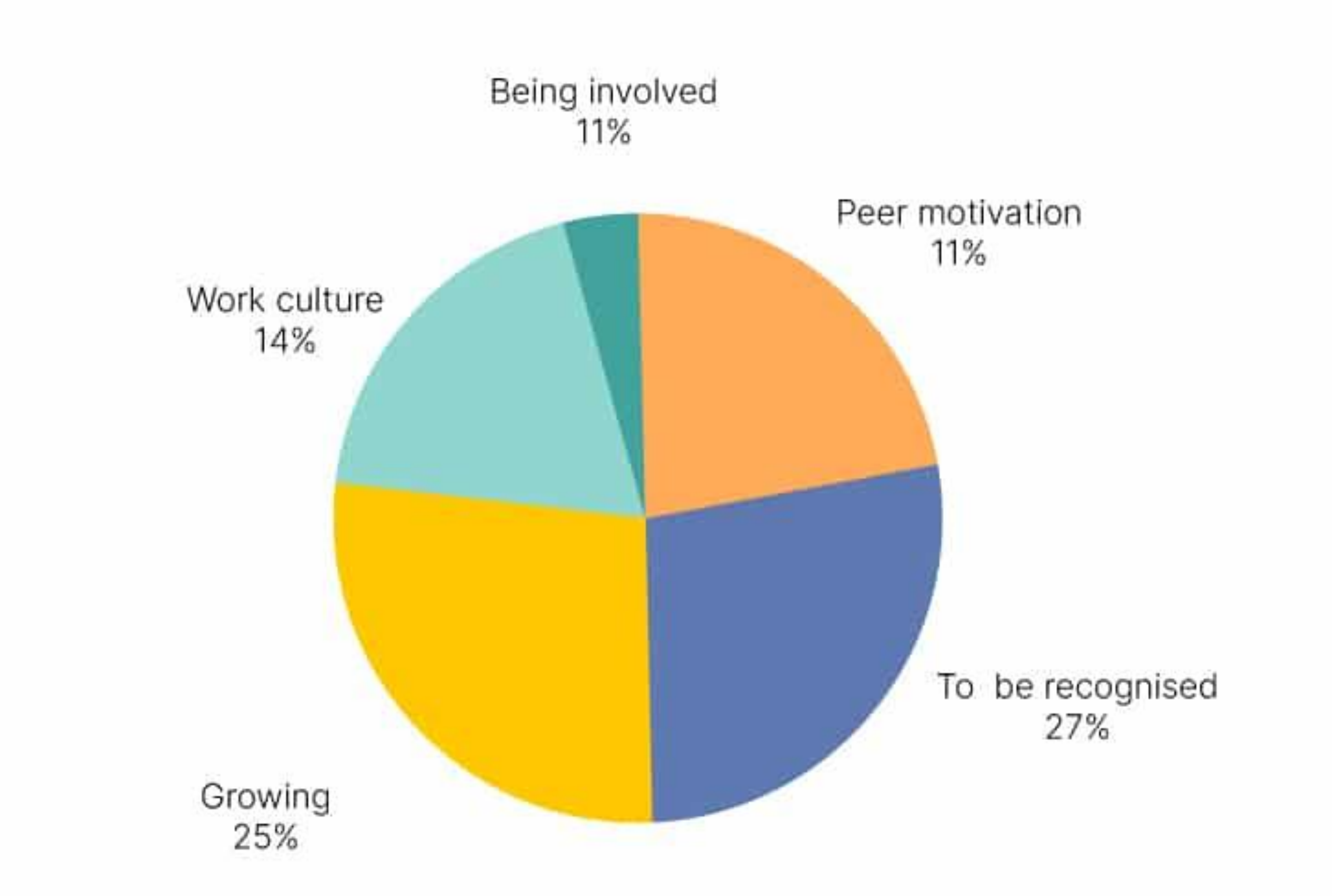
Machine Learning

1)Categorial data vs Numerical data

What is categorical data?

Categorical data refers to a data type that can be stored and identified based on the names or labels given to them. A process called matching is done, to draw out the similarities or relations between the data and then they are grouped accordingly.

The data collected in the categorical form is also known as qualitative data. Each dataset can be grouped and labelled depending on their matching qualities, under only one category. This makes the categories mutual exclusive.



Example: sexuality is categorical data, as a person can be straight, homosexual, heterosexual, etc. and they are grouped together depending on the common characteristics possessed by them.

By: Adham Magdy

There are two subtypes of categorical data namely: Nominal data and Ordinal data.

1. **Nominal data** – this is also called naming data. This is a type that names or labels the data and its characteristics are similar to a noun. Example: person’s name, gender, school name.

Questions to gather nominal data look like:

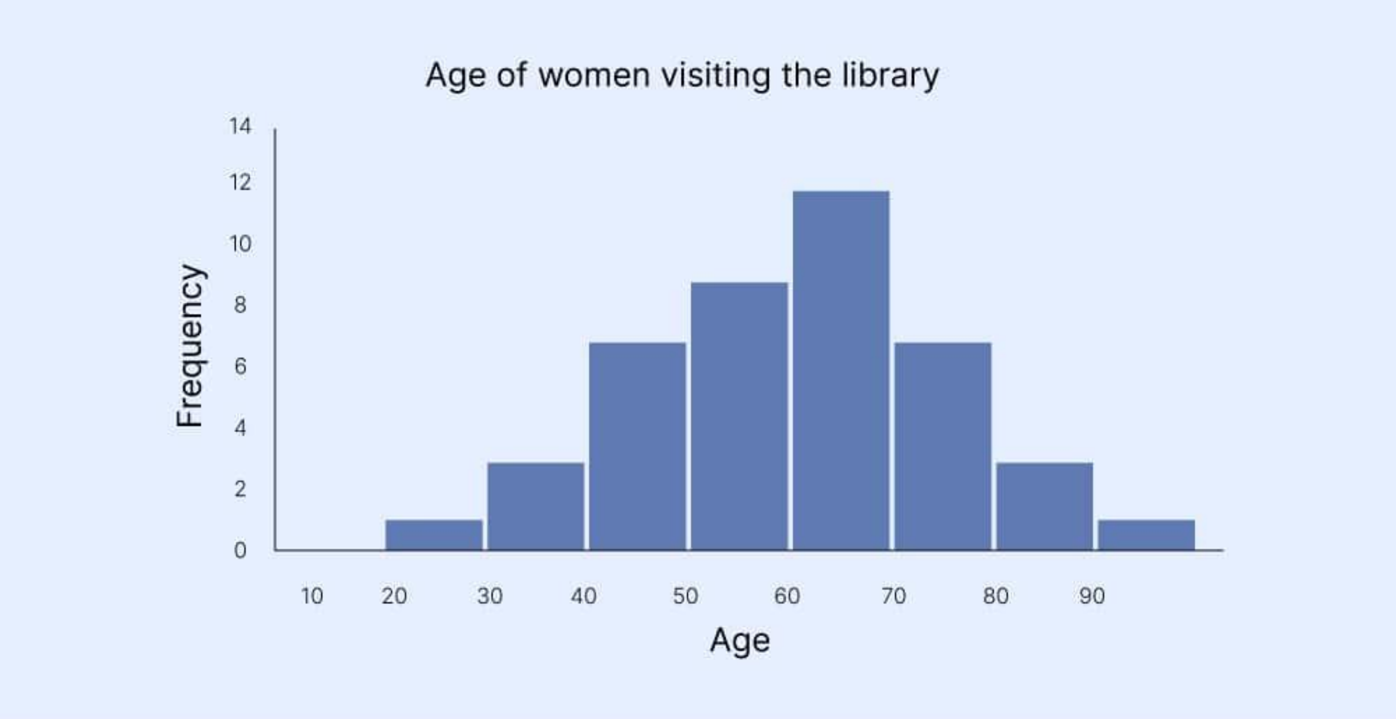
- What is your name?
- What is your pet’s name?
- What is your gender?

1. **Ordinal data** – this includes data or elements of data that is ranked, ordered or used on a rating scale. You can count and order ordinal data but it doesn’t allow you to measure it.

Example: seminar attendants are asked to rate their seminar experience on a scale of 1-5. Against each number, there will be options that will rate their satisfaction like “very good, good, average, bad, and very bad”.

What is numerical data?

