MODULE 1

OVERVIEW OF MACHINE LEARNING

Introduction to Machine Learning, Machine learning paradigms, supervised, semi-supervised, unsupervised, reinforcement learning. Supervised learning Input representation, Hypothesis class, Version space, Vapnik-Chervonenkis (VC) Dimension, Probably Approximately Correct Learning (PAC), Noise, Learning Multiple classes, Model Selection and Generalization

1. INTRODUCTION TO MACHINE LEARNING

Data Generation and Consumption

In the era of "big data," everyone generates data, not just companies. Personal computers and wireless communications have made individuals significant data producers through activities like purchasing products, browsing the web, and posting on social media. This vast amount of data can be utilized to tailor products and services to individual needs and preferences.

Business Example

Consider a supermarket chain that collects data on customer transactions. By analyzing this data, the supermarket aims to predict customer purchases to maximize sales and profit. Customers also benefit from finding products that meet their needs efficiently.

The Challenge

Predicting customer behavior isn't straightforward because behavior varies over time and location. However, purchasing patterns, like buying ice cream in the summer or beer with chips, suggest nonrandom behavior.

Algorithmic Solutions

Traditionally, solving problems on computers involves creating algorithms—a series of instructions to transform input into output. For instance, sorting a list of numbers can be done using welldefined algorithms.

Learning from Data

For tasks where algorithms are unknown, like predicting customer behavior or filtering spam emails, machine learning (ML) is used. ML leverages large datasets to "learn" patterns and make predictions. This process involves using example data to train models to recognize patterns and make decisions without explicit programming.

Approximation and Patterns

ML doesn't always provide perfect solutions but can construct useful approximations. These approximations reveal patterns that help make predictions about future behavior, assuming it resembles past behavior.

Data Mining

Applying ML to large datasets is known as data mining, which extracts valuable insights from vast amounts of data. It's used in various industries:

- Retail: Predicting customer purchases.
- Finance: Credit scoring, fraud detection.
- Manufacturing: Optimization and troubleshooting.
- Medicine: Diagnostic systems.
- Telecommunications: Network optimization.
- Science: Analyzing large datasets in fields like physics and biology.
- Web Search: Efficiently finding relevant information.

AI and Adaptability

ML is integral to artificial intelligence (AI), allowing systems to adapt to changing environments without preprogrammed solutions. It helps in solving problems in vision, speech recognition, and robotics.

Example: Face Recognition

Recognizing faces is a task humans do effortlessly but cannot easily explain how. ML models analyze patterns in facial features to recognize individuals, showcasing how ML captures and utilizes patterns in data.

2.MACHINE LEARNING PARADIGMS

Machine learning (ML) paradigms are different approaches or frameworks within which machine learning tasks are performed. These paradigms guide how models learn from data and how problems are structured. Here are the primary ML paradigms:

1. Supervised Learning

In supervised learning, models are trained on labeled data, meaning that each training example is paired with an output label. The model learns to map inputs to the correct outputs.

Types:

- **Regression**: Predicts continuous values. Example: Predicting house prices.
- **Classification**: Predicts categorical/Discrete values. Example: Email spam detection.

Examples:

Linear Regression: For predicting continuous outcomes.

Logistic Regression: For binary classification.

Support Vector Machines (SVM): For both classification and regression tasks.

Neural Networks: For complex patterns in both classification and regression.

2. Unsupervised Learning

Unsupervised learning involves training models on data without labeled responses. The model tries to find hidden patterns or intrinsic structures in the input data.

Types:

- **Clustering**: Groups similar data points together. Example: Customer segmentation.
- **Dimensionality Reduction**: Reduces the number of features. Example: Principal Component Analysis (PCA).

Examples:

k-Means Clustering: Partitions data into k clusters.

Hierarchical Clustering: Builds a hierarchy of clusters.

3. Semi-Supervised Learning

Semi-supervised learning uses a combination of labeled and unlabeled data for training. Typically, a small amount of labeled data is supplemented with a large amount of unlabeled data.

Applications: Useful in scenarios where labeling data is expensive or time-consuming, such as in medical imaging or natural language processing.

4. Reinforcement Learning

In reinforcement learning, an agent learns by interacting with an environment and receiving feedback in the form of rewards or penalties. The goal is to learn a policy that maximizes cumulative rewards over time.

Examples:

- **Q-Learning**: A value-based approach where the agent learns the value of actions.
- **Deep Q-Networks (DQN):** Combines Q-learning with deep neural networks.

- **Policy Gradient Methods:** Directly optimize the policy that the agent uses to select actions.
- **Applications:** Game playing (e.g., AlphaGo), robotics, autonomous vehicles.

5. Self-Supervised Learning

Self-supervised learning involves generating labels from the data itself. This paradigm uses the data's inherent structure to create supervised learning problems.

Examples:

Context Prediction: Predicting the context or future states in sequences (e.g., predicting the next word in a sentence).

Image Inpainting: Filling in missing parts of an image.

6. Transfer Learning

Transfer learning leverages knowledge from one domain (the source domain) to improve learning in another domain (the target domain). It is particularly useful when the target domain has limited data.

Examples:

Fine-Tuning Pretrained Models: Using a model trained on a large dataset (like ImageNet) and fine-tuning it for a specific task.

7. Active Learning

Active learning involves interactively querying a user or some other information source to label new data points. The model actively selects the most informative data points to be labeled.

Applications: Used in scenarios where labeled data is scarce and labeling is expensive, such as in medical diagnosis and anomaly detection.

8. Online Learning

Online learning involves training models incrementally as new data arrives, rather than all at once. This is useful for applications where data comes in a stream and it is not feasible to retrain the model from scratch frequently.

Applications: Real-time analytics, stock market prediction, adaptive filtering.

3.SUPERVISED LEARNING

In supervised learning, the machine is trained on a set of labeled data, which means that the input data is paired with the desired output. The machine then learns to predict the output for new input data. Supervised learning is often used for tasks such as classification, regression, and object detection.

Supervised learning is a type of machine learning algorithm that learns from labeled data. Labeled data is data that has been tagged with a correct answer or classification.

