Generalisation, Inductive Learning, Structured Data and When to Use ML?

Sagar Prakash Barad & Sajag Kumar

National Institute of Science Education and Research (NISER)
Bhubaneswar

January 16, 2023

Part I: Some Terminology

Part II: Generalization and Inductive Learning

Part III: Structured Data and When to Use Machine Learning

1 Structured and Unstructured Data 25

2 When to use Machine Learning? 28

3 Machine Learning (ML) Algorithms 31

Part I

Some Terminology

Artificial Intelligence

The **Oxford Dictionary of English** defines intelligence as ’the ability to acquire and apply knowledge and skills.‘

- Notice that to be called intelligent one should just be able to acquire some knowledge or skill and apply it. But in common usage, we refer to someone as intelligent when they are able to acquire and apply the knowledge and skills well.
- **Artificial intelligence** is the ability of machines to acquire and apply knowledge and skills.
- To quantify the intelligence of machines, the standard convention is to compare it with humans. We call a machine intelligent, if it is as good as humans at a certain task. If a machine is better than humans in all tasks, we call it **superintelligent** (such a machine does not exist).

Examples of Artificial Intelligence

1. AlphaZero, the famous chess-playing engine developed by DeepMind. It can defeat any human or other chess-playing engine quite easily, hence it is intelligent. However, it is not superintelligent despite being better than humans at chess because it cannot outperform humans in other tasks, for example, speech recognition.
2. Alexa, is a virtual assistant developed by Amazon. It recognises verbal instructions and, based on them, performs certain tasks. Alexa is as good as humans in speech recognition and hence is intelligent. But again not superintelligent because it cannot, for example, play chess.

Deterministic and Non-Deterministic Algorithms

In machine learning we will come across a number of algorithms. All of these algorithms can be broadly classified into two categories:

1. **Deterministic Algorithms**: Given an input, output is always the same, and the machine on which the algorithm is running when given same input produces the output in exactly the same way.
2. **Non-Deterministic Algorithms**: Given an input, the machine on which the algorithm is running produces the output in different ways, the output may or may not be the same for same input.

EXAMPLES I

1. **Deterministic Finite Automaton (DFA)**: The following DFA is an example of a deterministic machine. Given same input the machine goes through exactly the same states every time to produce the same output.

Accepts all the binary strings that contains 000 as a substring.

DFA

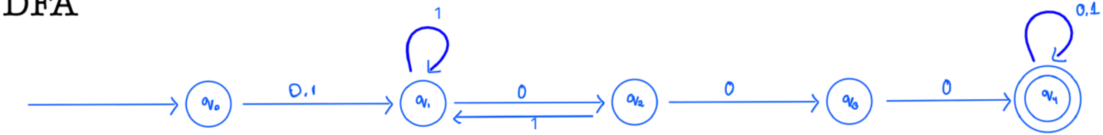

Figure 1. At state q_1 for input ’0’ and ’1’, one can determine the state to which machine will move.

Examples II

2. **Non-Deterministic Finite Automaton (NFA)**: The following NFA is an example of a non-deterministic machine. Given same input the machine produces the same output but it can go through different set of states.

Accepts all the binary strings that contains 000 as a substring.

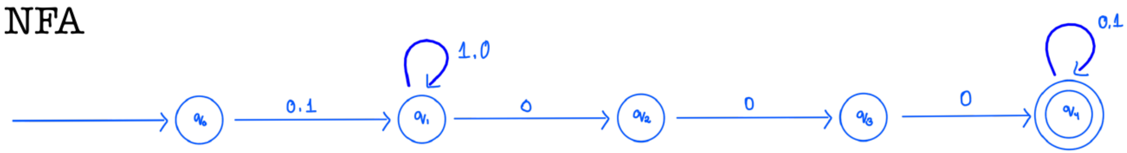

Figure 2. At state q_1 for input '0' machine can move to either state q_1 or q_0. Thus the exact state to which the machine will move cannot be determined.

