Unit I

1 Introduction to Machine Learning

Syllabus

Introduction to Machine Learning, Comparison of Machine learning with traditional programming, ML vs AI vs Data Science.
Types of learning : Supervised, Unsupervised, and semi-supervised, reinforcement learning techniques, Models of Machine learning : Geometric model, Probabilistic Models, Logical Models, Grouping and grading models, Parametric and non-parametric models.
Important Elements of Machine Learning - Data formats, Learnability, Statistical learning approaches.

Contents

1.1 Introduction to Machine Learning

SPPU : Oct.-19, Dec.-19

- Machine Learning (ML) is a sub-field of Artificial Intelligence (AI) which concerns with developing computational theories of learning and building learning machines.
- **Learning** is a phenomenon and process which has manifestations of various aspects. Learning process includes gaining of new symbolic knowledge and development of cognitive skills through instruction and practice. It is also discovery of new facts and theories through observation and experiment.
- **Machine Learning Definition :** A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.
- Machine learning is programming computers to optimize a performance criterion using example data or past experience. Application of machine learning methods to large databases is called **data mining.**
- It is very hard to write programs that solve problems like recognizing a human face. We do not know what program to write because we don't know how our brain does it. Instead of writing a program by hand, it is possible to collect lots of examples that specify the correct output for a given input.
- A machine learning algorithm then takes these examples and produces a program that does the job. The program produced by the learning algorithm may look very different from a typical hand-written program. It may contain millions of numbers. If we do it right, the program works for new cases as well as the ones we trained it on.
- Main goal of machine learning is to devise learning algorithms that do the learning automatically without human intervention or assistance. The machine learning paradigm can be viewed as "programming by example." Another goal is to develop computational models of human learning process and perform computer simulations.
- The goal of machine learning is to build computer systems that can adapt and learn from their experience.
- Algorithm is used to solve a problem on computer. An algorithm is a sequence of instruction. It should carry out to transform the input to output. For example, for addition of four numbers is carried out by giving four number as input to the algorithm and output is sum of all four numbers. For the same task, there may be various algorithms. It is interested to find the most efficient one, requiring the least number of instructions or memory or both.
- For some tasks, however, we do not have an algorithm.

Why is Machine Learning Important ?

- Machine learning algorithms can figure out how to perform important tasks by generalizing from examples.
- Machine Learning provides business insight and intelligence. Decision makers are provided with greater insights into their organizations. This adaptive technology is being used by global enterprises to gain a competitive edge.
- Machine learning algorithms discover the relationships between the variables of a system (input, output and hidden) from direct samples of the system.
- **Following are some of the reasons :**
  1. Some tasks cannot be defined well, except by examples. For example: recognizing people.
  2. Relationships and correlations can be hidden within large amounts of data. To solve these problems, machine learning and data mining may be able to find these relationships.
  3. Human designers often produce machines that do not work as well as desired in the environments in which they are used.
  4. The amount of knowledge available about certain tasks might be too large for explicit encoding by humans.
  5. Environments change time to time.
  6. New knowledge about tasks is constantly being discovered by humans.
- Machine learning also helps us find solutions of many problems in computer vision, speech recognition and robotics. Machine learning uses the theory of statistics in building mathematical models, because the core task is making inference from a sample.

How Machines Learn ?

- Machine learning typically follows three phases :
  1. **Training :** A training set of examples of correct behavior is analyzed and some representation of the newly learnt knowledge is stored. This is some form of rules.
  2. **Validation :** The rules are checked and, if necessary, additional training is given. Sometimes additional test data are used, but instead, a human expert may validate the rules, or some other automatic knowledge - based component may be used. The role of the tester is often called the opponent.
  3. **Application :** The rules are used in responding to some new situation.
- Fig. 1.1.1 shows phases of ML.

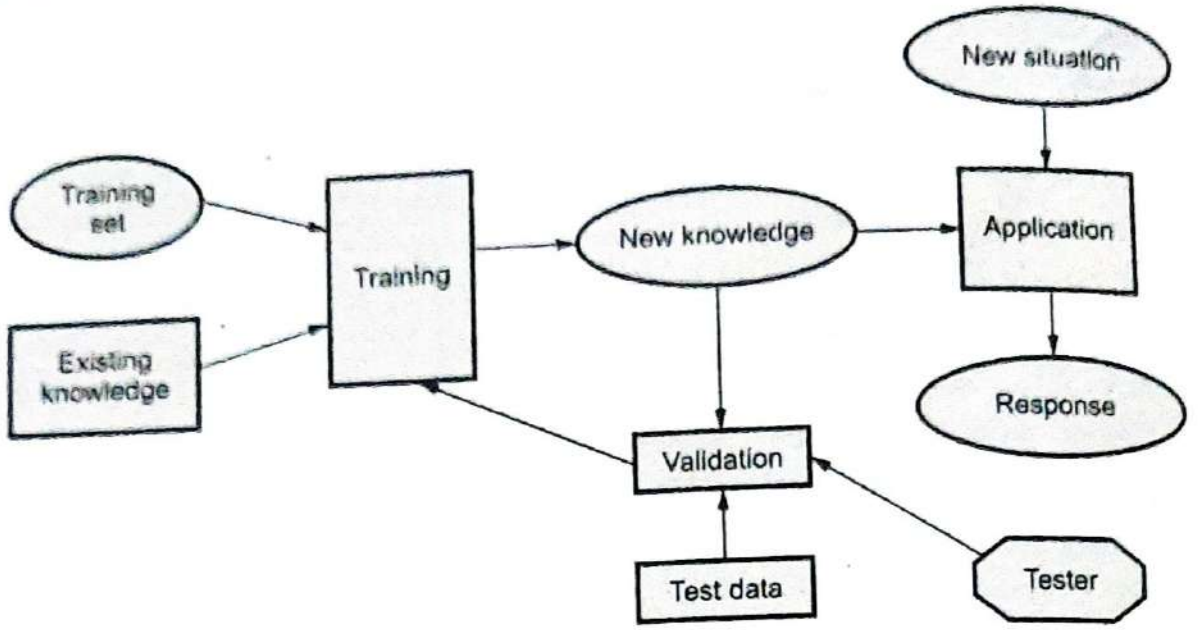


Fig. 1.1.1 Phases of ML

1.1.1 Why Machine Learning is Important ?

- Machine learning algorithms can figure out how to perform important tasks by generalizing from examples.
- Machine learning provides business insight and intelligence. Decision makers are provided with greater insights into their organizations. This adaptive technology is being used by global enterprises to gain a competitive edge.
- Machine learning algorithms discover the relationships between the variables of a system (input, output and hidden) from direct samples of the system.
- **Following are some of the reasons :**
  1. Some tasks cannot be defined well, except by examples. For example : Recognizing people.
  2. Relationships and correlations can be hidden within large amounts of data. To solve these problems, machine learning and data mining may be able to find these relationships.
  3. Human designers often produce machines that do not work as well as desired in the environments in which they are used.
  4. The amount of knowledge available about certain tasks might be too large for explicit encoding by humans.
  5. Environments change time to time.
  6. New knowledge about tasks is constantly being discovered by humans.

- Machine learning also helps us find solutions of many problems in computer vision, speech recognition and robotics. Machine learning uses the theory of statistics in building mathematical models, because the core task is making inference from a sample.
- Learning is used when :
  1. Human expertise does not exist (navigating on Mars),
  2. Humans are unable to explain their expertise (speech recognition)
  3. Solution changes in time (routing on a computer network)
  4. Solution needs to be adapted to particular cases (user biometrics)

1.1.2 Ingredients of Machine Learning

The ingredients of machine learning are as follows :

1. **Tasks :** The problems that can be solved with machine learning. A task is an abstract representation of a problem. The standard methodology in machine learning is to learn one task at a time. Large problems are broken into small, reasonably independent sub-problems that are learned separately and then recombined.
   - Predictive tasks perform inference on the current data in order to make predictions. Descriptive tasks characterize the general properties of the data in the database.
2. **Models :** The output of machine learning. Different models are geometric models, probabilistic models, logical models, grouping and grading.
   - The model-based approach seeks to create a modified solution tailored to each new application. Instead of having to transform your problem to fit some standard algorithm, in model-based machine learning you design the algorithm precisely to fit your problem.
   - Model is just made up of set of assumptions, expressed in a precise mathematical form. These assumptions include the number and types of variables in the problem domain, which variables affect each other, and what the effect of changing one variable is on another variable.
   - Machine learning models are classified as : Geometric model, Probabilistic model and Logical model.
3. **Features :** The workhorses of machine learning. A good feature representation is central to achieving high performance in any machine learning task.
   - Feature extraction starts from an initial set of measured data and builds derived values intended to be informative, non redundant, facilitating the subsequent learning and generalization steps.

- Feature selection is a process that chooses a subset of features from the original features so that the feature space is optimally reduced according to a certain criterion.

Review Questions

1. *Justify the following*
   i) Predict the height of a person. Is it a regression task.
   ii) Find the gender of a person by analyzing his writing style. Is it a classification task ?
   iii) Filter out spam emails. Is it a example of unsupervised learning. **SPPU : Oct.-19, Marks 5**
2. *What is machine learning ? Explain types of machine learning.* **SPPU : Dec.-19, Marks 5**

1.2 Comparison of Machine Learning with Traditional Programming

- Machine learning seeks to construct a model or logic for the problem by analyzing its input data and answers. In contrast, traditional programming is that programming aims to answer a problem using a predefined set of rules or logic
- Machine learning is the ability of machines to automate a learning process. The input of this learning process is data and the output is a model. Through machine learning, a system can perform a learning function with the data it ingests and thus it becomes progressively better at said function.
- Traditional programming is a manual process. It requires a programmer to create the rules or logic of the program. We have to manually come up with the rules and feed it to the computer alongside input data. The machine then processes the given data according to the coded rules and comes up with answers as output.
- Fig 1.2.1 shows machine learning and traditional programming.

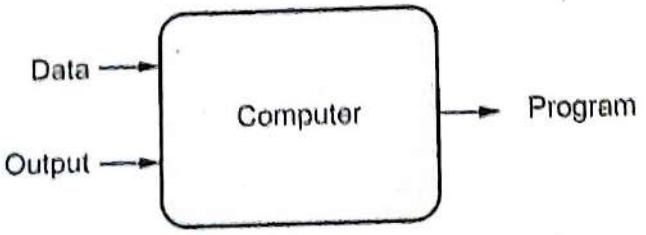


Fig. 1.2.1 (a) Machine learning process

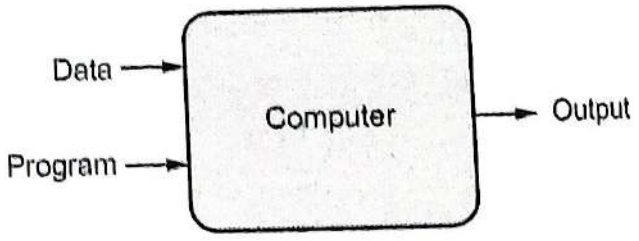


Fig. 1.2.1 (b) Traditional programming

- For projects that involve predicting output or identifying objects in images, **machine learning** has proven to be much more efficient. In traditional programming, the rule-based approach is preferred in situations where the problem is of an algorithmic manner and there are not so many parameters to consider when writing the logical rules.

- Machine Learning is a proven technique for helping to solve complex problems such as facial and voice recognition, recommendation systems, self-driving cars and email spam detection.

1.2.1 ML vs AI vs Data Science

- Artificial Intelligence (AI) is the broad concept of developing machines that can simulate human thinking, reasoning and behavior.
- Machine Learning (ML) is a subset of AI wherein computer systems learn from the environment and in turn, use these learnings to improve experiences and processes. All machine learning is AI, but not all AI is machine learning.
- Data Science is the processing, analysis and extraction of relevant assumptions from data. It's about finding hidden patterns in the data. A Data Scientist makes use of machine learning in order to predict future events.
- Fig. 1.2.2 shows relation between AI, ML and Data science.

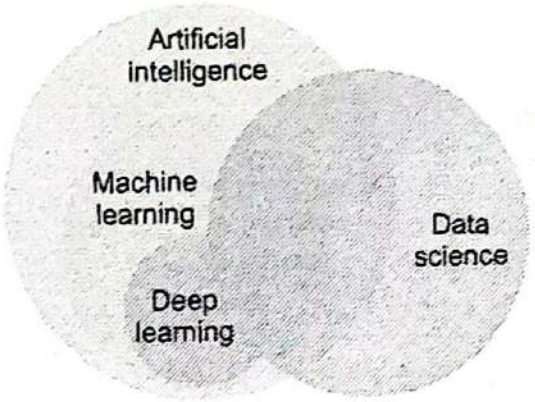


Fig. 1.2.2 Relation between AI, ML and Data science

- Machine Learning uses statistical models. Artificial Intelligence uses logic and decision trees. Data Science deals with structured data.
- Machine Learning : A form of analytics in which software programs learn about data and find patterns.
- AI : Development of computerized applications that simulate human intelligence and interaction.
- Data Science : The process of using advanced analytics to extract relevant information from data.

In Table Format :

Machine Learning	Artificial Intelligence	Data Science
Focuses on providing a means for algorithms and systems to learn from experience with data and use that experience to improve over time.	Focuses on giving machines cognitive and intellectual capabilities similar to those of humans.	Focuses on extracting information needles from data haystacks to aid in decision-making and planning.
Machine Learning uses statistical models.	Artificial Intelligence uses logic and decision trees.	Data Science deals with structured data.
A form of analytics in which software programs learn about data and find patterns.	Development of computerized applications that simulate human intelligence and interaction.	The process of using advanced analytics to extract relevant information from data.
Objective is to maximize accuracy.	Objective is to maximize the chance of success.	Objective is to extract actionable insights from the data.
ML can be done through supervised, unsupervised or reinforcement learning approaches.	AI encompasses a collection of intelligence concepts, including elements of perception, planning and prediction.	Uses statistics, mathematics, data wrangling, big data analytics, machine learning and various other methods to answer analytics questions.
ML is concerned with knowledge accumulation.	AI is concerned with knowledge dissemination and conscious machine actions.	Data science is all about data engineering.

1.3 Types of Learning

- Learning is essential for unknown environments, i.e. when designer lacks the omniscience. Learning simply means incorporating information from the training examples into the system.
- Learning is any change in a system that allows it to perform better the second time on repetition of the same task or on another task drawn from the same population. One part of learning is acquiring knowledge and new information; and the other part is problem-solving.
- Supervised and Unsupervised Learning are the different types of machine learning methods. A computational learning model should be clear about the following aspects :
  1. **Learner** : Who or what is doing the learning. For example : Program or algorithm.
  2. **Domain** : What is being learned ?
  3. **Goal** : Why the learning is done ?

4. **Representation :** The way the objects to be learned are represented.
5. **Algorithmic technology :** The algorithmic framework to be used.
6. **Information source :** The information (training data) the program uses for learning.
7. **Training scenario :** The description of the learning process.

- Learning is constructing or modifying representation of what is being experienced. Learn means to get knowledge of by study, experience or being taught.
- Machine learning is a scientific discipline concerned with the design and development of the algorithm that allows computers to evolve behaviors based on empirical data, such as form sensors data or database.
- Machine learning is usually divided into two main types : Supervised Learning and Unsupervised Learning.

Why do Machine Learning ?

1. To understand and improve efficiency of human learning.
2. Discover new things or structure that is unknown to humans (Example : Data mining).
3. Fill in skeletal or incomplete specifications about a domain.

- Fig. 1.3.1 shows types of machine learning.

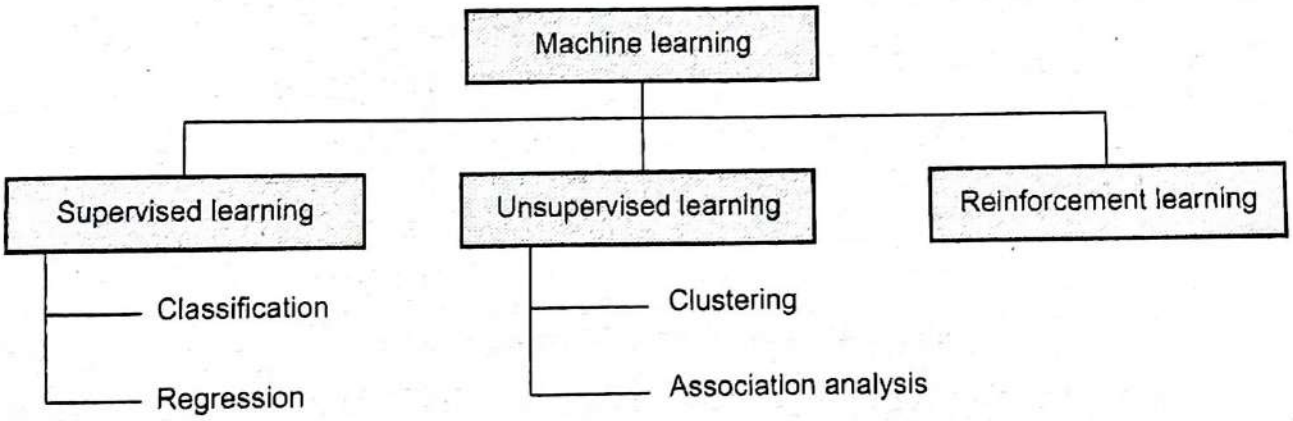


Fig. 1.3.1 Types of machine learning

1.4 Supervised Learning

SPPU : March-20, June-22

- Supervised learning is the machine learning task of inferring a function from supervised training data. The training data consist of a set of training examples. The task of the supervised learner is to predict the output behavior of a system for any set of input values, after an initial training phase.
- **Supervised learning** in which the network is trained by providing it with input and matching output patterns. These input-output pairs are usually provided by an external teacher.

- Human learning is based on the past experiences. A computer does not have experiences.
- A computer system learns from data, which represent some "past experiences" of an application domain.
- To learn a target function that can be used to predict the values of a discrete class attribute, e.g., approve or not-approved and high-risk or low risk. The task is commonly called : Supervised learning, Classification or inductive learning.
- Training data includes both the input and the desired results. For some examples the correct results (targets) are known and are given in input to the model during the learning process. The construction of a proper training, validation and test set is crucial. These methods are usually fast and accurate.
- Have to be able to generalize : give the correct results when new data are given in input without knowing a priori the target.
- Supervised learning is the machine learning task of inferring a function from supervised training data. The training data consist of a set of training examples. In supervised learning, each example is a pair consisting of an input object and a desired output value.
- A supervised learning algorithm analyzes the training data and produces an inferred function, which is called a classifier or a regression function. Fig. 1.4.1 shows supervised learning process.

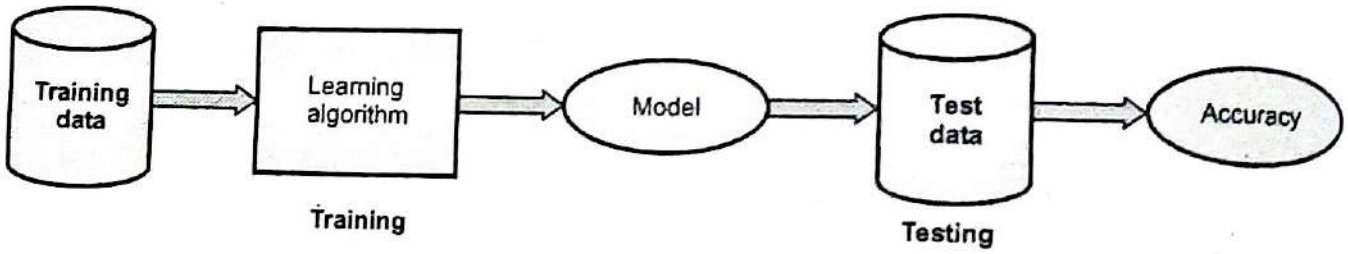


Fig. 1.4.1 Supervised learning process

- The learned model helps the system to perform task better as compared to no learning.
- Each input vector requires a corresponding target vector.
  Training Pair = (Input Vector, Target Vector)
- Fig. 1.4.2 shows input vector. (See Fig. 1.4.2 on next page)
- Supervised learning denotes a method in which some input vectors are collected and presented to the network. The output computed by the net-work is observed and the deviation from the expected answer is measured. The weights are corrected according to the magnitude of the error in the way defined by the learning algorithm.

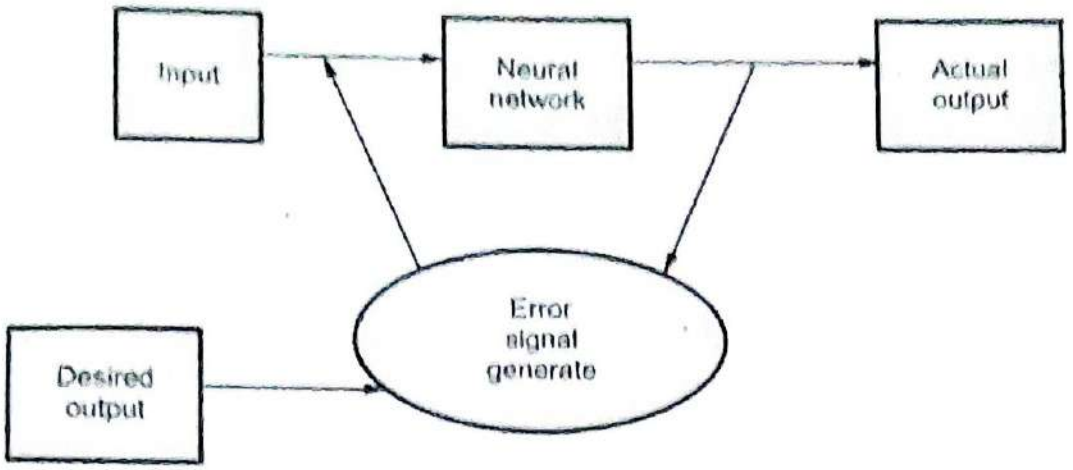


Fig. 1.4.2 Input vector

- Supervised learning is further divided into methods which use reinforcement or error correction. The perceptron learning algorithm is an example of supervised learning with reinforcement.