Supervised learning is when we teach or train the machine using data that is well-labelled. Which means some data is already tagged with the correct answer. After that, the machine is provided with a new set of examples(data) so that the supervised learning algorithm analyses the training data(set of training examples) and produces a correct outcome from labeled data.

Example:

Let's say you have a fruit basket that you want to identify. The machine would first analyze the image to extract features such as its shape, color, and texture. Then, it would compare these features to the features of the fruits it has already learned about. If the new image's features are most similar to those of an apple, the machine would predict that the fruit is an apple.

For instance, suppose you are given a basket filled with different kinds of fruits. Now the first step is to train the machine with all the different fruits one by one like this:

If the shape of the object is rounded and has a depression at the top, is red in color, then it will be labeled as –**Apple.**

If the shape of the object is a long curving cylinder having Green-Yellow color, then it will be labeled as –**Banana.**

Now suppose after training the data, you have given a new separate fruit, say Banana from the basket, and asked to identify it.

Since the machine has already learned the things from previous data and this time has to use it wisely. It will first classify the fruit with its shape and color and would confirm the fruit name as BANANA and put it in the Banana category. Thus the machine learns the things from training data(basket containing fruits) and then applies the knowledge to test data(new fruit).

3.1 TYPES OF SUPERVISED LEARNING

Supervised learning is classified into two categories of algorithms:

- **Regression**: A regression problem is when the output variable is a real value, such as “dollars” or “weight”.
- **Classification**: A classification problem is when the output variable is a category, such as “Red” or “blue” , “disease” or “no disease”.

3.1.1 REGRESSION

Regression is a type of supervised learning that is used to predict continuous values, such as house prices, stock prices, or customer churn. Regression algorithms learn a function that maps from the input features to the output value.

Some common regression algorithms include:

- Linear Regression
- Polynomial Regression
- Support Vector Machine Regression
- Decision Tree Regression
- Random Forest Regression

3.1.2 CLASSIFICATION

Classification is a type of supervised learning that is used to predict categorical values, such as whether a customer will churn or not, whether an email is spam or not, or whether a medical image shows a tumor or not. Classification algorithms learn a function that maps from the input features to a probability distribution over the output classes.

Some common classification algorithms include:

- Logistic Regression
- Support Vector Machines
- Decision Trees
- Random Forests
- Naive Baye

3.2 APPLICATIONS OF SUPERVISED LEARNING

Supervised learning can be used to solve a wide variety of problems, including:

- **Spam filtering**: Supervised learning algorithms can be trained to identify and classify spam emails based on their content, helping users avoid unwanted messages.
- **Image classification**: Supervised learning can automatically classify images into different categories, such as animals, objects, or scenes, facilitating tasks like image search, content moderation, and image-based product recommendations.

- **Medical diagnosis**: Supervised learning can assist in medical diagnosis by analyzing patient data, such as medical images, test results, and patient history, to identify patterns that suggest specific diseases or conditions.
- **Fraud detection**: Supervised learning models can analyze financial transactions and identify patterns that indicate fraudulent activity, helping financial institutions prevent fraud and protect their customers.
- **Natural language processing (NLP)**: Supervised learning plays a crucial role in NLP tasks, including sentiment analysis, machine translation, and text summarization, enabling machines to understand and process human language effectively.

ADVANTAGES OF SUPERVISED LEARNING

- Supervised learning allows collecting data and produces data output from previous experiences.
- Helps to optimize performance criteria with the help of experience.
- Supervised machine learning helps to solve various types of real-world computation problems.
- It performs classification and regression tasks.
- It allows estimating or mapping the result to a new sample.
- We have complete control over choosing the number of classes we want in the training data.

DISADVANTAGES OF SUPERVISED LEARNING

- Classifying big data can be challenging.
- Training for supervised learning needs a lot of computation time. So, it requires a lot of time.
- Supervised learning cannot handle all complex tasks in Machine Learning.
- Computation time is vast for supervised learning.
- It requires a labelled data set.
- It requires a training process.

4. SEMI-SUPERVISED LEARNING

Semi-supervised learning is a type of machine learning that falls in between supervised and unsupervised learning. It is a method that uses a small amount of labeled data and a large amount of unlabeled data to train a model. The goal of semi-supervised learning is to learn a function that can accurately predict the output variable based on the input variables, similar to supervised learning. However, unlike supervised learning, the algorithm is trained on a dataset that contains both labeled and unlabeled data.

Semi-supervised learning is particularly useful when there is a large amount of unlabeled data available, but it's too expensive or difficult to label all of it.

Intuitively, one may imagine the three types of learning algorithms as Supervised learning where a student is under the supervision of a teacher at both home and school, Unsupervised

learning where a student has to figure out a concept himself and Semi-Supervised learning where a teacher teaches a few concepts in class and gives questions as homework which are based on similar concepts.

Examples of Semi-Supervised Learning

Text classification: In text classification, the goal is to classify a given text into one or more predefined categories. Semi-supervised learning can be used to train a text classification model using a small amount of labeled data and a large amount of unlabeled text data.

Image classification: In image classification, the goal is to classify a given image into one or more predefined categories. Semi-supervised learning can be used to train an image classification model using a small amount of labeled data and a large amount of unlabeled image data.

Anomaly detection: In anomaly detection, the goal is to detect patterns or observations that are unusual or different from the norm

4.1 APPLICATIONS OF SEMI-SUPERVISED LEARNING

Speech Analysis: Since labeling audio files is a very intensive task, Semi-Supervised learning is a very natural approach to solve this problem.

Internet Content Classification: Labeling each webpage is an impractical and unfeasible process and thus uses Semi-Supervised learning algorithms. Even the Google search algorithm uses a variant of Semi-Supervised learning to rank the relevance of a webpage for a given query.

Protein Sequence Classification: Since DNA strands are typically very large in size, the rise of Semi-Supervised learning has been imminent in this field.

5. UNSUPERVISED LEARNING

Unsupervised learning is a branch of machine learning that deals with unlabeled data. Unlike supervised learning, where the data is labeled with a specific category or outcome, unsupervised learning algorithms are tasked with finding patterns and relationships within the data without any prior knowledge of the data's meaning.