Part II

Generalization

What is learning? I

- Consider a teacher T, student S and a subject S1. The teacher T, wants to know if the student S, learned the subject S1.
- T in order to teach S1 provides some notes to S.
- T wants to know if S learned from the notes.
- A parameter for learning would be how good S does on an exam.
- T can take three kinds of exam to judge if S actually learned S1:
  1. Put all the questions directly from the notes.
  2. Put questions from a subject other than S1.
  3. Put questions based on the notes but not directly from them.

- Lets see which of these exams can tell us if S learned S1 based on their performance.
  1. If S does well on the first exam it would mean that they are good at memorising. They memorised everything from the notes and reproduced on the exam.
  2. If S does well on the second exam. Either they are extremely lucky or they are God. This type of an exam is not a good way of gauging whether S actually learned anything.
  3. If S does well on the third exam it would imply they have learned from the notes. Because they could answer questions based on what they learned from the notes.

WHAT IS LEARNING? II

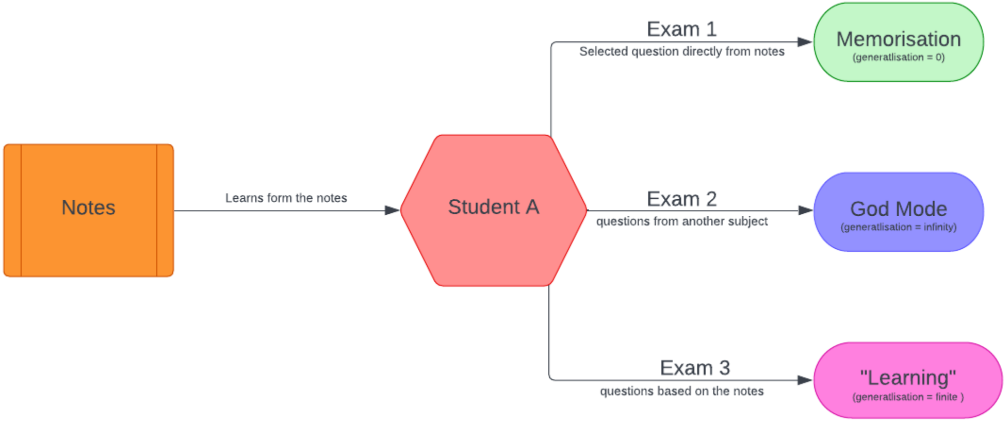
Notes
Learns form the notes
Student A
Exam 1
Selected question directly from notes
Memorisation
(generatlisation = 0)
Exam 2
questions from another subject
God Mode
(generatlisation = infinity)
Exam 3
questions based on the notes
"Learning"
(generatlisation = finite)

Figure 3. Flow chart summarizing the previous example.

Generalization

Generalization is the process of S learning from the notes and then doing well on an exam with questions based on the notes.

- If S does well on an exam with questions based on the notes but not directly from them we say that the knowledge or the method of learning of S has generalised well.

Why is generalisation important?

To see why generalisation is important we take two examples. One natural example from our life and one example from machine learning.

1. Consider a dog. While playing it hits a cactus tree and got hurt. So next time when it gets near a tree with needles it would be extra careful so as not to hurt itself. This would be an example of his knowledge generalising well. If for example the dog thinks that only that cactus tree can hurt it, it would keep getting hurt by other trees with needles. This is an example where the knowledge did not generalise well.
2. Suppose we have an algorithm for detecting whether an email is spam. We give the algorithm a bunch of emails classified as spam or not spam. Given a new email if the algorithm classifies it as spam or not spam based on the words or phrases present in the seen emails, and it produces correct results, we would say the algorithm generalised well. However given a new email if the algorithm just checks if the new email is exactly the same as the old spam email we would say the algorithm did not generalise at all.

Degrees of Generalisation

We would like to have some way of quantifying how generalised is a learning algorithm. Consider the example at the beginning of this part,

- the generalization of the learning method for doing well on the first exam is zero. S can just memorise everything. The generalization of memorisation is zero.
- the generalization of learning method for doing well on the second exam is infinity. S learned from the notes for S1 but could answer questions from S2. The generalization of this god-mode of learning is infinity.
- the generalization of learning method for doing well on the third exam depends on the performance of S. The generalization is definitely between zero and infinity and is directly proportional to how well S did on the exam.