Data formats in supervised learning :

- Supervised learning always uses a dataset to define finite set of real vectors with m features each :

$$X = \{\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n\} \text{ where } \bar{x}_i \in \mathbb{R}^m$$

- Considering that user approach is always probabilistic, we need to consider each X as drawn from a statistical multivariate distribution D. It is also useful to add an important condition upon the whole dataset X. Here we consider that all samples to be independent and identically distributed. This means all variables belong to the same distribution D and considering an arbitrary subset of m values, it happens that :

$$P(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_m) = \prod_{i=1}^{m} P(\bar{x}_i)$$

- The corresponding output values can be both numerical - continuous and categorical. In the first case, the process is called regression, while in the second, it is called classification.
- Example : Dataset contains city populations by year for the past 100 years and user want to know what the population of a specific city will be four years from now. The outcome uses labels that already exist in the data set : **population, city and year.**
- In order to solve a given problem of supervised learning, following steps are performed :
  1. Find out the type of training examples.

2. Collect a training set.
3. Determine the input feature representation of the learned function.
4. Determine the structure of the learned function and corresponding learning algorithm.
5. Complete the design and then run the learning algorithm on the collected training set.
6. Evaluate the accuracy of the learned function. After parameter adjustment and learning, the performance of the resulting function should be measured on a test set that is separate from the training set.

- Supervised learning is divided into two types : Classification and Regression.

1. Classification :

- Classification predicts categorical labels (classes), prediction models continuous - valued functions. Classification is considered to be supervised learning.
- Classifies data based on the training set and the values in a classifying attribute and uses it in classifying new data. **Prediction** means models continuous-valued functions, i.e., predicts unknown or missing values.
- Preprocessing of the data in preparation for classification and prediction can involve data cleaning to reduce noise or handle missing values, relevance analysis to remove irrelevant or redundant attributes and data transformation, such as generalizing the data to higher level concepts or normalizing data.
- Numeric prediction is the task of predicting continuos values for given input. For example, we may wish to predict the salary of college employee with 15 years of work experience or the potential sales of a new product given its price.
- Some of the classification methods like back - propagation, support vector machines and k-nearest-neighbor classifiers can be used for prediction.

2. Regression :

- For an input x, if the output is continuous, this is called a regression problem. For example, based on historical information of demand for tooth paste in supermarket, user are asked to predict the demand for the next month.
- Regression is concerned with the prediction of continuous quantities. Linear regression is the oldest and most widely used predictive model in the field of machine learning. The goal is to minimize the sum of the squared errors to fit a straight line to a set of data points.
- Regression algorithm used in supervised learning is linear regression, Bayesian linear regression, polynomial regression, regression tree etc.

1.4.1 Advantages and Disadvantages of Supervised Learning

1. Advantages of supervised learning

- 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.

2. Disadvantages of supervised learning

- 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.

Review Questions

1. *Explain supervised learning with example.* **SPPU : March-20, In Sem, Marks 5**
2. *Explain data formats for supervised learning problem with example.* **SPPU: June-22, End Sem, Marks 6**

1.5 Unsupervised Learning

- Unsupervised learning is a type of machine learning in which models are trained using unlabeled dataset and are allowed to act on that data without any supervision.
- In unsupervised learning, a dataset is provided without labels and a model learns useful properties of the structure of the dataset. The main goal of unsupervised learning is to discover hidden and interesting patterns in unlabeled data.
- They are called unsupervised because they do not need a teacher or super-visor to label a set of training examples. Only the original data is required to start the analysis.
- Unsupervised learning tasks typically involve grouping similar examples together, dimensionality reduction and density estimation.
- Common algorithms used in unsupervised learning include clustering, anomaly detection, neural networks.

- Fig. 1.5.1 shows unsupervised learning.

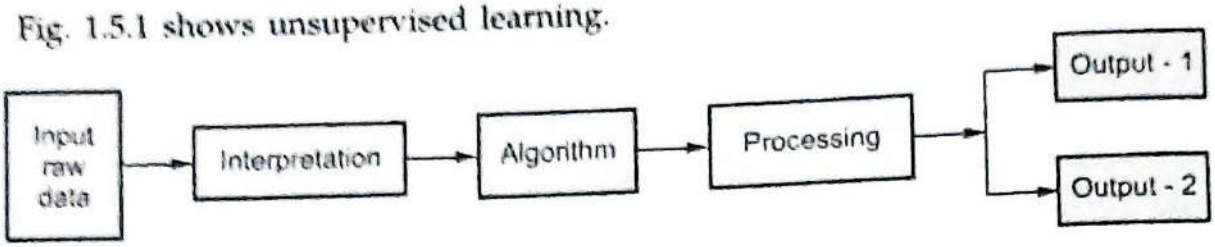


Fig. 1.5.1 Unsupervised learning

- The most common unsupervised learning method is cluster analysis, which applies clustering methods to explore data and find hidden patterns or groupings in data. Unsupervised learning is typically applied before supervised learning, to identify features in exploratory data analysis and establish classes based on groupings.
- Unsupervised machine learning is mainly used to :
  1. Cluster datasets on similarities between features or segment data.
  2. Understand relationship between different data point such as automated music recommendations.
  3. Perform initial data analysis.
- Unsupervised learning algorithms have the capability of analyzing large amounts of data and identifying unusual points among the dataset. Once those anomalies have been detected, they can be brought to the awareness of the user, who can then decide to act or not on this warning.
- Anomaly detection can be very useful in the financial and banking sectors. Indeed, financial fraud has become a daily problem, due to the ease with which credit card details can be stolen. Using unsupervised learning models, unauthorized or fraudulent transactions on a bank account can be identified as it will most often constitute a change in the user's normal pattern of spending.
- Example : Using customer data and user want to create segments of customers who like similar products. The data that user are providing is not labeled and the labels in the outcome are generated based on the similarities that were discovered between data points.
- Types of unsupervised learning is clustering and association analysis.
- There is a wide range of algorithms that can be deployed under unsupervised learning. A few of them includes : K-means clustering, Principal component analysis, Hierarchical clustering and Dendrogram.

1.5.1 Advantages and Disadvantages of Unsupervised Learning

1. Advantages of unsupervised learning

- It does not require a training data to be labelled.

- Dimensionality reduction can be easily accomplished using unsupervised learning.
- Capable of finding previously unknown patterns in data.

2. Disadvantages of unsupervised learning

- Difficult to measure accuracy or effectiveness due to lack of predefined answers during training.
- The results often have lesser accuracy.
- The user needs to spend time interpreting and label the classes which follow that classification.

1.5.2 Difference between Supervised and Unsupervised Learning

Sr. No.	Supervised learning	Unsupervised learning
1.	Desired output is given.	Desired output is not given.
2.	It is not possible to learn larger and more complex models than with supervised learning.	It is possible to learn larger and more complex models with unsupervised learning.
3.	Use training data to infer model.	No training data is used.
4.	Every input pattern that is used to train the network is associated with an output pattern.	The target output is not presented to the network.
5.	Trying to predict a function from labeled data.	Try to detect interesting relations in data.
6.	Supervised learning requires that the target variable is well defined and that a sufficient number of its values are given.	For unsupervised learning typically either the target variable is unknown or has only been recorded for too small a number of cases.
7.	Example : Optical character recognition.	Example : Find a face in an image.
8.	We can test our model.	We can not test our model.
9.	Supervised learning is also called classification.	Unsupervised learning is also called clustering.

1.6 Semi-supervised Learning

- Semi-supervised learning uses both labeled and unlabeled data to improve supervised learning. The goal is to learn a predictor that predicts future test data better than the predictor learned from the labeled training data alone.
- Semi-supervised learning is motivated by its practical value in learning faster, better and cheaper.

- In many real world applications, it is relatively easy to acquire a large amount of unlabeled data x.
- For example, documents can be crawled from the Web, images can be obtained from surveillance cameras, and speech can be collected from broadcast. However, their corresponding labels y for the prediction task, such as sentiment orientation, intrusion detection and phonetic transcript, often requires slow human annotation and expensive laboratory experiments.
- In many practical learning domains, there is a large supply of unlabeled data but limited labeled data, which can be expensive to generate. For example : text processing, video-indexing, bioinformatics etc.
- Semi-supervised Learning makes use of both labeled and unlabeled data for training, typically a small amount of labeled data with a large amount of unlabeled data. When unlabeled data is used in conjunction with a small amount of labeled data, it can produce considerable improvement in learning accuracy.
- Semi-supervised learning sometimes enables predictive model testing at reduced cost.
- **Semi-supervised classification :** Training on labeled data exploits additional unlabeled data, frequently resulting in a more accurate classifier.
- **Semi-supervised clustering :** Uses small amount of labeled data to aid and bias the clustering of unlabeled data.

1.6.1 Comparision between Supervised, Unsupervised, Semi-supervised Learning

Sr. No.	Supervised learning	Unsupervised learning	Semi-supervised learning
1.	Input data is labeled.	Input data is unlabeled.	A large amount of input data is unlabeled while a small amount is labeled.
2.	Trying to predict a specific quantity.	Trying to understand the data.	Using unsupervised methods to improve supervised algorithm.
3.	Used in Fraud detection.	Used in Identity management.	Used in spam detection.
4.	Subtype : Classification and regression.	Subtype : Clustering and association.	Subtype : Classification, regression, clustering and association.
5.	Higher accuracy.	Lesser accuracy.	Lesser accuracy.

1.7 Reinforcement Learnings

SPPU : March-20

- 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.
- A good example of using reinforcement learning is a robot learning how to walk. The robot first tries a large step forward and falls. The outcome of a fall with that big step is a data point the reinforcement learning system responds to. Since the feedback was negative, a fall, the system adjusts the action to try a smaller step. The robot is able to move forward. This is an example of reinforcement learning in action.
- Reinforcement learning is learning what to do and how to map situations to actions. The learner is not told which actions to take. Fig. 1.7.1 shows concept of reinforced learning.

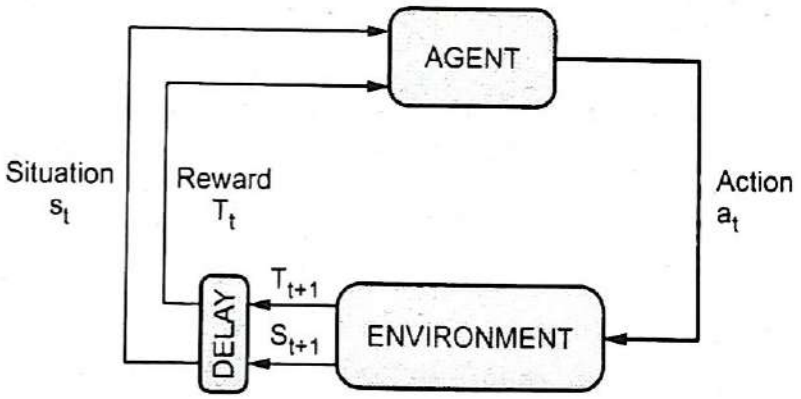


Fig. 1.7.1 Reinforced learning

- Reinforced learning is deals with agents that must sense and act upon their environment. It combines classical Artificial Intelligence and machine learning techniques.
- It allows machines and software agents to automatically determine the ideal behavior within a specific context, in order to maximize its performance. Simple reward feedback is required for the agent to learn its behavior; this is known as the reinforcement signal.
- Two most important distinguishing features of reinforcement learning is trial-and-error and delayed reward.

- With reinforcement learning algorithms an agent can improve its performance by using the feedback it gets from the environment. This environmental feedback is called the reward signal.
- Based on accumulated experience, the agent needs to learn which action to take in a given situation in order to obtain a desired long term goal. Essentially actions that lead to long term rewards need to reinforced. Reinforcement learning has connections with control theory, Markov decision processes and game theory.

1.7.1 Elements of Reinforcement Learning

- Reinforcement learning elements are as follows :
  1. Policy 2. Reward function
  3. Value function 4. Model of the environment
- Fig. 1.7.2 shows elements of RL.

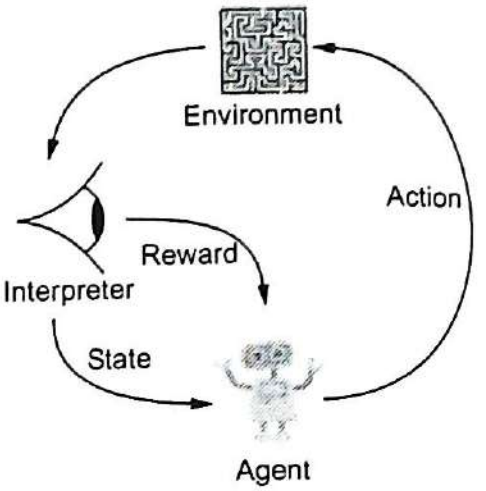


Fig. 1.7.2 : Elements of reinforcement learning

- **Policy** : Policy defines the learning agent behavior for given time period. It is a mapping from perceived states of the environment to actions to be taken when in those states.
- **Reward function** : Reward function is used to define a goal in a reinforcement learning problem. It also maps each perceived state of the environment to a single number.
- **Value function** : Value functions specify what is good in the long run. The value of a state is the total amount of reward an agent can expect to accumulate over the future, starting from that state.
- **Model of the environment** : Models are used for planning.
- **Credit assignment problem** : Reinforcement learning algorithms learn to generate an internal value for the intermediate states as to how good they are in leading to the goal.
- The learning decision maker is called the agent. The agent interacts with the environment that includes everything outside the agent.
- The agent has sensors to decide on its state in the environment and takes an action that modifies its state.
- The reinforcement learning problem model is an agent continuously interacting with an environment. The agent and the environment interact in a sequence of time steps. At each time step t, the agent receives the state of the environment and a scalar numerical reward for the previous action, and then the agent then selects an action.

- Reinforcement learning is a technique for solving Markov decision problems.
- Reinforcement learning uses a formal framework defining the interaction between a learning agent and its environment in terms of states, actions, and rewards. This framework is intended to be a simple way of representing essential features of the artificial intelligence problem.

1.7.2 Application of Reinforcement Learning

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.

1.7.3 Advantages and Disadvantages of Reinforcement Learning

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

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

Review Question

1. *Discuss the reinforcement learning and write the brief applications.*

SPPU : March-20, In Sem, Marks 5

1.8 Models of Machine Learning

- A machine learning model is a program that can find patterns or make decisions from a previously unseen dataset. For example, in natural language processing, machine learning models can parse and correctly recognize the intent behind previously unheard sentences or combinations of words.

- A machine learning model can perform such tasks by having it 'trained' with a large dataset. During training, the machine learning algorithm is optimized to find certain patterns or outputs from the dataset, depending on the task. The output of this process - often a computer program with specific rules and data structures - is called a machine learn.
- For classification and regression problem, there are different choices of Machine Learning Models each of which can be viewed as a black box that solve the same problem. However, each model come from a different algorithm approaches and will perform differently under different data set. The best way is to use cross - validation to determine which model performs best on test data.
- The model - based approach seeks to create a modified solution tailored to each new application. Instead of having to transform user problem to fit some standard algorithm, in model-based machine learning user design the algorithm precisely to fit problem.
- The core idea at the heart of model - based machine learning is that all the assumptions about the problem domain are made explicit in the form of a model.
- Model is just made up of set of assumptions, expressed in a precise mathematical form. These assumptions include the number and types of variables in the problem domain, which variables affect each othe and what the effect of changing one variable is on another variable.
- Machine learning models are classified as :
  1. Geometric model (Using the Geometry of the instance space)
  2. Probabilistic model (Using Probability to classify the instance space)
  3. Logical model (Using a Logical expression)

1.8.1 Geometric Model

- Here, we consider models that define similarity by considering the geometry of the instance space. In Geometric models, features could be described as points in two dimensions (x - and y - axis) or a three-dimensional space (x, y, and z).
- Geometric models are constructed directly in instance space, using geometric concepts like lines, planes and distances.
- Even when features are not intrinsically geometric, they could be modelled in a geometric manner (for example, temperature as a function of time can be modelled in two axes).
- This models use intuitions from geometry such as separating hyper planes, linear transformations and distance metric. The main goal of this method is to find a set of representative features of geometric form to represent an object by collecting

geometric features from images and learning them using efficient machine learning methods.

- In geometric models, there are two ways we could impose similarity. We could use geometric concepts like lines or planes to segment (classify) the instance space. These are called **linear models.**
- Linear models are parametric, which means that they have a fixed form with a small number of numeric parameters that need to be learned from data. Linear models have low variance and high bias. This implies that linear models are less likely to overfit the training data than some other models.
- In other method, we can use the geometric notion of distance to represent similarity. In this case, if two points are close together, they have similar values for features and thus can be classed as similar. We call such models as **Distance - based models.**
- Examples of distance - based models include the nearest - neighbour models, which use the training data.
- Geometric learning methods can not only solve recognition problems but also predict subsequent actions by analyzing a set of sequential input sensory images, usually some extracting features of images.
- Example of Geometric models : K - nearest neighbors, linear regression, support vector machine, logistic regression.
- Geometric features :
  1. Corners : Corners is a very simple but significant feature of objects. Especially, Complex objects usually have different corner features with each other. Corners of an object can be extracted by using the technique, calling corner detection.
  2. Edges : Edges are one-dimensional structure features of an image. They represent the boundary of different image regions. The outline of an object can be easily detected by finding the edge using the technique of edge detection.
  3. Blobs : Blobs represent regions of images, which can be detected using blob detection method.
  4. Ridges : From a practical viewpoint, a ridge can be thought of as a one - dimensional curve that represents an axis of symmetry, i.e. Ridges detection method.

1.8.2 Probabilistic Models

- Probabilistic models view learning as a process of reducing uncertainty, modeled by means probability distributions. A model describes data that one could observe

from a system. If we use the mathematics of probability theory to express all forms of uncertainty and noise associated with our model.

- Probabilistic models see features and target variables as random variables. The process of modelling represents and manipulates the level of uncertainty with respect to these variables.
- Fig. 1.8.1 shows Probabilistic logic learning.

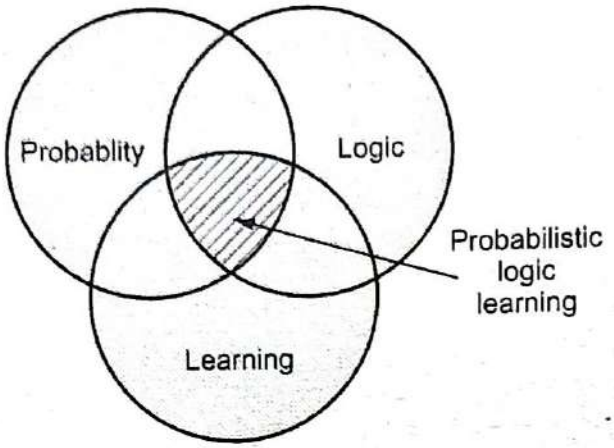


Fig. 1.8.1 Probabilistic logic learning as the intersection of probability logic and learning

- There are two types of probabilistic models : Predictive and Generative.
- Predictive probability models use the idea of a conditional probability distribution P (Y |X) from which Y can be predicted from X. Generative models estimate the joint distribution P (Y, X).
- Once we know the joint distribution for the generative models, we can derive any conditional or marginal distribution involving the same variables. Thus, the generative model is capable of creating new data points and their labels, knowing the joint probability distribution.
- The joint distribution looks for a relationship between two variables. Once this relationship is inferred, it is possible to infer new data points. Naïve Bayes is an example of a probabilistic classifier.
- Example of Probabilistic models : Naïve Bayes, Gaussian process regression, conditional random field.
- Probabilistic modeling is a statistical technique used to take into account the impact of random events or actions in predicting the potential occurrence of future outcomes.
- In machine learning, we train the system by using a limited data set called 'training data' and based on the confidence level of the training data we expect the machine learning algorithm to depict the behaviour of the larger set of actual data.

- Probability theory provides a mathematical foundation for quantifying uncertainty of the knowledge.
- ML is focused on making predictions as accurate as possible, while traditional statistical models are aimed at inferring relationships between variables.
- We make observations using the sensors in the world. Based on the observations, we intend to make decisions. Given the same observations, the decision should be the same. However, the world changes, observations change, our sensors change, the output should not change.
- We build models for predictions; can we trust them? Are they certain? Many applications of machine learning depend on good estimation of the uncertainty :
  a) Forecasting
  b) Decision making
  c) Learning from limited, noisy, and missing data
  d) Learning complex personalised models
  e) Data compression
  f) Automating scientific modelling, discovery, and experiment design
- A signal is called random if its occurrence can not be predicted. Such signal can not be by any mathematical equation.
- The random signals are represented collectively by a random variable takes its value will be taken at particular time is not known.
- The random variables are analyzed statistically with the help of probability, probability density functions and statistical averages such as mean, variance etc.