Numerical data refers to the data that is in the form of numbers, and not in any language or descriptive form. Often referred to as quantitative data, numerical data is collected in number form and stands different from any form of number data types due to its ability to be statistically and arithmetically calculated.



It doesn’t involve any natural language description and is quantitative in nature and it is used to measure quantities like a person’s height, age, IQ, etc.

It also has two subtypes known as Discrete data and Continuous data.

1. **Discrete data** – Discrete data is used to represent countable items. It can take both numerical and categorical forms and group them into a list. This list can be finite or infinite too.

Discrete data basically takes countable numbers like 1, 2, 3, 4, 5, and so on. In the case of infinity, these numbers will keep going on.

Example: counting sugar cubes from a jar is finite countable. But counting sugar cubes from all over the world is infinite countable.

1. **Continuous data** – As the name says, this form has data in the form of intervals. Or simply said ranges. Continuous numerical data represent measurements and their intervals fall on a number line. Hence, it doesn’t involve taking counts of the items.

Example: in a school exam, students who scored 80%-100% come under distinction, 60%-80% have first-class and below 60% are second class.

Continuous data is further divided into two categories: Interval and Ratio.

- **Interval data** – interval data type refers to data that can be measured only along a scale at equal distances from each other. The numerical values in this data type can only undergo add and subtract operations. **Example**: body temperature can be measured in degree Celsius and degree Fahrenheit and neither of them can be 0.

- **Ratio data** – unlike interval data, ratio data has zero points. Being similar to interval data, zero point is the only difference they have. **Example**: in the body temperature, the zero point temperature can be measured in Kelvin.

15 differences between Categorical data and Numerical data

Features	Categorical data	Numerical data
Definition	Categorical data refers to a data type that can be stored and identified based on the names or labels given to them.	Numerical data refers to the data that is in the form of numbers, and not in any language or descriptive form.
Alias	Also known as qualitative data as it qualifies data before classifying it.	Also known as quantitative data as it represents quantitative values to perform arithmetic operations on them.
Examples	What is your gender? • Male • Female • Other	What is your test score out of 20? • Below 5 • 5-10 • 10-15 • 15-20 • 20
Types	Nominal data and Ordinal data.	Discrete data and Continuous data.
Characteristics	• No order scale • Natural language description • Can take numerical values but with qualitative properties • Can be visualized using bar charts and pie charts	• Has an ordered scale • Not use of natural language description • Takes numeric values with numeric qualities • Can be visualized using bar charts and pie charts

By: Adham Magdy

User-friendly design	Can include long surveys and has a chance of pushing respondents away.	Survey interaction is easy and short, hence fewer survey abandonment issues.
Data collection method	Nominal data: open-ended questions Ordinal data: multiple-choice questions	Mostly collected through multiple-choice questions and sometimes through open-ended questions.
Data collection tools	Questionnaires, surveys, and interviews	Questionnaires, surveys, interviews, focus groups and observations
Analysis and interpretation	Median and mode Eg: univariate statistics, bivariate statistics, regression analysis	Descriptive and inferential statistics Eg: measures of central tendency, turf analysis, text analysis, conjoint analysis, trend analysis
Uses	Used when a study requires respondents' personal information, opinions and experiences. Commonly used in business research	Used for statistical calculations as a result of the potential performance of arithmetic operations
Advantages	• Provides intuitive representation of the data • To gain a deeper knowledge of a topic or a population • Respondent dependent data	• Supports statistical calculations • Commonly used by researchers
Disadvantages	• Large data to process and analyze • The researcher may have to handle irrelevant data	• Standardized performance limits investigation • Significant factor can be eliminated which alters the results • The researcher has more control over the data than respondents

By: Adham Magdy

Compatibility	It is not compatible with most statistical analysis methods, hence researchers avoid using it most of the times	It is compatible with most statistical calculation methods.
Visualization	Can be visualized using only bar graphs and pie charts.	Can be visualized using bar graphs, pie charts as well as scatter plots.
Structure	Is known as unstructured or semi-structured data It can use indexing methods to structure data like Google, Bing, etc.	It is structured data and can be quickly organized and made sense of

2)Basic Statistics

Mean :

Mean is also known as average of all the numbers in the data set which is calculated by below equation.

$$\text{Mean} = \frac{\text{Sum of all data values}}{\text{Number of data values}}$$

Symbolically,

$$\bar{x} = \frac{\sum x}{n}$$

where $\bar{x}$ (read as 'x bar') is the mean of the set of x values,

$\sum x$ is the sum of all the x values, and

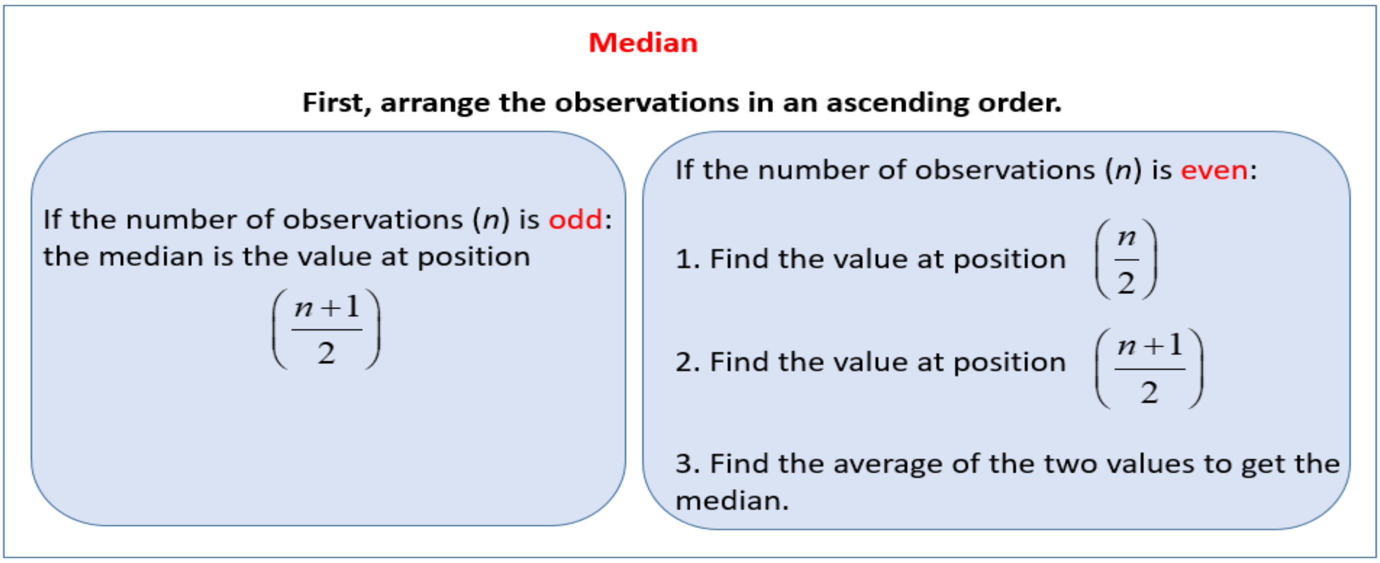
n is the number of x values.

Lets say we have below heights of persons.

heights=[168,170,150,160,182,140,175,191,152,150]

Median :

Median is mid value in this ordered data set.



Median

First, arrange the observations in an ascending order.

If the number of observations (n) is odd: the median is the value at position $\left(\frac{n+1}{2}\right)$

If the number of observations (n) is even:

1. Find the value at position $\left(\frac{n}{2}\right)$
2. Find the value at position $\left(\frac{n+1}{2}\right)$
3. Find the average of the two values to get the median.

Mode :

Mode is the number which occur most often in the data set

Variance :

Variance is the numerical values that describe the variability of the observations from its arithmetic mean and denoted by sigma-squared(σ2)

Variance measure how far individuals in the group are spread out, in the set of data from the mean.

$$\sigma^2 = \frac{\sum_{i=1}^{n}(X_i - \mu)^2}{n}$$

Where

Xi : Elements in the data set

mu : the population mean

Standard Deviation :

It is a measure of dispersion of observation within dataset relative to their mean.It is square root of the variance and denoted by Sigma (σ) .