Overfitting vs Underfitting I

Let us take the example of a machine learning algorithm which predicts the price of a house based on its area.

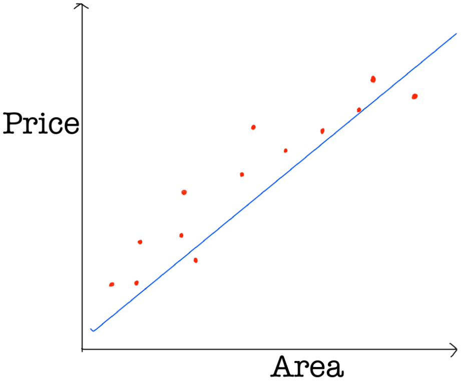

Figure 4. This is an example of underfitting. Very few data points are near or on the fitted line. The learning method which led to this result is similar to god-mode or the lucky escape method which helped student S do well in the second exam.

OVERFITTING VS UNDERFITTING II

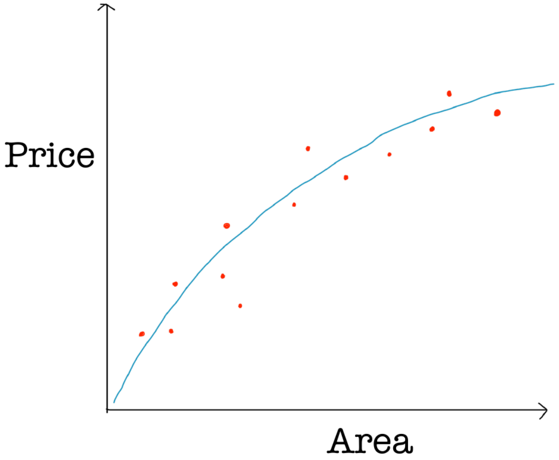

Figure 5. This is a good fit. The learning method which led to this result generalised well.

Overfitting vs Underfitting III

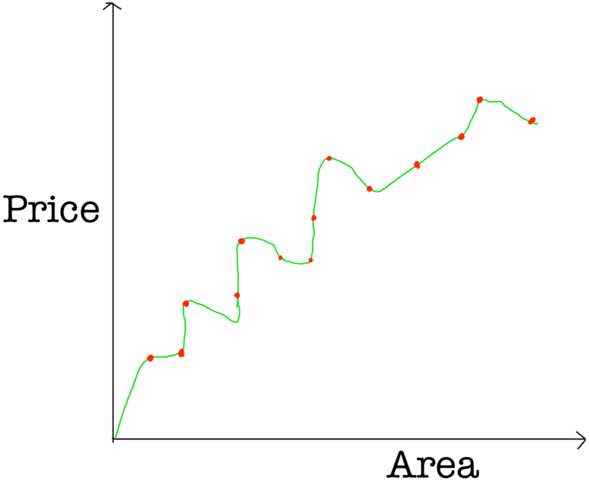

Figure 6. This is an example of overfitting. This situation is equivalent to memorisation. The learning method did not generalise at all. While the algorithm exactly predicts the price given any area it has already seen it would fail to make good predictions given new values of area.

Bias-Variance Trade-off

As seen in the previous example doing both too well and too bad on fitting the data points resulted in undesirable outcomes. To quantify a model's fitting capacity we need to understand the concept of bias and variance.

- Bias of a model is proportional to the difference in the value of a quantity as predicted by it and the actual value of the quantity. A model has high bias if the values predicted by it are way off actual values. Whereas it has low bias if the values predicted by it are close to the actual values.
- Variance of a model is the measure of the spread of the predicted values. A model with high variance has a very complicated fit to the data. A model with low variance has a simple fit to the data.