Relative frequency : For event 'A' relative frequency is defined as,

$$\text{Relative frequency} = \frac{\text{Number of times event occurs } (N_A)}{\text{Total number of trials}} = \frac{N_A}{N}$$

As number of trials approach infinity, relative frequency is called probability.

Probability of event 'A' is defined as the ratio of number of possible favourable outcomes to total number of outcomes i.e.,

$$\text{Probability; } P(A) = \lim_{N \to \infty} \frac{N_A}{N}$$

$$= \frac{\text{Number of possible favourable outcomes}}{\text{Total number of outcomes}}$$

Permutations and Combinations

Combination of 'n' taken 'r' at a time, ${}^{n}C_{r} = \frac{n!}{(n-r)!\,r!}$

Permutation of 'n' taken 'r' at a time, ${}^{n}P_{r} = \frac{n!}{(n-r)!}$

1.8.2.1 Naive Bayes Classifier

- Naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong independence assumptions between the features. It is highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem.
- A Naive Bayes Classifier is a program which predicts a class value given a set of attributes.
- For each known class value,
  1. Calculate probabilities for each attribute, conditional on the class value.
  2. Use the product rule to obtain a joint conditional probability for the attributes.
  3. Use Bayes rule to derive conditional probabilities for the class variable.
- Once this has been done for all class values, output the class with the highest probability.
- Naive bayes simplifies the calculation of probabilities by assuming that the probability of each attribute belonging to a given class value is independent of all other attributes. This is a strong assumption but results in a fast and effective method.
- The probability of a class value given a value of an attribute is called the conditional probability. By multiplying the conditional probabilities together for each attribute for a given class value, we have a probability of a data instance belonging to that class.

Conditional Probability

- Let A and B be two events such that P(A) > 0. We denote P(B | A) the probability of B given that A has occurred. Since A is known to have occurred, it becomes the new sample space replacing the original S. From this, the definition is ,

$$P(B/A) \equiv \frac{P(A \cap B)}{P(A)}$$

OR

$$P(A \cap B) = P(A)\,P(B/A)$$

- The notation P(B | A) is read **"the probability of event B given event A"**. It is the probability of an event B given the occurrence of the event A.
- We say that, the probability that both A and B occur is equal to the probability that A occurs times the probability that B occurs given that A has occurred. We call **P(B | A) the conditional probability of B given A**, i.e., the probability that B will occur given that A has occurred.
- Similarly, the conditional probability of an event A, given B by,

$$P(A/B) \equiv \frac{P(A \cap B)}{P(B)}$$

- The probability P(A | B) simply reflects the fact that the probability of an event A may depend on a second event B. If A and B are mutually exclusive $A \cap B = \phi$ and P(A | B) = 0.
- Another way to look at the conditional probability formula is :

$$P(\text{Second}/\text{First}) = \frac{P\ (\text{First choice and second choice})}{P\ (\text{First choice})}$$

- Conditional probability is a defined quantity and cannot be proven.
- The key to solving conditional probability problems is to :
  1. Define the events.
  2. Express the given information and question in probability notation.
  3. Apply the formula.

Joint Probability

- A joint probability is a probability that measures the likelihood that two or more events will happen concurrently.
- If there are two independent events A and B, the probability that A and B will occur is found by multiplying the two probabilities. Thus for two events A and B, the special rule of multiplication shown symbolically is :

  P(A and B) = P(A) P(B).
- The general rule of multiplication is used to find the joint probability that two events will occur. Symbolically, the general rule of multiplication is,

  P(A and B) = P(A) P(B | A).
- The probability $P(A \cap B)$ is called the joint probability for two events A and B which intersect in the sample space. Venn diagram will readily shows that

$$P(A \cap B) = P(A) + P(B) - P\ (A \cup B)$$

Equivalently :

$$P(A \cap B) = P(A) + P(B) - P(A \cap B) \leq P(A) + P(B)$$

- The probability of the union of two events never exceeds the sum of the event probabilities.
- A tree diagram is very useful for portraying conditional and joint probabilities. A tree diagram portrays outcomes that are mutually exclusive.

Bayes Theorem

- Bayes' theorem is a method to revise the probability of an event given additional information. Bayes's theorem calculates a conditional probability called a posterior or revised probability.
- Bayes' theorem is a result in probability theory that relates conditional probabilities. If A and B denote two events, $P(A \mid B)$ denotes the conditional probability of A occurring, given that B occurs. The two conditional probabilities $P(A \mid B)$ and $P(B \mid A)$ are in general different.
- Bayes theorem gives a relation between $P(A \mid B)$ and $P(B \mid A)$. An important application of Bayes' theorem is that it gives a rule how to update or revise the strengths of evidence-based beliefs in light of new evidence a posteriori.
- A **prior probability** is an initial probability value originally obtained before any additional information is obtained.
- A **posterior probability** is a probability value that has been revised by using additional information that is later obtained.
- Suppose that $B_1, B_2, B_3 \ldots B_n$ partition the outcomes of an experiment and that A is another event. For any number, k, with $1 \leq k \leq n$, we have the formula :

$$P(B_k/A) = \frac{P(A/B_k) \cdot P(B_k)}{\sum_{i=1}^{n} P(A/B_i) \cdot P(B_i)}$$

1.8.3 Logical Models

- Logical models are defined in terms of easily interpretable logical expansions. Logical models use a logical expression to divide the instance space into segments and hence construct grouping models.
- A logical expression is an expression that returns a Boolean value, i.e., a True or False outcome. Models involving logical statements easily translated into human - understandable rules.

- Once the data is grouped using a logical expression, the data is divided into homogeneous groupings for the problem we are trying to solve. For example, for a classification problem, all the instances in the group belong to one class.
- There are mainly two kinds of logical models : Tree models and Rule models.
- Rule models consist of a collection of implications or IF-THEN rules. For tree - based models, the 'if - part' defines a segment and the 'then - part' defines the behavior of the model for this segment. Rule models follow the same reasoning.
- Example of Logical models : Decision tree, random forest.

1.9 Distance-based Models

SPPU : Dec.-19

- Learning a good distance metric in feature space is crucial in real-world application. Good distance metrics are important to many computer vision tasks, such as image classification and content-based image retrieval.
- The arithmetic and geometric means, usually used to average a finite set of positive numbers, generalize naturally to a finite set of Symmetric Positive-Definite Matrices. This generalization is based on the key observation that a mean has a various characterization.
- The **arithmetic mean** minimizes the sum of the squared Euclidean distances to given positive numbers. The **geometric mean** minimizes the sum of the squared hyperbolic distances to the given positive numbers.
- Examples of distance-based algorithms are Hierarchical Agglomerative Clustering (HAC) and K-nearest neighbor algorithm (KNN) for prediction. These algorithm works on arbitrary types of structured data. They require a distance function on the underlying data type.
- The distance calculation on complex/structured types requires three types of functions :
  1. A function to generate pairs of objects of the simpler constitutive types i.e. pairing function.
  2. Distance functions on the simpler types.
  3. An aggregation function that is applied to the distance values obtained from above steps.
- Depending on the availability of the training examples, algorithms for distance metric learning can be divided into two categories : **Supervised distance metric learning and unsupervised distance metric learning.**
- Unlike most supervised learning algorithms where training examples are given class labels, the training examples of supervised distance metric learning is cast into pair wise constraints : The equivalence constraints where pairs of data points

that belong to the same classes and in-equivalence constraints where pairs of data points belong to different classes.

- The supervised, distance metric learning can be further divided into two categories : the global distance metric learning and the local distance metric learning. The first one learns the distance metric in a global sense, i.e., to satisfy all the pair-wise constraints simultaneously. The second approach is to learn a distance metric in a local setting, i.e., only to satisfy local pair-wise constraints.
- The main ingredients of distance-based models are distance metrics, which can be Euclidean, Manhattan, Minkowski or Mahalanobis, among many others.

1.9.1 Euclidean Distance

- The Euclidean distance is the most common distance metric used in low dimensional data sets. It is also knows the L_2 norm. The Euclidean distance is the ususal manner in which distance is measured in real world.

  $$d_{euclidean}(x, y) = \sqrt{\sum i(x_i - y_i)^2}$$

  where x and y are m-dimensional vectors and denoted by $x = (x_1, x_2, x_3, \ldots, x_m)$ and $y = (y_1, y_2, y_3, \ldots, y_m)$ represent the m attribute values of two records.
- While Euclidean metric is useful in low dimensions, it doesn't work well in high dimensions and for categorical variables. The drawback of Euclidean distance is that it ignores the similarity between attributes. Each attributes is treated as totally different from all of the attributes.

1.9.2 Manhattan

- Mahalanobis distance is also called **quadratic distance.**
- Mahalanobis distance is a distance measure between two points in the space defined by two or more correlated variables. Mahalanobis distance takes the correlations within a data set between the variableinto considerations.
- If there are two non-correlated variables, the Mahalanobis distance between the points of the variables in a 2D scatter plot is same as Euclidean distance.
- The Mahalanobis distance is the distance between an observation and the center for each group in m-dimensional space defined by m variables and their covariance. Thus, a small value of Mahalanobis distance increases the chance of an observation to be closer to the group's center and the more likely it is to be assigned to that group.
- Mahalanobis distance between two samples (x, y) of a random variable is defined as,

$$d_{Mahalanobis}(x, y) = \sqrt{(x-y)^T \sum^{-1} (x-y)}$$

- The Mahalanobis metric is defined in independence of the data matrix.
- No pre - processing of labeled data samples is needed before using KNN algorithm. A dominated class label in K-nearest neighbors of a data point is assigned as class label to that the point. A tie occurs when neighborhood has same amount of labels from multiple classes.
- To break the tie, the distances of neighbors can be summed up in each class that is tied and vector f is assigned to the class with minimal distance. Or, the class can be chosen with the nearest neighbor. Clearly, tie is still possible here, in which case an arbitrary assignment is taken.
- Mahalanobis distance that takes into account the correlation S of the dataset :

$$L_m(x, y) = \sqrt{(x-y)S^{-1}(x-y)}$$

1.9.3 Hamming Distance

- Hamming bits are inserted into the message at the random locations. Hamming code is a single error correcting code. It is most complex from the stand point of creating and interpreting the error bits. Let us consider a frame which consists of m data bits and r check bits. The total length of message is then n = m + r. An n-bit unit containing data and checkbits is often referred to as an **n-bit codeword.**
- If 10001001 and 10110001 are two codewords, then the corresponding bits differ in these two codewrods is 3 bits. The number of bit positions in which two codewords differ is called the **hamming distance.**
- If two codewords are a hamming distance d apart, it will require d single bit errors to convert one into the other.
- Determining the placement and binary value of the hamming bits can be implemented using hardware, but it is often more practical to implement them using software.
- The number of bits in the message are counted and used to determine the number of hamming bits to be used. The equation is used to count the number of hamming bits.

$$2^H \geq M + H + 1$$

where M = Number of bits in a message

H = Hamming bits

- The parity bits are inserted into the message. Position of the parity bit is calculated as follows. Create a 4 bit binary number $b_4b_3b_2$ and b_1 where

$b_i = 0$ if the parity check for P_i succeeds

$b_i = 1$ otherwise

for i = 1, 2, 3 or 4.

1) The parity bit P_1 is inserted at bit position 1 for even parity for bit positions 1, 3, 5, 7, 9, 10. In these bit positions it contains even number of 0s or 1s.
2) The parity bit P_2 is inserted at bit position 2, for even parity for bit positions 2, 3, 6, 7, 10, 11.
3) The parity bit P_3 is inserted at bit position 4, for even parity for bit positions 4, 5, 6, 7, 12.
4) The parity bit P_4 is inserted at bit position 8, for even parity for bit positions 8, 9, 10, 11, 12.

- For inserting the parity bit even or odd parity can be used. Each parity bit is determined by the data bits it checks. When a receiver gets a transmitted frame, it performs each of the parity checks.
- The combination of failures and successes then determines whether there was no error or in which position an error occurred. Once the receiver knows where the error occurred, it changes the bit value in that position and the error is corrected.

Minimum hamming distance (d_{min}) :

- The minimum hamming distance is the smallest hamming distance between all possible pairs in a set of words.
- To find the value of d_{min}, we find the hamming distances between all words and select the smallest one.

1.9.4 Minkowski Distance Metric

- Minkowski Distance is the generalized form of Euclidean and Manhattan Distance.
- The Minkowski distance between two variabes X and Y is defined as

$$D = \left(\sum_{i=1}^{n} |p_i - q_i|^p \right)^{1/p}$$

- The case where p = 1 is equivalent to the Manhattan distance and the case where p = 2 is equivalent to the Euclidean distance.
- Although p can be any real value, it is typically set to a value between 1 and 2. For values of p less than 1, the formula above does not define a valid distance metric since the triangle inequality is not satisfied.

Review Question

1. *What do you mean by distance metric and exemplar ? Explain different types of distances, measures.* **SPPU : Dec.-19, Marks 9**

1.10 Tree Based Model

SPPU : Dec.-18

1.10.1 Decision Trees

- A decision tree is a simple representation for classifying examples. A decision tree or a classification tree is a tree in which each internal node is labeled with an input feature. The arcs coming from a node labeled with a feature are labeled with each of the possible values of the feature. Each leaf of the tree is labeled with a class or a probability distribution over the classes.
- In this method a set of training examples is broken down into smaller and smaller subsets while at the same time an associated decision tree get incrementally developed. At the end of the learning process, a decision tree covering the training set is returned.
- The key idea is to use a decision tree to partition the data space into cluster (or dense) regions and empty (or sparse) regions.
- Decision tree consists of

1. **Nodes :** Test for the value of a certain attribute.
2. **Edges :** Correspond to the outcome of a test and connect to the next node or leaf.
3. **Leaves :** Terminal nodes that predict the outcome.

- In Decision Tree Learning, a new example is classified by submitting it to a series of tests that determine the class label of the example. These tests are organized in a hierarchical structure called a decision tree.
- **Learn trees in a Top-Down fashion :**

1. Divide the problem in subproblems.
2. Solve each problem.

Basic Divide-and-Conquer Algorithm :

1. Select a test for root node. Create branch for each possible outcome of the test.
2. Split instances into subsets. One for each branch extending from the node.

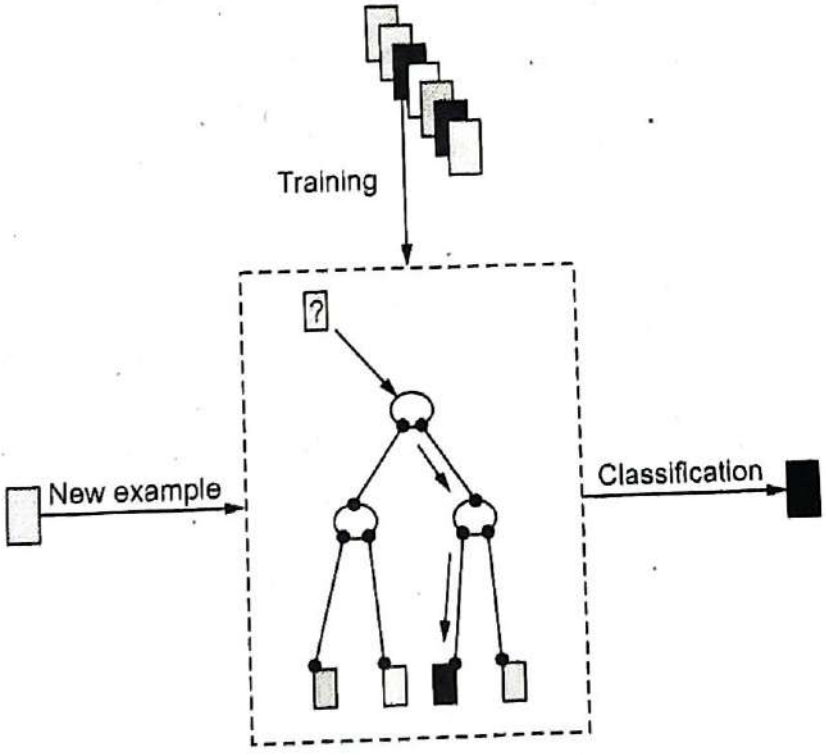


Fig. 1.10.1

3. Repeat recursively for each branch, using only instances that reach the branch.
4. Stop recursion for a branch if all its instances have the same class.

1.10.2 Ranking and Probability Estimation Trees

- Decision trees are one of the most effective and widely used classification methods. Many applications require class probability estimates and Probability Estimation Trees (PET) have the same attractive features as classification trees. But decision trees have been found to provide poor probability estimates.
- A tree is defined as a set of logical conditions on attributes ; a leaf represents the subset of instances corresponding to the conjunction of conditions along its branch or path back to the root. A simple approach to ranking is to estimate the probability of an instance's membership in a class and assign that probability as the instance's rank. A decision tree can easily be used to estimate these probabilities.
- Rule learning is known for its descriptive and therefore comprehensible classification models which also yield good class predictions. In some application areas, we also need good class probability estimates.
- For different classification models, such as decision trees, a variety of techniques for obtaining good probability estimates have been proposed and evaluated.
- In classification rule mining one search for a set of rules that describes the data as accurately as possible. As there are many different generation approaches and types of generated classification rules.
- A probabilistic rule is an extension of a classification rule, which does not only predict a single class value, but a set of class probabilities, which form a probability distribution over the classes. This probability distribution estimates all probabilities that a covered instance belongs to any of the class in the data set, so we get one class probability per class.
- Error rate does not consider the probability of the prediction, so it is consider in PET. Instead of predicting a class, the leaves give a probability. It is very useful when we do not want just the class, but examples most likely to belong to a class. No additional effort in learning PET compared to decision tree.
- Building decision trees with accurate probability estimates, called probability estimation trees. A small tree has a small number of leaves, thus more examples will have the same class probability. That prevents the learning algorithm from building an accurate PET.
- On the other hand, if the tree is large, not only may the tree overfit the training data, but the number of examples in each leaf is also small and thus the probability estimates would not be accurate and reliable. Such a contradiction does exist in traditional decision trees.

- Decision trees acting as probability estimators, however, are often observed to produce bad probability estimates. There are two types of probability estimation trees - a single tree estimator and an ensemble of multiple trees.
- Applying a learned PET involves minimal computational effort, which makes the tree-based approach particularly suited for a fast reranking of large candidate sets.
- For simplicity, all attributes are assumed to be numeric. For n attributes, each input datum is then given by an n-tuple $X = (x_1, \ldots, x_n) \in R^n$
- Let $X = \{x^{(1)}, \ldots, x^{(R)}\} \subset R^n$ be the set of training items.
- A probability estimation tree is introduced as a binary tree T with $s \geq 0$ inner nodes $D^T = \{d_1, d_2, \ldots, d_s\}$ and leaf nodes $E^T = \{e_0, e_1, \ldots, e_s\}$ with $E^T \cap D^T = \varnothing$ Each inner node $d_i, i \in \{1, 2, \ldots, s\}$ is labeled by an attribute $\alpha_i^T \in \{1, \ldots, n\}$, while each leaf node $e_j, j \in \{1, 2, \ldots, s\}$ is labeled by a probability $p_j^T \in [0, 1]$.
- The arcs in A^T correspond to conditions on the inputs. Since it is a binary tree and every inner node has exactly two children. By splitting inputs at each decision node until a leaf is reached, the PET partitions the input space into n-dimensional cartesian blocks :

$$H_j^T = \prod_{\alpha=1}^{n} h(l_{j,\alpha}^T, u_{j,\alpha}^T)$$

1.10.3 Regression Tree

- Regression tree models are known for their simplicity and efficiency when dealing with domains with large number of variables and cases. Regression trees are obtained using a fast divide and conquer greedy algorithm that recursively partitions the given training data into smaller subsets.
- When the complexity of the model is dependant on the learning sample size, both bias and variance decrease with the learning sample size. E.g. regression trees. Small bias, a tree can approximate any non linear function.
- Regression trees are among the machine learning method that present the highest variance. Even a small change of the learning sample can result in a very different tree. Even small trees have a high variance.
- Possible sources of variance :
  1. **Discretization of numerical attributes :** The selected threshold has a high variance.
  2. **Structure choice :** Sometimes, attribute scores are very close.
  3. **Estimation at leaf nodes :** Because of the recursive partitioning, prediction at leaf nodes are based on very small samples of objects.

- Regression trees are constructed using a Recursive Partitioning (RP) algorithm. This algorithm builds a tree by recursively splitting the training sample into smaller subsets. The RP algorithm receives as input a set of n data points and if certain termination criteria are not met it generates a test node t, whose branches are obtained by applying the same algorithm with two subsets of the input data points.
- At each node the best split test is chosen according to some local criterion which means that this is a greedy hill-climbing algorithm.

Algorithm : Recursive Partitioning Algorithm

Input : A set of n datapoints

Output : A regression tree

IF termination criterion THEN

Create Leaf Node and assign it a Constant Value

Return Leaf Node

ELSE

*Find Best Splitting Test s**

*Create Node t with s**

Left_branch (t) = RecursivePartitioningAlgorithm $(\{< x_i, y_i > : x_i \rightarrow s^*\})$

Right_branch (t) = RecursivePartitioningAlgorithm $(\{< x_i, y_i > : x_i \rightarrow s^*\})$

Return Node t

ENDIF

- The algorithm has three main components :
  1. A way to select a split test
  2. A rule to determine when a tree node is terminal.
  3. A rule for assigning a value to each terminal node.