In artificial intelligence, machine learning that takes place in the absence of human supervision is known as unsupervised machine learning. Unsupervised machine learning models, in contrast to supervised learning, are given unlabeled data and allow discover patterns and insights on their own—without explicit direction or instruction.

Unsupervised machine learning analyzes and clusters unlabeled datasets using machine learning algorithms. These algorithms find hidden patterns and data without any human intervention, i.e., we don't give output to our model. The training model has only input parameter values and discovers the groups or patterns on its own.

Unsupervised learning works by analyzing unlabeled data to identify patterns and relationships. The data is not labeled with any predefined categories or outcomes, so the algorithm must find these patterns and relationships on its own. This can be a challenging task, but it can also be very rewarding, as it can reveal insights into the data that would not be apparent from a labeled dataset.
EXAMPLE
Mall data that contains information about its clients that subscribe to them. Once subscribed they are provided a membership card and the mall has complete information about the customer and his/her every purchase. Now using this data and unsupervised learning techniques, the mall can easily group clients based on the parameters we are feeding in.

The input to the unsupervised learning models is as follows:

Unstructured data: May contain noisy(meaningless) data, missing values, or unknown data

Unlabeled data: Data only contains a value for input parameters, there is no targeted value(output). It is easy to collect as compared to the labeled one in the Supervised approach.

5.1 ADVANTAGES OF UNSUPERVISED LEARNING

- **No labeled data required**: Unlike supervised learning, unsupervised learning does not require labeled data, which can be expensive and time-consuming to collect.
- **Can uncover hidden patterns**: Unsupervised learning algorithms can identify patterns and relationships in data that may not be obvious to humans.
- **Can be used for a variety of tasks**: Unsupervised learning can be used for a variety of tasks, such as clustering, dimensionality reduction, and anomaly detection.
- **Can be used to explore new data**: Unsupervised learning can be used to explore new data and gain insights that may not be possible with other methods.

5.2 DISADVANTAGES OF UNSUPERVISED LEARNING

- **Difficult to evaluate:** It can be difficult to evaluate the performance of unsupervised learning algorithms, as there are no predefined labels or categories against which to compare results.
- **Can be difficult to interpret**: It can be difficult to understand the decision-making process of unsupervised learning models.
- **Can be sensitive to the quality of the data**: Unsupervised learning algorithms can be sensitive to the quality of the input data. Noisy or incomplete data can lead to misleading or inaccurate results.
- **Can be computationally expensive**: Some unsupervised learning algorithms, particularly those dealing with high-dimensional data or large datasets, can be computationally expensive

5.3 APPLICATIONS OF UNSUPERVISED LEARNING

- **Customer segmentation:** Unsupervised learning can be used to segment customers into groups based on their demographics, behavior, or preferences. This can help businesses to better understand their customers and target them with more relevant marketing campaigns.
- **Fraud detection**: Unsupervised learning can be used to detect fraud in financial data by identifying transactions that deviate from the expected patterns. This can help to prevent fraud by flagging these transactions for further investigation.
- **Recommendation systems**: Unsupervised learning can be used to recommend items to users based on their past behavior or preferences. For example, a recommendation system might use unsupervised learning to identify users who have similar taste in movies, and then recommend movies that those users have enjoyed.
- **Natural language processing (NLP):** Unsupervised learning is used in a variety of NLP tasks, including topic modeling, document clustering, and part-of-speech tagging.
- **Image analysis**: Unsupervised learning is used in a variety of image analysis tasks, including image segmentation, object detection, and image pattern recognition.

6. REINFORCEMENT LEARNING

Reinforcement learning is an area of Machine Learning. It is about taking suitable action to maximize reward in a particular situation. Reinforcement learning differs from supervised learning in a way that in supervised learning the training data has the answer key with it so the model is trained with the correct answer itself whereas in reinforcement learning, there is no answer but the reinforcement agent decides what to do to perform the given task. In the absence of a training dataset, it is bound to learn from its experience.

Reinforcement Learning (RL) is the science of decision making. It is about learning the optimal behavior in an environment to obtain maximum reward. In RL, the data is accumulated from machine learning systems that use a trial-and-error method. Data is not part of the input that we would find in supervised or unsupervised machine learning.

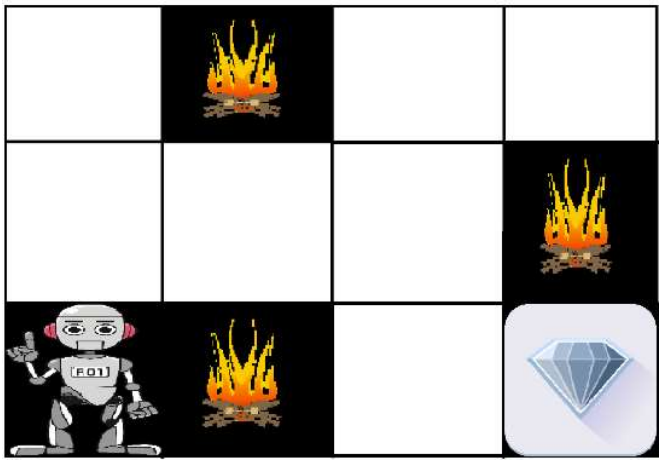
Reinforcement learning uses algorithms that learn from outcomes and decide which action to take next. After each action, the algorithm receives feedback that helps it determine whether the choice it made was correct, neutral or incorrect. It is a good technique to use for automated systems that have to make a lot of small decisions without human guidance.

Reinforcement learning is an autonomous, self-teaching system that essentially learns by trial and error. It performs actions with the aim of maximizing rewards, or in other words, it is learning by doing in order to achieve the best outcomes.

Example

The problem is as follows: We have an agent and a reward, with many hurdles in between. The agent is supposed to find the best possible path to reach the reward. The following problem explains the problem more easily.

The below image shows the robot, diamond, and fire. The goal of the robot is to get the reward that is the diamond and avoid the hurdles that are fired. The robot learns by trying all the possible paths and then choosing the path which gives him the reward with the least hurdles. Each right step will give the robot a reward and each wrong step will subtract the reward of the robot. The total reward will be calculated when it reaches the final reward that is the diamond.



6.1 APPLICATION OF REINFORCEMENT LEARNINGS

1. Robotics: Robots with pre-programmed behavior are useful in structured environments, such as the assembly line of an automobile manufacturing plant, where the task is repetitive in nature.

