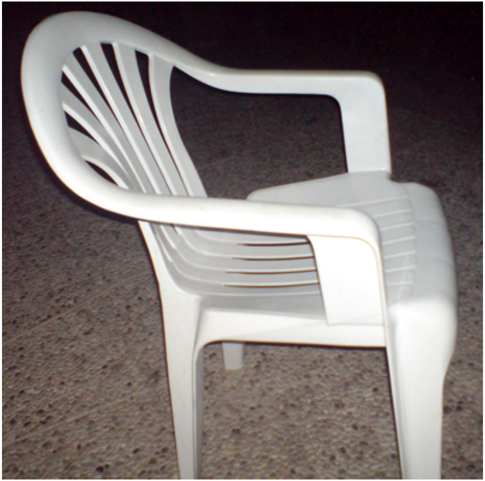


Figure: Image by Tjako van Schie - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=5159950.

Machine learning

- Can a computer, having observed lots of chairs, identify what constitutes a chair?
- When you see a new design of chair; you identify it as a chair
- Can we make machines do the same?
- Short answer: yes (may be qualified!)
- Long answer: rest of this course!!

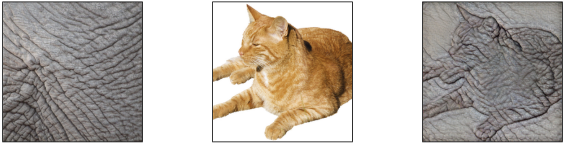
Types of learning

- Supervised learning, unsupervised learning, reinforcement learning
- Supervised: give both data (chair images) and target response (chair) for training
- Unsupervised: give data (chair images) and allow the algorithm to pick the pattern
- Reinforcement: unsupervised but with feedback (no; that is not a chair but is a stool; and, yes; this is a chair!)

Idea behind machine learning

- We can represent reality using a mathematical function
- Machine: gets the data but the mathematical function is unknown
- Can the machine identify the mathematical function based on the data?
- Example: can the machine label all the chairs in a given image?
- *Feature:* characteristics which help the machine label it appropriately! May be the height of a chair!
- Machine learning: supposing there exists a *target function* (of characteristics), can we get it or its approximation using data?
- Answer: Qualified yes!

Why qualified yes?



(a) Texture image

81.4%	**`Indian elephant`**
10.3%	`indri`
8.2%	`black swan`

(b) Content image

71.1%	**`tabby cat`**
17.3%	`grey fox`
3.3%	`Siamese cat`

(c) Texture-shape cue conflict

63.9%	**`Indian elephant`**
26.4%	`indri`
9.6%	`black swan`

Figure 1: Classification of a standard ResNet-50 of **(a)** a texture image (elephant skin: only texture cues); **(b)** a normal image of a cat (with both shape and texture cues), and **(c)** an image with a texture-shape cue conflict, generated by style transfer between the first two images.

Figure: Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness, R Geirhos et al, a conference paper at ICLR 2019.

See *Where We See Shapes, AI Sees Textures*: Jordana Cepelewicz in Quanta magazine for a popular exposition!

`https://www.quantamagazine.org/where-we-see-shapes-ai-sees-textures-20190701/`

Machine learning as hypothesis testing and optimization

- *Mapping:* Representation process of observing output and constructing the target function
- *Hypothesis space:* All potential functions that the learning algorithm can figure out
- Why hypothesis? Given the data, we are testing the hypothesis that the data is a result of the given target function!
- The result of ML: classifier and its parameters is called a *hypothesis*
- Which hypothesis we accept? Best in terms of mapping the features to classes during the training!
- Thus, ML is an optimization process

ML: some more ideas

- Note: hypothesis space should contain target function or its approximation
- *Hyper-parameters:* parameters that are not learnable from the model but which are needed for the ML algorithm
- ML algorithm: should be supplied with data and hyper-parameters
- Data for an ML algorithm: examples with features (chairs, and their shapes, sizes, colours, ...)
- Data: needs curation and processing before handing to ML algorithm

Data preprocessing

Data: importance

- Problems with ML: most of the times can be traced to noisy data and lack of data
- Data: foundation on which ML is built
- Bad data: irrespective of how sophisticated is your ML algorithm, you will get bad resuls
- GIGO: Garbage in - Garbage out!
- Curating and cleaning data: first step
- Spend time doing exploratory data analysis (EDA)
- Efforts and time doing EDA: totally worth it!
- Data: subject matter expertise!