1.10.4 Impurity Measures - Gini Index and Entropy

- One of the decision tree algorithms is CART (Classification and Regression Tree).
- Classification Tree : When decision or target variable is categorical, the decision tree is classification decision tree.
- Regression Tree : When the decision or target variable is continuous variable, the decision tree is called regression decision tree.
- CART algorithm can be used for building both Classification and Regression Decision Trees. The impurity measure used in building decision tree in CART is Gini Index. The decision tree built by CART algorithm is always a binary decision tree.

- Gini impurity is a measure of how often a randomly chosen element from the set would be incorrectly labeled if it was randomly labeled according to the distribution of labels in the subset.
- Gini index, entropy and twoing rule are some of the frequently used impurity measures.
- Gini Index for a given node t :

$$GINI(t) = \sum_j p(j|t)(1-p(j|t)) = \sum_j p(j|t)^2$$

Maximum of $1-1/n_c$ (number of classes) when records are equally distributed among all classes = Maximal impurity.

- Minimum of 0 when all records belong to one class = Complete purity.
- Entropy at a given node by :

$$\text{Entropy } (t) = \sum_j p(j|t) \log p(j|t)$$

- Maximum ($\log n_c$) when records are equally distributed among all classes(maximal impurity).
- Minimum (0.0) when all records belongs to one class (maximal purity).
- Entropy is the only function that satisfies all of the following three properties
  1. When node is pure, measure should be zero
  2. When impurity is maximal (i.e. all classes equally likely), measure should be maximal
  3. Measure should obey multistage property
- When a node p is split into k partitions (children), the quality of the split is computed as a weighted sum :

$$GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i) = \sum_j p(j|t)^2$$

where n_i = number of records at child i, and n = number of records at node p.

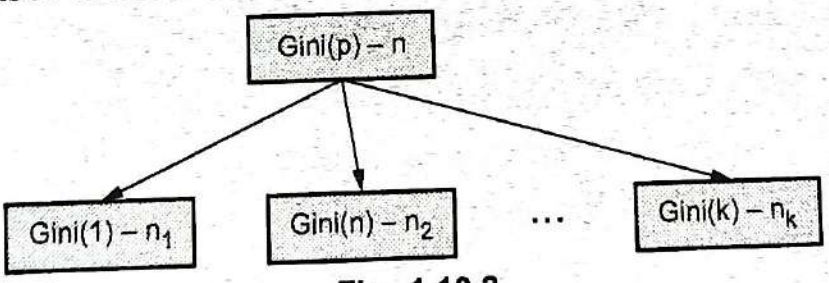


Fig. 1.10.2

- A problem with all impurity measures is that they depend only on the number of (training) patterns of different classes on either side of the hyperplane. Thus, if we change the class regions without changing the effective areas of class regions on either side of a hyperplane, the impurity measure of the hyperplane will not change.

- Thus the impurity measures do not really capture the geometric structure of class distributions. Also, all the algorithms need to optimize on some average of impurity of the child nodes and often it is not clear what kind of average is proper.

1.10.5 Feature Tree

- A feature tree is a tree such that each internal node is labelled with a feature, and each edge emanating from an internal node is labelled with a literal. The set of literals at a node is called a split.
- Each leaf of the tree represents a logical expression, which is the conjunction of literals encountered on the path from the root of the tree to the leaf. The extension of that conjunction is called the instance space segment associated with the leaf.
- Algorithm GrowTree(D, F)

  Input : data D; set of features F .

  Output : feature tree T with labelled leaves.

  1. if Homogeneous(D) then return Label(D);
  2. S ← BestSplit(D, F) ;
  3. split D into subsets D_i according to the literals in S ;
  4. for each i do
  5. if $D_i \neq \phi$; then $T_i \leftarrow$ GrowTree(D_i, F) ;
  6. else T_i is a leaf labelled with Label(D);
  7. end
  8. return a tree whose root is labeled with S and whose children are T_i.
- Algorithm gives the generic learning procedure common to most tree learners. It assumes that the following three functions are defined :
  1. Homogeneous(D) returns true if the instances in D are homogeneous enough to be labelled with a single label, and false otherwise;
  2. Label(D) returns the most appropriate label for a set of instances D;
  3. BestSplit(D, F) returns the best set of literals to be put at the root of the tree.
- These functions depend on the task at hand : for instance, for classification tasks a set of instances is homogeneous if they are of a single class, and the most appropriate label would be the majority class. For clustering tasks a set of instances is homogenous if they are close together, and the most appropriate label would be some exemplar such as the mean.

1.10.6 Information Gain and Entropy

- Entropy measures the impurity of a collection. Information gain is defined in terms of entropy.
- Information gain tells us how important a given attribute of the feature vectors is,
- Information gain of attribute A is the reduction in entropy caused by partitioning the set of examples S.

$$\text{Gain}(S, A) \equiv \text{Entropy}(S) - \sum_{v \in \text{values}} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

where values (A) is the set of all possible values for attributes A and S_v is the subset of S for which attribute A has value v.

Pruning by information gain :

- The simplest technique is to prune out portions of the tree that result in the least information gain.
- This procedure does not require any additional data and only bases the pruning on the information that is already computed when the tree is being built from training data.
- The process of information gain based pruning required us to identify "twigs", nodes whose children are all leaves.
- "Pruning" a twig removes all of the leaves which are the children of the twig and makes the twig a leaf.
- The algorithm for pruning is as follows :
  1. Catalog all twigs in the tree.
  2. Count the total number of leaves in the tree.
  3. While the number of leaves in the tree exceeds the desired number :
     a) Find the twig with the least information gain
     b) Remove all child nodes of the twig
     c) Relabel twig as a leaf
     d) Update the leaf count.

1.10.7 Advantages and Disadvantages of Decision Tree

Advantages :

1. Rules are simple and easy to understand.
2. Decision trees can handle both nominal and numerical attributes.
3. Decision trees are capable of handling datasets that may have errors.

4. Decision trees are capable of handling datasets that may have missing values.
5. Decision trees are considered to be a nonparametric method.
6. Decision trees are self - explantory.

Disadvantages :

1. Most of the algorithms require that the target attribute will have only discrete values.
2. Some problem are difficult to solve like XOR.
3. Decision trees are less appropriate for estimation tasks where the goal is to predict the value of a continuous attribute.
4. Decision trees are prone to errors in classification problems with many class and relatively small number of training examples.

1.10.8 Tree Pruning

- If the classifier fits the training instances too closely, it may fit noisy instances and that reduces its usefulness. This phenomenon is called overfitting.
- Pruning simplifies a classifier by merging disjuncts that are adjacent in instance space. This can improve the classier's performance by eliminating error-prone components.
- Pruning of the decision tree is done by replacing a whole sub-tree by a leaf node. The replacement takes place if a decision rule establishes that the expected error rate in the sub-tree is greater than in the single leaf.
- For example :

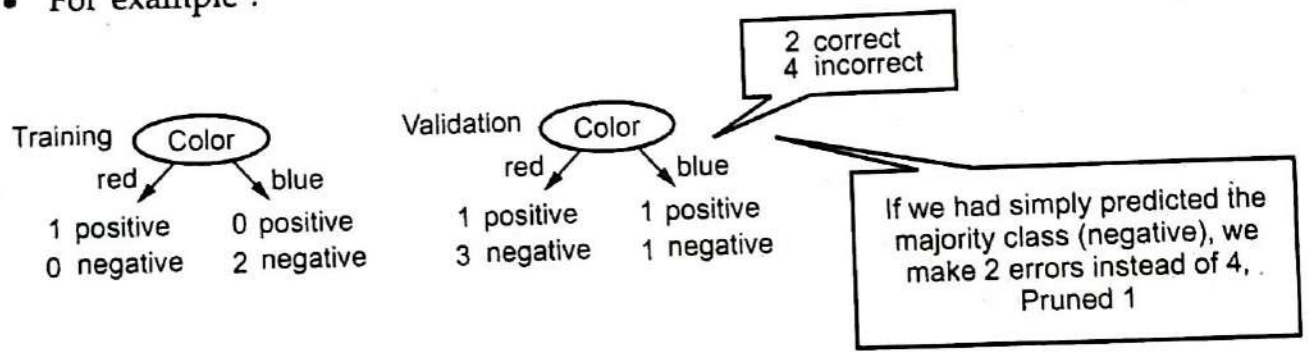


Fig. 1.10.3

1.10.9 ID3

- ID3 stands for Iterative Dichotomiser 3. This algorithm used to generate a decision tree. ID3 uses Entropy function and Information gain as metrics.
- The ID3 follows the Occam's razor principle. It attempts to create the smallest possible decision tree.
- The calculation for information gain is the most difficult part of this algorithm.

- ID3 performs a search whereby the search states are decision trees and the operator involves adding a node to an existing tree. It uses information gain to measure the attribute to put in each node, and performs a greedy search using this measure of worth.
- The algorithm goes as follows : Given a set of examples (S), categorised in categories c_i, then :
  1. Choose the root node to be the attribute, A, which scores the highest for information gain relative to S.
  2. For each value v that A can possibly take, draw a branch from the node.
  3. For each branch from A corresponding to value v, calculate S_v. Then :
     i. If S_v is empty, choose the category c default which contains the most examples from S, and put this as the leaf node category which ends that branch.
     ii. If S_v contains only examples from a category c, then put c as the leaf node category which ends that branch.
     iii. Otherwise, remove A from the set of attributes which can be put into nodes. Then put a new node in the decision tree, where the new attribute being tested in the node is the one which scores highest for information gain relative to S_v.

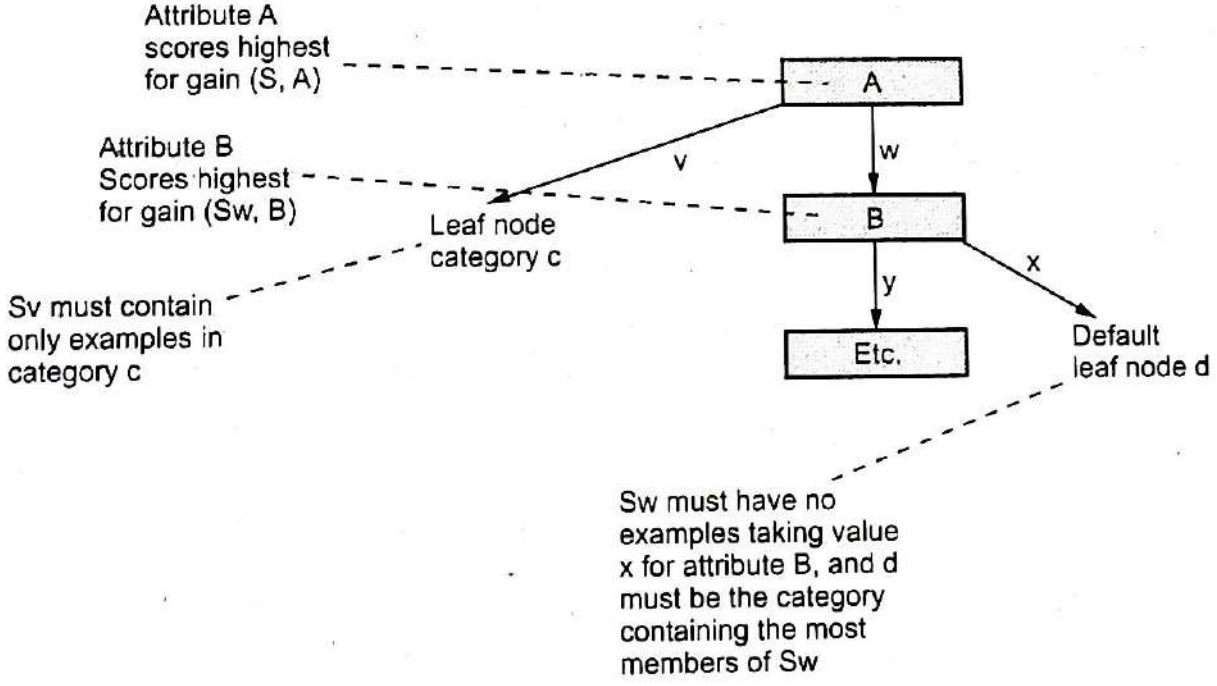


Fig. 1.10.4

- The ID3 algorithm is a classic data mining algorithm for classifying instances. The input is a set of training data for building a decision tree.
- By applying the ID3 algorithm, a decision tree is created. To create the decision tree, we have to choose a target attribute.

- Take all unused attributes and calculates their entropies. Chooses attribute that has the lowest entropy is minimum or when information gain is maximum. Makes a node containing that attribute.

Capabilities and Limitations of ID3

1. Hypothesis space is a complete space of all discrete valued functions.
2. Cannot determine how many alternative trees are consistent with training data.
3. ID3 in its pure form performs no backtracking.
4. ID3 uses all training examples at each step to make statistically based decisions regarding how to refine its current hypothesis.

Example 1.10.1 *If S is a collection of 14 examples with 9 YES and 5 NO examples then calculate entropy.*

Solution :

$$\text{Entropy(S)} = \Sigma - p(I) \log_2 p(I)$$

Where p(I) is the proportion of S belonging to class I.

Σ is over c.

$$\text{Entropy(S)} = -\left(\frac{9}{14}\right)\log_2\left(\frac{9}{14}\right) - \left(\frac{5}{14}\right)\log_2\left(\frac{5}{14}\right) = \mathbf{-0.940}$$

Example 1.10.2 *Consider the following table :*

Weekend (Example)	Wheather	Parents	Money	Decision (Category)
W1	Sunny	Yes	Rich	Cinema
W2	Sunny	No	Rich	Tennis
W3	Windy	Yes	Rich	Cinema
W4	Rainy	Yes	Poor	Cinema
W5	Rainy	No	Rich	Stay in
W6	Rainy	Yes	Poor	Cinema
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W9	Windy	Yes	Rich	Cinema
W10	Sunny	No	Rich	Tennis

Calculate Entropy and Gain.

Solution :

$$\text{Entropy}(S) = -p_{\text{cinema}} \log_2(p_{\text{cinema}}) - p_{\text{tennis}} \log_2(p_{\text{tennis}}) - p_{\text{shopping}} \log_2(p_{\text{shopping}}) - p_{\text{stay\_in}} \log_2(p_{\text{stay\_in}})$$

$$=. -(6/10) * \log_2(6/10) - (2/10) * \log_2(2/10) - (1/10) * \log_2(1/10) - (1/10) * \log_2(1/10)$$

$$= -(6/10) * -0.737 - (2/10) * -2.322 - (1/10) * -3.322 - (1/10) * -3.322$$

$$= 04422 + 0.4644 + 0.3322 + 0.3322 = 1.571$$

and we need to determine the best of :

$$\text{Gain(S, weather)} = 1.571 - (\text{IS}_{\text{sun}}\ \text{I}/10) * \text{Entropy}(S_{\text{sun}}) - (\text{IS}_{\text{wind}}\ \text{I}/10) * \text{Entropy}(S_{\text{wind}}) - (\text{IS}_{\text{rain}}\ \text{I}/10) * \text{Entropy}(S_{\text{rain}})$$

$$= 1.571 - (0.3) * \text{Entropy}(S_{\text{sun}}) - (0.4) * \text{Entropy}(S_{\text{wind}}) - (0.3) * \text{Entropy}(S_{\text{rain}})$$

$$= 1.571 - (0.3) * (0.918) - (0.4) * (0.81125) - (0.3) * (0.918) = 0.70$$

$$\text{Gain(S, parents)} = 1.571 - (\text{IS}_{\text{yes}}\ \text{I}/10) * \text{Entropy}(S_{\text{yes}}) - (\text{IS}_{\text{no}}\ \text{I}/10) * \text{Entropy}(S_{\text{no}})$$

$$= 1.571 - (0.5) * 0 - (0.5) * 1.922 = 1.571 - 0.961 = 0.61$$

$$\text{Gain(S, money)} = 1.571 - (\text{IS}_{\text{rich}}\ \text{I}/10) * \text{Entropy}(S_{\text{rich}}) - (\text{IS}_{\text{poor}}\ \text{I}/10) * \text{Entropy}(S_{\text{poor}})$$

$$= 1.571 - (0.7) * (1.842) - (0.3) * 0 = 1.571 - 1.2894 = 0.2816$$

- This means that the first node in the decision tree will be the weather attribute. From the weather node, we draw a branch for the values that weather can take : Sunny, windy and rainy :

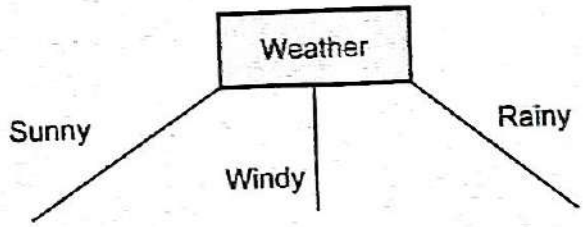


Fig. 1.10.5

- Now we look at the first branch. S_{sunny} = {W1, W2, W10}. This is not empty, so we do not put a default categorization leaf node here.
- The categorisations of W1, W2 and W10 are Cinema, Tennis and Tennis respectively. As these are not all the same, we cannot put a categorisation leaf node here. Hence we put an attribute node here, which we will leave blank for the time being.
- Looking at the second branch, S_{windy} = {W3, W7, W8, W9}. Again, this is not empty and they do not all belong to the same class, so we put an attribute node

here, left blank for now. The same situation happens with the third branch, hence our amended tree looks like this :

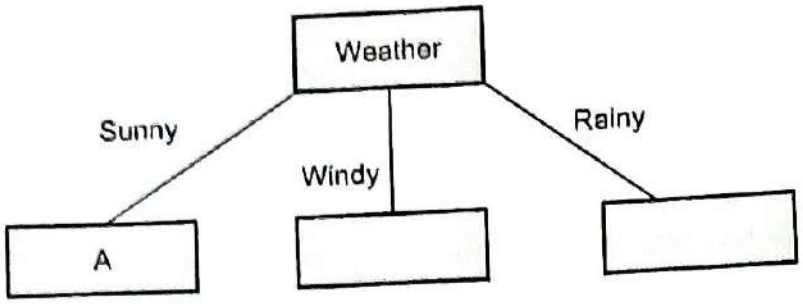


Fig. 1.10.6

- In effect, we are interested only in this part of the table :

Weekend (Example)	Wheather	Parents	Money	Decision (Category)
W1	Sunny	Yes	Rich	Cinema
W2	Sunny	No	Rich	Tennis
W10	Sunny	No	Rich	Tennis

Hence we can calculate :

$$\text{Gain}(S_{\text{sunny}}, \text{parents}) = 0.918 - (IS_{\text{yes}}\ I/ISI) * \text{Entropy}(S_{\text{yes}}) - (IS_{\text{no}}\ I/ISI) * \text{Entropy}(S_{\text{no}})$$

$$= 0.918 - (1/3) * 0 - (2/3) * 0 = 0.918$$

$$\text{Gain}(S_{\text{sunny}}, \text{money}) = 0.918 - (IS_{\text{rich}}\ I/ISI) * \text{Entropy}(S_{\text{rich}}) - (IS_{\text{poor}}\ I/ISI) * \text{Entropy}(S_{\text{poor}})$$

$$= 0.918 - (3/3) * 0.918 - (0/3) * 0 = 0.918 - 0.918 = 0$$

Example 1.10.3 *Consider following splits having four features :*

Length = [3, 4, 5] *[2 +, 0 –] [1+, 3 –] [2+, 2–]*

Gills = [Yes, No] *[0+, 4 –] [5+, 1–]*

Beak = [Yes, No] *[5+, 3 –] [0+, 2–]*

Teeth = [many, few] *[3+, 4 –] [2+, 1–]*

Find : Total weighted entropy and gini-index of all features.

SPPU : Dec.-18, In Sem, Marks 8

Solution :

- Lets calculate the impurity of the first split. We have three segments : the first one is pure and so has entropy 0;
- The second one has entropy

$$= -(1/4)\log_2(1/4) - (3/4)\log_2(3/4) = 0.5 + 0.31$$

$$= 0.81$$; the third one has entropy

- Similar calculations for the other three features give the following entropies :

$$\text{Gills} = (4/10)\times 0 + (6/10)\times(-5/6)\log_2(5/6) - (1/6)\log_2(1/6)) = 0.39$$

$$\text{Beak} = (8/10)\times(-5/8\log_2(5/8) - (3/8)\log_2(3/8)) + (2/10)\times 0 = 0.76$$

$$\text{Teeth} = (7/10)\times(-3/7\log_2(3/7) - (4/7)\log_2(4/7)) + (3\times 10)\times(-2/3)\log_2(2/3) - (1/3)\log_2(1/3)0 = 0.97$$

We thus clearly see that 'Gills' is an excellent feature to split on ; 'Teeth' is poor and the other two are somewhere in between.

1.11 Grouping and Grading Models

- Grading vs grouping is an orthogonal categorization to geometric - probabilistic - logical-compositional model. Difference between grouping and grading models is the way they handle the instance space.