2. A master chess player makes a move. The choice is informed both by planning, anticipating possible replies and counter replies.

3. An adaptive controller adjusts parameters of a petroleum refinery's operation in real time.

RL can be used in large environments in the following situations:

A model of the environment is known, but an analytic solution is not available;

Only a simulation model of the environment is given (the subject of simulation-based optimization)

The only way to collect information about the environment is to interact with it.

6.2 ADVANTAGES OF REINFORCEMENT LEARNING

1. Reinforcement learning can be used to solve very complex problems that cannot be solved by conventional techniques.

2. The model can correct the errors that occurred during the training process.

3. In RL, training data is obtained via the direct interaction of the agent with the environment

4. Reinforcement learning can handle environments that are non-deterministic, meaning that the outcomes of actions are not always predictable. This is useful in real-world applications where the environment may change over time or is uncertain.

5. Reinforcement learning can be used to solve a wide range of problems, including those that involve decision making, control, and optimization.

6. Reinforcement learning is a flexible approach that can be combined with other machine learning techniques, such as deep learning, to improve performance.

6.3 DISADVANTAGES OF REINFORCEMENT LEARNING

1. Reinforcement learning is not preferable to use for solving simple problems.

2. Reinforcement learning needs a lot of data and a lot of computation

3. Reinforcement learning is highly dependent on the quality of the reward function. If the reward function is poorly designed, the agent may not learn the desired behavior.

4. Reinforcement learning can be difficult to debug and interpret. It is not always clear why the agent is behaving in a certain way, which can make it difficult to diagnose and fix problems.

7. SUPERVISED LEARNING

7.1 INPUT REPRESENTATION

The general classification problem is concerned with a class label to an unknown instance from instance of known assignment of labels .In a real world problem, a given situation or an object will have large number of features may contribute to the assignments of the labels.But in practice not all the features may be equally relevant or important only those which who are significant needed be considered as inputs for assigning the class labels

These features are referred to as the input features for the problem. they are also said to constitute an input representation

Example:

Consider the problem of assigning the label “family car” or “not family car” to cars. Let as assume that the features that separate a family car from other cars are the price and engine power .These attributes or features constitute the input representation for the problem. While deciding on this input representation we are ignoring various other attribute like seating capacity or colour irrelevant.

7.2 HYPOTHESIS SPACE

In the following discussions we consider only “binary classification” problems; that is, classification problems with only two class labels. The class labels are usually taken as “1” and “0”. The label “1” may indicate “True”, or “Yes”, or “Pass”, or any such label. The label “0” may indicate “False”, or “No” or “Fail”, or any such label. The examples with class labels 1 are called “positive examples” and examples with labels “0” are called “negative examples”.

7.2.1 Definition

1. Hypothesis

In a binary classification problem, a hypothesis is a statement or a proposition to explain a given set of facts or observations.

2. Hypothesis space

The hypothesis space for a binary classification problem is a set of hypotheses for the problem that might possibly be returned by it.

3. Consistency and satisfying

Let x be an example in a binary classification problem and let c(x) denote the class label

assigned to x (c(x) is 1 or 0). Let D be a set of training examples for the problem. Let h be a hypothesis for the problem and h(x) be the class label assigned to x by the hypothesis h

(a) We say that the hypothesis h is consistent with the set of training examples D if h(x) =c(x) for all x ∈ D.
(b) We say that an example x satisfies the hypothesis h if h(x) = 1.

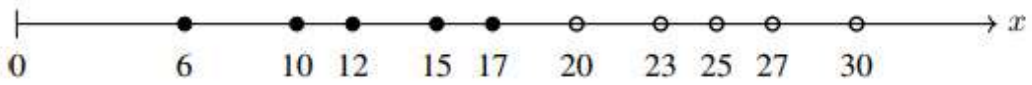
7.2.2 Examples

1. Consider the set of observations of a variable x with the associated class labels given in Table 2.1

x	27	15	23	20	25	17	12	30	6	10
Class	1	0	1	1	1	0	0	1	0	0

Table 2.1: Sample data to illustrate the concept of hypotheses

Figure 2.1 shows the data plotted on the x-axis.



Looking at Figure 2.1, it appears that the class labelling has been done based on the following rule.

h′: IF x ≥ 20 THEN “1” ELSE “0”. (2.1)

Note that h′ is consistent with the training examples in Table 2.1. For example, we have:

h′(27) = 1, c(27) = 1, h′(27) = c(27)h′(15) = 0, c(15) = 0, h′(15) = c(15)

Note also that, for x = 5 and x = 28 (not in training data),

h′(5) = 0, h′(28) = 1.

The hypothesis h′ explains the data. The following proposition also explains the

data:

h″ : IF x ≥ 19 THEN “0” ELSE “1”. (2.2)

It is not enough that the hypothesis explains the given data; it must also predict correctly the class label of future observations. So we consider a set of such hypotheses and choose the “best” one. The set of hypotheses can be defined using a parameter, say m, as given below:

hm : IF x ≥ m THEN “1” ELSE ”0”. (2.3)

The set of all hypotheses obtained by assigning different values to m constitutes the hypothesis

space H; that is,H = {hm ： m is a real number}. (2.4)

For the same data, we can have different hypothesis spaces. For example, for the data in Table2.1, we may also consider the hypothesis space defined by the following proposition:

h′m : IF x ≤ m THEN “0” ELSE “1”.

2. Consider a situation with four binary variables x1, x2, x3, x4 and one binary output variable y. Suppose we have the following observations.

x_1	x_2	x_3	x_4	y
0	0	0	1	1
0	1	0	1	0
1	1	0	0	1
0	0	1	0	0

The problem is to predict a function f of x1, x2, x3, x4 which predicts the value of y for any combination of values of x1, x2, x3, x4. In this problem, the hypothesis space is the set of all possible functions f. It can be shown that the size of the hypothesis space is $2^{(2^4)} = 65536$.

3. Consider the problem of assigning the label "family car" or "not family car" to cars. For convenience, we shall replace the label "family car" by "1" and "not family car" by "0".