Standard deviation is expressed in the same unit as the values in the dataset so it measure how much observations of the data set differs from its mean.

$$\sigma = \sqrt{\frac{\sum(x_i - \mu)^2}{N}}$$

σ = population standard deviation

N = the size of the population

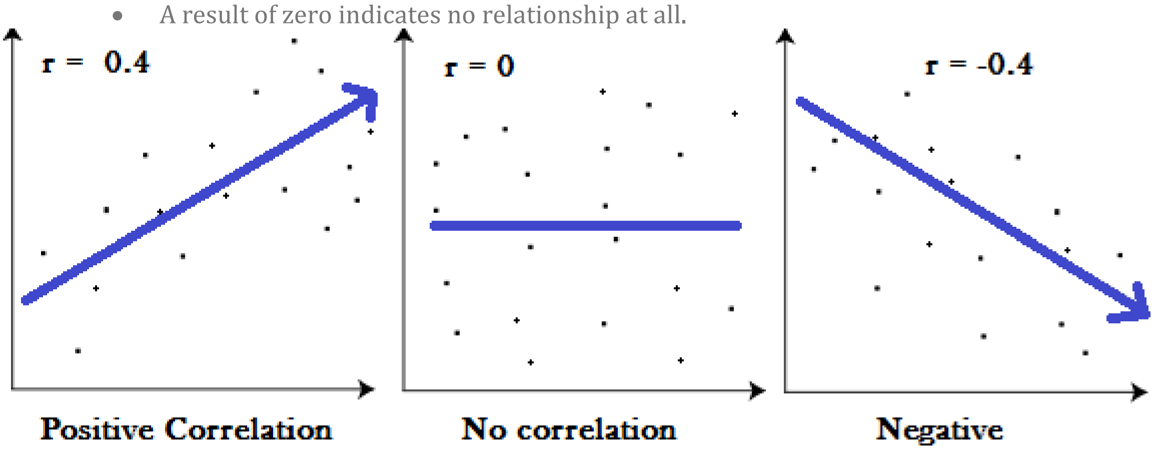
x_i = each value from the population

μ = the population mean

Correlation Coefficient Formula: Definition

Correlation coefficient formulas are used to find how strong a relationship is between data. The formulas return a value between -1 and 1, where:

- 1 indicates a strong positive relationship.
- -1 indicates a strong negative relationship.
- A result of zero indicates no relationship at all.



Graphs showing a correlation of -1, 0 and +1

By: Adham Magdy

Meaning

- A correlation coefficient of 1 means that for every positive increase in one variable, there is a positive increase of a fixed proportion in the other. For example, shoe sizes go up in (almost) perfect correlation with foot length.
- A correlation coefficient of -1 means that for every positive increase in one variable, there is a negative decrease of a fixed proportion in the other. For example, the amount of gas in a tank decreases in (almost) perfect correlation with speed.
- Zero means that for every increase, there isn't a positive or negative increase. The two just aren't related.

The absolute value of the correlation coefficient gives us the relationship strength. The larger the number, the stronger the relationship. For example, |-.75| = .75, which has a stronger relationship than .65.

Like the explanation? Check out the Practically Cheating Statistics Handbook, which has hundreds of step-by-step, worked out problems!

Types of correlation coefficient formulas.

There are several types of correlation coefficient formulas.

One of the most commonly used formulas is Pearson's correlation coefficient formula. If you're taking a basic stats class, this is the one you'll probably use:

$$r = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{\sqrt{[n\Sigma x^2 - (\Sigma x)^2][n\Sigma y^2 - (\Sigma y)^2]}}$$

Pearson correlation coefficient

Two other formulas are commonly used: the sample correlation coefficient and the population correlation coefficient.

Sample correlation coefficient

$$r_{xy} = \frac{s_{xy}}{s_x s_y}$$

S_x and s_y are the sample standard deviations, and s_{xy} is the sample covariance.

Population correlation coefficient

$$\rho_{xy} = \frac{\sigma_{xy}}{\sigma_x \sigma_y}$$

The population correlation coefficient uses σ_x and σ_y as the population standard deviations, and σ_{xy} as the population covariance.

By: Adham Magdy

3) supervised vs unsupervised

What is supervised learning?

Supervised machine learning requires labelled input and output data during the training phase of the machine learning lifecycle. This training data is often labelled by a data scientist in the preparation phase, before being used to train and test the model. Once the model has learned the relationship between the input and output data, it can be used to classify new and unseen datasets and predict outcomes.

The reason it is called supervised machine learning is because at least part of this approach requires human oversight. The vast majority of available data is unlabelled, raw data. Human interaction is generally required to accurately label data ready for supervised learning. Naturally, this can be a resource intensive process, as large arrays of accurately labelled training data is needed.

Supervised machine learning is used to classify unseen data into established categories and forecast trends and future change as a predictive model. A model developed through supervised machine learning will learn to recognise objects and the features that classify them. Predictive models are also often trained with supervised machine learning techniques. By learning patterns between input and output data, supervised machine learning models can predict outcomes from new and unseen data. This could be in forecasting changes in house prices or customer purchase trends.

Supervised machine learning is often used for:

- Classifying different file types such as images, documents, or written words.
- Forecasting future trends and outcomes through learning patterns in training data.

What is unsupervised learning?

Unsupervised machine learning is the training of models on raw and unlabelled training data. It is often used to identify patterns and trends in raw datasets, or to cluster similar data into a specific number of groups. It's also often an approach used in the early exploratory phase to better understand the datasets.

As the name suggests, unsupervised machine learning is more of a hands-off approach compared to supervised machine learning. A human will set model hyperparameters such as the number of cluster points, but the model will process huge arrays of data effectively and without human oversight. Unsupervised machine learning is therefore suited to answer questions about unseen trends and relationships within data itself. But because of less human oversight, extra consideration should be made for the explainability of unsupervised machine learning.

The vast majority of available data is unlabelled, raw data. By grouping data along similar features or analysing datasets for underlying patterns, unsupervised learning is a powerful tool used to gain insight from this data. In contrast, supervised machine learning can be resource intensive because of the need for labelled data.

Unsupervised machine learning is mainly used to:

- Cluster datasets on similarities between features or segment data
- Understand relationship between different data point such as automated music recommendations
- Perform initial data analysis

Supervised vs unsupervised learning compared

The main difference between supervised vs unsupervised learning is the need for labelled training data. Supervised machine learning relies on labelled input and output training data, whereas unsupervised learning processes unlabelled or raw data. In supervised machine learning the model learns the relationship between the labelled input and output data. Models are finetuned until they can accurately predict the outcomes of unseen data. However, labelled training data will often be resource intensive to create. Unsupervised machine learning on the other hand learns from unlabelled raw training data. An unsupervised model will learn relationships and patterns within this unlabelled dataset, so is often used to discover inherent trends in a given dataset.

So overall, supervised and unsupervised machine learning are different in the approach to training and the data the model learns from. But as a result, they also differ in their final application and specific strengths. Supervised machine learning models are generally used to predict outcomes for unseen data. This could be predicting fluctuations in house prices or understanding the sentiment of a message.

Models are also used to classify unseen data against learned patterns. On the other hand, unsupervised machine learning techniques are generally used to understand patterns and trends within unlabelled data. This could be clustering data due to similarities or differences, or identifying underlying patterns within datasets. Unsupervised machine learning can be used to cluster customer data in marketing campaigns, or to detect anomalies and outliers.

The main differences of supervised vs unsupervised learning include:

- The need for labelled data in supervised machine learning.
- The problem the model is deployed to solve. Supervised machine learning is generally used to classify data or make predictions, whereas unsupervised learning is generally used to understand relationships within datasets.
- Supervised machine learning is much more resource-intensive because of the need for labelled data.
- In unsupervised machine learning it can be more difficult to reach adequate levels of explainability because of less human oversight.

Supervised vs unsupervised learning examples

A main difference between supervised vs unsupervised learning is the problems the final models are deployed to solve. Both types of machine learning model learn from training data, but the strengths of each approach lie in different applications. Supervised machine learning will learn the relationship between input and output through labelled training data, so is used to classify new data using these learned patterns or in predicting outputs.

Unsupervised machine learning on the other hand is useful in finding underlying patterns and relationships within unlabelled, raw data. This makes it particularly useful for exploratory data analysis, segmenting or clustering of datasets, or projects to understand how data features connect to other features for automated recommendation systems.