Grouping Model :

- Grouping models break the instance space up into groups or segments and in each segment apply a very simple method. Example : Decision tree, KNN.
- Grouping models have fixed resolution. They cannot distinguish instances beyond a resolution. At the finest resolution, grouping models assign the majority class to all instances that fall into the segment. Find the right segments and label all the objects in that segment.

Grading Model :

- Grading models form one global model over the instance space. They don't use the notion of segment.
- Grading models are usually able to distinguish between arbitrary instances, no matter how similar they are.

1.12 Parametric Models

- Model can be represented using a pre - determined number of parameters. These methods in Machine Learning typically take a model - based approach. We make an assumption there with respect to the form of the function to be guessed. Then we choose an appropriate model based on this assumption correct to estimate the set of parameters.
- The advantage of the parametric approach is that the model is defined up to a small number of parameters, for example mean and variance, the sufficient statistics of the distribution. Once those parameters are estimated from the sample, the whole distribution is known.

- We estimate the parameters of the distribution from the given sample, plug in these estimates to the assumed model and get an estimated distribution, which we then use to make a decision. The method we use to estimate the parameters of a distribution is maximum likelihood estimation.
- Examples of parametric machine learning algorithms are Logistic Regression, Linear Discriminant Analysis, Perceptron, Naive Bayes and Simple Neural Networks.
- **Advantages** :
  1. These methods are simpler and easier to understand.
  2. These models are very rapid to learn from data.
  3. They do not need as much training data.
  4. The methods are well - matched to simpler problems.

1.12.1 Maximum Likelihood Estimation

- Maximum - Likelihood Estimation (MLE) is a method of estimating the parameters of a statistical model. When applied to a data set and given a statistical model, maximum - likelihood estimation provides estimates for the model's parameters. $X_1, X_2, X_3, \cdots, X_n$ have joint density denoted $f_\theta(x_1, x_2, \cdots, x_n) = f(x_1, x_2, \cdots, x_n|\theta)$. Given observed values $X_1 = X_1, x_2 = x_2, \cdots, X_n = x_n$,

  $$\text{lik}(\theta) = f(x_1, x_2, \cdots, x_n|\theta)$$

Considered as a function of θ.

- If the distribution is discrete, f will be the frequency distribution function.
- The maximum likelihood estimate of θ is that value of that maximises lik(θ) : It is the value that makes the observed data the most probable.

Examples of maximizing likelihood :

- A random variable with this distribution is a formalization of a coin toss. The value of the random variable is 1 with probability θ and 0 with probability $1 - \theta$. Let X be a Bernoulli random variable and let x be an outcome of X, then we have

  $$P(X = x) = \begin{bmatrix} \theta & \text{if } x = 1 \\ 1-\theta & \text{if } x = 0 \end{bmatrix}$$

- Usually, we use the notation P(.) for a probability mass and the notation P(.) for a probability density. For mathematical convenience write P(X) as

  $$P(X = x) = \theta^x (1-\theta)^{1-x}$$

1.13 Nonparametric Methods

- Size of the model depends on data, cannot be represented using a pre-determined number of parameters.
- Instead, non-parametric methods state to a set of algorithms. That does not make any primary assumptions with respect to the form of the function to be assessed. These methods are accomplished by approximating the unidentified function f that could be of any form.
- In machine learning, nonparametric methods are also called instance - based or memory - based learning algorithms.
- Density estimation is the problem of reconstructing the probability density function using a set of given data points. Namely, we observe $X_1;\ldots;X_n$ and we want to recover the underlying probability density function generating our dataset.
- Here we discuss, histogram methods for density estimation. A histogram is a chart that plots the distribution of a numeric variable's values as a series of bars. Each bar typically covers a range of numeric values called a bin or class; a bar's height indicates the frequency of data points with a value within the corresponding bin.
- For simplicity, we assume that $X_i \in [0; 1]$, so p(x) is non - zero only within [0; 1]. We also assume that p(x) is smooth and $|p'(X)| \leq L$ for all x. The histogram is to partition the set [0; 1] into several bins and using the count of the bin as a density estimate.
- When we have M bins, this yields a partition

$$B_1 = \left[0, \frac{1}{M}\right), B_2 = \left[\frac{1}{M}, \frac{2}{M}\right), \ldots, B_{M-1} = \left[\frac{M-2}{M}, \frac{M-1}{M}\right), B_M = \left[\frac{M-1}{M}, 1\right]$$

- In such case, then for a given point $x \in B_\ell$, the density estimator from the histogram will be

$$\hat{p}_n(x) = \frac{\text{Number of observations within } B_\ell}{n} \times \frac{1}{\text{Length of the bin}}$$

$$= \frac{M}{n} \sum_{i=1}^{n} I(X_i \in B_\ell)$$

- The intuition of this density estimator is that the histogram assign equal density value to every points within the bin. So for B_ℓ, that contains x, the ratio of observations within this bin is $\frac{1}{n}\sum_{i=1}^{n} I(X_i \in B_\ell)$ which should be equal to the density estimate times the length of the bin.
- Non - parametric methods lean towards additional precision because they try to find the best fit for the data points. Though, this approaches at the cost of needing

a very huge amount of observations. That is desired so as to approximate the unidentified function (f) exactly.

- Non - parametric methods can occasionally present overfitting. They can on occasion learn the errors and noise in a way that they cannot simplify well to new, unseen data points as these methods tend to be more flexible.
- Examples of non - parametric methods are k-Nearest Neighbors, Decision Trees like CART and C4.5, Support Vector Machines.
- **Advantages of nonparametric methods :**

1. Accomplished in fitting a huge number of functional forms.
2. There are no assumptions about the original function.
3. They may outcome in higher performance models for prediction.

Limitations of nonparametric methods

1. They require a lot more training data to estimate the mapping function.
2. Overfitting : Extra risk to overfit the training data.

1.13.1 Difference between Non-parametric Methods and Parametric Methods

Sr. No.	Non-parametric method	Parametric methods
1.	Algorithms that do not make particular assumptions about the kind of mapping function are known as non-parametric algorithms.	Parametric model is a learner that summarizes data through a collection of parameters.
2.	Non-parametric analysis to test group medians.	Parametric analysis to test group means.
3.	It can be used on small samples.	Tend to need larger samples.
4.	No information about the population is available.	Information about population is completely known.
5.	It can be used on ordinal and nominal scale data.	Used mainly on interval and ratio scale data.
6.	Not necessarily the samples are independent.	Samples are independent.
7.	K-nearest neighbors is an example of a non - parametric algorithm.	Examples of parametric models include logistic regression and linear SVM.

1.14 Important Elements of Machine Learning

- Machine learning typically follows three phases :
  1. Training : A training set of examples of correct behavior is analyzed and some representation of the newly learnt knowledge is stored. This is some form of rules.
  2. Validation : The rules are checked and, if necessary, additional training is given. Sometimes additional test data are used, but instead, a human expert may validate the rules, or some other automatic knowledge - based component may be used. The role of the tester is often called the opponent.
  3. Application : The rules are used in responding to some new situation

1.14.1 Data Formats

- Supervised learning always use a dataset, defined as a finite set of real vectors with m features.
- In training data, data are assign the labels. In test data, data labels are unknown but not given. The training data consist of a set of training examples.
- The real aim of supervised learning is to do well on test data that is not known during learning. Choosing the values for the parameters that minimize the loss function on the training data is not necessarily the best policy.
- The training error is the mean error over the training sample. The test error is the expected prediction error over an independent test sample.
- **Training set :** A set of examples used for learning, where the target value is known.
- **Test set :** It is used only to assess the performances of a classifier. It is never used during the training process so that the error on the test set provides an unbiased estimate of the generalization error.
- Training data is the knowledge about the data source which we use to construct the classifier.
- Data format is expressed as

$$X = \{\bar{x}_1, \bar{x}_2, \ldots\ldots, \bar{x}_n\}$$

- Machine learning can be summarized as learning a function (f) that maps input variables (X) to output variables (Y).

$$Y = f(x)$$

- An algorithm learns this target mapping function from training data.

- The form of the function is unknown, so our job as machine learning practitioners is to evaluate different machine learning algorithms and see which is better at approximating the underlying function.
- Different algorithms make different assumptions or biases about the form of the function and how it can be learned.
- Parametric : "A learning model that summarizes data with a set of parameters of fixed size 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." Basically it includes normal distribution and other known distributions.
- Non-parametric : "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."
- Parametric : Data are drawn from a probability distribution of specific form up to unknown parameters.
- Nonparametric : Data are drawn from a certain unspecified probability distribution.

1.14.2 Learnability

- Parametric model are of two types : Static structure and a dynamic structure.
- Although the learning task is to determine a hypothesis (h) identical to the target concept cover the entire set of instances (X), the only information available about c is its value over the training examples.
- Inductive learning algorithms can at best guarantee that the output hypothesis fits the target concept over the training data
- Hypothesis representation
(constraints on instance attributes) :

<Sky, AirTemp, Humidity, Wind, Water, Forecast>

1. Any value is acceptable is represented by ?
2. No value is acceptable is represented by Ø

- The goal of a parametric learning process is to find the best hypothesis whose corresponding prediction error is minimum and the residual generalization ability is enough to avoid overfitting.
- Classification is the process of finding a model that describes and distinguishes data classes or concepts, for the purpose of being able to use the model to predict the class of objects whose class label is unknown. The derived model is based on the analysis of a set of training data.

- Even when the data set is linearly separable, there may be many solutions, and which one is found will depend on the initialization of the parameters and on the order of presentation of the data points.
- Furthermore, for data sets that are not linearly separable, the perceptron learning algorithm will never converge.
- Consider two hypotheses :

  h1 = (Sunny, ?, ?, Strong, ?, ?)

  h2 = (Sunny, ?, ?, ?, ?, ?)
- Now consider the sets of instances that are classified positive by hl and by h2. Because h2 imposes fewer constraints on the instance, it classifies more instances as positive.
- In fact, any instance classified positive by hl will also be classified positive by h2. Therefore, we say that h2 is more general than hl.
- One learning method is to determine the most specific hypothesis that matches all the training data

1.14.3 Underfitting and Overfitting

Overfitting

- Training error can be reduced by making the hypothesis more sensitive to training data, but this may lead to overfitting and poor generalization.
- Overfitting occurs when a statistical model describes random error or noise instead of the underlying relationship.
- Overfitting is when a classifier fits the training data too tightly. Such a classifier works well on the training data but not on independent test data. It is a general problem that plagues all machine learning methods.
- Overfitting generally occurs when a model is excessively complex, such as having too many parameters relative to the number of observations.
- We can determine whether a predictive model is underfitting or overfitting the training data by looking at the prediction error on the training data and the evaluation data. Fig. 1.14.1 shows overfiting.

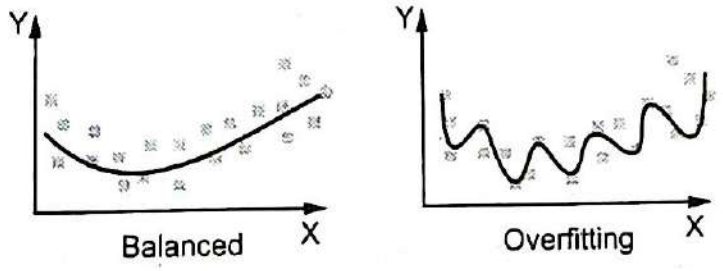


Fig. 1.14.1 : Overfitting

- Reasons for overfitting
  1. Noisy data
  2. Training set is too small
  3. Large number of features
- To prevent over-fitting we have several options :
  1. Restrict the number of adjustable parameters the network has - e.g. by reducing the number of hidden units, or by forcing connections to share the same weight values.
  2. Stop the training early, before it has had time to learn the training data too well.
  3. Add some form of regularization term to the error/cost function to encourage smoother network mappings.
  4. Add noise to the training patterns to smear out the data points
- Often several heuristic are developed in order to avoid overfitting, for example, when designing neural networks one may :
  1. Limit the number of hidden nodes
  2. Stop training early to avoid a perfect explanation of the training set, and
  3. Apply weight decay to limit the size of the weights, and thus of the function class implemented by the network

Underfitting

- Underfitting happens when the learner has not found a solution that fits the observed data to an acceptable level.
- Underfitting : If we put too few variables in the model, leaving out variables that could help explain the response, we are **underfitting**. Consequences :
  1. Fitted model is not good for prediction of new data - prediction is biased
  2. Regression coefficients are biased
  3. Estimate of error variance is too large
- Underfitting examples :
  1. The learning time may be prohibitively large, and the learning stage was prematurely terminated.
  2. The learner did not use a sufficient number of iterations.
  3. The learner tries to fit a straight line to a training set whose examples exhibit a quadratic nature.
- Because of overfitting, low error on training data and high error on test data. Overfitting occurs when a model begins to memorize training data rather than learning to generalize from trend.

- The more difficult a criterion is to predict, the more noise exists in past information that need to be ignored. The problem is determining which part to ignore. Fig. 1.14.2 shows underfitting.

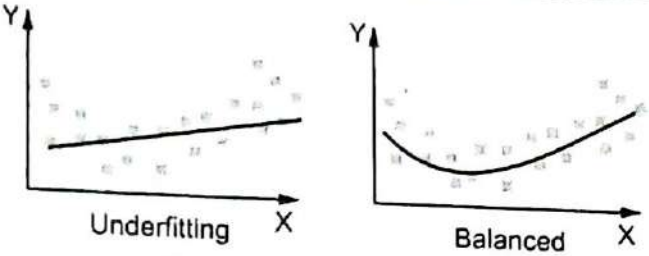


Fig. 1.14.2 : Underfitting

- In the machine learning the more complex model is said to show signs of overfitting, while the simpler model underfitting.
- A learner that underfits the training data will miss important aspects of the data, and this will negatively impact its performance in making accurate predictions on new data it has not seen during training.
- How do we know if we are underfitting or overfitting?
  1. If by increasing capacity we decrease generalization error, then we are underfitting, otherwise we are overfitting.
  2. If the error in representing the training set is relatively large and the generalization error is large, then underfitting;
  3. If the error in representing the training set is relatively small and the generalization error is large, then overfitting;
  4. There are many features but relatively small training set.
- To prevent under-fitting we need to make sure that :
  1. The network has enough hidden units to represent the required mappings.
  2. The network is trained for long enough that the error/cost function is sufficiently minimized

Signs of Underfitting/Overfitting

- As we increase capacity, training error can be reduced, but the difference between training and generalization error increases.
- At some point, the increase in capacity causes an increase in generalization error, and we enter the overfitting zone, where capacity is too large, above the optimal capacity. Fig. 1.14.3 shows optimal capacity.

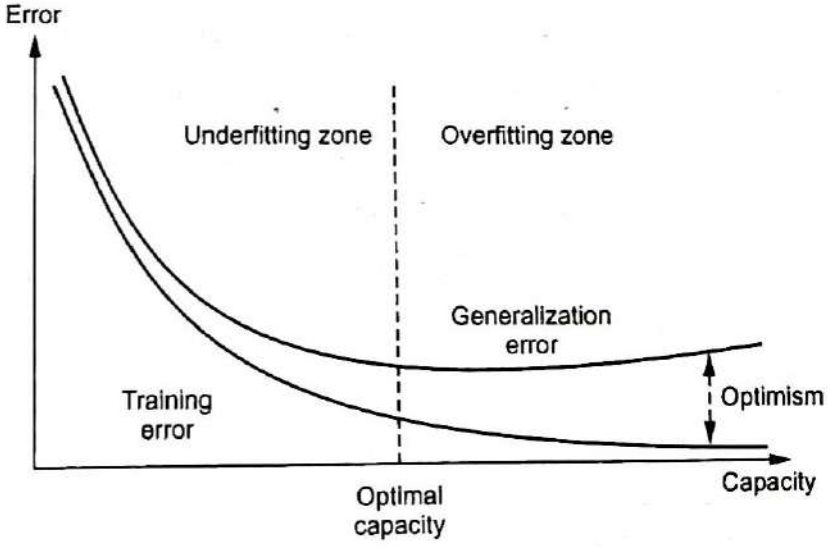


Fig. 1.14.3 : Optimal capacity

- Optimal capacity should increase with the number of training examples; in other words, the more training examples you have, the higher capacity you can afford.
- Optimal capacity does not always increase with the number of training examples. For example, increasing the capacity for a quadratic training set would not increase the optimal capacity threshold.

1.14.4 Probably Approximately Correct (PCA) Learning

- PCA is a nice formalism for deciding how much data you need to collect in order for a given classifier to achieve a given probability of correct predictions on a given fraction of future test data.
- To understand what this model is all about, it's probably easiest just to give an example. Say there's a hidden line on the chalk board.
- Given a point on the board, we need to classify whether it's above or below the line. To help, we'll get some sample data, which consists of random points on the board and whether each point is above or below the line.
- After seeing, say, twenty points, you won't know exactly where the line is, but you'll probably know roughly where it is. And using that knowledge, you'll be able to predict whether most future points lie above or below the line.
- Suppose we have agreed that predicting the right answer "most of the time" is okay. Is any random choice of twenty points going to give you that ability? No, because you could get really unlucky with the sample data, and it could tell you almost nothing about where the line is. Hence the "Probably" in PAC.
- **X** is the set of all possible examples. *D* is the distribution from which the examples are drawn
- **H** is the set of all possible hypotheses, $c \in \mathbf{H}$
- *m* is the number of training examples. Then

error(h) = Pr(h(x) ≠ c(x) | x is drawn from X with D)

where h is approximately correct if *error(h)* $\leq \varepsilon$

- *Hypothesis h(X) is consistent with m examples and has an error of at most with probability* $1 - \delta$. This is a worst-case analysis. Note that the result is independent of the distribution *D*.
- **Curse of dimensionality :** If the number of features d is large, the number of samples n, may be too small for accurate parameter estimation.
- For accurate estimation, n should be much bigger than d^2, otherwise model is too complicated for the data, overfitting

1.14.5 Statistical Learning Approaches

- Statistical learning theory explores ways of estimating functional dependency from a given collection of data. It covers important topics in classical statistics such as discriminant analysis, regression methods, and the density estimation problem
- Statistical learning is a kind of statistical inference, also called inductive statistics.
- For instance, in image analyses it is straightforward to consider each data point (image) as a point in a n-dimensional space, where n is the number of pixels of each image. Therefore, dimensionality reduction may be necessary in order to discard redundancy and simplify further computational operations.
- Bayesian learning formulates learning as a form of probabilistic inference, using the observations to update a prior distribution over hypotheses.
- Maximum A Posteriori (MAP) selects a single most likely hypothesis given the data.
- Maximum likelihood simply selects the hypothesis that maximizes the likelihood of the data.
- Many learning approaches such as neural network learning, linear regression, and polynomial curve fitting try to learn a continuous-valued target function.
- Under certain assumptions any learning algorithm that minimizes the squared error between the output hypothesis predictions and the training data will output a MAXIMUM LIKELIHOOD HYPOTHESIS.
- Learner L considers an instance space X and a hypothesis space H consisting of some class of real-valued functions defined over X.
- The problem faced by L is to learn an unknown target function f drawn from H.
- A set of m training examples is provided, where the target value of each example is corrupted by random noise drawn according to a normal probability distribution
- The task of the learner is to output a maximum likelihood hypothesis, or, equivalently, a MAP hypothesis assuming all hypotheses are equally probable a priori.

Maximum likelihood

- Maximum-Likelihood Estimation (MLE) is a method of estimating the parameters of a statistical model. When applied to a data set and given a statistical model, maximum-likelihood estimation provides estimates for the model's parameters.

  $x_1, x_2, x_3, \ldots x_n$ have joint density denoted.

  $f_0 (x_1, x_2, \ldots x_n) = f (x_1, 2, \ldots x_n|\theta)$

Given observed values $X_1 = x_1$, $X_2 = x_2$, ... ,$X_n = x_n$, the likelihood of θ is the function

$$\text{lik}(\theta) = f(x_1, x_2, \ldots x_n|\theta)$$

considered as a function of θ

- If the distribution is discrete, f will be the frequency distribution function.
- The maximum likelihood estimate of θ is that value of that maximises lik (θ) :

It is the value that makes the observed data the most probable.

Examples of maximizing likelihood

- A random variable with this distribution is a formalization of a coin toss. The value of the random variable is 1 with probability θ and 0 with probability $1-\theta$. Let X be a Bernoulli random variable, and let x be an outcome of X, then we have.