How To: prepare data

- Obtain *meaningful* data
- Acquire *enough* data
- Arrange data in proper format
- Deal with missing data, redundancies, anamolies, skewed data, outliers, biased data, etc
- Create new features

Data: bias and size!

- How big should the data be to be enough?
- Errors due to bias and variance: need to know
- Data: Sampling from the true distribution
- Data sampling: random?
- Suppose the sample you choose is not random: data could be biased
- Survivorship bias: bullet holes in planes during world war II
- In general, however big the data set be, there are always biases!!

Sample

- Training of ML algorithm: in-sample
- Checking the ML model: requires out-of-sample
- One solution: split the given data as in- and out-of-samples
- Use in-sample for training and out-of-sample for testing
- Cross-validation: process of splitting the total data in many different ways for training and testing

ML: uncertainties

- Suppose we have clean, unbiased data
- Does not guarantee good ML model
- ML model is probabilistic: hence, there is always uncertainty

ML: issues of bias

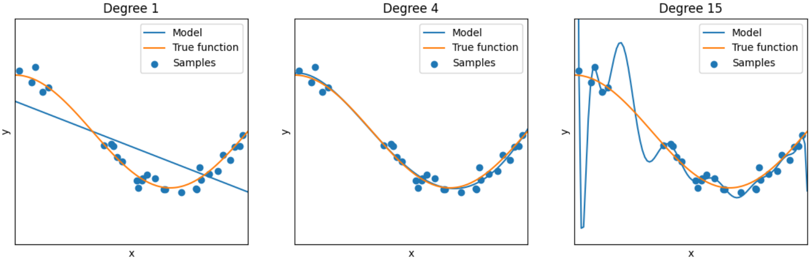


Figure: Underfitting, unbiased fit, and overfitting. Example from `scikit-learn` manual.

Origin of bias

- Bias: due to the fact that a complex mathematical target function is approximated by a simpler one!
- Variance: a simple mathematical target function is approximated by a complex one!
- Underfitting: complex target function is represented by a simple function – high bias scenario
- Overfitting: simple target function is represented by a complex function – high variance scenario

Bias-Variance: trade-off

ML models: optimization in the sense of simplicity (higher bias) versus complexity (higher variance) trade-off too!

An aside: `scikit-learn`

From https://scikit-learn.org/0.15/index.html

- `scikit-learn`: for Machine Learning in Python
- Simple and efficient tools for data mining and data analysis
- Built on NumPy, SciPy, and matplotlib
- Open source

Learning curves

- *Learning curve:* Performance of one or more ML algorithm with respect to quantity of data used for training
- Visualization of the degree to which the ML algorithm suffers from bias or variance with respect to the given data and data problem

Example learning curves

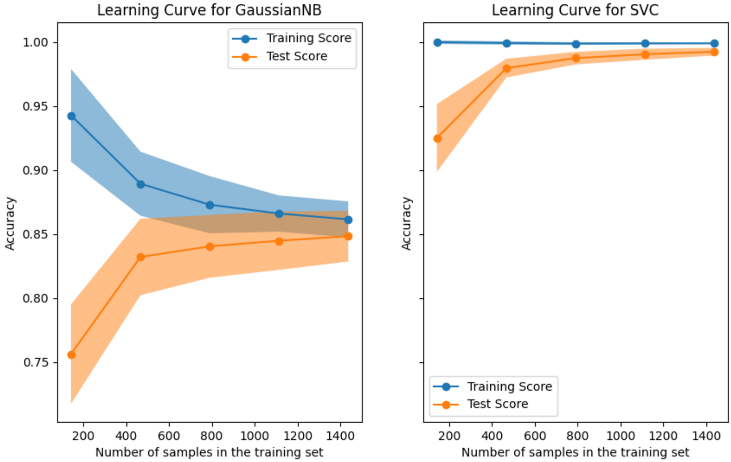


Figure: Learning curves example from `scikit-learn` manual.