Examples of supervised machine learning include:

- Classification, identifying input data as part of a learned group.
- Regression, predicting outcomes from continuously changing data.

Examples of unsupervised machine learning include:

- Clustering, grouping together data points with similar data.
- Association, understanding how certain data features connect with other features.

Here we explore the main applications of supervised vs unsupervised learning, including examples of specific algorithms in action today.

Examples of supervised learning classification

A classification problem in machine learning is when a model is used to classify whether data belongs to a known group or object class. Models will assign a class label to the data it processes, which is learned by the algorithm through training on labelled training data. The input and output of the data has been labelled, so the model can understand which features will classify an object or data point with different class labels. The need for labelled data in the training phase means this is a supervised machine learning process.

Examples of how classification models are used include:

- Spam detection as part of an email firewall.
- Identifying and classifying objects in an image file type.
- Speech recognition and facial recognition software.
- Automated classification of documents and writing.
- Analysing the sentiment of written language and messages.

There are different types of classification problems, which are generally different depending on the count of class labels that are applied to the data in a live environment.

The main classification problems include:

- Binary classification
- Multiple class classification
- Multiple label classification

Binary classification

Binary classification is when a model can apply only two class labels. A popular use of a binary classification would be in detecting and filtering junk emails. A model can be trained to label incoming emails as either junk or safe, based on learned patterns of what constitutes a spam email.

Binary classification is commonly performed by algorithms such as:

- Logistic Regression
- Decision Trees
- Naïve Bayes

Multiple class classification

Multiple class classification is when models reference more than the two class labels found in binary classification. Instead, there could be a huge array of possible class labels that could be applied to the object or data. An example would be in facial recognition software, where a model may analyse an image against a huge range of possible class labels to identify the individual.

Multiple class classification is commonly performed by algorithms such as:

- Random Forest
- k-Nearest Neighbors
- Naive Bayes

Multiple label classification

Multiple label classification is when an object or data point may have more than one class label assigned to it by the machine learning model. In this case the model will usually have multiple outputs. An example could be in image classification which may contain multiple objects. A model can be trained to identify, classify and label a range of subjects in one image.

Multiple label classification is commonly performed by algorithms such as:

- Multiple label Gradient Boosting
- Multiple label Random Forests
- Using different classification algorithms for each class label

Examples of supervised learning regression

Another common use of supervised machine learning models is in predictive analytics. Regression is commonly used as the process for a machine learning model to predict continuous outcomes. A supervised machine learning model will learn to identify patterns and relationships within a labelled training dataset. Once the relationship between input data and expected output data is understood, new and unseen data can be processed by the model. Regression is therefore used in predictive machine learning models, which could be used to:

- Forecast stock or trading outcomes and market fluctuations, a key role of machine learning in finance.
- Predict the success of marketing campaigns so organisations can assign and refine resources.

- Forecast changes in market value in sectors like retail or the housing market.
- Predict changes in health trends in a demographic or area.

Common algorithms used in supervised learning regression include:

- Simple Linear Regression
- Decision tree Regression

Simple Linear Regression

Simple Linear Regression is a popular type of regression approach and is used to predict target output from an input variable. A linear connection between the input and target output should be present. Once a model has been trained on the relationship between the input and target output, it can be used to make predictions on new data. Examples might be predicting salary based on age and gender.

Decision Tree Regression

As the name suggests, Decision Tree models take the structure of a tree in which the model incrementally branches. Decision Trees are a popular form of supervised machine learning, and can be used for both regression and classification. The dataset is broken down into incremental subsets, and can be used to understand the correlation between independent variables. The resulting model can then be used to predict output based on new data.

Examples of unsupervised learning clustering

Clustering is the grouping together of data points into a determined number of categories depending on similarities (or differences) between data points. This way raw and unlabelled data can be processed and clustered depending on the patterns within the dataset. Hyperparameters set by the data scientist will usually define the overall count of clusters.

Clustering is a popular use of unsupervised learning models and can be used to understand trends and groupings in raw data. The approach can also highlight data points that sit outside of the groupings, making it an important tool for anomaly detection.

Clustering as an approach can be used to:

- Segment audience or customer data into groups in marketing environments.
- Perform initial exploratory analysis on raw datasets to understand the grouping of data points.
- Detect outliers and anomalies that sit outside of clustered data.

Common approaches to unsupervised learning clustering include:

- K-means clustering
- Gaussian Mixture Models

K-means clustering

K-means clustering is a popular method for clustering data. K represents the count of clusters, set by the data scientist. Clusters are defined by the distance from the centre of each grouping. A higher count of clusters means more granular groupings, and a lower count of clusters means less granular groupings. This method can be used to identify exclusive or overlapping clusters. Exclusive clustering means each data point can belong to only one cluster. Overlapping clustering means data can be within multiple clusters.

Gaussian Mixture Models

Gaussian Mixture Models is an example of an approach to probabilistic clustering, in which data points are grouped based on the probability that they belong to a defined grouping. This approach uses probabilities in the data to map data points to each cluster, in contrast to K-means clustering which uses distance from the centre of the cluster.

Examples of unsupervised learning association rules

Association is the discovery of the relationships between different variables, to understand how data point features connect with other features. This means that the relationship between different data points can be mapped and understood. A key example is in the automated recommendation tools found in ecommerce or news websites. An unsupervised algorithm can be used to analyse customer or user behaviour and recommend products to similar users.

A popular method of forming association rules is the Apriori algorithm, which works by identifying trends in a database based on frequency. This approach can be applied to retail product purchases or engagement with film streaming services.

Unsupervised machine learning association rules can be used to:

- Recommend products and services to customers depending on their buying habits.
- Recommend media like songs, films, or TV programmes based on user interests or behaviour.
- Understand habits and interests of customers to inform e-commerce or marketing campaigns.

4) Genetic algorithm

A genetic algorithm is a search-based algorithm used for solving optimization problems in machine learning. This algorithm is important because it solves difficult problems that would take a long time to solve. It has been used in various real-life applications such as data centers, electronic circuit design, code-breaking, image processing, and artificial creativity.

This article will take the reader through the basics of this algorithm and explains how it works. It also explains how it has been applied in various fields and highlights some of its limitations.

Genetic algorithm (GA) explained

The following are some of the basic terminologies that can help us to understand genetic algorithms:

- **Population:** This is a subset of all the probable solutions that can solve the given problem.
- **Chromosomes:** A chromosome is one of the solutions in the population.
- **Gene:** This is an element in a chromosome.
- **Allele:** This is the value given to a gene in a specific chromosome.
- **Fitness function:** This is a function that uses a specific input to produce an improved output. The solution is used as the input while the output is in the form of solution suitability.
- **Genetic operators:** In genetic algorithms, the best individuals mate to reproduce an offspring that is better than the parents. Genetic operators are used for changing the genetic composition of this next generation.

A genetic algorithm (GA) is a heuristic search algorithm used to solve search and optimization problems. This algorithm is a subset of evolutionary algorithms, which are used in computation. Genetic algorithms employ the concept of genetics and natural selection to provide solutions to problems.

These algorithms have better intelligence than random search algorithms because they use historical data to take the search to the best performing region within the solution space.

GAs are also based on the behavior of chromosomes and their genetic structure. Every chromosome plays the role of providing a possible solution. The fitness function helps in providing the characteristics of all individuals within the population. The greater the function, the better the solution.

Advantages of genetic algorithm

- It has excellent parallel capabilities.
- It can optimize various problems such as discrete functions, multi-objective problems, and continuous functions.
- It provides answers that improve over time.
- A genetic algorithm does not need derivative information.

How genetic algorithms work

Genetic algorithms use the evolutionary generational cycle to produce high-quality solutions. They use various operations that increase or replace the population to provide an improved fit solution.

Genetic algorithms follow the following phases to solve complex optimization problems:

Initialization

The genetic algorithm starts by generating an initial population. This initial population consists of all the probable solutions to the given problem. The most popular technique for initialization is the use of random binary strings.

Fitness assignment

The fitness function helps in establishing the fitness of all individuals in the population. It assigns a fitness score to every individual, which further determines the probability of being chosen for reproduction. The higher the fitness score, the higher the chances of being chosen for reproduction.