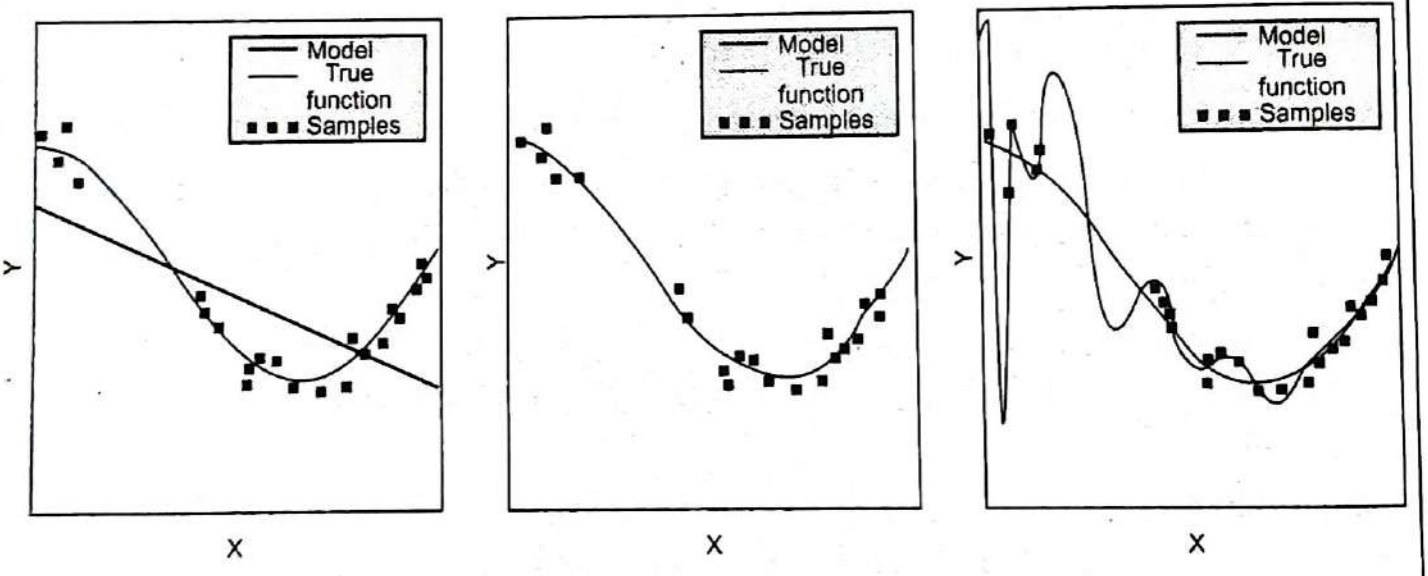
$$P(X = x) = \begin{bmatrix} \theta & \text{if } x = 1 \\ 1-\theta & \text{if } x = 0 \end{bmatrix}$$

- Usually, we use the notation P(.) for a probability mass, and the notation p(.) for a probability density. For mathematical convenience write P(X) as

$$P(X = x) = \theta^x(1-\theta)^{1-x}$$

Review Questions

1. *Which one of these is underfit and overfit ? Why ? Comment with respect to bias and variance.*



(a) Degree 1 (b) Degree 4 (c) Degree 15

Fig. 1.14.4

2. *Explain overfitting and underfitting.*

1.15 Application of Machine Learning

SPPU : March-20

- Examples of successful applications of machine learning :

Here are several examples :

1 **Optical character recognition :** Categorize images of handwritten characters by the letters represented.

2 **Face detection :** Find faces in images (or indicate if a face is present)

3 **Spam filtering :** Identify email messages as spam or non-spam topic spotting: categorize news articles (say) as to whether they are about politics, sports, entertainment, etc.

4 **Spoken language understanding :** Within the context of a limited domain, determine the meaning of something uttered by a speaker to the extent that it can be classified into one of a fixed set of categories.

1.15.1 Face Recognition and Medical Diagnosis

Face recognition

- Face recognition task is effortlessly and every day we recognize our friends, relative and family members. We also recognition by looking at the photographs. In photographs, they are in different pose, hair styles, background light, makeup and without makeup.
- We do it subconsciously and cannot explain how we do it. Because we can't explain how we do it, we can't write an algorithm.
- Face has some structure. It is not a random collection of pixel. It is symmetric structure. It contains predefined components like nose, mouth, eye, ears. Every person face is a pattern composed of a particular combination of the features. By analyzing sample face images of a person, a learning program captures the pattern specific to that person and users it to recognize if a new real face or new image belongs to this specific person or not.
- Machine learning algorithm creates an optimized model of the concept being learned based on data or past experience.
- In the case of face recognition, the input is an image, the classes are people to be recognized and the learning program should learn to associate the face images to identities. This problem is more difficult than optical character recognition because there are more classes, input image is larger and a face is 3D and differences in pose and lighting cause significant changes in the image.

Medical diagnosis

- In medical diagnosis, the input are the relevant information about the patient and the classes are the illness. The inputs contain the age of patient's, gender, past medical history and current symptoms.
- Some tests may not have been applied to the patient and thus these inputs would be missing. Tests take time, may be costly and may inconvience the patient so we do not want to apply them unless we believe that they will give us valuable information.
- In the case of a medical diagnosis, a wrong decision may lead to a wrong or no treatment and in cases of doubt it is preferable that the classifier reject, and defer decision to a human expert.

1.15.2 Google Home and Amazon Alexa

Amazon Alexa / Siri

- Every time Alexa or Siri make a mistake when responding to our request, it uses the data it receives based on how it responded to the original query to improve the next time. If an error was made, it takes that data and learns from it. If the response was favourable, the system notes that as well.
- Data and machine learning are responsible for the explosive growth of digital voice assistants. They continue o get better with the more experiences they have and the data they accumulate.
- When user make a request of Alexa, the microphone on the device records command. This recording is sent to over the internet to the cloud. If user are talking to Alexa, the recording is sent to Alexa Voice Services (AVS). This cloud-based service will review the recording and interpret user request. Then, the system will send a relevant response back to the device.
- Amazon breaks down user "orders" into individual sounds. It then consults a database containing various words' pronunciations to find which words most closely correspond to the combination of individual sounds.
- It then identifies important words to make sense of the tasks and carry out corresponding functions. For instance, if Alexa notices words like "sport" or "basketball", it would open the sports app.
- Amazon's servers send the information back to our device and Alexa may speak. If Alexa needs to say anything back, it would go through the same process described above, but in reverse order.

Google Home :

- Google services such as its image search and translation tools use sophisticated machine learning which allow computers to see, listen and speak in much the same way as human do.
- To perform its functions, Google Assistant relies on Artificial Intelligence (AI) technologies such as natural language processing and machine learning to understand what the user is saying and to make suggestions or act on that language input.
- The Google Home can play music, but it's primarily designed as a vehicle for Google Assistant -- Google's voice - activated virtual helper that's connected to the internet.
- The Google Home is always listening to its environment, but it won't record what we are saying or respond to our commands until we speak one of its pre-programmed wake words -- either "OK, Google" or "Hey, Google."
- TF-IDF, is a numerical statistic that is intended to reflect how important a word is to a document from collection corpus. It is often used as a weighting factor for searches of information retrieval, text mining and user modeling.
- The TD-IDF value increases proportionally to the number of times a word appears in the document but it is often offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently.

1.15.3 Unmanned Vehicles

- An Unmanned Aerial Vehicle (UAV), sometimes known as a drone, is an aircraft or airborne system that is controlled remotely by an onboard computer or a human operator. The ground control station, aircraft components, and various types of sensors make up the UAV system.
- UAVs are categorized depending on their endurance, weight and altitude range. They can be used for multiple commercial and military applications.
- Machine learning is the process of using, storing and finding patterns within massive amounts of data, which can eventually be fed into algorithms. It's basically a process of using the data accumulated by the machine or device that allows computers to develop their own algorithm so that humans won't have to create challenging algorithms manually.
- Unmanned ground vehicles are classified into two broad types, remotely operated and autonomous.
- Autonomous unmanned ground vehicles comprise several technologies that allow the machine to be self - acting and self - regulating, sans human intervention. The

technology was initially developed to aid ground forces in the transfer of heavy equipment.

- However, the technology has witnessed significant evolution over the years, giving rise to more tactical vehicles designed to assist in surveillance or IED search-and-destroy missions.
- For example, unmanned ships in the course of the voyage, the default route is to ensure the obstruction of the premise of a straight line navigation.

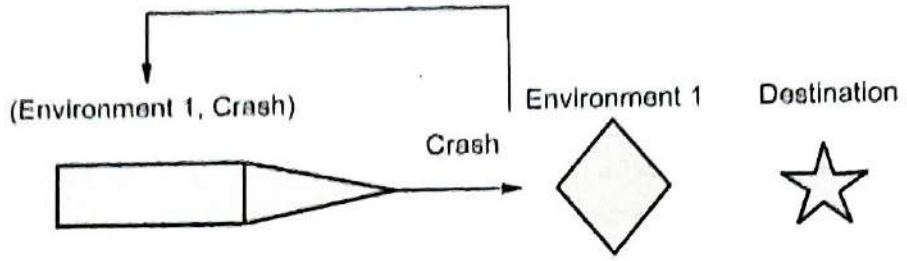


Fig. 1.15.1 First time lane

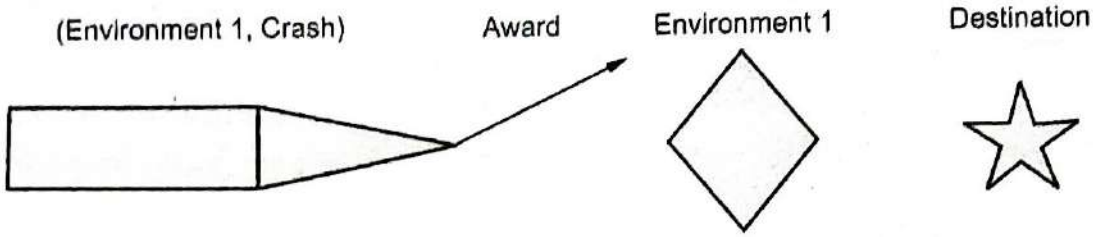


Fig. 1.15.2 The second time

- During the course of the voyage, the hull is changed by the intensity and direction of the waves and is unpredictable. It is clear for the unmanned boat itself.
- Therefore, unmanned ships in the process of navigation, continue to train the perception of the surrounding environment and make the appropriate strategy, if the results of the implementation of the strategy in line with the default route to be rewarded.

Review Question

1. *Describe the role of machine learning in the following applications :*
   a) Google home or Alexa b) Unmanned Vehicles. **SPPU : March-20, In Sem, Marks 10**

❑❑❑

Unit II

2 Feature Engineering

Syllabus

Concept of Feature, Preprocessing of data : Normalization and Scaling, Standardization, Managing missing values, Introduction to Dimensionality Reduction, Principal Component Analysis (PCA), Feature Extraction : Kernel PCA, Local Binary Pattern. Introduction to various Feature Selection Techniques, Sequential Forward Selection, Sequential Backward Selection. Statistical feature engineering : count-based, Length, Mean, Median, Mode etc. based feature vector creation, Multidimensional Scaling, Matrix Factorization Techniques..

Contents

2.1 Concept of Feature

- In machine learning, features are individual independent variables that act like input in your system. Feature is an attribute of a data set and used in a machine learning process. Selection of the subset of features which are meaningful for machine learning is a sub-area of feature engineering.
- The features in a data set are also called its dimensions. So a data set having 'n' features is called an n-dimensional data set.
- A good feature representation is central to achieving high performance in any machine learning task.
- Consider an example of text categorization. Assume that we need to train a model for classifying a given document as spam and not spam. If we represent a document as a bag of words, the feature space consists of a vocabulary of all unique words present in all the documents in the training set.
- For a collection of 100,000 to 1,000,000 documents, we can easily expect hundreds of thousands of features. If we further extend this document model to include all possible bigrams and trigrams, we could easily get over a million features.
- A feature tree is a tree such that each internal node is labelled with a feature, and each edge emanating from an internal node is labelled with a literal. The set of literals at a node is called a split. Each leaf of the tree represents a logical expression, which is the conjunction of literals encountered on the path from the root of the tree to the leaf. The extension of that conjunction is called the instance space segment associated with the leaf.
- Two features are redundant if they are highly correlated, regardless of whether they are correlated with the task or not.
- Feature engineering is the process of creating features (also called "attributes") that don't already exist in the dataset. This means that if your dataset already contains enough "useful" features, you don't necessarily need to engineer additional features.
- Feature engineering refers to the process of translating a data set into features such that these features are able to represent the data set more effectively and result in a better learning performance.
- If feature engineering is performed properly, it helps to improve the power of prediction of machine learning algorithms by creating the features using the raw data that facilitate the machine learning process.
- Elements of feature engineering is **feature transformation and feature subset selection.**

2.1.1 Feature Transformation

- **Feature transformation** transforms the data, structured or unstructured, into a new set of features which can represent the underlying problem which machine learning is trying to solve.

- There are two distinct goals of feature transformation :
  1. Achieving best reconstruction of the original features in the data set.
  2. Achieving highest efficiency in the learning task.
- There are two variants of feature transformation :
  1. Feature construction.
  2. Feature extraction.

2.1.2 Feature Construction

- Feature construction involves transforming a given set of input features to generate a new set of more powerful features which can then used for prediction.
- Feature construction methods may be applied to pursue two distinct goals : Reducing data dimensionality and improving prediction performance.
- **Steps :**
  1. Start with an initial feature space F_0 .
  2. Transform F_0 to construct a new feature space F_N .
  3. Select a subset of features F_i from F_N .
  4. If some terminating criteria is achieved : Go back to step 3 otherwise set $F_T = F_i$.
  5. F_T is the newly constructed feature space.
- Feature construction process discovers missing information about the relationships between features and augments the feature space by creating additional features.
- Hence, if there are 'n' features or dimensions in a data set, after feature construction 'm' more features or dimensions may get added. So at the end, the data set will become 'n + m' dimensional.
- The task of constructing appropriate features is often highly application specific and labour intensive. Thus building auto-mated feature construction methods that require minimal user effort is challenging. In particular we want methods that :
  1. Generate a set of features that help improve prediction accuracy.
  2. Are computationally efficient.
  3. Are generalizable to different classifiers.
  4. Allow for easy addition of domain knowledge.
- Genetic programming is an evolutionary algorithm-based technique that starts with a population of individuals, evaluates them based on some fitness function and constructs a new population by applying a set of mutation and crossover operators on high scoring individuals and eliminating the low scoring ones.

- In the feature construction paradigm, genetic programming is used to derive a new feature set from the original one. Individuals are often tree like representations of features, the fitness function is usually based on the prediction performance of the classifier trained on these features while the operators can be applications specific.
- The method essentially performs a search in the new feature space and helps generate a high performing subset of features. The newly generated features may often be more comprehensible and intuitive than the original feature set, which makes GP-related methods well-suited for such tasks.
- In decision trees, the model explicitly selects features that are highly correlated with the label. In particular, by limiting the depth of the decision tree, one can at least hope that the model will be able to throw away irrelevant features.

2.1.3 Feature Extraction

- Feature extraction is a process that extracts a set of new features from the original features through some functional mapping. Feature extraction method creates a new feature set.
- Feature extraction increases the accuracy of learned models by extracting features from the input data. This phase of the general framework reduces the dimensionality of data by removing the redundant data.
- A characteristic of these large data sets is a large number of variables that require a lot of computing resources to process.
- Feature extraction is the name for methods that select and combine variables into features, effectively reducing the amount of data that must be processed, while still accurately and completely describing the original data set.
- The process of feature extraction is useful when you need to reduce the number of resources needed for processing without losing important or relevant information.
- Feature extraction can also reduce the amount of redundant data for a given analysis. Also, the reduction of the data and the machine's efforts in building variable combinations (features) facilitate the speed of learning and generalization steps in the machine learning process.

2.1.4 Feature Selection

- Feature selection is a process that chooses a subset of features from the original features so that the feature space is optimally reduced according to a certain criterion.
- Feature selection is a critical step in the feature construction process. In text categorization problems, some words simply do not appear very often. Perhaps the word "groovy" appears in exactly one training document, which is positive. Is

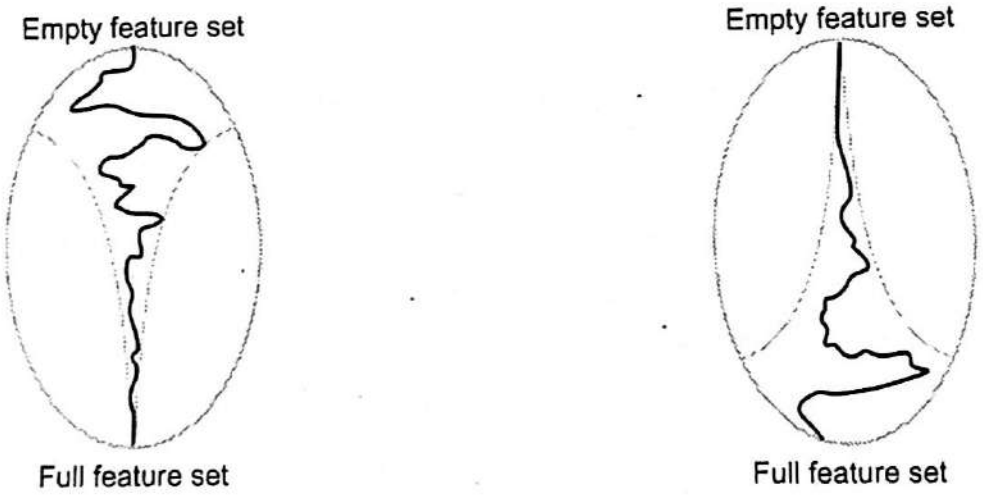
it really worth keeping this word around as a feature ? It's a dangerous endeavor because it's hard to tell with just one training example if it is really correlated with the positive class or is it just noise. You could hope that your learning algorithm is smart enough to figure it out. Or you could just remove it.

- There are three general classes of feature selection algorithms : Filter methods, wrapper methods and embedded methods.
- The role of feature selection in machine learning is,
  1. To reduce the dimensionality of feature space.
  2. To speed up a learning algorithm.
  3. To improve the predictive accuracy of a classification algorithm.
  4. To improve the comprehensibility of the learning results.
- Features Selection Algorithms are as follows :
  1. **Instance based approaches :** There is no explicit procedure for feature subset generation. Many small data samples are sampled from the data. Features are weighted according to their roles in differentiating instances of different classes for a data sample. Features with higher weights can be selected.
  2. **Nondeterministic approaches :** Genetic algorithms and simulated annealing are also used in feature selection.
  3. **Exhaustive complete approaches :** Branch and Bound evaluates estimated accuracy and ABB checks an inconsistency measure that is monotonic. Both start with a full feature set until the preset bound cannot be maintained.

2.1.5 Subset Selection

- Finding the best subset of the set of features is main aim of subset selection. The best subset contains the least number of dimensions that most contribute to accuracy.
- Using a suitable error function, this can be used in both regression and classification problems. There are 2d possible subsets of d variables, but we cannot test for all of them unless d is small and we employ heuristics to get a reasonable (but not optimal) solution in reasonable (polynomial) time.
- Subset selection are of two types : Forward and backward selection.
  1. **Forward selection :** It start without variables and add them one by one, at each step adding the one that decreases the error the most, until any further addition does not decrease the error.
  2. **Backward selection :** It start with all variables and remove them one by one, at each step removing the one that decreases the error the most, until any further removal increases the error significantly.

- **Sequential Forward Selection (SFS) :** SFS is the simplest greedy search algorithm. It start from the empty set, sequentially add the feature x^+. SFS performs best when the optimal subset is small.
- The main disadvantage of SFS is that it is unable to remove features that become obsolete after the addition of other features.
- **Sequential Backward Selection (SBS) :** It works in the opposite direction of SFS. Starting from the full set, sequentially remove the feature x^- that least reduces the value of the objective function.
- SBS works best when the optimal feature subset is large, since SBS spends most of its time visiting large subsets. The main limitation of SBS is its inability to reevaluate the usefulness of a feature after it has been discarded.
- SFS is performed from the empty set. SBS is performed from the full set.
- There are two floating methods :
  1. Sequential Floating Forward Selection (SFFS) starts from the empty set. After each forward step, SFFS performs backward steps as long as the objective function increases.
  2. Sequential Floating Backward Selection (SFBS) starts from the full set. After each backward step, SFBS performs forward steps as long as the objective function increases.
- Subset selection is supervised in that outputs are used by the regressor or classifier to calculate the error, but it can be used with any regression or classification method.



(a) Sequential forward selection **(b) Sequential backward selection**

Fig. 2.1.1

- In an application like face recognition, feature selection is not a good method for dimensionality reduction because individual pixels by themselves do not carry much discriminative information; it is the combination of values of several pixels

together that carry information about the face identity. This is done by feature extraction methods.

2.2 Preprocessing of Data

SPPU : April-19

- Data pre-processing is a data mining technique that involves transforming raw data into an understandable format. Aim to reduce the data size, find the relation between data and normalized them.
- Data pre-processing is a proven method of resolving such issues. Data pre-processing prepares raw data for further processing.

Why Data Pre-processing ?

- Data which capture from various source is not pure. It contains some noise. It is called dirty data or incomplete data. In this data, there is lacking attribute values, lacking certain attributes of interest, or containing only aggregate data. For example : occupation= " "
- Noisy data which contains errors or outliers. For example: Salary="-10"
- Inconsistent data which contains discrepancies in codes or names. For example : Age="51" Birthday="03/08/1998"
- Incomplete, noisy, and inconsistent data are commonplace properties of large real-world databases and data warehouses. Incomplete data can occur for a number of reasons.
- **Steps during pre-processing :**
  1. **Data cleaning :** Data is cleansed through processes such as filling in missing values, smoothing the noisy data, or resolving the inconsistencies in the data.
  2. **Data integration :** Data with different representations are put together and conflicts within the data are resolved.
  3. **Data transformation :** Data is normalized, aggregated and generalized.
  4. **Data reduction :** This step aims to present are duced representation of the data in a datawarehouse.
  5. **Data discretization :** Involves the reduction of a number of values of a continuous attribute by dividing the range of attribute intervals.