Bias-Variance Trade-off

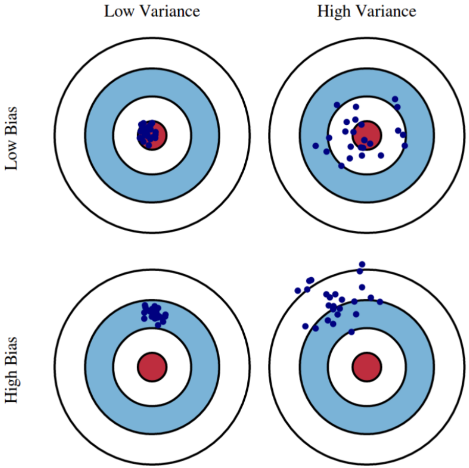

Figure 7. The bias-variance trade-off is of fundamental importance to machine learning. The ideal model will have low bias and low variance. A high variance model with low bias predicts value closer to the actual value but the spread of the predicted values is large. A high bias model with low variance predicts values which have a small spread but are far from the actual values. A high variance, high bias model is the worst, it predicts values with large spread which are far from the actual values.

INDUCTIVE LEARNING

Consider the teacher T, student S and subject S1 example again. The example can be summarised in the following flow-chart.

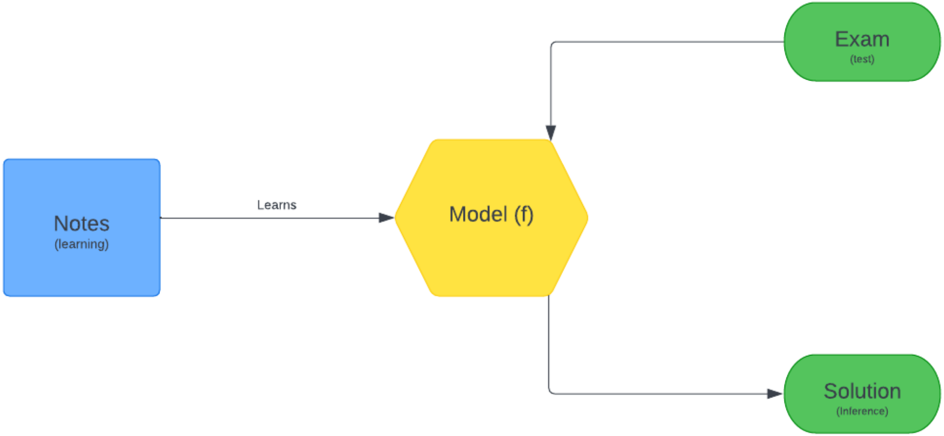

Figure 8. S uses the notes to learn. An exam is given to S to test its learning. S provides a solution to the exam based on his learning.

Inductive Learning

Inductive learning refer to the general learning algorithm described by the following flow-chart.

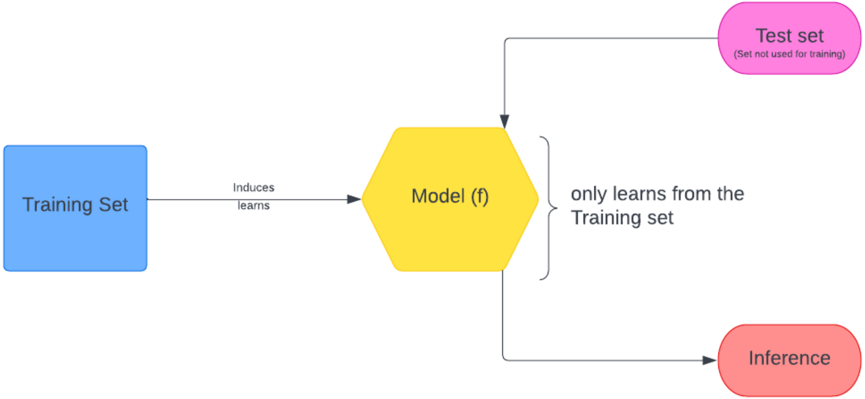
Test set
(Set not used for training)
Training Set
Induces
learns
Model (f)
only learns from the Training set
Inference

Figure 9. We give a set of examples as input called the **training set**. The model learns or **induces** from the training set and come up with a function f called the **hypothesis**. Then we test the model's performance on a few examples which we call the **test set**. Given an example from the test set the model using the hypothesis it induced from the training set produces an **inference**.

