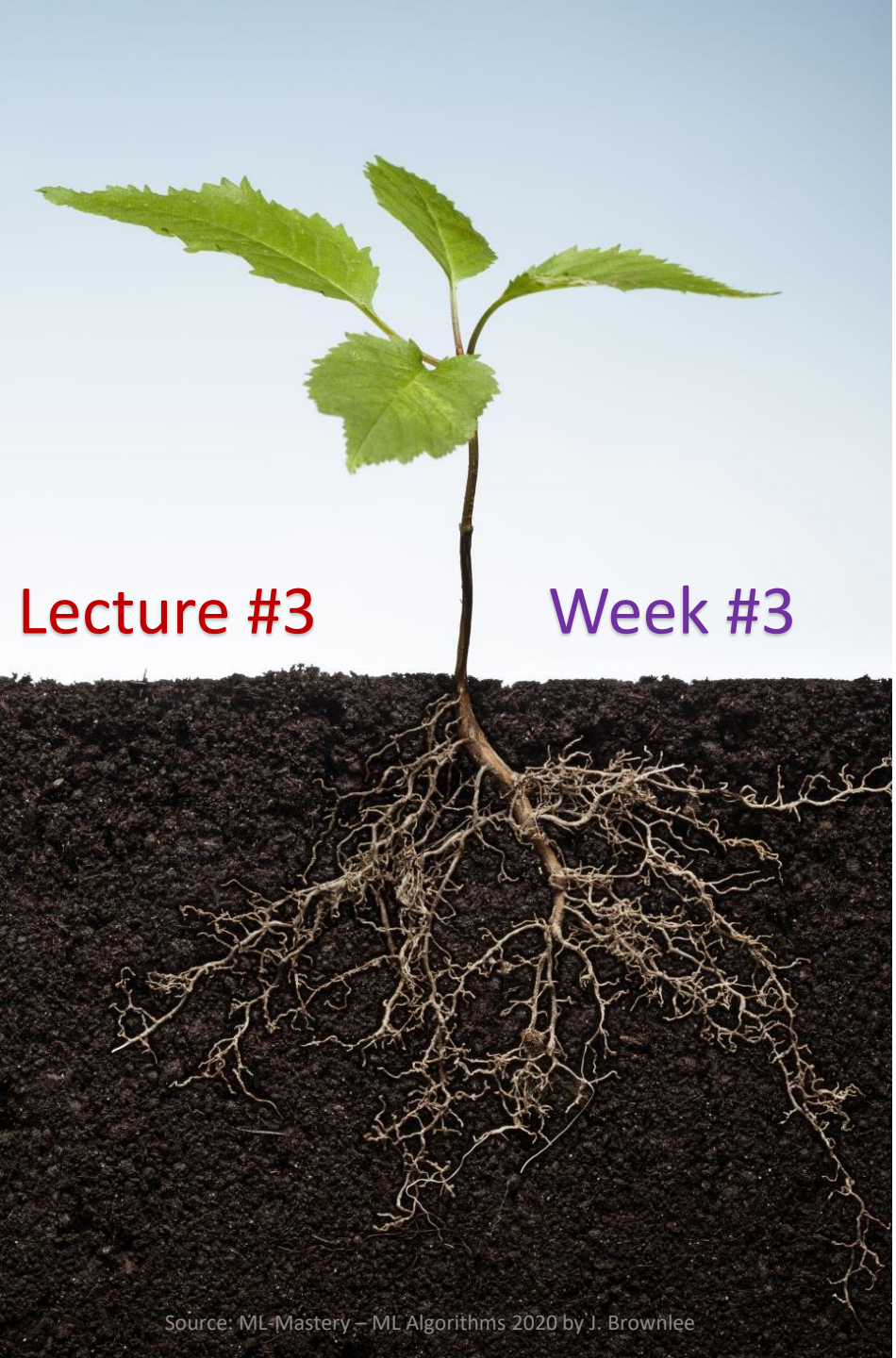


Pattern Recognition

ECSE 4410/6410 CAPA

Spring 2021

The Basics of Pattern Recognition / ML

Course Instructor - Thirimachos Bourlai

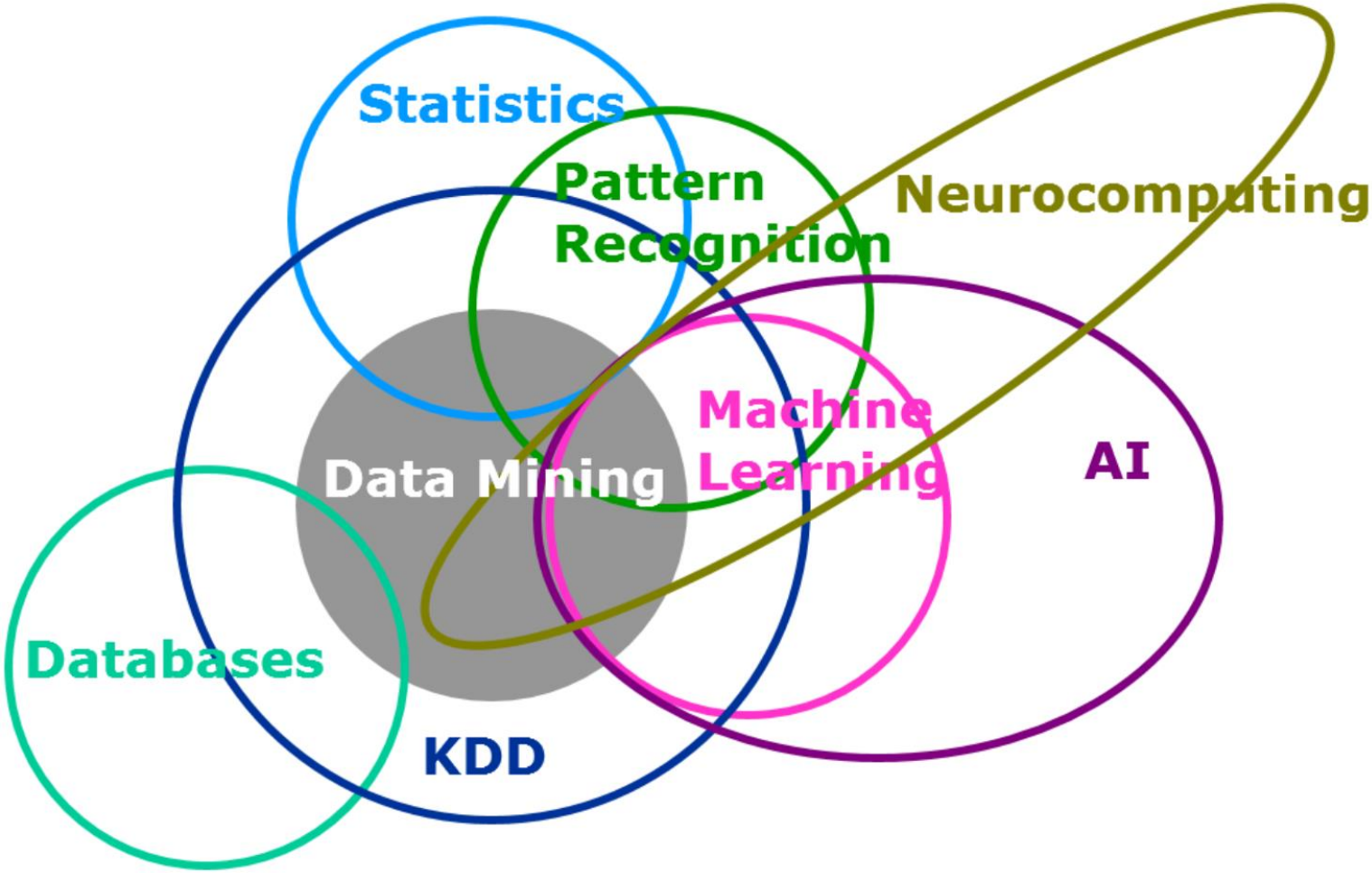
January to May 2021

Please Check the 2021 Syllabus

OVERVIEW

- How To Talk About Data in Machine Learning
- Data As you Know It
- Statistical Learning Perspective
- Computer Science Perspective
- Models and Algorithms
- Algorithms Learn a Mapping From Input to Output
- Parametric and Nonparametric Machine Learning Algorithms
- Supervised, Unsupervised and Semi-Supervised Learning
- The Bias-Variance Trade-Off
- Overfitting and Underfitting
- A Good Fit in Machine Learning

OVERVIEW



Venn diagram on the coverage of machine learning and statistical modeling in the universe of data science (Reference: SAS institute)

PART 1

DATA in Pattern Recognition

Introduction

- Data is very important in PR
- It is important to understand and use the right terminology
- In this lecture we will talk about data terminology

Learning outcomes: We will understand the differences below…

- Data terminology used in:
  - ❖ General / spreadsheets
  - ❖ Statistics
  - ❖ Computer/Data science and Engineering

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

PART 1

DATA in PR – Viewpoint #1

What we know – Tabular Data

	A	B	C	D	E	F
1		Column 1	Column 2	Column 3	Column 4	Column 5
2	Row 1	1	2	3	4	5
3	Row 2	3	4	5	6	7
4	Row 3	5	6	7	8	9
5	Row 4	7	8	9	10	11
6	Row 5	9	10	11	12	13

- Column: Data Type; Attribute; Feature
- Rows: Observations; Examples
- Cells: Single value (numeric; categorical)

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

DATA in PR – Viewpoint #2

PART 1

Statistics

	A	B	C	D
	x1	**x2**	**x3**	**Y**
1	1.1	2.23	4.1	1
2	1.2	2.21	4.3	1
3	1.1	2.01	4.1	0
4	1	2	4.7	0

Output = Function (Input)