2.2.1 Normalization and Scaling

- Normalization is a data preparation technique that is frequently used in machine learning. The process of transforming the columns in a dataset to the same scale is referred to as normalization. Every dataset does not need to be normalized for machine learning.

- Normalization makes the features more consistent with each other, which allows the model to predict outputs more accurately. The main goal of normalization is to make the data homogenous over all records and fields.
- Normalization refers to rescaling real-valued numeric attributes into a 0 to 1 range. Data normalization is used in machine learning to make model training less sensitive to the scale of features.
- Normalization is important in such algorithms as k-NN, support vector machines, neural networks, and principal components. The type of feature preprocessing and normalization that's needed can depend on the data.
- The most widely used types of normalization in machine learning are :
  1. **Min-max scaling :** Subtract the minimum value from each column's highest value and divide by the range. Each new column has a minimum value of 0 and a maximum value of 1.
  2. **Standardization scaling :** The term "standardization" refers to the process of centering a variable at zero and standardizing the variance at one. Subtracting the mean of each observation and then dividing by the standard deviation is the procedure :
     - Scaling of the data comes under the set of steps of data pre-processing when we are performing machine learning algorithms in the data set. In the machine learning algorithms if the values of the features are closer to each other there are chances for the algorithm to get trained well and faster instead of the data set where the data points or features values have high differences with each other will take more time to understand the data and the accuracy will be lower.
     - The machine learning models provide weights to the input variables according to their data points and inferences for output. In that case, if the difference between the data points is so high, the model will need to provide the larger weight to the points and in final results, the model with a large weight value is often unstable. This means the model can produce poor results or can perform poorly during learning.
     - Whether the data is categorical, numerical, textual, or time series, normalization can bring all the data on a single scale.

2.2.2 Standardization

- Standardization entails scaling data to fit a standard normal distribution. A standard normal distribution is defined as a distribution with a mean of 0 and a standard deviation of 1. This technique also tries to scale the data point between zero to one but in it, we don't use max or minimum.

- When your data has variable dimensions and the technique you're using (like logistic regression, linear regression, linear discriminant analysis) standardization is useful.
- Standardization is used for feature scaling when your data follows Gaussian distribution. It is most useful for :
  1. Optimizing algorithms such as gradient descent.
  2. Clustering models or distance-based classifiers like K-Nearest neighbors
  3. High variance data ranges such as in principle component analysis
- Data standardization is the process of placing dissimilar features on the same scale. Standardized data in other words can be defined as rescaling the attributes in such a way that their mean is 0 and standard deviation becomes 1.
- Let x be an individual feature value and min(x) and max(x), respectively, be the minimum and maximum values of this feature over the entire dataset. Min-max scaling squeezes all feature values to be within the range of [0, 1].
- Feature standardization is defined as :

$$\overline{X} = \frac{X - \text{Mean}(X)}{\sqrt{\text{var}(X)}}$$

- It subtracts off the mean of the feature (over all data points) and divides by the variance. Hence, it can also be called variance scaling. The resulting scaled feature has a mean of 0 and a variance of 1. If the original feature has a Gaussian distribution, then the scaled feature does too.
- It subtracts off the mean of the feature (over all data points) and divides by the variance. Hence, it can also be called variance scaling. The resulting scaled feature has a mean of 0 and a variance of 1. If the original feature has a Gaussian distribution, then the scaled feature does too.

Review Question

1. *With reference to feature engineering, explain data scaling and normalization tasks.*
   SPPU : April-19, In Sem, Marks 5

2.3 Data Cleaning

- Sometimes real-world data is incomplete, noisy, and inconsistent. Data cleaning methods are used for making useable data.
- Data cleaning tasks are as follows :
  1. Data acquisition and metadata
  2. Fill in missing values

3. Unified date format
4. Converting nominal to numeric
5. Identify outliers and smooth out noisy data
6. Correct inconsistent data

- Data cleaning is a first step in data pre-processing techniques which is used to find the missing value, smooth noise data, recognize outliers and correct inconsistent.

2.3.1 Missing Value

These dirty data will affects on miming procedure and led to unreliable and poor output. Therefore it is important for some data cleaning routines.

How to handle noisy data in data mining ?

- Following methods are used for handling noisy data :
  1. **Ignore the tuple :** Usually done when the class label is missing. This method is not good unless the tuple contains several attributes with missing values.
  2. **Fill in the missing value manually :** It is time-consuming and not suitable for a large data set with many missing values.
  3. **Use a global constant to fill in the missing value :** Replace all missing attribute values by the same constant.
  4. **Use the attribute mean to fill in the missing value :** For example, suppose that the average salary of staff is Rs 65000/- . Use this value to replace the missing value for salary.
  5. Use the attribute mean for all samples belonging to the same class as the given tuple
  6. Use the most probable value to fill in the missing value

2.3.2 Noisy Data

- **Noise :** Random error or variance in a measured variable
- For numeric values, box plots and scatter plots can be used to identify outliers. To deal with these anomalous values, data smoothing techniques are applied, which are described below.
  1. **Binning :** Using binning methods smooths sorted value by using the values around it. The sorted values are then divided into 'bins'. There are various approaches to binning. Two of them are smoothing by bin means where each bin is replaced by the mean of bin's values, and smoothing by bin medians where each bin is replaced by the median of bin's values.

Binning methods for data smoothing :

a) **In smoothing by bin means :** Each value in a bin is replaced by the mean value of the bin. For example, the mean of the values 5, 9, and 13 in Bin is 9. Therefore, each original value in this bin is replaced by the value 9.

b) **Smoothing by bin medians** can be employed, in which each bin value is replaced by the bin median.

c) **Smoothing by bin boundaries :** The minimum and maximum bin values are stored at the boundary while intermediate bin values are replaced by the boundary value to which it is more closer.

2. **Regression :** Linear regression and multiple linear regression can be used to smooth the data, where the values are conformed to a function.

3. **Outlier analysis :** Approaches such as clustering can be used to detect outliers and deal with them.

2.4 Introduction to Dimensionality Reduction

- In machine learning, "dimensionality" simply refers to the number of features (i.e. input variables) in your dataset.
- When the number of features is very large relative to the number of observations in your dataset, certain algorithms struggle to train effective models. This is called the "Curse of Dimensionality," and it's especially relevant for clustering algorithms that rely on distance calculations.
- Dimensionality reduction is the process of reducing the number of random variables under consideration, by obtaining a set of principal variables. It can be divided into feature selection and feature extraction.
- Classification problem example : We have an input data $\{X_1, X_2, X_3, ..., X_N\}$ such that $x_i = (x_i^1, x_i^2, ..., x_i^d)$ and a set of corresponding output labels. Assume the dimension d of the data point x is very large and we want to classify x.
  - Problem with high dimensional input vectors are large number of parameters to learn, if a dataset is small, this can result in overfit and large variance of estimates.
  - Solution to this problem is as follows :
    1. Selection of a smaller subset of inputs from a large set of inputs; train classifier on the reduced input set.
    2. Combination of high dimensional inputs to a smaller set of features $\phi_k(x)$; train classifier on new features.

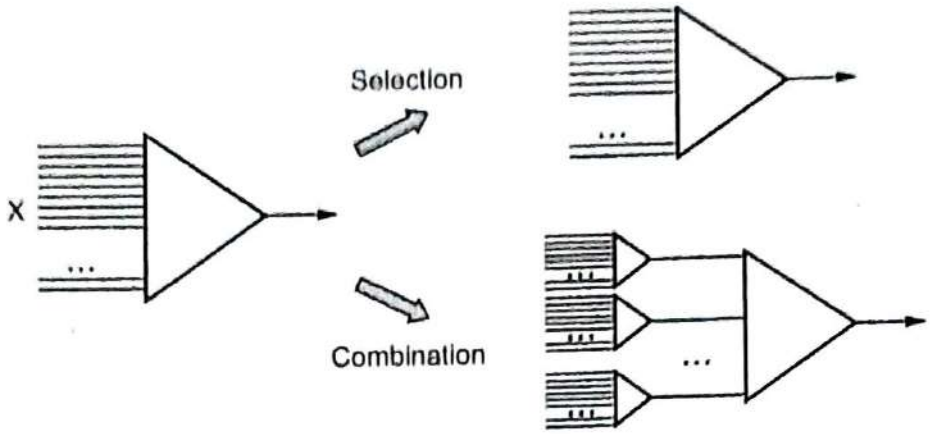


Fig. 2.4.1 Dimensionality reduction

There are two components of dimensionality reduction :

1. **Feature selection :** User try to find a subset of the original set of variables, or features, to get a smaller subset which can be used to model the problem. It usually involves three ways ; Filter, wrapper and embedded.
2. **Feature extraction :** This reduces the data in a high dimensional space to a lower dimension space, i.e. a space with lesser number of dimensions.

There are many methods to perform dimension reduction.

1. **Missing Values :** While exploring data, if we encounter missing values, what we do ? Our first step should be to identify the reason then impute missing values/drop variables using appropriate methods. But, what if we have too many missing values ? Should we impute missing values or drop the variables ?
2. **Low Variance :** Let's think of a scenario where we have a constant variable in our data set.
3. **Decision Trees :** It can be used as a ultimate solution to tackle multiple challenges like missing values, outliers and identifying significant variables.
4. **Random Forest :** Similar to decision tree is random forest.
5. **High Correlation :** Dimensions exhibiting higher correlation can lower down the performance of model. Moreover, it is not good to have multiple variables of similar information or variation also known as "multicollinearity".
6. **Backward Feature Elimination :** In this method, we start with all n dimensions. Compute the sum of square of error (SSR) after eliminating each variable (n times). Then, identifying variables whose removal has produced the smallest increase in the SSR and removing it finally, leaving us with n–1 input features.

2.4.1 Advantages and Disadvantages of Dimensionality Reduction

Advantages of Dimensionality Reduction

It helps in data compression, and hence reduced storage space.

It reduces computation time.

- It also helps remove redundant features, if any.

- **Disadvantages of Dimensionality Reduction**
  - It may lead to some amount of data loss.
  - PCA tends to find linear correlations between variables, which is sometimes undesirable.
  - PCA fails in cases where mean and covariance are not enough to define datasets.
  - We may not know how many principal components to keep - in practice, some thumb rules are applied.

2.5 Principal Component Analysis (PCA)

SPPU : April-19

- This method was introduced by Karl Pearson. It works on a condition that while the data in a higher dimensional space is mapped to data in a lower dimension space, the variance of the data in the lower dimensional space should be maximum.
- Principal Component Analysis (PCA) is to reduce the dimensionality of a data set by finding a new set of variables, smaller than the original set of variables, retains most of the sample's information and useful for the compression and classification of data.

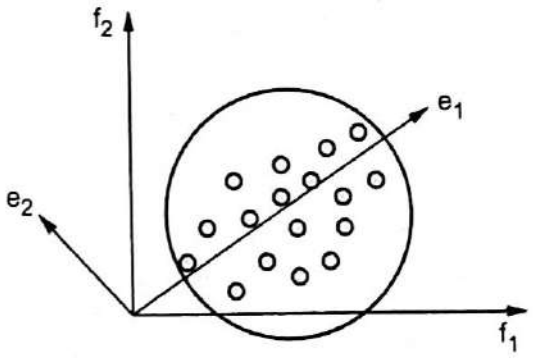


Fig. 2.5.1 PCA

- In PCA, it is assumed that the information is carried in the variance of the features, that is, the higher the variation in a feature, the more information that feature carries.
- Hence, PCA employs a linear transformation that is based on preserving the most variance in the data using the least number of dimensions.
- It involves the following steps :
  1. Construct the covariance matrix of the data.
  2. Compute the eigenvectors of this matrix.

3. Eigenvectors corresponding to the largest eigen values are used to reconstruct a large fraction of variance of the original data.

- The data instances are projected onto a lower dimensional space where the new features best represent the entire data in the least squares sense.
- It can be shown that the optimal approximation, in the least square error sense, of a d-dimensional random vector $x_2 < d$ by a linear combination of independent vectors is obtained by projecting the vector x onto the eigenvectors e_i corresponding to the largest eigen values l_i of the covariance matrix (or the scatter matrix) of the data from which x is drawn.
- The eigenvectors of the covariance matrix of the data are referred to as principal axes of the data, and the projection of the data instances on to these principal axes are called the principal components. Dimensionality reduction is then obtained by only retaining those axes (dimensions) that account for most of the variance, and discarding all others.
- In the Fig. 2.5.1, Principal axes are along the eigenvectors of the covariance matrix of the data. There are two principal axes shown in the figure, first one is closed to origin, the other is far from origin.
- If $X = X_1, X_2, \ldots\ldots, X_N$ is the set of n patterns of dimension d, the sample mean of the data set is

$$m = \frac{1}{n}\sum_{i=1}^{n} X_i$$

- The sample covariance matrix is

$$C = (X - m)(X - m)^T$$

- C is a symmetric matrix. The orthogonal basis can be calculated by finding the eigenvalues and eigenvectors.
- The eigenvectors g_i and the corresponding eigenvalues λ_i are solutions of the equation

$$C * g_i = \lambda_i * g_i \quad i = 1, \ldots\ldots, d$$

- The eigenvector corresponding to the largest eigenvalue gives the direction of the largest variance of the data. By ordering the eigenvectors according to the eigenvalues, the directions along which there is maximum variance can be found.
- If E is the matrix consisting of eigenvectors as row vectors, we can transform the data X to get Y .

$$Y = E(X - m)$$

- The original data X can be got from Y as follows :

$$X = E^t Y + m$$

- Instead of using all d eigenvectors, the data can be represented by using the first k eigenvectors where $k < d$.
- If only the first k eigenvectors are used represented by E_K, then

$$Y = E_K (X - m) \quad \text{and} \quad X' = E_k^t Y + m$$

2.5.1 Non Negative Matrix Factorization (NMF)

- Nonnegative Matrix Factorization is a matrix factorization method where we constrain the matrices to be nonnegative. In order to understand NMF, we should clarify the underlying intuition between matrix factorization.
- Suppose we factorize a matrix X into two matrices W and H so that X = W H.
- There is no guarantee that we can recover the original matrix, so we will approximate it as best as we can.
- Now, suppose that X is composed of m rows, $x_1, x_2, \ldots x_m$, W is composed of k rows $w_1, w_2, \ldots w_k$, H is composed of m rows $h_1, h_2, \ldots h_m$.
- Each row in X can be considered a data point. For instance, in the case of decomposing images, each row in X is a single image, and each column represents some feature,

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_k \end{bmatrix}, \quad W = \begin{bmatrix} w_1 \\ w_2 \\ \cdots \\ w_k \end{bmatrix}, \quad H = \begin{bmatrix} h_1 \\ h_2 \\ \cdots \\ h_k \end{bmatrix}$$

- Take the i^{th} row in X, x_i. If you think about the equation, you will find that x_i can be written as

$$x_i = \sum_{j=1}^{k} w_{ij} \times h_i$$

- Basically, we can interpret x_i to be a weighted sum of some components, where each row in H is a component, and each row in W contains the weights of each component.

How Does it Work ?

- NMF decomposes multivariate data by creating a user-defined number of features. Each feature is a linear combination of the original attribute set; the coefficients of these linear combinations are non-negative.
- NMF decomposes a data matrix V into the product of two lower rank matrices W and H so that V is approximately equal to W times H.

- NMF uses an iterative procedure to modify the initial values of W and H so that the product approaches V. The procedure terminates when the approximation error converges or the specified number of iterations is reached
- During model apply, an NMF model maps the original data into the new set of attributes (features) discovered by the model.

2.5.2 Difference between PCA and NMF

Sr. No.	PCA	NMF
1.	It uses unsupervised dimensionality reduction.	It also uses unsupervised dimensionality reduction.
2.	Orthogonal vectors with positive and negative coefficients.	Non-negative coefficients.
3.	Difficult to interpret.	Easier to interpret.
4.	PCA is non-iterative.	NMF is iterative.
5.	Designed for producing optimal basis images.	Designed for producing coefficients with a specific property.

2.5.3 Sparse PCA

- In sparse PCA one wants to get a small number of features which still capture most of the variance. Thus one needs to enforce sparsity of the PCA component, which yields a trade-off between explained variance and sparsity.
- To address the non-sparsity issue of traditional PCA, sparse PCA imposes additional constraint on the number of non-zero element in the vector v.
- This is achieved through the l_0 norm, which gives the number of non-zero element in the vector v. A sparse PCA with at most k non-zero loadings can then be formulated as the following optimization problem.
- Optimization problems with l_0 norm constraint is in general NP-hard. Therefore, most methods for sparse PCA relaxes the l_0 norm constraint with l_1 norm appended to the objective function.

2.5.4 Kernel PCA

- Kernel PCA is the nonlinear form of PCA, which better exploits the complicated spatial structure of high-dimensional features.
- It can extract up to n (number of samples) nonlinear principal components without expensive computations.

- The standard steps of kernel PCA dimensionality reduction can be summarized as :
  1. Construct the kernel matrix K from the training data set
  2. Compute the gram matrix
  3. Solve N-dimensional column vector
  4. Compute the kernel principal components
- Kernel PCA supports both transform and inverse_transform.
- Fig 2.5.2 (a), (b) shows PCA and KPCA.

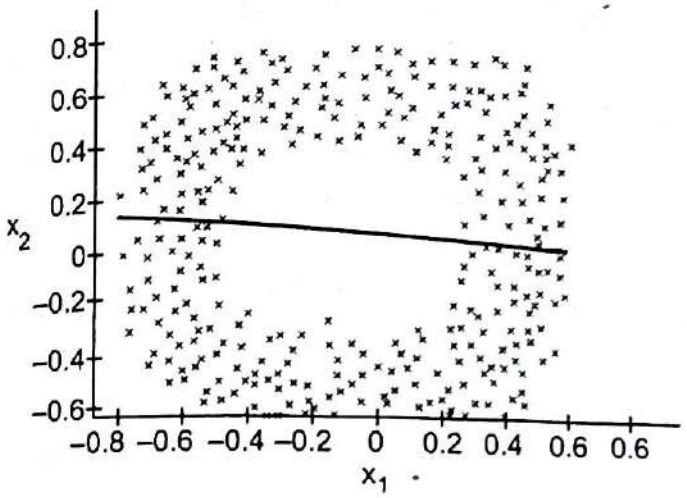


Fig. 2.5.2 (a) PCA

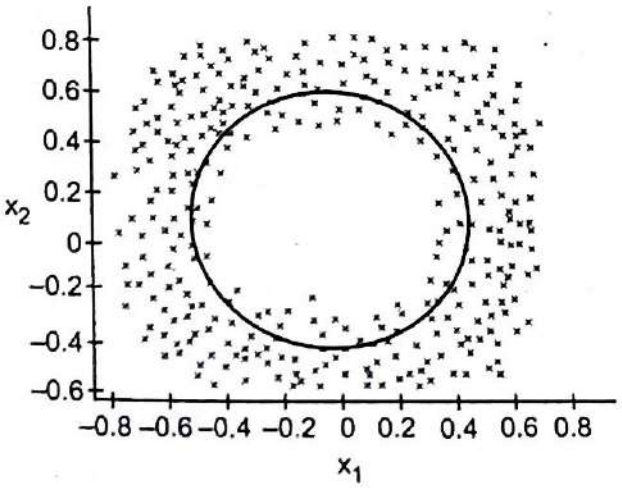


Fig. 2.5.2 (b) KPCA

Preliminaries :

```
# Load libraries
from sklearn.decomposition import PCA, KernelPCA
from sklearn.datasets import make_circles
```

Create Linearly Inseparable Data :

```
# Create linearly inseparable data
X, _ = make_circles(n_samples=1000, random_state=1, noise=0.1, factor=0.1)
```

Conduct Kernel PCA :

```
# Apply kernal PCA with radius basis function (RBF) kernel
kpca = KernelPCA(kernel="rbf", gamma=15, n_components=1)
X_kpca = kpca.fit_transform(X)
```

Review Question

1. What is Principal Component Analysis (PCA), when it is used.

SPPU : April-19, In Sem, Marks 5

2.6 Local Binary Pattern

- Local Binary Pattern (LBP) is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number.
- The LBP is one of the non-parametric descriptors for extracting the image local information. It is implicitly defined at the position of the pixel, LBP operator is expressed by a sequence of binary values obtained by comparing the central pixel with its neighbors in a circular manner.
- For this, all the pixels of the image take an operator value which is calculated as a function of the P neighborhood pixels and the neighborhood threshold which is based on the central pixel. The pixel less than to the central pixel are given the binary value 0 and 1 otherwise. Then all the calculated binary values are concatenated, and the decimal value equivalent to the binary code represents the LBP-label.
- This operator works with the eight neighbors of a pixel, using the value of this center pixel as a threshold. If a neighbor pixel has a higher gray value than the center pixel (or the the same gray value) than a one is assigned to that pixel, else it gets a zero. The LBP code for the center pixel is then produced by concatenating the eight ones or zeros to a binary code.