Suppose we choose the features "price ('000 $)" and "power (hp)" as the input representation for the problem. Further, suppose that there is some reason to believe that for a car to be a family car, its price and power should be in certain ranges. This supposition can be formulated in the form of the following proposition:

IF (p1 < price < p2) AND (e_1 < power < e_2) THEN "1" ELSE "0" (2.5)

for suitable values of p1, p2, e1 and e2. Since a solution to the problem is a proposition of the form Eq.(2.5) with specific values for p1, p2, e1 and e2, the hypothesis space for the problem is the set of all such propositions obtained by assigning all possible values for p1, p2, e_1 and e_2

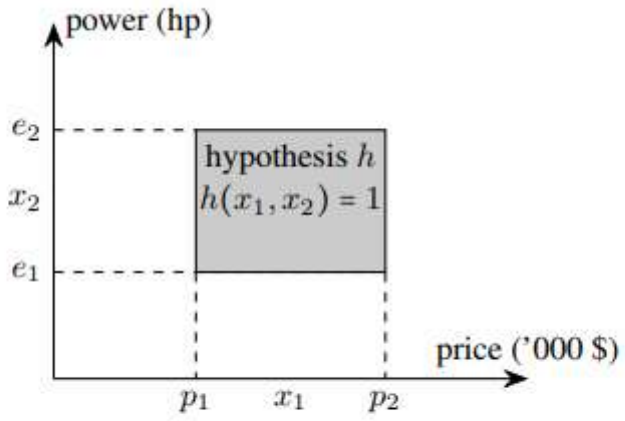


Figure 2.2: An example hypothesis defined by Eq. (2.5)

It is interesting to observe that the set of points in the power–price plane which satisfies the condition

(p1 < price < p2) AND (e1 < power < e2)

defines a rectangular region (minus the boundary) in the price–power space as shown in Figure 2.2. The sides of this rectangular region are parallel to the coordinate axes. Such a rectangle is called an axis-aligned rectangle If h is the hypothesis defined by Eq.(2.5), and (x1, x2) is any point in the price–power plane, then

h(x1, x2) = 1 if and only if (x1, x2) is within the rectangular region.

Hence we may identify the hypothesis h with the rectangular region. Thus, the hypothesis space for the problem can be thought of as the set of all axis-aligned rectangles in the price–power plane

7.2.3 Ordering of hypotheses

Definition Let X be the set of all possible examples for a binary classification problem and let h ′ and h ″ be two hypotheses for the problem.

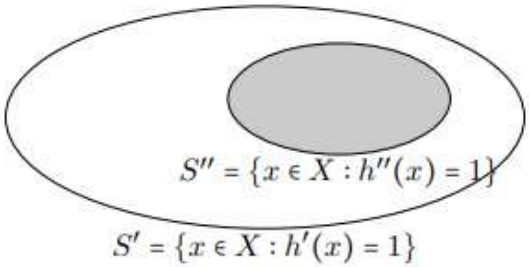


Figure 2.3: Hypothesis h' is more general than hypothesis h'' if and only if $S'' \subseteq S'$

1. We say that h′ is more general than h″
   if and only if for every x ∈ X, if x satisfies h″ then x satisfies h′also;
   that is, if h″(x) = 1 then h′(x) = 1 also. The relation “is more general than”defines a partial ordering relation in hypothesis space.

2. We say that h′ is more specific than h″, if h″ is more general than h′.

3. We say that h′ is strictly more general than h″ if h′ is more general than h″ and h″ is not

more general than h′.

4. We say that h′ is strictly more specific than h″ if h′ is more specific than h″ and h″ is not

more specific than h′.

Example Consider the hypotheses h ′ and h ″ defined in Eqs.(2.1),(2.2). Then it is easy to check that if h ′ (x) = 1 then h ″(x) = 1 also. So, h ″ is more general than h ′ . But, h ′ is not more general than h ″ and so h ″ is strictly more general than h ′ .

7.3 Version Space

Definition

Consider a binary classification problem. Let D be a set of training examples and H a hypothesis space for the problem. The version space for the problem with respect to the

set D and the space H is the set of hypotheses from H consistent with D; that is, it is the set

$VS_{D,H} = \{h \in H : h(x) = c(x) \text{ for all } x \in D\}$.

2.4.1 Examples

Example 1

Consider the data D given in Table 2.1 and the hypothesis space defined by Eqs.(2.3)-(2.4).

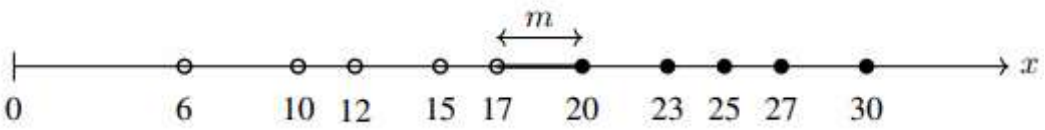


Figure 2.4: Values of m which define the version space with data in Table 2.1 and hypothesis space defined by Eq.(2.4)

From Figure 2.4 we can easily see that the hypothesis space with respect to this dataset D and hypothesis space H is as given below:

$VS_{D,H} = \{hm : 17 < m \leq 20\}$

Example 2

Consider the problem of assigning the label "family car" (indicated by "1") or "not family car" (indicated by "0") to cars. Given the following examples for the problem and assuming that the hypothesis space is as defined by Eq. (2.5), the version space for the problem

x_1: Price in '000 ($)	32	82	44	34	43	80	38
x_2: Power (hp)	170	333	220	235	245	315	215
Class	0	0	1	1	1	0	1

x_1	47	27	56	28	20	25	66	75
x_2	260	290	320	305	160	300	250	340
Class	1	0	0	0	0	0	0	0

Solution Figure 2.5 shows a scatter plot of the given data. In the figure, the data with class label "1" (family car) is shown as hollow circles and the data with class labels "0" (not family car) are shown as solid dots.

A hypothesis as given by Eq.(2.5) with specific values for the parameters p1, p2, e1 and e2 specifies an axis-aligned rectangle as shown in Figure 2.2. So the hypothesis space for the problem can be thought as the set of axis-aligned rectangles in the price-power plane.