- Function (f) is the hypothesis that the ML algorithm needs to learn.
- The question is: given some input (features, characteristics, variables, attributes), i.e. X (x1,x2,x3…), what is the predicted output (Y), and how close is this output to reality

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

DATA in PR – Viewpoint #2

PART 1

Statistics

	A	B	C	D
	x1	**x2**	**x3**	**Y**
1	1.1	2.23	4.1	1
2	1.2	2.21	4.3	1
3	1.1	2.01	4.1	0
4	1	2	4.7	0

Basic Concept

- Output = Function (Input)

Typically:

- Output_Variable = Function (Input Vector)

As we know in Statistics

- Output*DEPENDENT*_Variable =
  = Function (Input*INDEPENDENT*Variables)

Based on the framing of *the prediction problem*:
"the output is dependent", also called - "a function of" the input

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

DATA in PR – Viewpoint #2

PART 1

Statistics

	A	B	C	D
	x1	**x2**	**x3**	**Y**
1	1.1	2.23	4.1	1
2	1.2	2.21	4.3	1
3	1.1	2.01	4.1	0
4	1	2	4.7	0

To describe our problem, we need to have data

We need to have a set of algorithms and equations to describe and process the data

In statistics:

- INPUT VARIABLES : we describe them as capital x (X)
- OUTPUT VARIABLE : we describe them as capital y (Y)
- Equation used : Y = f(X)
- Multiple input variables : formed as an input vector, for example X1, X2 and X3 above.

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

DATA in PR – Viewpoint #3

Computer Science

	A	B	C	D
	x1	**x2**	**x3**	**Y**
1	1.1	2.23	4.1	1
2	1.2	2.21	4.3	1
3	1.1	2.01	4.1	0
4	1	2	4.7	0

- CS and Statistics terminology → overlap

PART 1

	A	B	C	D
	Attribute 1	**Attribute 2**	**Attribute 3**	**OUTPUT**
Observation 1	1.1	2.23	4.1	1
Observation 2	1.2	2.21	4.3	1
Observation 3	1.1	2.01	4.1	0
Observation 4	1	2	4.7	0

	A	B	C	D
	Fearure 1	**Fearure 2**	**Fearure 3**	**OUTPUT**
Instance 1	1.1	2.23	4.1	1
Instance 2	1.2	2.21	4.3	1
Instance 3	1.1	2.01	4.1	0
Instance 4	1	2	4.7	0

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

DATA in PR – Viewpoint #3

PART 1

Computer Science

	A	B	C	D
	Attribute 1	**Attribute 2**	**Attribute 3**	**OUTPUT**
Observation 1	1.1	2.23	4.1	1
Observation 2	1.2	2.21	4.3	1
Observation 3	1.1	2.01	4.1	0
Observation 4	1	2	4.7	0

	A	B	C	D
	Fearure 1	**Fearure 2**	**Fearure 3**	**OUTPUT**
Instance 1	1.1	2.23	4.1	1
Instance 2	1.2	2.21	4.3	1
Instance 3	1.1	2.01	4.1	0
Instance 4	1	2	4.7	0

Computer Science perspective:

$$Output\_Attribute = Program(Input\_Attributes)$$

- **Row**: **Observation** about an entity -- of different **Attributes**
- **Column**: each column → **all Observations** about each **Attribute**

OR

$$Output\_Feature = Program(Input\_Features)$$

- **Row**: Observation about an entity -- of different **Features**
- **Column**: each column → all Observations about each **Feature**

Attributes or Features → must be extracted from RAW DATA

DATA in PR – Viewpoint #3

Computer Science

	A	B	C	D
	Attribute 1	**Attribute 2**	**Attribute 3**	**OUTPUT**
Observation 1	1.1	2.23	4.1	1
Observation 2	1.2	2.21	4.3	1
Observation 3	1.1	2.01	4.1	0
Observation 4	1	2	4.7	0

	A	B	C	D
	Fearure 1	**Fearure 2**	**Fearure 3**	**OUTPUT**
Instance 1	1.1	2.23	4.1	1
Instance 2	1.2	2.21	4.3	1
Instance 3	1.1	2.01	4.1	0
Instance 4	1	2	4.7	0

PART 1

Computer Science perspective:

$$Prediction = Program(Instance)$$

- Input = a set of observations or instances (data rows) that is consider “an instance”
- Each data row is considered: “a single example” or “single instance” of data observed or generated by the data source

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

PART 1

Models and Algorithms

There is a confusion about the difference

- “A mathematical model is an abstract model that uses mathematical language to describe the behavior of a system.”
- “Eykhoff (1974) defined a mathematical model as 'a representation of the essential aspects of an existing system (or a system to be constructed) which presents knowledge of that system in usable form’ ”

Source HERE

PART 1

Models and Algorithms

In Computer Science – Engineering:

Model = the representation learned from data
Algorithm = the process for learning this representation

$$Model = Algorithm(Data)$$

EXAMPLE:

- Model = A decision tree
  - Algorithm = C5.0
- Models = a set of coefficients
  - Algorithm = the Least Squares Linear Regression

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

PART 1

DATA in Pattern Recognition

What have we learned thus far?