Selection

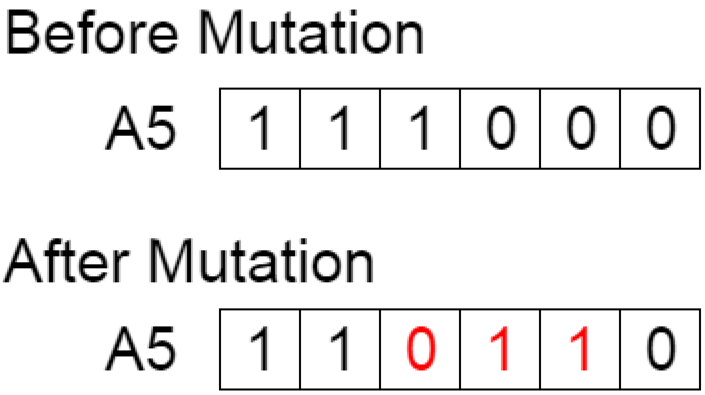
In this phase, individuals are selected for the reproduction of offspring. The selected individuals are then arranged in pairs of two to enhance reproduction. These individuals pass on their genes to the next generation.

The main objective of this phase is to establish the region with high chances of generating the best solution to the problem (better than the previous generation). The genetic algorithm uses the fitness proportionate selection technique to ensure that useful solutions are used for recombination.

Reproduction

This phase involves the creation of a child population. The algorithm employs variation operators that are applied to the parent population. The two main operators in this phase include crossover and mutation.

1. **Crossover:** This operator swaps the genetic information of two parents to reproduce an offspring. It is performed on parent pairs that are selected randomly to generate a child population of equal size as the parent population.
2. **Mutation:** This operator adds new genetic information to the new child population. This is achieved by flipping some bits in the chromosome. Mutation solves the problem of local minimum and enhances diversification. The following image shows how mutation is done.



Before Mutation

A5	1	1	1	0	0	0

After Mutation

A5	1	1	0	1	1	0

By: Adham Magdy

Replacement

Generational replacement takes place in this phase, which is a replacement of the old population with the new child population. The new population consists of higher fitness scores than the old population, which is an indication that an improved solution has been generated.

Termination

After replacement has been done, a stopping criterion is used to provide the basis for termination. The algorithm will terminate after the threshold fitness solution has been attained. It will identify this solution as the best solution in the population.

Application areas

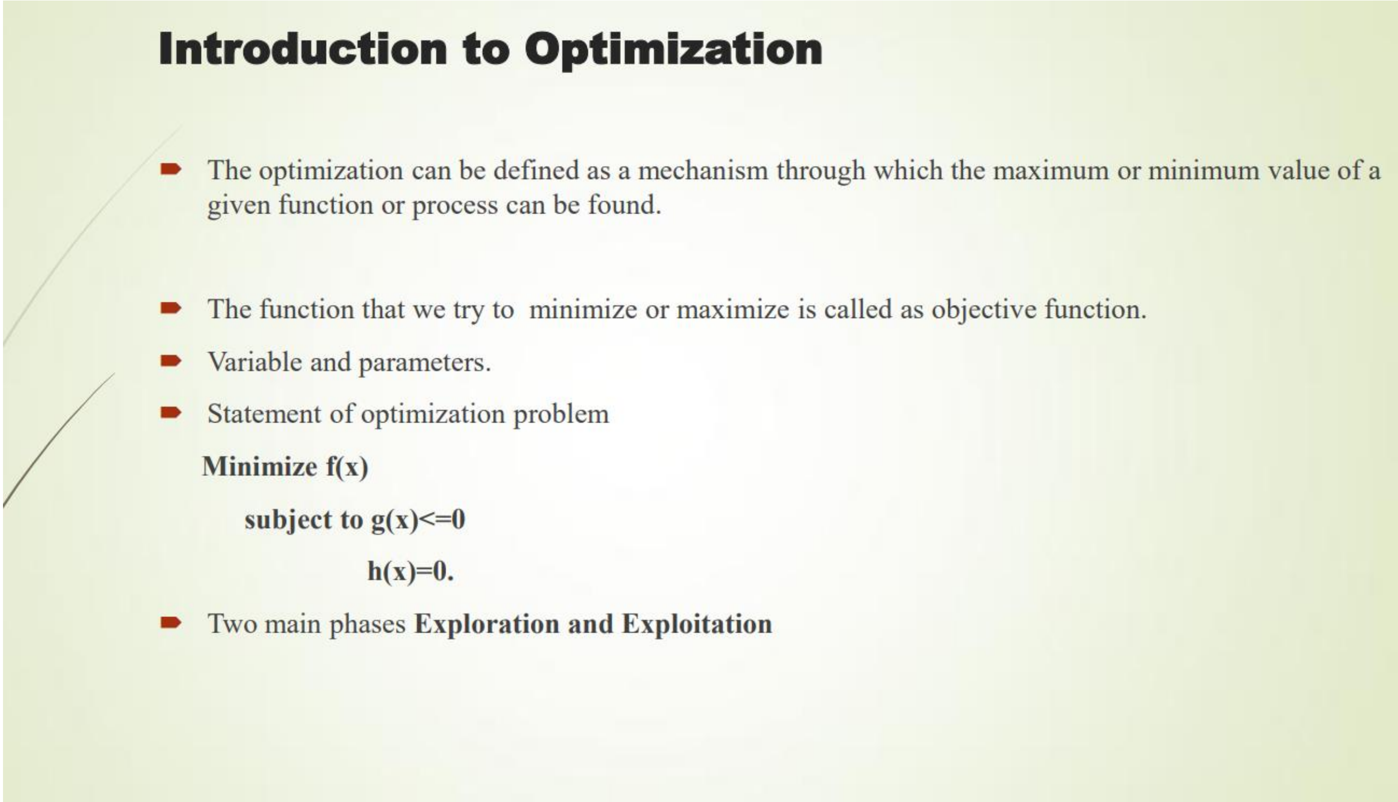
Genetic algorithms are applied in the following fields:

- **Transport:** Genetic algorithms are used in the traveling salesman problem to develop transport plans that reduce the cost of travel and the time taken. They are also used to develop an efficient way of delivering products.
- **DNA Analysis:** They are used in DNA analysis to establish the DNA structure using spectrometric information.
- **Multimodal Optimization:** They are used to provide multiple optimum solutions in multimodal optimization problems.
- **Aircraft Design:** They are used to develop parametric aircraft designs. The parameters of the aircraft are modified and upgraded to provide better designs.
- **Economics:** They are used in economics to describe various models such as the game theory, cobweb model, asset pricing, and schedule optimization.

Limitations of genetic algorithms

- They are not effective in solving simple problems.
- Lack of proper implementation may make the algorithm converge to a solution that is not optimal.
- The quality of the final solution is not guaranteed.
- Repetitive calculation of fitness values may make some problems to experience computational challenges.

5) PSO algorithm



Introduction to Optimization

- The optimization can be defined as a mechanism through which the maximum or minimum value of a given function or process can be found.
- The function that we try to minimize or maximize is called as objective function.
- Variable and parameters.
- Statement of optimization problem

Minimize f(x)