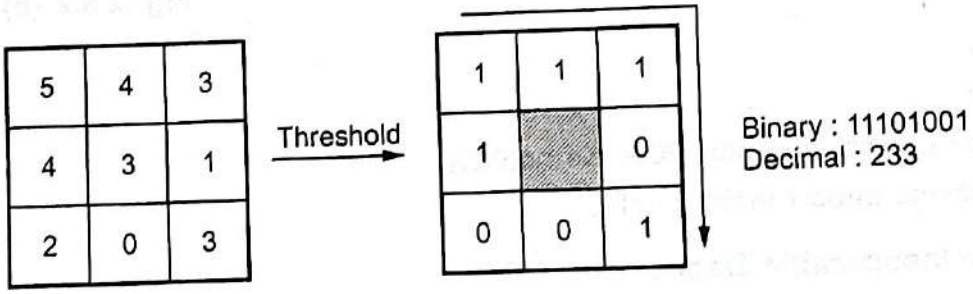


Fig. 2.6.1 LBP code

- Simple LBP feature vector is created in the following manner :

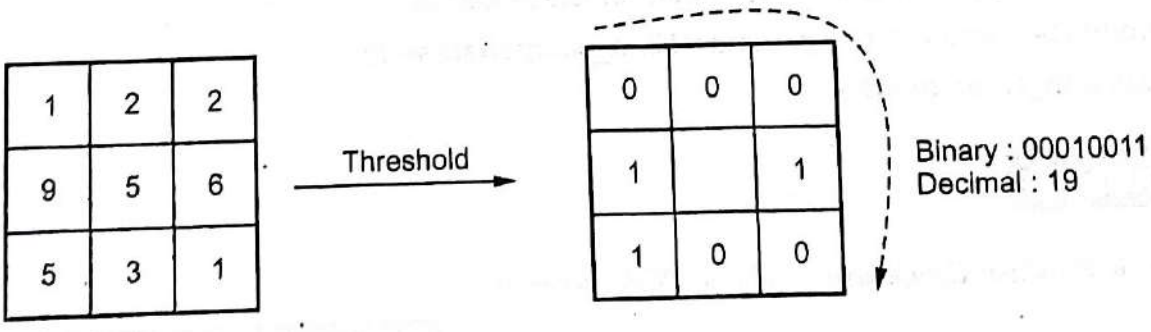


Fig. 2.6.2 LBP feature vector

- Divide the examined window into cells (e.g. 4 x 4 pixels for each cell). For each pixel in a cell, compare the pixel to each of its 8 neighbors (on its left-top, left-middle, left-bottom, right-top, etc.). Follow the pixels along a circle, i.e. clockwise or counterclockwise.
- In the above step, the neighbours considered can be changed by varying the radius of the circle around the pixel, R and the quantization of the angular space P. Where the center pixel's value is greater than the neighbor's value, write "0". Otherwise, write "1". This gives an 8-digit binary number.
- Compute the histogram, over the cell, of the frequency of each "number" occurring. This histogram can be seen as a 256-dimensional feature vector.

2.7 Introduction to Various Feature Selection Techniques

- Feature selection techniques are fundamental to predictive modeling tasks; one can not create predictive models without selecting the features correctly. Feature selection is the method of reducing the input variable to user model by using only relevant data and getting rid of noise in data.
- The role of feature selection in machine learning is
  1. to reduce the dimensionality of feature space
  2. to speed up a learning algorithm
  3. to improve the predictive accuracy of a classification algorithm
  4. to improve the comprehensibility of the learning results
- There are three types of approach for feature selection :
  1. Filter approach 2. Wrapper approach 3. Embedded approach

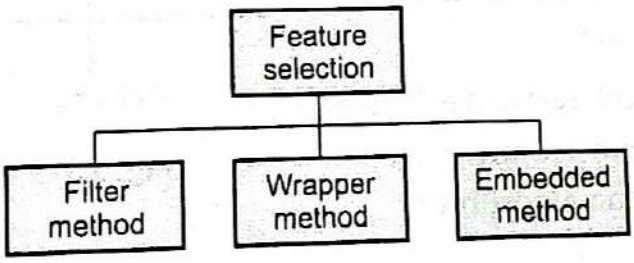


Fig. 2.7.1

1. Filter method :

- Filter methods are generally used as a preprocessing step. The selection of features is independent of any machine learning algorithms. Instead, features are selected on the basis of their scores in various statistical tests for their correlation with the outcome variable.

- Fig. 2.7.2 shows filter method.
- The filter feature selection methods make use of statistical techniques to predict the relationship between each independent input variable and the output (target) variable which assigns scores for each feature.
- Correlation based Feature Selection (CFS) is a simple filter algorithm that ranks feature subsets according to a correlation based heuristic evaluation function.
- Example : Co-orelation, chi-square test, ANOVA, information gain etc.

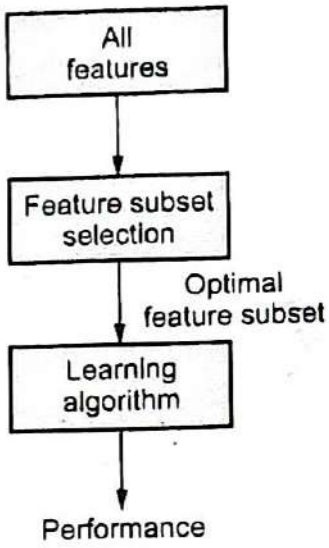


Fig. 2.7.2 Filter method

2. Wrapper methods :

- In wrapper methods, the Learner is considered a black-box. Interface of the black-box is used to score subsets of variables according to the predictive power of the learner when using the subsets.
- Fig. 2.7.3 shows wrapper method.
- The feature selection algorithm searches for a good feature subset using the induction algorithm itself as a part of the evaluation function.
- Wrapper methods are recursive feature elimination, sequential feature selection algorithms and genetic algorithms.

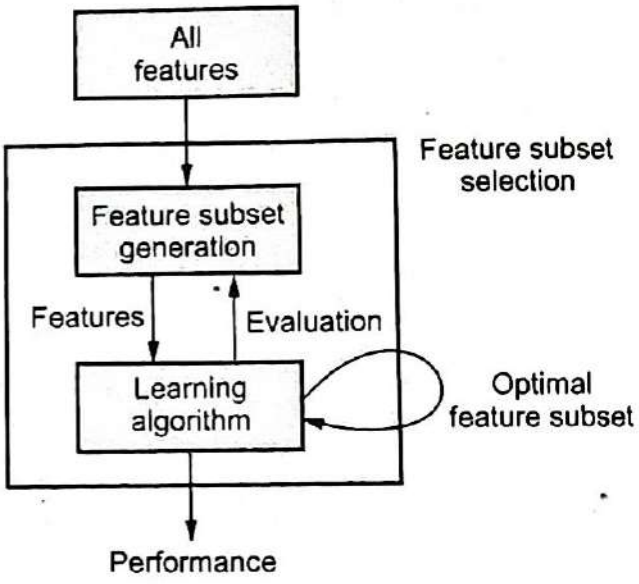


Fig. 2.7.3 Wrapper method

3. Embedded methods :

- Embedded methods, are quite similar to wrapper methods since they are also used to optimize the objective function or performance of a learning algorithm or model.
- It's implemented by algorithms that have their own feature selection methods in them.

- A learning algorithm takes advantage of its own variable selection process and performs feature selection and classification/regression at the same time.
- The most Common embedded technique are the tree algorithm's like Random Forest. LASSO with the L1 penalty and Ridge with the L2 penalty for constructing a linear model.
- Tree algorithms select a feature in each recursive step of the tree growth process and divide the sample set into smaller subsets. The more child nodes in a subset are in the same class, the more informative the features are.

2.7.1 Difference between Filter, Wrapper and Embedded Method

Filter methods	Wrapper methods	Embedded methods
Generic set of methods which do not incorporate a specific machine learning algorithm.	Evaluates on a specific machine learning algorithm to find optimal features.	Embeds (fix) features during model building process. Feature selection is done by observing each iteration of model training phase.
Much faster compared to Wrapper methods in terms of time complexity.	High computation time for a dataset with many features.	Sits between Filter methods and Wrapper methods in terms of time complexity.
Less prone to over-fitting.	High chances of over-fitting because it involves training of machine learning models with different combination of features.	Generally used to reduce over-fitting by penalizing the coefficients of a model being too large.
Examples - Correlation, Chi-square test, ANOVA, Information gain, etc.	Examples - Forward Selection, Backward elimination, Stepwise Selection, etc.	Examples - LASSO, Elastic Net, Ridge Regression, etc.

2.8 Statistical Feature Engineering

- Feature engineering refers to the process of using domain knowledge to select and transform the most relevant variables from raw data when creating a predictive model using machine learning or statistical modeling. The goal of feature engineering and selection is to improve the performance of machine learning algorithms.
- Feature engineering typically includes feature creation, feature transformation, feature extraction, and feature selection.
- **Count-based** feature selection is a simple yet relatively powerful way of finding information about predictors. The basic idea underlying count-based featurization is simple: by calculating counts of individual values within a column, user can get an idea of the distribution and weight of values, and from this, understand which columns contain the most important information.

2.8.1 Mean, Median and Mode

1. Mean :

- The mean of a data set is the average of all the data values. The sample mean $\bar{x}$ is the point estimator of the population mean μ.

$$\text{Sample mean } \bar{x} = \frac{\text{Sum of the values of then observations}}{\text{Number of observations in the sample}} = \frac{\sum x_i}{n}$$

$$\text{Population mean } \mu = \frac{\text{Sum of the values of then N observations}}{\text{Number of observations in the population}} = \frac{\sum x_i}{n}$$

2. Median :

- The median of a data set is the value in the middle when the data items are arranged in ascending order. Whenever a data set has extreme values, the median is the preferred measure of central location.
- The median is the measure of location most often reported for annual income and property value data. A few extremely large incomes of property values can inflate the mean.
- For an odd number of observations :

7 observations = 26, 18, 27, 12, 14, 29, 19

Numbers in ascending order = 12, 14, 18, 19, 26, 27, 29

- The median is the middle value.

Median = 19

- For an even number of observations :

8 observations = 26, 18, 29, 12, 14, 27, 30, 19

Numbers in ascending order = 12, 14, 18, 19, 26, 27, 29, 30

The median is the average of the middle two values.

$$\text{Median} = \frac{(19+26)}{2} = 22.5$$

3. Mode :

- The mode of a data set is the value that occurs with greatest frequency. The greatest frequency can occur at two or more different values. If the data have exactly two modes, the data are bimodal. If the data have more than two modes, the data are multimodal.

Example 2.8.1 *Time between failures (in hours) of a wire cutter used in a cookie manufacturing oven is given in table. The function of the wire-cut is to cut the dough into cookies of desired size.*

The between failure of wire-cut (in hours)

2	22	32	39	46	56	76	79	88	93
3	24	33	44	46	66	77	79	89	99
5	24	34	45	47	67	77	86	89	99
9	26	37	45	55	67	78	86	89	99
21	31	39	46	56	75	78	87	90	102

a) Calculate the mean, median and mode of time between failures of wire-cuts.

Solution :

	2	22	32	39	46	56	76	79	88	93
	3	24	33	44	46	66	77	79	89	99
	5	24	34	45	47	67	77	86	89	99
	9	26	37	45	55	67	78	86	89	99
	21	31	39	46	56	75	78	87	90	102
Total	**40**	**127**	**175**	**228**	**250**	**331**	**386**	**417**	**445**	**492**

$$\text{Mean} = \frac{40+127+175+228+250+331+386+417+445+492}{50} = \mathbf{57.82}$$

Median = 56

Mode = 24, 39, 45, 46, 56, 67, 77, 78, 79, 86, 89, 99

2.9 Multidimensional Scaling

- Multidimensional Scaling (MDS) is a non-linear technique for embedding data in a lower-dimensional space. It maps points residing in a higher-dimensional space to a lower-dimensional space while preserving the distances between those points as much as possible. Because of this, the pairwise distances between points in the lower-dimensional space are matched closely to their actual distances.
- Multidimensional scaling techniques are used for dimensionality reduction when the input data is not linearly arranged or it is not known whether a linear relationship exists or not. They are typically iterative and aim to minimize the

difference between the distances between the pairs of points in the original input data and the distances between the corresponding pairs of points in the lower-dimensional output data.

- MDS can be used as a preprocessing step for dimensionality reduction in classification and regression problems.
- Normally the distance measure used in MDS is the Euclidean distance, however, any other suitable dissimilarity metric can be used when applying MDS. There are two main ways to implement MDS :
  1. **Metric MDS/Classical MDS** : This version of MDS aims to preserve the pairwise distance/dissimilarity measure as much as possible.
  2. **Non-Metric MDS** : This method is applicable when only the ranks of a dissimilarity metric are known. MDS then maps the objects so that the ranks are preserved as much as possible.
- Suppose there are n-entities and $n(n-1)/2$ pairs with each pair having a measure of distance. The distance is a function of many variables for each entity. MDS takes the pairwise distances between the entities and finds best-fit representations of the points in all lower dimensional spaces.
- Consider n-entities, i = 1, ... , n and K-variables, k = 1, ..., K. A simple standardization is achieved by :

$$x_{k,i} = \frac{z_{k,i} - \mu_k}{\sigma_k}$$

where $z_{k,i}$ denotes the k^{th} variable of the i^{th} data in original units, $x_{k,i}$ is the standardized data, μ_k and σ_k^2 are the mean and variance of the k = 1, ..., K variables. Once standardized, each variable has a mean of zero and a standard deviation of one.

- The distance between the different entities can be calculated by the Euclidean distance, correlation coefficients, or another method. The Euclidean distance is common :

$$d_{ij} = \sqrt{\sum_{k=1}^{K} (x_{k,i} - x_{k,j})^2} \text{ for } i, j = 1, \ldots, n$$

where d_{jj} is the Euclidean distance between entity-i and entity-j for the K variables being considered.

2.10 Matrix Factorization Techniques

SPPU : June-22

- Matrix factorization techniques have become a dominant methodology within collaborative filtering recommenders. Experience with datasets such as the Netflix

Prize data has shown that they deliver accuracy superior to classical nearest-neighbor techniques.

- Recommender systems rely on different types of input data, which are often placed in a matrix with one dimension representing users and the other dimension representing items of interest. Most websites like Amazon, YouTube, and Netflix use collaborative filtering as a part of their sophisticated recommendation systems
- Recommender systems are based on one of two strategies : Content filtering and collaborative filtering.
- The content filtering approach creates a profile for each user or product to characterize its nature. Collaborative filtering analyzes relationships between users and interdependencies among products to identify new user-item associations.
- Matrix factorization is a way to generate latent features when multiplying two different kinds of entities. Collaborative filtering is the application of matrix factorization to identify the relationship between items' and users' entities. With the input of users' ratings on the shop items, we would like to predict how the users would rate the items so the users can get the recommendation based on the prediction.
- Matrix factorization is the act of decomposing a matrix into the product of two or more matrices. Matrices appear all over the place in data science applications. For example, the input to many classical models such as random forest is generally a structured two-dimensional dataset, with the columns representing the different features and the rows representing the samples. This dataset can be thought of as a two-dimensional matrix.
- Matrix factorization models are superior to classic nearest-neighbor techniques for producing product recommendations, allowing the incorporation of additional information such as implicit feedback, temporal effects, and confidence levels.
- Matrix factorization characterizes both items and users by vectors of factors inferred from item rating patterns. High correspondence between item and user factors leads to a recommendation.
- When a user gives feedback to a certain movie they saw, this collection of feedback can be represented in a form of a matrix. Where each row represents each users, while each column represents different movies. Obviously the matrix will be sparse since not everyone is going to watch every movies. Fig. 2.10.1 shows Matrix factorization.

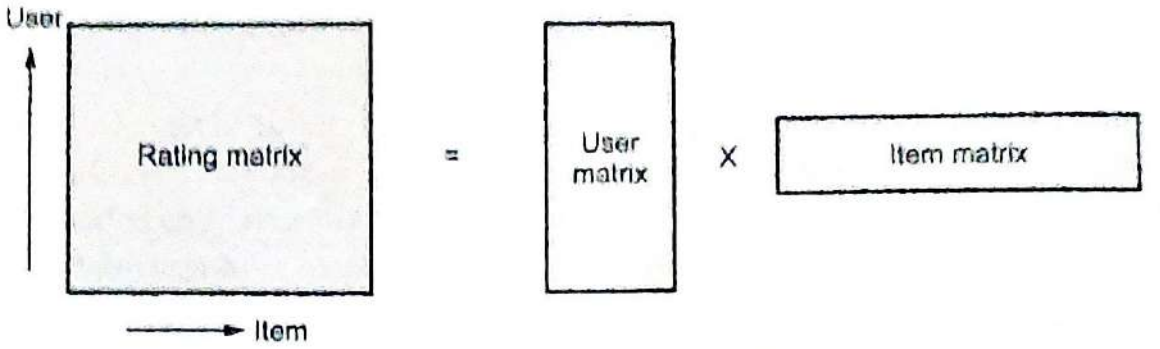


Fig. 2.10.1 Matrix factorization

- Matrix factorization can be seen as breaking down a large matrix into a product of smaller ones. This is similar to the factorization of integers, where 16 can be written as 8×2 or 4×4. In the case of matrices, a matrix A with dimensions $m \times n$ can be reduced to a product of two matrices X and Y with dimensions $m \times p$ and $p \times n$ respectively.
- The two columns in the user matrix and the two rows in the item matrix are called latent factors and are an indication of hidden characteristics about the users or the items. The reduced matrices actually represent the users and items individually.
- The m rows in the first matrix represent the m users, and the p columns tell you about the features or characteristics of the users. The same goes for the item matrix with n items and p characteristics. Here's an example of how matrix factorization looks :

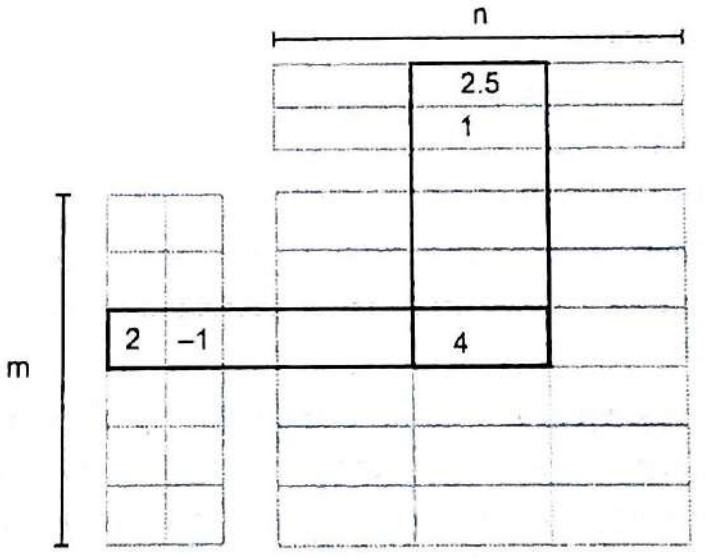


Fig. 2.10.2 Working of matrix factorization

- In the above Fig. 2.10.2, the matrix is reduced into two matrices. The one on the left is the user matrix with m users, and the one on top is the item matrix with n items. The rating 4 is reduced or factorized into :
  1. A user vector (2, – 1)
  2. An item vector (2.5, 1)

- The two columns in the user matrix and the two rows in the item matrix are called latent factors and are an indication of hidden characteristics about the users or the items. A possible interpretation of the factorization could look like this :
  a) Assume that in a user vector (u, v), u represents how much a user likes the horror movie and v represents how much they like the romance movie.
  b) The user vector (2, – 1) thus represents a user who likes horror movies and rates them positively and dislikes movies that have romance and rates them negatively.
  c) Assume that in an item vector (i, j), i represents how much a movie belongs to the Horror movie, and j represents how much that movie belongs to the Romance movigenre.
  d) The movie (2.5, 1) has a Horror rating of 2.5 and a Romance rating of 1. Multiplying it by the user vector using matrix multiplication rules give, $(2 * 2.5) + (-1 * 1) = 4$.
- So, the movie belonged to the Horror genre, and the user could have rated it 5, but the slight inclusion of Romance caused the final rating to drop to 4.

2.10.1 Naïve User Based Recommendation System

- Naïve Bayes is a probabilistic approach to inductive learning, and belongs to the general class of Bayesian classifiers. These approaches generate a probabilistic model based on previously observed data.
- Let us assume, a set of users represented by feature vectors.

$$U = \{\bar{u}_1, \bar{u}_2, ..., \bar{u}_n\} \text{ where } \bar{u}_n \in R^n$$

- The set of items are represented by

$$I = \{i_1, i_2, ..., i_m\}$$

- Let's assume also that there is a relation which associates each user with a subset of items, items for which an explicit action or feedback has been performed :

$$g(\bar{u}) \rightarrow \{i_1, i_2, ..., i_k\} \text{ where } k \in (0, m)$$

- In a user-based system, the users are periodically clustered and therefore, considering a generic user u, we can immediately determine the ball containing all the users who are similar to our sample :

$$B_R(\bar{u}) = \{\bar{u}_1, \bar{u}_2, ..., \bar{u}_k\}$$

- Each user has expressed an opinion for some items :
  a) Explicit opinion : Rating score
  b) Implicit : Purchase records or listen to tracks

- Target (or Active) user for whom the CF recommendation task is performed.
  1. Identify set of items rated by the target user.
  2. Identify which other users rated 1+ items in this set (neighborhood formation).
  3. Compute how similar each neighbor is to the target user (similarity function)
  4. In case, select k most similar neighbors.
  5. Predict ratings for the target user's unrated items (prediction function)
  6. Recommend to the target user the top N products based on the predicted ratings

Review Question

1. *Explain with example Naïve user based recommendation systems.*

SPPU : June-22, End Sem, Marks 8

❑❑❑