A. Tabular data are seen in a spreadsheet as _____

B. The statistical terms of input and output variables that may be denoted as ____ and ____ respectively.

C. The computer science terms of variable are names as:

_____ , _____ and _____ .

D. When talking of models, we mean the _____.

D. When talking of algorithms, we mean the _____.

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

PART 2

Algorithms: I/O Mapping

What are we going to learn

1. The mapping problem
2. What Predictive Modeling is
3. Different ML algorithms represent different strategies for learning the mapping function

PART 2

Algorithms: I/O Mapping

Learning a Function

Machine learning algorithms:

- Learn a target function (f) that maps X → Y

$$Y = f(X) \quad (1)$$

What is the "learning task"?

- To make predictions, i.e. future (Y) when new/unseen examples of input variables (X) are available

- The function (f) is unknown and that is why we need to learn it from data using these ML algorithms

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

PART 2

Algorithms: I/O Mapping

Learning a Function

- The mapping is not perfect → thus there is an error (e)

$$Y = f(X) + e \quad (2)$$

- **Error**:
  - Its independent of the input (X) data
  - It exists due to unavailability of enough features that can support the best mapping

PART 2

Algorithms: I/O Mapping

Making Predictions

Predictive modeling or predictive analytics

- Learn the mapping Y = f(X) to make predictions of Y for new X
- Goal: make the most accurate predictions possible

We do **not** care about:

- The shape/form of the function (f) we need to learn

We care about:

- Function (f) that needs to make the most accurate predictions

Statistical Inference (SI)

- **SI Goal**: if we care about learning the mapping of Y = f(X) to learn more about data relationships → use simple methods
- **In SI we care more about** "understanding the learned model and its form" rather that to make the MOST accurate predictions

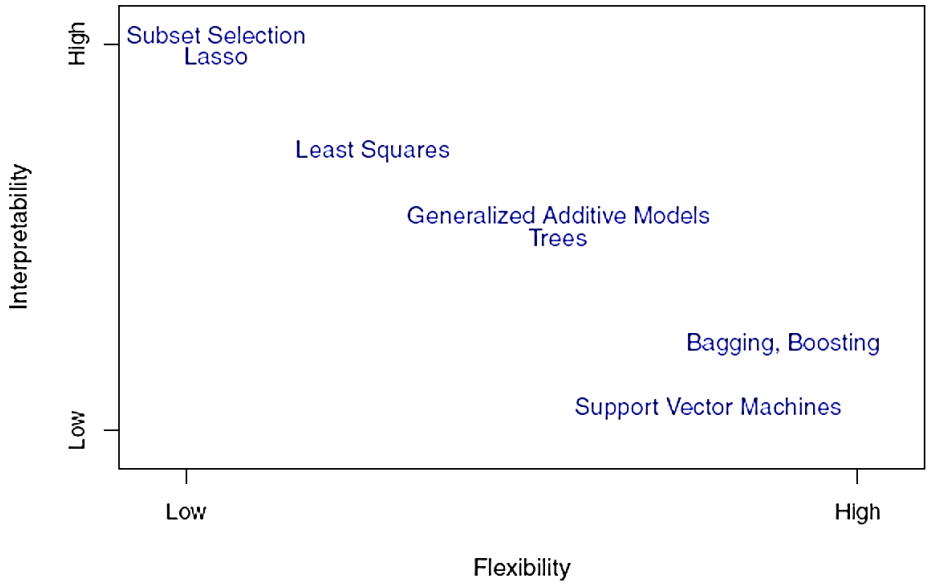
Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

Algorithms: I/O Mapping

Making Predictions

PART 2

- "The major **difference** between **machine learning** and **statistics** is their purpose.
- **Machine learning** models are designed to make the most accurate predictions possible.
- **Statistical** models are designed for **inference** about the relationships between variables."



Source: https://healthcare.ai/machine-learning-versus-statistics-use/

PART 2

Algorithms: I/O Mapping

Learning a Function

- ML = techniques for estimating the target function (f) to predict the output variable Y given X
- **Different models** → different assumptions about **f** (linear or nonlinear)
- **Different ML algorithms** → different assumptions about the function type needed to optimize the model/hypothesis to approximate it

- **What do we need to do**?
  - Try different ML algorithms
  - Study the problem to determine which algorithms have been used before to efficiently solve the problem under investigation

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

Algorithms: I/O Mapping

PART 2

What have we learned in Part Two ?