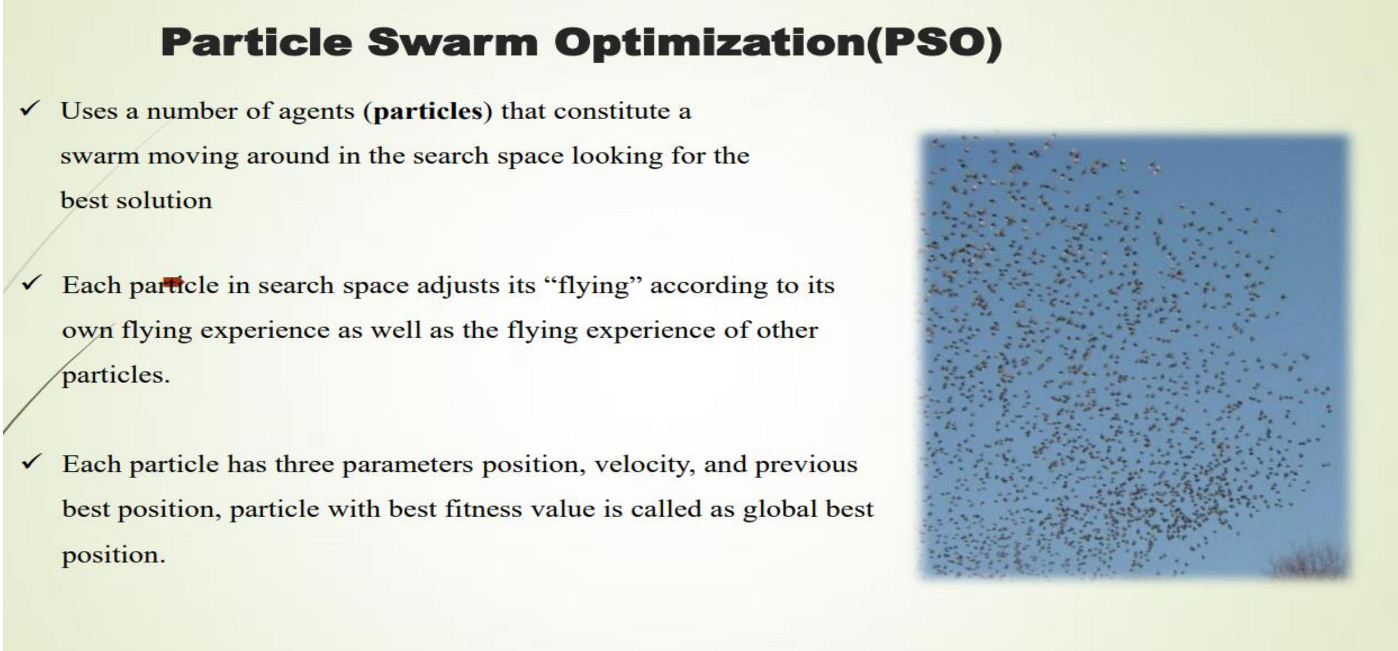
subject to g(x)<=0

h(x)=0.

- Two main phases **Exploration and Exploitation**

Particle Swarm Optimization(PSO)

- Inspired from the nature social behavior and dynamic movements with communications of insects, birds and fish.

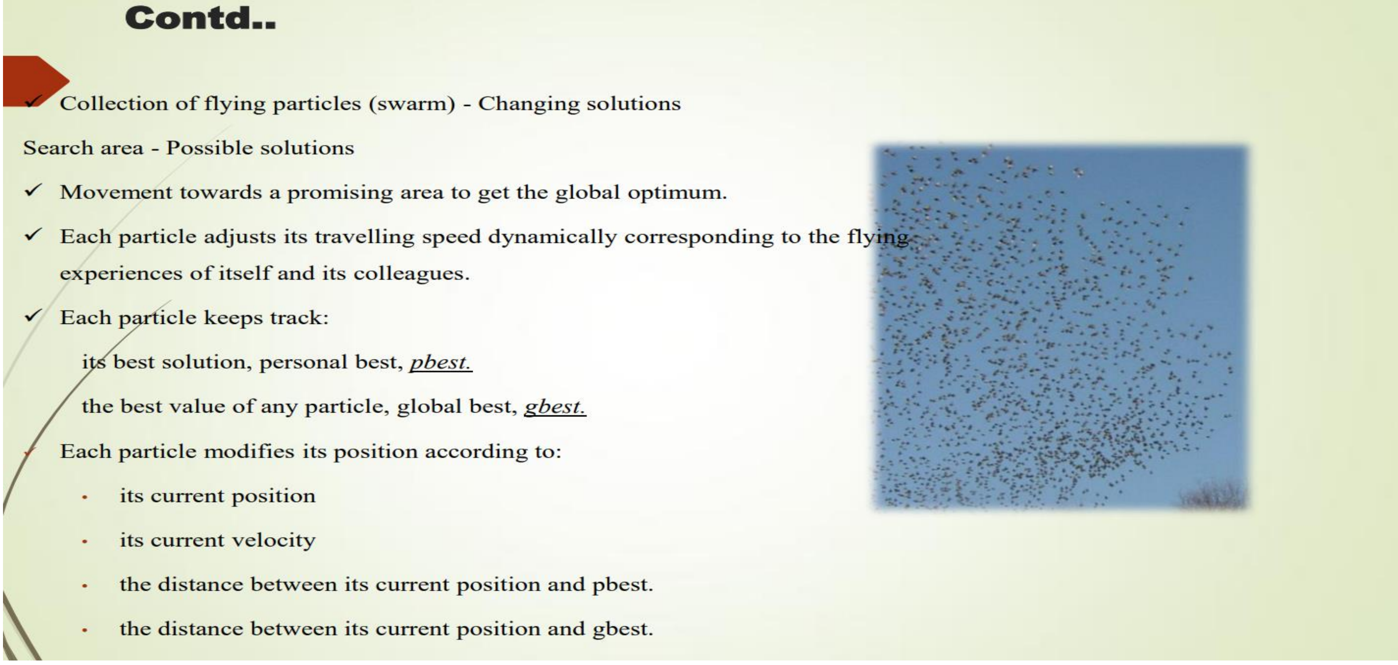


Particle Swarm Optimization(PSO)

✓ Uses a number of agents (**particles**) that constitute a swarm moving around in the search space looking for the best solution

✓ Each particle in search space adjusts its "flying" according to its own flying experience as well as the flying experience of other particles.

✓ Each particle has three parameters position, velocity, and previous best position, particle with best fitness value is called as global best position.



Contd..

✓ Collection of flying particles (swarm) - Changing solutions

Search area - Possible solutions

✓ Movement towards a promising area to get the global optimum.

✓ Each particle adjusts its travelling speed dynamically corresponding to the flying experiences of itself and its colleagues.

✓ Each particle keeps track:

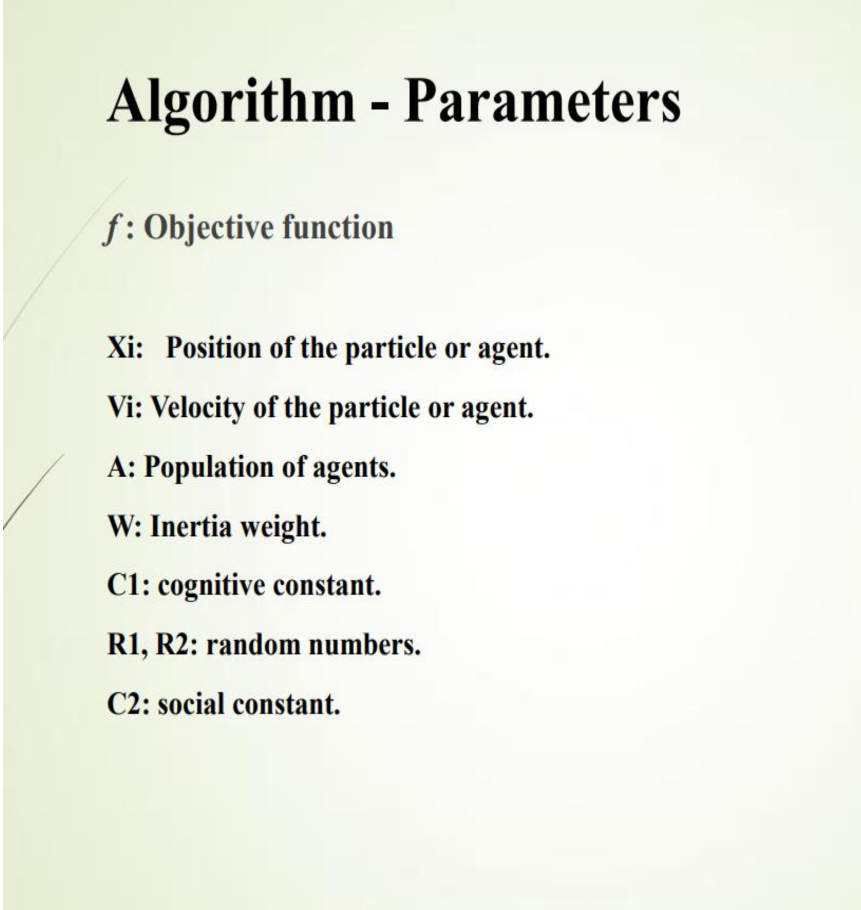
its best solution, personal best, *pbest.*

the best value of any particle, global best, *gbest.*

✓ Each particle modifies its position according to:

- its current position
- its current velocity
- the distance between its current position and pbest.
- the distance between its current position and gbest.