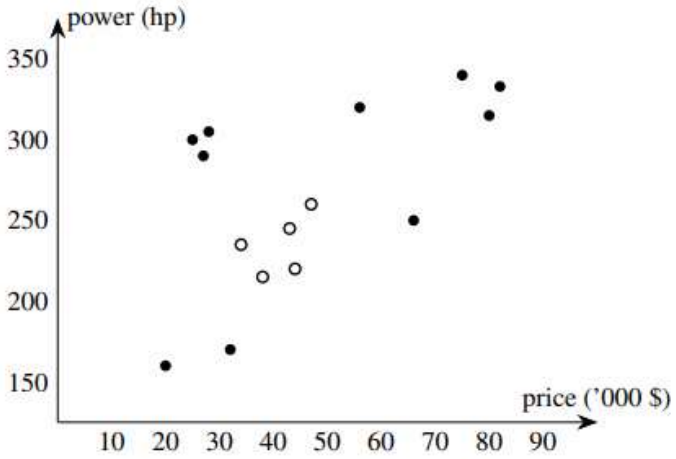


Figure 2.5: Scatter plot of price-power data (hollow circles indicate positive examples and solid dots indicate negative examples)

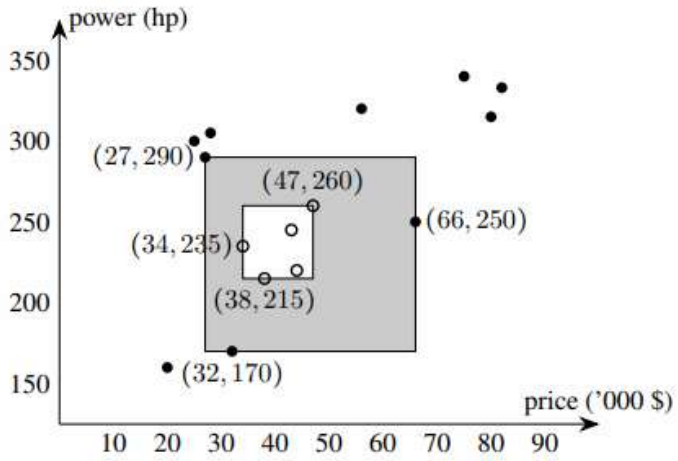


Figure 2.6: The version space consists of hypotheses corresponding to axis-aligned rectangles contained in the shaded region

The version space consists of all hypotheses specified by axis-aligned rectangles contained in the shaded region in Figure 2.6. The inner rectangle is defined by

(34 < price < 47) AND (215 < power < 260)

and the outer rectangle is defined by

(27 < price < 66) AND (170 < power < 290).

7.4 Noise

7.4.1 Noise and their sources

Noise is any unwanted anomaly in the data ([2] p.25). Noise may arise due to several factors:

1. There may be imprecision in recording the input attributes, which may shift the data points in the input space.

2. There may be errors in labeling the data points, which may relabel positive instances as negative and vice versa. This is sometimes called teacher noise.

3. There may be additional attributes, which we have not taken into account, that affect the label of an instance. Such attributes may be hidden or latent in that they may be unobservable. The effect of these neglected attributes is thus modelled as a random component and is included in "noise."

7.4.2 Effect of noise

Noise distorts data. When there is noise in data, learning problems may not produce accurate results. Also, simple hypotheses may not be sufficient to explain the data and so complicated hypotheses may have to be formulated. This leads to the use of additional computing resources and the needless wastage of such resources.

For example, in a binary classification problem with two variables, when there is noise, there may not be a simple boundary between the positive and negative instances and to separate them. A rectangle can be defined by four numbers, but to define a more complicated shape one needs a more complex model with a much larger number of parameters. So, when there is noise, we may make a complex model which makes a perfect fit to the data and attain zero error; or, we may use a simple model and allow some error.

7.5 Learning multiple classes

So far we have been discussing binary classification problems. In a general case there may be more than two classes. Two methods are generally used to handle such cases. These methods are known by the names "**one-against-all**" and "**one-against-one**".

7.5.1 Procedures for learning multiple classes

"One-against all" method

Consider the case where there are K classes denoted by C1, . . . , CK. Each input instance belongs to exactly one of them.

We view a K-class classification problem as K two-class problems. In the i-th two-class problem, the training examples belonging to Ci are taken as the positive examples and the examples of all other classes are taken as the negative examples. So, we have to find K hypotheses h1, . . . , hK

where hi is defined by

$$h_i(x) = \begin{cases} 1 & \text{if } x \text{ is in class } C_i \\ 0 & \text{otherwise} \end{cases}$$

For a given x, ideally only one of hi(x) is 1 and then we assign the class Ci to x. But, when no, or, two or more, hi(x) is 1, we cannot choose a class. In such a case, we say that the classifier rejects such cases

“One-against-one” method

In the one-against-one (OAO) (also called one-vs-one (OVO)) strategy, a classifier is constructed for each pair of classes. If there are K different class labels, a total of

$K(K-1)/2$ classifiers are constructed. An unknown instance is classified with the class getting the most votes. Ties are broken arbitrarily.

For example, let there be three classes, A, B and C. In the OVO method we construct

$3(3-1)/2 = 3$ binary classifiers. Now, if any x is to be classified, we apply each of the three classifiers to x. Let the three classifiers assign the classes A, B, B respectively to x. Since a label to x is assigned by the majority voting, in this example, we assign the class label of B to x.

7.6 Model selection

As we have pointed earlier in Section 1.1.1, there is no universally accepted definition of the term “model”. It may be understood as some mathematical expression or equation, or some mathematical structures such as graphs and trees, or a division of sets into disjoint subsets, or a set of logical “if . . . then . . . else . . .” rules, or some such thing.

In order to formulate a hypothesis for a problem, we have to choose some model and the term “model selection” has been used to refer to the process of choosing a model. However, the term has been used to indicate several things. In some contexts it has been used to indicates the process of choosing one particular approach from among several different approaches. This may be choosing an appropriate algorithms from a selection of possible algorithms, or choosing the sets of features to be used for input, or choosing initial values for certain parameters. Sometimes “model selection” refers to the process of picking a particular mathematical model from among different mathematical models which all purport to describe the same data set. It has also been described as the process of choosing the right inductive bias

7.6.1 Inductive bias

In a learning problem we only have the data. But data by itself is not sufficient to find the solution. We should make some extra assumptions to have a solution with the data we have. The set of assumptions we make to have learning possible is called the inductive bias of the learning algorithm.