A. ML algorithms estimate the _____ of output variables (Y) given input variables (X)

B. Different ML algorithms make _____about the form of the underlying function.

C. When we don't know much about the form of the target function, we must try _____ to see what works best.

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

PART 3

Algorithms: Parametric vs. Non-Parametric

What are we going to learn

1. PML algorithms “simplify the mapping to a known functional form”
2. Non-PML algorithms → learn any I/O mapping
3. All ML algorithms are organized into PML and Non-PML categories

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

PART 3

Algorithms: Parametric vs. Non-Parametric

Parametric ML Algorithms

"A learning model that **summarizes** data with **a set of parameters** of fixed size (*independent of the number of training examples*) is called a parametric model. No matter how much data you throw at a parametric model, it won't change its mind about how many parameters it needs."

Artificial Intelligence: A Modern Approach, page 737

Parametric ML Algorithms:

- Step 1: Select a form for function (f)
- Step 2: Use the training data to learn the function's coefficients

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

PART 3

Algorithms: Parametric vs. Non-Parametric

Parametric ML Algorithms

Hypothesis: the functional form of a line is used to simplify the learning process of a problem

Basic mapping function (as we will see - used in linear regression) is a line:

$$a_0 + a_1 \cdot X_1 + a_2 \cdot X_2 = 0$$

- a0, a1 and a2: line coefficients (control the intercept and slope)
- X_1 and X_2 are two input variables

Need to do:
- Estimate a0, a1 and a2 to have a **predictive model** for the problem

PART 3

Algorithms: Parametric vs. Non-Parametric

Parametric ML Algorithms

The hypothesis is often assumed to have a functional form that is a linear combination of the input variables

- Thus, …… **Parametric** …… **machine learning algorithms**
  → **(called) Linear** ……**machine learning algorithms**

CHALLENGE:

- The actual underlying function that best maps the problem **may not be** a linear function
- It may **almost be** linear but require some minor transformation of the input data to best represent the problem (data)
- It may **NOT be** a line and thus, this hypothesis will result in poor system performance

Parametric vs. Non-Parametric

PART 3

Parametric ML Algorithms - **OVERVIEW**

Example Algorithms	Benefits	Limitations
• Logistic Regression • Linear Discriminant Analysis • Perceptron	• **Easy** to understand • **Fast** to learn • **Less Data** to Train • **Work well** even with an approxi-mate fit	• Constrained to a specific form • Not suited for complicated problems • Poor Fit to the underline function

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

Parametric vs. Non-Parametric

Non-Parametric ML Algorithms

PART 3

- **"Algorithms** that do **not** make strong assumptions about the form of the mapping function are called **Non-Parametric ML Algorithms"**

- "By **not** making assumptions, they are free to learn any functional form from the training data"

*"Nonparametric methods are good when you have **a lot of data** and **no prior knowledge**, and when you **don't want to worry** too much **about choosing just the right features**"*

— Artificial Intelligence: A Modern Approach, page 757

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

Parametric vs. Non-Parametric

Non-Parametric ML Algorithms

PART 3

- **"Algorithms** that do **not** make strong assumptions about the form of the mapping function are called **Non-Parametric ML Algorithms"**

- "By **not** making assumptions, they are free to learn any functional form from the training data"

*"Nonparametric methods are good when you have **a lot of data** and **no prior knowledge**, and when you **don't want to worry** too much **about choosing just the right features**"*

— Artificial Intelligence: A Modern Approach, page 757

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

Parametric vs. Non-Parametric

Non-Parametric ML Algorithms

- **Goal**: to **best fit the training data** in ***constructing*** *the mapping function*, while, at the same time they *try to* ***generalize*** *to unseen data*

- Non-Parametric ML Algorithm **can fit** many functional forms

- Example: k-NN algorithm

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

Parametric vs. Non-Parametric

PART 3

Non-Parametric ML Algorithms - **OVERVIEW**

Example Algorithms	Benefits	Limitations
• Decision Trees • Naive Bayes • SVMs • NNs	• Flexibility • No assumptions about the underlying function • Higher Performance prediction models	• Require more training data • More params → Slower to train • Overfitting training data is a high risk

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

PART 3

Parametric vs. Non-Parametric

What have we learned in Part Three ?

A. What is a **parametric** and a **non-parametric** function

B. Examples, benefits and limitations of **parametric** functions

C. Examples, benefits and limitations of **non-parametric** functions

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

Supervised, Unsupervised and Semi-Supervised Learning

PART 4

What we are going to learn

A. About the classification and regression **supervised** learning problems

B. About the clustering and association **unsupervised** learning problems

C. Example algorithms used for supervised and unsupervised problems

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

PART 4

Supervised Learning

$$Y = f(X)$$

SL GOAL:
Approximate the mapping function well → predict well the output data (Y) from the input data (X)

- SL algorithms are learning from the training data → supervise the training

SL algorithms are grouped into:
- **Classification**: When the output variable is a category
- **Regression**: when the output variable is a real value

SL examples:
- Linear regression → Regression
- Random forest → Classification/regression
- SVM → Classification problems

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

PART 4

Unsupervised Learning

USL:

- Only have input data (X) → no corresponding output
- There is no teacher to supervise anything
- The algorithms discover natural grouping of data on their own

USL algorithms are grouped into:

- **Clustering**: Discover the inherent data groupings
- **Association**: Discover rules that describe large portions of your data

USL examples:

- K-means → clustering problems
- Apriori algorithm → association problems

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

PART 4

Semi-Supervised Learning

Example: photo archive with some labeled images but all the rest are unlabeled

- **Labeling data**: Expensive or time consuming
- **Unlabeled data**: low cost; easy to collect; easy to store.