Interpreting learning curves

- Data set: should be large enough so that training and testing of data sets gives you convergence in accuracy / error
- High error in converged values: bias in your ML model
- Note: high error in coverged values – can not be improved by throwing more data at the problem!
- Curves converge but test curve is not monotonic: high variance scenario

Cleaning data

- Choose appropriate data
- Is the data large enough? We need to use learning curve
- ML: iterative process
- Important skill: to interpret the results, to know when to take follow-up action and when to stop
- Follow-up action / stop decision: both based on knowledge of ML and data (subject matter expertise)
- Large enough data but unclean: needs cleaning
- What do we mean unclean?

Repairing data

- Missing data: common problem
- Repair missing data: how?
- Drop variable if data is missing – good idea only if the data is missing in large number of cases – say 90%
- Missing data is random: replace by average, median,...
- If not random, data repair is more involved – including collect the missing data

Replacement strategies

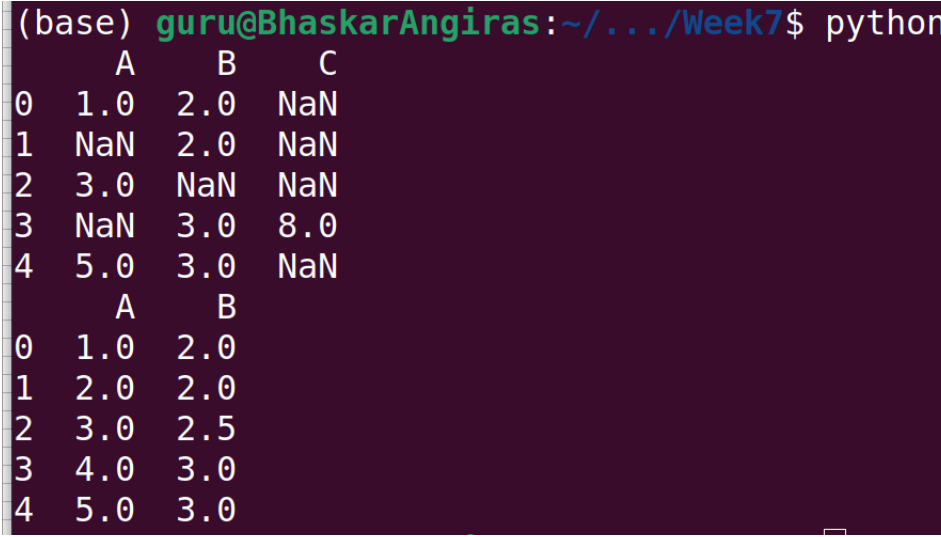
- Replace with a computed constant: mean, median, ...
- Replace with a value outside the feature range: works well for decision trees and qualitative variables
- Replace by zero: works well for regression models and standardised variables
- Interpolate: works well for quantitative values in a sequence
- Give values based on information from other predictor features

An example

```
import pandas as pd
Data = pd.read_csv("TestData.csv")
print(Data)
Data.drop(’C’,axis=1,inplace=True)
Data[’B’].fillna(Data[’B’].mean(),inplace=True)
Data[’A’].interpolate(method=’linear’,inplace=True)
print(Data)
```

From Machine learning in python and R for dummies, J P Mueller and L Massaron

Original and repaired data



```
(base) guru@BhaskarAngiras:~/.../Week7$ python
     A    B    C
0  1.0  2.0  NaN
1  NaN  2.0  NaN
2  3.0  NaN  NaN
3  NaN  3.0  8.0
4  5.0  3.0  NaN
     A    B
0  1.0  2.0
1  2.0  2.0
2  3.0  2.5
3  4.0  3.0
4  5.0  3.0
```

Data anomalies and their identification

- Outliers in data: called anomaly if we are certain that they are because of errors or mistakes (for example, age as a negative number!)
- Outliers: can cause ML algorithms misbehave
- First do Exploratory Data Analysis (EDA) and identify outliers
- They might need cleaning before running ML algorithm on them!
- Or, while interpreting the results, we have to pay attention to these outliers, anomalies / novelties!