Examples

- Consider the housing price prediction example. The training set constitutes of actual prices of houses with their area. The algorithm will then come up with a line through these known data points. The equation of this line is the hypothesis of the model. We can then test the working of the model by feeding in some points that were not in the training set. Note that if we test the model on the points in the training set it may appear to perform well on the test set but fail on newer examples (the model may have overfit).
- In the spam detection example. The training set constitutes of emails and tagged as spam or not spam. The model will then form a hypothesis based on the training examples. The hypothesis here can be complicated it may classify an email as spam based on the words the sender and the phrases used. Once the algorithm has induced the hypothesis from the training set we can use some examples of known spam and not-spam emails not in the training set to test the hypothesis.

Part III

Structured Data and When to Use Machine Learning?

Structured and Unstructured Data

- Any information about a problem is data. Data on which we can ask statistical queries is called structured data. Examples of statistical queries is average, standard deviation, minimum value, maximum value, range etc.
- We have very good algorithms for statistical queries on structured data. For example, NumPy is a python library containing many efficient functions for statistical queries.
- But most of the data that we have is unstructured. We usually have data in the form of pictures or texts or audio files. These are all unstructured data.
- The general structure for structured data is a table or matrix. But this also depends on the problem in hand. For example, we want to do digit classification and we are given an image as a matrix, this matrix would be unstructured data, because we cannot ask statistical queries of interest on this data.
- Given unstructured data converting it to structured data is very difficult. We use machine learning for this task.

THE GOAL OF MACHINE LEARNING

The goal of machine learning algorithms is:

- To make unstructured data structured.
- Tidy the structured data.

A structured data is called tidy if it satisfies the following three criteria:

1. Every row represents one information.
2. Every column represents one unique value.
3. Every cell has only one value.

EXAMPLE

Suppose we are given several X-ray images and we want to tell whether the bones are broken or not. Lets say we have a few X-ray images for which we know the percent of damage and doctor's decision of whether the bones are broken.

Picture #	Damage Percentage	Doctor's Note
1	25	Not Broken
2	65	Not Broken
3	75	Broken

Table 1. Training set for our machine learning model.

Once our model is trained on this training set, it would be able to convert X-ray images (which is unstructured data) given to it to a table similar to table 1 (which is tidy structured data).

When to use Machine Learning (ML)? I

Following is a checklist for using machine learning to solve a problem. If any of these is not fulfilled for the problem in hand, then applying machine learning is not advisable.

- The problem does not have an efficient analytical solution from domain knowledge.
    - If an efficient analytical solution exists, using ML to solve the problem is a waste of time. At best, the ML algorithm will rediscover the already-known analytical solution. For example, using machine learning to find roots of a quadratic equation is not a good idea because we already have a formula for that.
    - However if the analytical solution available is not very efficient then ML algorithms can help. For example, determining the mechanism of an organic chemistry reaction has analytical solutions but these solutions require a lot of computation and generally very expensive to perform. In this case an ML algorithm can really help.

When to use Machine Learning (ML)? II

- The problem does not have any analytical solution from domain knowledge or the domain knowledge is scarce.
  - If this is the case our only option to solve the problem is to use machine learning. For example, if we want to find the roots of polynomial of degree 5. It is mathematically proven that for such polynomials an analytical solution is not possible. But there are machine learning algorithms which can predict roots of such polynomials given their coefficients.
  - In such cases we must make sure that we have enough data. Successful applications of machine learning usually requires a lot of data for the model to train on. We should use ML if we have a good amount of data or we can simulate a good amount of data for the problem. For example, in particle physics a lot of data is generated in big experiments such as LHC at the CERN, due to availability of huge data many machine learning algorithms are successfully deployed for particle physics experiments for classifying different particles, on the other hand in materials science the amount of data that we have is low and even simulating new data is very difficult because of which machine learning algorithms are still not very useful for materials science applications.

ML Algorithms

We use machine learning to solve mainly four types of problems, i.e regression, classification (multi or binary), and ranking. The problems mentioned above are solved using ML algorithms where our model learns from labeled or unlabeled training data, and the desired output may or may not be known.

- Regression: A regression algorithm is used to predict a continuous numeric value. It is used to model the relationship between a dependent variable (also known as the target or output variable) and one or more independent variables (also known as the predictor or input variables). For example, a model might be trained to predict the price of a house based on its size, location, and other features. Linear regression and polynomial regression are common algorithms used for regression tasks.