SSL: When we have a lot of input data (X) and some labeled data (Y)

- **STEP 1**
  - USL can be used → Discover the structure in the input variables and make good predictions for the unlabeled data
- **STEP 2**
  - SL can be the grouped data as an input (training) to a SL algorithm
  - Use the trained model by the SL to make predictions on new unseen data

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

SL, USL and Semi-SL

What have we learned in Part Four ?

A. _____: All data is (labeled/unlabeled)?; Algorithms learn the input's data **inner structure**

B. _____: All data is (labeled/unlabeled)? and the algorithms learn to **predict the output** from the input data

C. _____: Some data is (labeled/unlabeled)? but most of it is (labeled/unlabeled)? and a mixture of SL and USL can be used.

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

PART 5

Bias - Variance

What have we going to learn?

A. Learning error → is broken down into **bias** or **variance error**

B. Bias → simplifying algorithm assumptions so that the problem can be solved easier

C. Variance → model sensitivity to training data changes

D. Model prediction (applied ML) is better understood via the **bias** vs. **variance trade-off**

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

Prediction Error

- For any ML algorithm→ Prediction Error depends on:
  - Bias Error
  - Variance Error
  - Irreducible Error

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

Bias Error

PART 5

Bias → simplifying algorithm assumptions so that the hypothesis/target function can be solved easier

- **Low Bias**
  - **Less** assumptions about the form of the hypothesis function
- **High-Bias**
  - **More** assumptions about the form of the hypothesis function

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

PART 5

Bias Error

High-bias ML algorithms:

- Linear Regression
- Linear Discriminant Analysis
- Logistic Regression

Linear algorithms

- **High bias**
- Fast to learn
- Easier to understand but generally less flexible.

BUT:

- They have **lower predictive** performance (complex problems)
- Do not meet the simplifying assumptions of the algorithmic bias.

Low-bias ML algorithms:

- Decision Trees
- k-Nearest Neighbors
- Support Vector Machines

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

PART 5

Variance Error

Variance → model sensitivity to training data changes

- Training data is used to estimate the hypothesis/target function
- To do that we need a ML algorithm
- ML algorithms are expected to have some variance

- **IDEAL SCENARIO** - When changing datasets, the ML algorithm used will be able to pick out the hidden underlying I/O variable mapping
- **REAL WORLD** - When changing datasets, the ML algorithm picks out the hidden underlying I/O variable mapping with an error that depends on many factors

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

PART 5

Variance Error

- **Low Variance**: When changing the training dataset, we can also have **small changes** made, *when estimating the hypothesis/target function*
- **High Variance**: Suggests **large changes**…

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

PART 5

Variance Error

High Variance (Low Bias) ML algorithms:

- Decision Trees
- k-Nearest Neighbors
- Support Vector Machines

Nonlinear ML algorithms

- High Variance
- A lot of flexibility

For example:

- **Decision trees** have a **high variance → large changes**
- **Pruning → lowers down variance** before they decision trees are used

Low Variance (High Bias) ML algorithms:

- Linear Regression
- Linear Discriminant Analysis
- Logistic Regression

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

PART 5

Bias vs. Variance TRADE-OFF

What are the GOAL of supervised ML algorithms?

- Achieve "LOW Bias" and "LOW Variance"
- Achieve **Good Prediction Performance**
- Perform **Algorithm Parameterization** to ***balance out*** *bias and variance (**increasing** one → results in **decreasing** the other and **vice versa – NO WAY-OUT RELATIONSHIP**)*

REMEMBER:

- **Linear** ML algorithms
  - High bias
  - How variance
- **Nonlinear** ML algorithms
  - Low bias
  - High variance

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

PART 5

Bias vs. Variance TRADE-OFF

Examples with specific algorithms:

k-NN:

- **Low** bias (less assumptions); **High** variance (large changes to HTH when changing training data)
- How to change the TRADE-OFF?
  - Increase k → increases the # of neighbors that contribute to the prediction → increases model bias

SVMs:

- **Low** bias and **High** variance
- How to change the TRADE-OFF?
  - Increasing the C parameter (impacts the # of margin violations allowed in the training data) → **increases** the **bias** and **decreases** the **variance**

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

PART 5

Bias vs. Variance Reality

The algorithms we choose and their configuration (e.g. kNN with large k) → impact the trade-off

In practice:

- Calculating the **real** bias and **real** variance **error** is not possible → the actual underlying target function is unknown
- However, the bias / variance framework can help us understand the behavior of ML algorithms when aiming to determine the highest predictive performance.