One way we introduce inductive bias is when we assume a hypothesis class.

Examples

- In learning the class of family car, there are infinitely many ways of separating the positive examples from the negative examples. Assuming the shape of a rectangle is an inductive bias.
- In regression, assuming a linear function is an inductive bias.

The model selection is about choosing the right inductive bias.

7.6.2 Advantages of a simple model

Even though a complex model may not be making any errors in prediction, there are certain advantages in using a simple model.

1. A simple model is easy to use.

2. A simple model is easy to train. It is likely to have fewer parameters. It is easier to find the corner values of a rectangle than the control points of an arbitrary shape.

3. A simple model is easy to explain

4. A simple model would generalize better than a complex model. This principle is known as Occam's razor, which states that simpler explanations are more plausible and any unnecessary complexity should be shaved off.

7.7 Generalisation

How well a model trained on the training set predicts the right output for new instances is called generalization.

Generalization refers to how well the concepts learned by a machine learning model apply to specific examples not seen by the model when it was learning. The goal of a good machine learning model is to generalize well from the training data to 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

are the two biggest causes for poor performance of machine learning algorithms. The model should be selected having the best generalisation. This is said to be the case if these problems are avoided.

• Underfitting

Underfitting is the production of a machine learning model that is not complex enough to

accurately capture relationships between a datasetâA˘ Zs features and a target variable. ´

• Overfitting

Overfitting is the production of an analysis which corresponds too closely or exactly to a

particular set of data, and may therefore fail to fit additional data or predict future observations reliably.

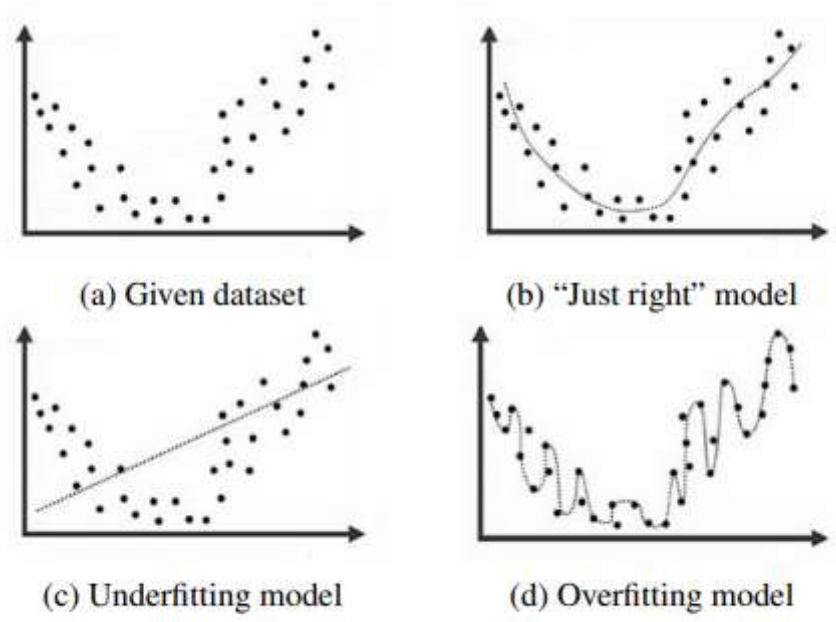


Figure 2.7: Examples for overfitting and overfitting models

Consider a dataset shown in Figure 2.7(a). Let it be required to fit a regression model to the data. The graph of a model which looks "just right" is shown in Figure 2.7(b). In Figure 2.7(c)we have a linear regression model for the same dataset and this model does seem to capture the essential features of the dataset. So this model suffers from underfitting. In Figure 2.7(d) we have a regression model which corresponds too closely to the given dataset and hence it does not account for small random noises in the dataset. Hence it suffers from overfitting.

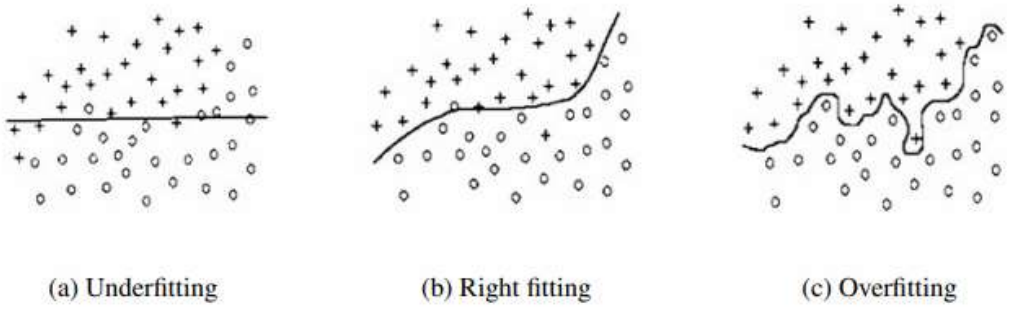


Figure 2.8: Fitting a classification boundary

Suppose we have to determine the classification boundary for a dataset two class labels. An example situation is shown in Figure 2.8 where the curved line is the classification boundary. The three figures illustrate the cases of underfitting, right fitting and overfitting

7.8 Vapnik-Chervonenkis (VC) dimensions and PAC Learning

7.8.1 Vapnik-Chervonenkis dimension

The VC dimension of a hypothesis set H is the largest number of points that can be shattered by H. A hypothesis set H shatters a set of points S if, for every possible labeling of the points in S, there exists a hypothesis in H that correctly classifies the points. In other words, a hypothesis set shatters a set of points if it can fit any possible labeling of those points.

Let H be the hypothesis space for some machine learning problem. The Vapnik-Chervonenkis dimension of H, also called the VC dimension of H, and denoted by V C(H), is a measure of the complexity (or, capacity, expressive power, richness, or flexibility) of the space H. To define the VC dimension we require the notion of the shattering of a set of instances.