By: Adham Magdy



Algorithm - Parameters

f : Objective function

Xi: Position of the particle or agent.

Vi: Velocity of the particle or agent.

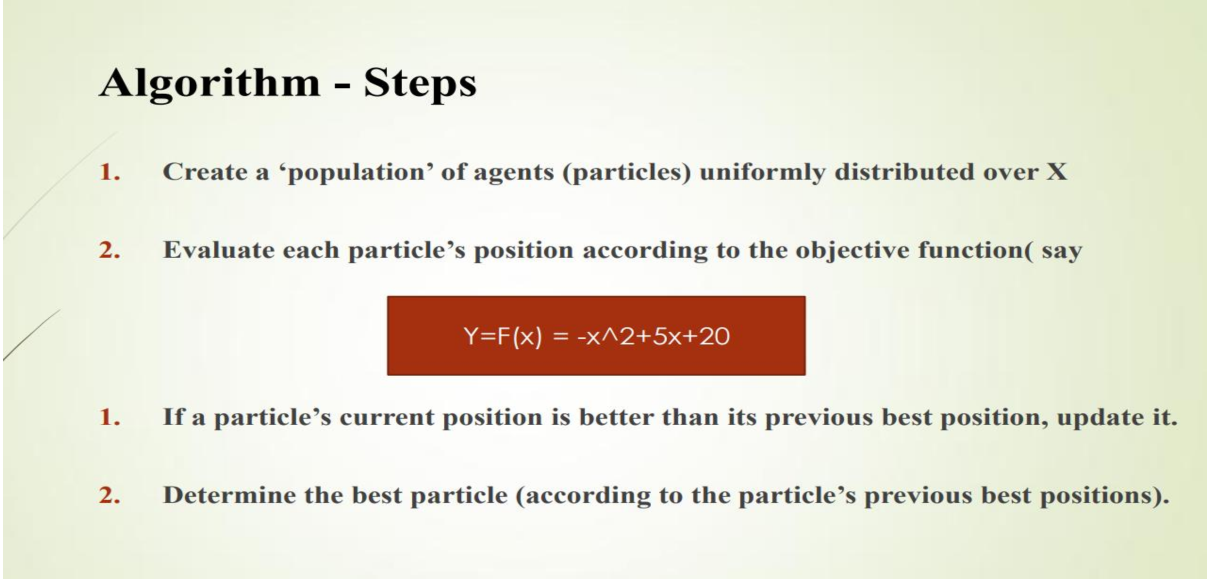
A: Population of agents.

W: Inertia weight.

C1: cognitive constant.

R1, R2: random numbers.

C2: social constant.



Algorithm - Steps

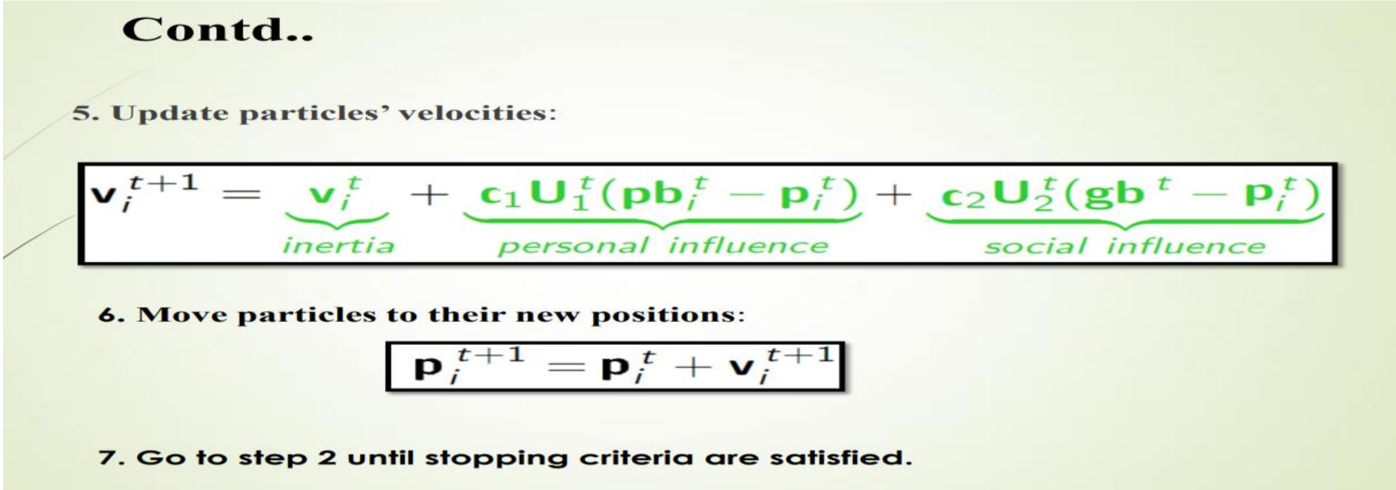
1. Create a ‘population’ of agents (particles) uniformly distributed over X

2. Evaluate each particle’s position according to the objective function(say

Y=F(x) = -x^2+5x+20

1. If a particle’s current position is better than its previous best position, update it.

2. Determine the best particle (according to the particle’s previous best positions).



Contd..

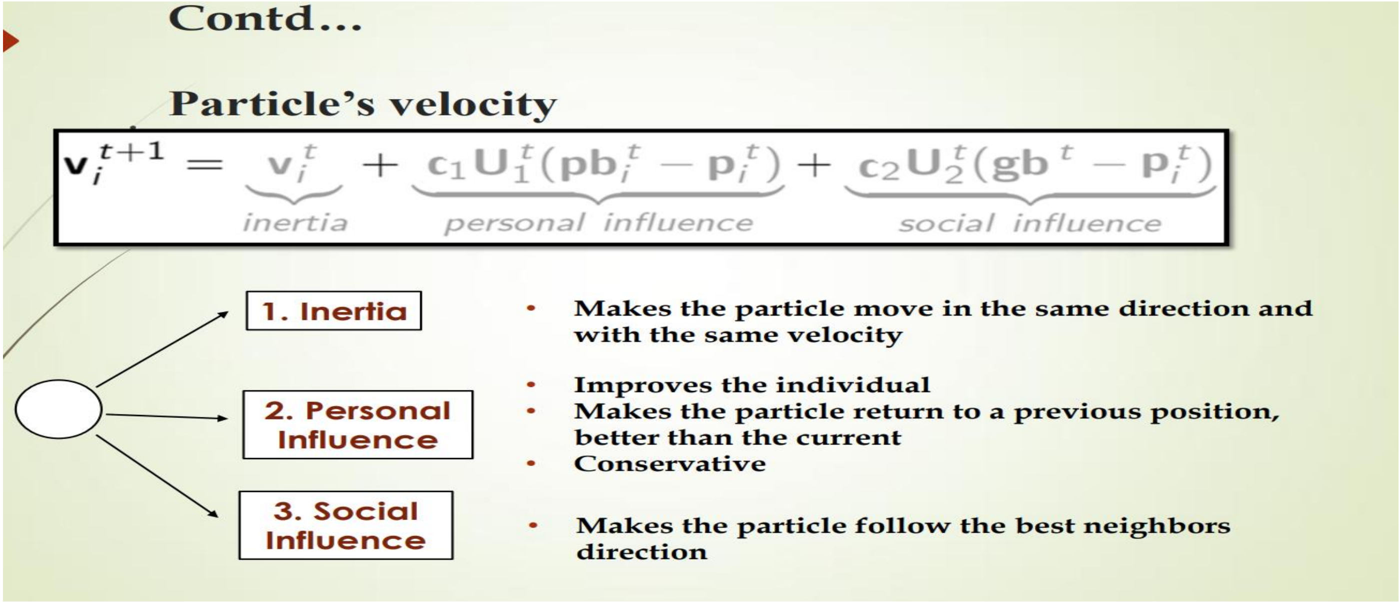
5. Update particles’ velocities:

$$\boxed{\mathbf{v}_i^{t+1} = \underbrace{\mathbf{v}_i^t}_{inertia} + \underbrace{c_1 \mathbf{U}_1^t(\mathbf{pb}_i^t - \mathbf{p}_i^t)}_{personal\ influence} + \underbrace{c_2 \mathbf{U}_2^t(\mathbf{gb}^t - \mathbf{p}_i^t)}_{social\ influence}}$$