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

BIAS - VARIANCE

What have we learned in Part Five ?

A. "The simplifying assumptions made by the model to make the target function easier to approximate" is called_____

B. "The amount that the estimate of the target function will change when changing training data" is called_____

C. The B and V Trade-off relationship can be avoided → YES / NO?

PART 6

Overfitting and Underfitting

What have we going to learn?

A. Overfitting: **learning** the training data **too well** and thus, not being able to perform well using new data

B. Underfitting: **insufficiently learning** how to solve the problem from the training data

C. Overfitting is in practice the **most common problem** in practice and the ways to address it.

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

PART 6

Learn about Generalization

- **Inductive learning**: using the training data to learn the target function.
- **Induction**: learning general concepts from specific examples (supervised learning)

- [**Deduction**: learning specific concepts from general rules]

- **Generalization**: how well the concepts learned by a model apply to specific examples not seen by the model when it was learning.

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

PART 6

Overfitting and Underfitting

Machine learning model GOAL: generalize well, i.e. the model built from the training data should also work with any data from the problem domain"

- This allows us to make predictions in the future on data the model has never seen.

- **Overfitting and underfitting**: terminology used in ML when we talk about how well a model learns and generalizes to new data.

- **Overfitting and underfitting**: main causes for poor performance of ML algorithms.

EXTRA READING: https://towardsdatascience.com/what-are-overfitting-and-underfitting-in-machine-learning-a96b30864690

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

PART 6

Statistical Fit

- “Goodness of fit test is a statistical hypothesis test to see how well sample data fit a distribution from a population with a normal distribution”

- “The test shows if your sample data represents the data you would expect to find in the actual population or if it is somehow skewed”

https://www.investopedia.com/terms/g/goodness-of-fit.asp

PART 6

Statistical Fit

- "Goodness-of-fit **establishes the discrepancy between** the **observed** values and those that would be **expected** of the model *in a normal distribution case*"

Methods: "There are multiple methods for determining goodness-of-fit. Some of the **most popular methods** used in statistics include the chi-square, the Kolmogorov-Smirnov test, the Anderson-Darling test and the Shipiro-Wilk test"

https://www.investopedia.com/terms/g/goodness-of-fit.asp

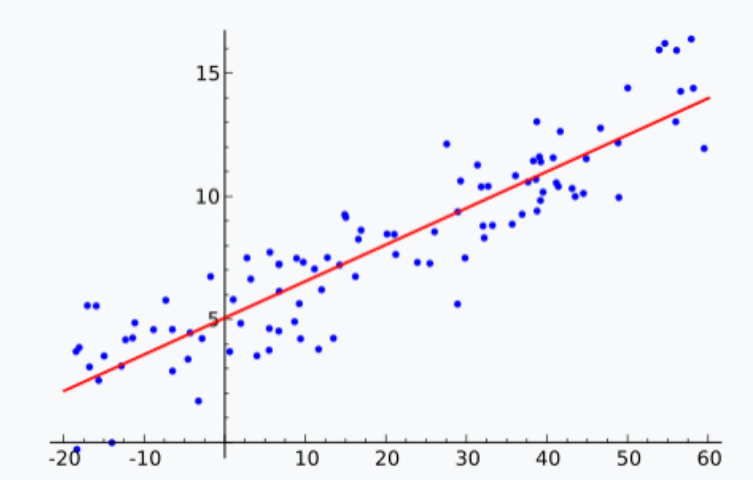
PART 6

Statistical Fit – ML Language

- A statistical fit (SFit) refers to the **measures used** to estimate **how well a target function is approximated**

Machine Learning:

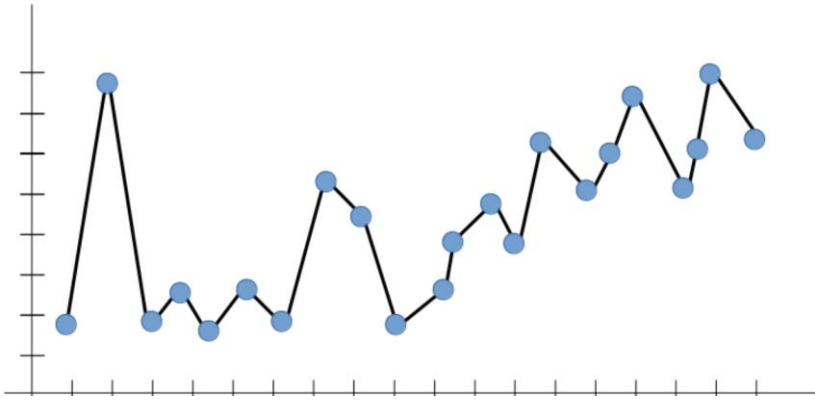
- SL algorithms "seek to approximate the unknown underlying mapping function for the output variables given the input variables"
- While *SFit methods are useful in ML* (e.g. calculating the residual errors), **some of them assume we know the form of the target function** we are approximating, which is **not** the case in ML.



Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

PART 6

Overfitting



- Overfitting: “a modeling error that occurs when a function is too closely fit to a limited set of data points. Overfitting the model generally takes the form of making an overly complex model to explain idiosyncrasies in the data under study.”

- **In Practice**: the data is often **impacted** with **some degree of error or random noise** within it → when we use a model that conforms too closely to slightly inaccurate (training) data can **impact the model** with substantial errors and reduce its predictive power (performance) on new data and negatively impact the model’s ability to **generalize**.

https://www.investopedia.com/terms/o/overfitting.asp

PART 6

Overfitting – Where?

- More likely with **nonparametric** and **nonlinear** models that have more flexibility when learning a target function
- Many such nonlinear algorithms include parameters or techniques to **limit and constrain** how much detail the model learns.

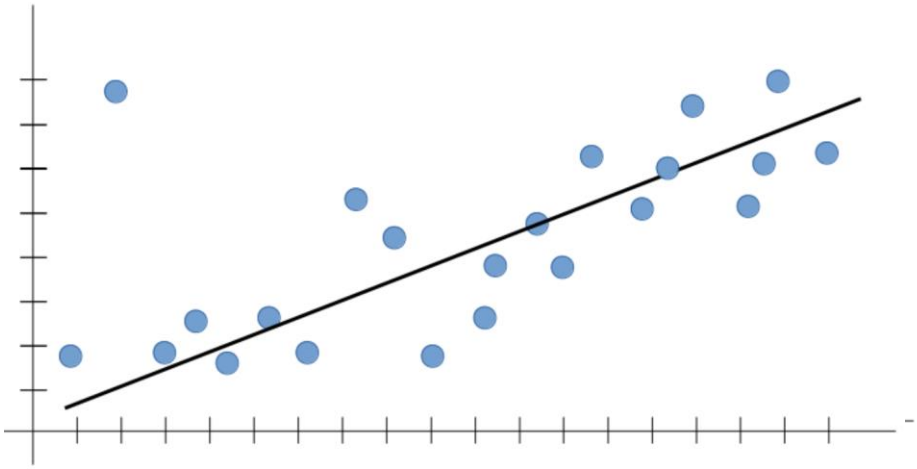
Example – Decision Trees (nonparametric algorithm)

- DTs are flexible and subject to overfitting
- By **pruning a tree (after it has learned)** in order to remove some of the detail it has picked up, **overfitting is controlled.**

https://www.investopedia.com/terms/o/overfitting.asp

PART 6

Underfitting



- An Ufit model can **neither** model the training data **nor** generalize to new data
- An underfit model is not a suitable model → has poor performance **on the training data**
- It is often not discussed as it is easy to detect given a good performance metric
- The remedy is to move on and try alternate machine learning algorithms
- It provides a good contrast to the problem of concept of overfitting

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

PART 6

What is a Good Fit in ML?

In Practice

- We need a model between underfitting and overfitting → results in the highest performance on NEW DATA
- Is it possible? Is it easy?

How do we achieve this goal:

- **Monitor** the **over time performance** of the ML algorithm we selected (as it is learning the training data).
- **Plot** the **performance** over time (training data; validation; test data)

As the algorithm learns:

1. The model error on the **training data** goes down
2. The model error on the **test data**……. goes down

Challenge – Generalization is at stake – if we train too long

(1) Error continues to decrease … thus, the model overfits

(2) At a certain point, the test error starts increasing

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

PART 6

What is a Good Fit in ML?

Not good practice

- For every ML algorithm: we do not stop learning, right before error on the test set starts to increase, and thus we expect to work well on unseen data.
- The test set is no longer unseen or **a standalone objective measure → knowledge leaks into the** training process.

Good practice – techniques to use for the model to work well on unseen data:

- Use resampling methods
- Use a validation dataset

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

PART 6

Dealing with Overfitting

Overfitting & underfitting → poor model performance

Overfitting is by far the most common problem in practical ML – does not work as expected on unseen data.

When evaluating ML algorithms, you can limit overfitting by:

1. Using resampling techniques (to estimate model accuracy)
2. Use a validation set

We will revisit all this topic again….

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

PART 6

Dealing with Overfitting

k-Fold Cross-Validation: The most popular resampling technique

- "It allows you to train and test your model k-times on different subsets of training data and build up an estimate of the performance of a machine learning model on unseen data"

Validation dataset: a training data subset that you hold back to further tune your model.

- ❖ Step 1: Training - tuning your ML algorithms
- ❖ Step 2: Evaluate the learned models on the validation dataset
- ❖ Step 3: Assess model on test set

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

Overfitting and Underfitting

What have we learned in Part SIX ?

A. _____: Good performance on the training data, poor generalization to other data.

B. _____: Poor performance on the training data and poor generalization to other data.

Source: ML-Mastery – Master Machine Learning Algorithms, 2020 by J. Brownlee

Questions?

THANK YOU!