7.8.2 Shattering of a set

Let D be a dataset containing N examples for a binary classification problem with class labels 0 and 1. Let H be a hypothesis space for the problem. Each hypothesis h in H partitions D into two disjoint subsets as follows:

$\{x \in D \mid h(x) = 0\}$ and $\{x \in D \mid h(x) = 1\}$.

Such a partition of S is called a "dichotomy" in D. It can be shown that there are 2 N possible dichotomies in D. To each dichotomy of D there is a unique assignment of the labels "1" and "0" to the elements of D. Conversely, if S is any subset of D then, S defines a unique hypothesis h as follows:

$$h(x) = \begin{cases} 1 & \text{if } x \in S \\ 0 & \text{otherwise} \end{cases}$$

Thus to specify a hypothesis h, we need only specify the set $\{x \in D \mid h(x) = 1\}$. Figure 3.1 shows all possible dichotomies of D if D has three elements. In the figure, we have shown only one of the two sets in a dichotomy, namely the set $\{x \in D \mid h(x) = 1\}$. The circles and ellipses represent such sets.

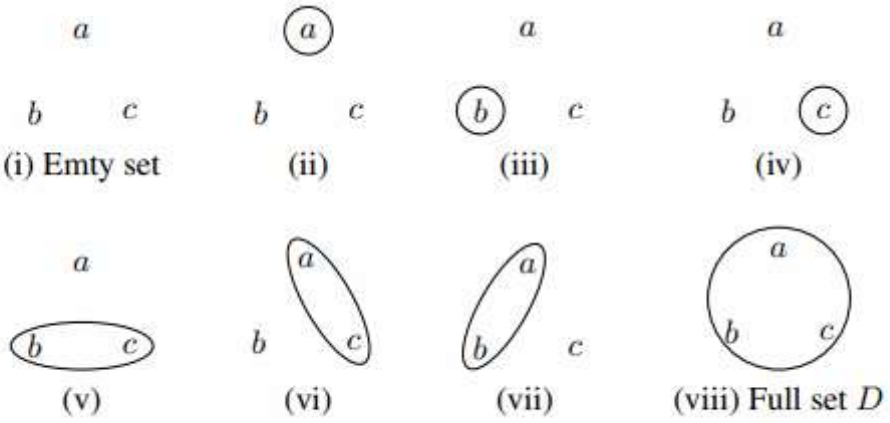


Figure 3.1: Different forms of the set $\{x \in S : h(x) = 1\}$ for $D = \{a, b, c\}$

We require the notion of a hypothesis consistent with a set of examples introduced in Section 2.4 in the following definition.

Definition

A set of examples D is said to be shattered by a hypothesis space H if and only if for very dichotomy of D there exists some hypothesis in H consistent with the dichotomy of D.

Vapnik-Chervonenkis dimension

The following example illustrates the concept of Vapnik-Chervonenkis dimension. Example Let the instance space X be the set of all real numbers. Consider the hypothesis space defined by Eqs.(2.3)-(2.4):

H = {hm ： m is a real number}

where

hm ： IF x ≥ m THEN ”1” ELSE “0”.

i) Let D be a subset of X containing only a single number, say, D = {3.5}. There are 2 dichotomies for this set. These correspond to the following assignment of class labels:

x	3.25
Label	0

x	3.25
Label	1

h4 ∈ H is consistent with the former dichotomy and h3 ∈ H is consistent with the latter. So, to every dichotomy in D there is a hypothesis in H consistent with the dichotomy. Therefore, the set D is shattered by the hypothesis space H.

ii) Let D be a subset of X containing two elements, say, D = {3.25, 4.75}. There are 4 dichotomies in D and they correspond to the assignment of class labels shown in Table 3.1. In these dichotomies, h5 is consistent with (a), h4 is consistent with (b) and h3 is consistent with (d). But there is no hypothesis hm ∈ H consistent with (c). Thus the two-element set D is not shattered by H. In a similar way it can be shown that there is no two-element subset of X which is shattered by H.

It follows that the size of the largest finite subset of X shattered by H is 1. This number is the VC dimension of H.

x	3.25	4.75
Label	0	0

(a)

x	3.25	4.75
Label	0	1

(b)

x	3.25	4.75
Label	1	0

(c)

x	3.25	4.75
Label	1	1

(d)

Table 3.1: Different assignments of class labels to the elements of {3.25, 4.75}

Definition

The Vapnik-Chervonenkis dimension (VC dimension) of a hypothesis space H defined over an instance space (that is, the set of all possible examples) X, denoted by V C(H), is the size of the largest finite subset of X shattered by H. If arbitrarily large subsets of X can be shattered by H, then we define V C(H) = ∞.

Remarks

It can be shown that V C(H) ≤ log2 (|H|) where H is the number of hypotheses in H.

7.9 Probably approximately correct learning

In computational learning theory, probably approximately correct learning (PAC learning) is a framework for mathematical analysis of machine learning algorithms. It was proposed in 1984 by Leslie Valiant.

In this framework, the learner (that is, the algorithm) receives samples and must select a hypothesis from a certain class of hypotheses. The goal is that, with high probability (the "probably" part), the selected hypothesis will have low generalization error (the "approximately correct" part).

In this section we first give an informal definition of PAC-learnability. After introducing a few more notions, we give a more formal, mathematically oriented, definition of PAC-learnability. At the end, we mention one of the applications of PAC-learnability.

7.9.1 PAC-learnability

To define PAC-learnability we require some specific terminology and related notations.

• Let X be a set called the instance space which may be finite or infinite. For example, X may be the set of all points in a plane.

• A concept class C for X is a family of functions c : X → {0, 1}. A member of C is called a concept. A concept can also be thought of as a subset of X. If C is a subset of X, it defines a unique function μC : X → {0, 1} as follows:

$$\mu_C(x) = \begin{cases} 1 & \text{if } x \in C \\ 0 & \text{otherwise} \end{cases}$$

• A hypothesis h is also a function h : X → {0, 1}. So, as in the case of concepts, a hypothesis can also be thought of as a subset of X. H will denote a set of hypotheses.

• We assume that F is an arbitrary, but fixed, probability distribution over X.

• Training examples are obtained by taking random samples from X. We assume that the samples are randomly generated from X according to the probability distribution F. Now, we give below an informal definition of PAC-learnability.