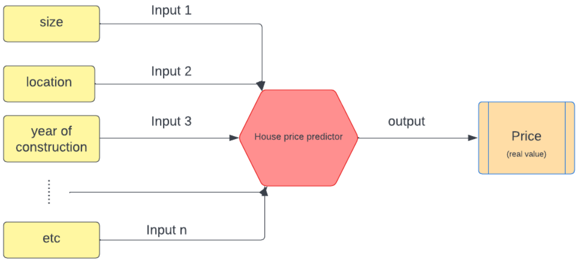

Figure 10. Model for housing price predictor.

ML Algorithms

- Binary classification: A Binary classification algorithm is used to classify items into one of two classes. It aims to learn from labeled data and make predictions about new, unseen data. Overall, a binary classification algorithm is used to predict one of two possible outcomes by learning from labeled data and making predictions on new unseen data. For example, a model might be trained to predict whether a customer will default on a loan or not. Logistic regression, k-nearest neighbors, and decision trees are common algorithms used for classification tasks.

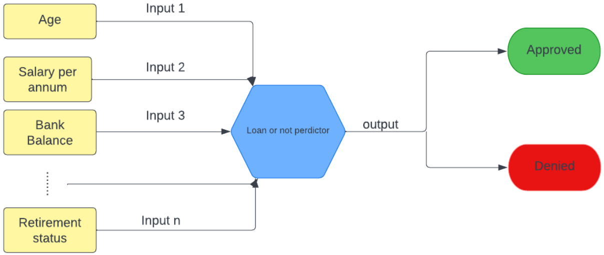

Figure 11. Model to predict if a borrower will default or not.

ML Algorithms

- Multi class classification:A multi-class classification algorithm is used to classify items into one of multiple classes or categories. The goal of a multi-class classification algorithm is to learn from labeled data and make predictions about new, unseen data. Overall, a multi-class classification algorithm is used to predict one of the multiple possible outcomes, by learning from labeled data and making predictions on new unseen data. For example, a model might be trained to predict the type of fruit in an image from a set of several different fruit types.

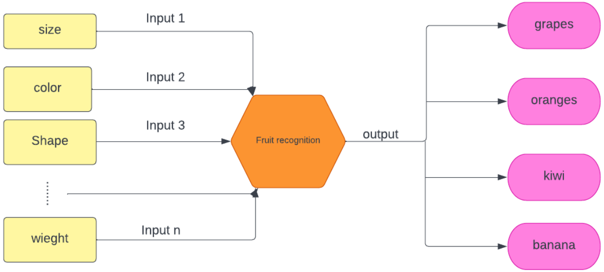

Figure 12. A model of fruit predictor using ML.

ML Algorithms

- Ranking:A ranking algorithm is a type of machine learning algorithm that is used to predict the order of items. The goal of a ranking algorithm is to learn from labeled data and make predictions about the relative order of new, unseen items. For example, a model might be trained to predict which articles are most likely to be read by a user, or which products are most likely to be purchased by a customer.

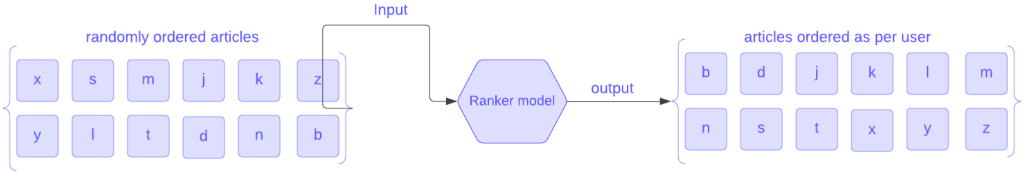

Figure 13. A model that recommends articles in order of what the reader would like the most.

References I

1. CS460 (Machine Learning) 2023 lectures, Subhankar Mishra.
2. CS460/C660 (Machine Learning) 2021, youtube playlist.
3. Deterministic algorithm, Wikipedia.
4. Artificial Intelligence, Wikipedia.
5. CS4780/CS5780 (Machine Learning for Intelligent Systems), Cornell University.
6. Introduction to Machine Learning, Coursera, Andrew Ng.