Tranformations

- Useful to transform features and response variables: to make cost function minimize better the error of the predictions and for faster convergence
- Logarithm to response variable: reduces the weight to extreme errors
- If logarithm is not possible (needs positive values), take cubeth root that preserves the sign or invert the variable
- Z-score : you already know how to transform!
- Min-max: subtract minimum value and divide by the range

Feature creation

- Feature: data
- Feature creation: manipulating existing data to a form that is more amenable to ML
- For example, one way to overcome texture bias is to create images with random superimposed textures and use such data to train models!
- Feature creation: more art than science: use common sense, common knowledge and subject matter expertise!
- Automatic feature creation: multiplying, by creating powers, by creating polynomials etc
- Multiplication: noise and price of cars!

Compression of data

- Recall the notion of basis for a space
- Basis set: we want it to be orthonormal
- Too many correlated features: collinear and multilinear features
- Data: redundant – same information is spread across multiple features
- Unique variance: ideal – feature is correlated to response and hence canbe a predictor for the response
- Shared variance: not good – ML algoithm has difficulty in picking feature that corresponds to response
- Random noise: sometimes can be misleading
- All data: consist of these three components to varying degrees!

PCA

- Principal Comonent Analysis (PCA): a technique to create a new set of features (components) that are uncorrelated (principal) and ordered according to their informative component: think in terms of ordering the vector components!
- Compare: image compression using Fourier and wavelet transforms!

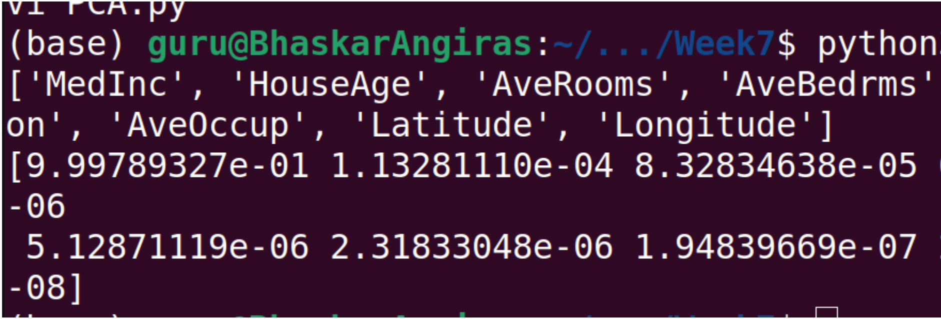
PCA example

```
from sklearn.datasets import fetch_california_housing
from sklearn.decomposition import PCA
from sklearn.preprocessing import scale
import numpy as np

cal = fetch_california_housing()
print(cal.feature_names[0:8])
X,y = cal.data, cal.target
pca = PCA().fit(X)
print(pca.explained_variance_ratio_)
```

Note: ethical issues with Boston housing data set (load_boston)

Output of PCA.py



```
vi PCA.py
(base) guru@BhaskarAngiras:~/.../Week7$ python
['MedInc', 'HouseAge', 'AveRooms', 'AveBedrms'
on', 'AveOccup', 'Latitude', 'Longitude']
[9.99789327e-01 1.13281110e-04 8.32834638e-05
-06
 5.12871119e-06 2.31833048e-06 1.94839669e-07
-08]
```

Figure: Output of PCA script. Note that a single component of the newly created data set can account for more than 99.99% of the variation in data.

What is the mathematics behind PCA? You'll learn in the theory lectures of this course!

Thank You!

Questions, clarifications, comments?