6. Move particles to their new positions:

$$\boxed{\mathbf{p}_i^{t+1} = \mathbf{p}_i^t + \mathbf{v}_i^{t+1}}$$

7. Go to step 2 until stopping criteria are satisfied.



Contd…

Particle's velocity

$$\boxed{\mathbf{v}_i^{t+1} = \underbrace{\mathbf{v}_i^{t}}_{inertia} + \underbrace{c_1\mathbf{U}_1^{t}(\mathbf{pb}_i^{t} - \mathbf{p}_i^{t})}_{personal\ \ influence} + \underbrace{c_2\mathbf{U}_2^{t}(\mathbf{gb}^{t} - \mathbf{p}_i^{t})}_{social\ \ influence}}$$

1. Inertia

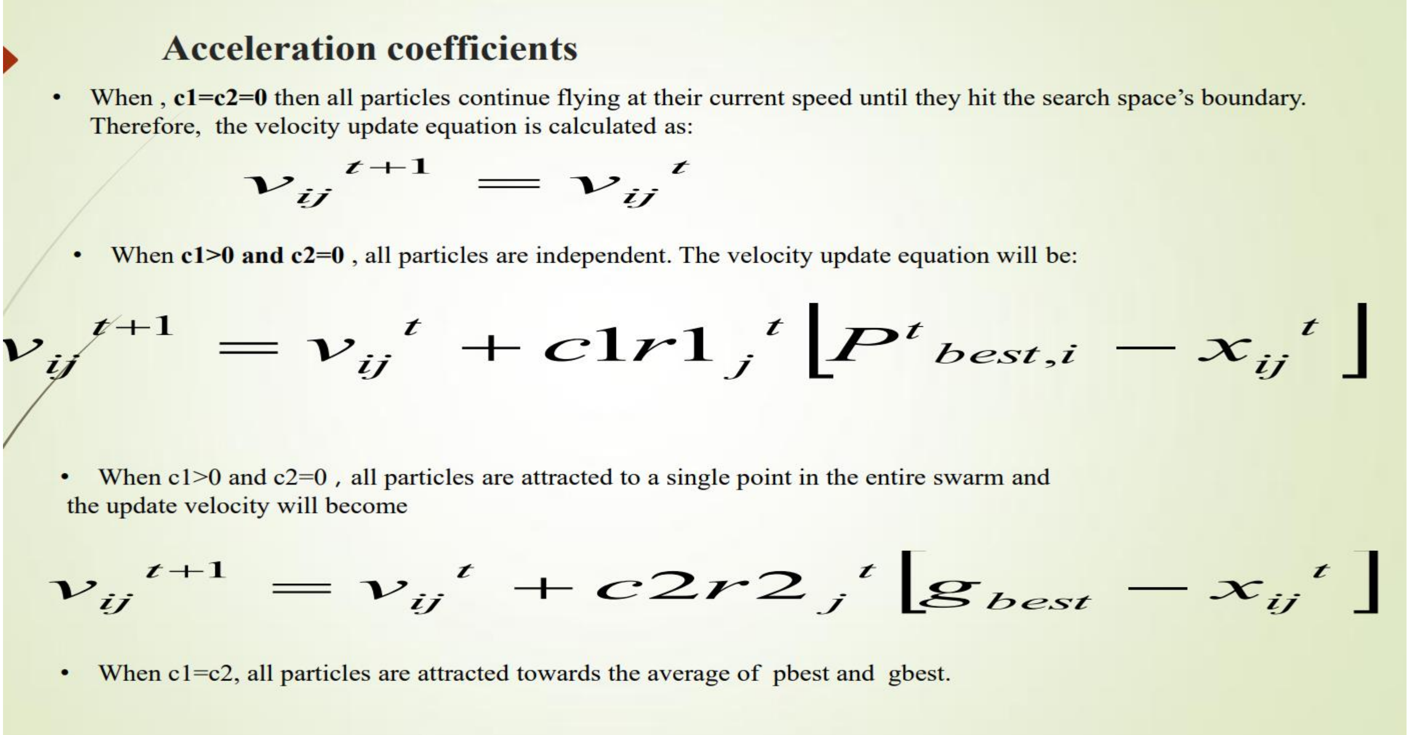
- **Makes the particle move in the same direction and with the same velocity**

2. Personal Influence

- **Improves the individual**
- **Makes the particle return to a previous position, better than the current**
- **Conservative**

3. Social Influence

- **Makes the particle follow the best neighbors direction**



Acceleration coefficients

- When , **c1=c2=0** then all particles continue flying at their current speed until they hit the search space's boundary. Therefore, the velocity update equation is calculated as:

$$v_{ij}^{\ t+1} = v_{ij}^{\ t}$$

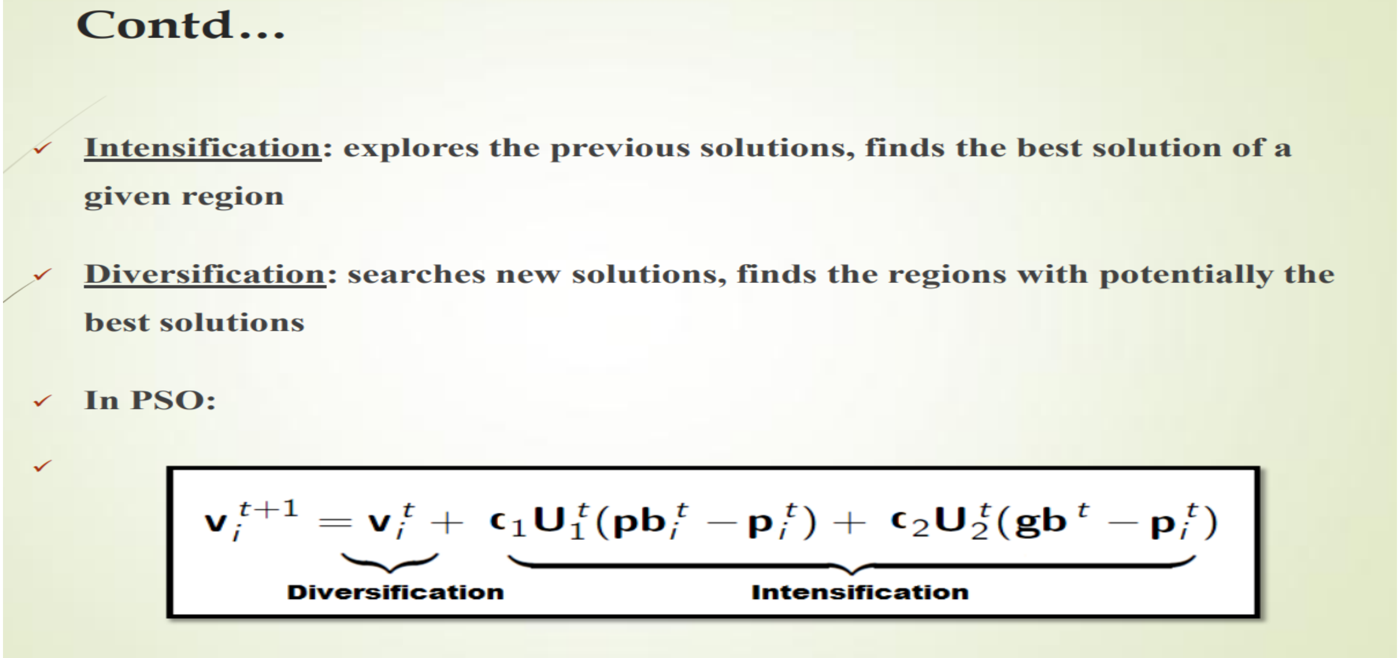
- When **c1>0 and c2=0** , all particles are independent. The velocity update equation will be:

$$v_{ij}^{\ t+1} = v_{ij}^{\ t} + c1r1_{j}^{\ t}\left[P^{t}{}_{best,i} - x_{ij}^{\ t}\right]$$

- When c1>0 and c2=0 , all particles are attracted to a single point in the entire swarm and the update velocity will become

$$v_{ij}^{\ t+1} = v_{ij}^{\ t} + c2r2_{j}^{\ t}\left[g_{best} - x_{ij}^{\ t}\right]$$

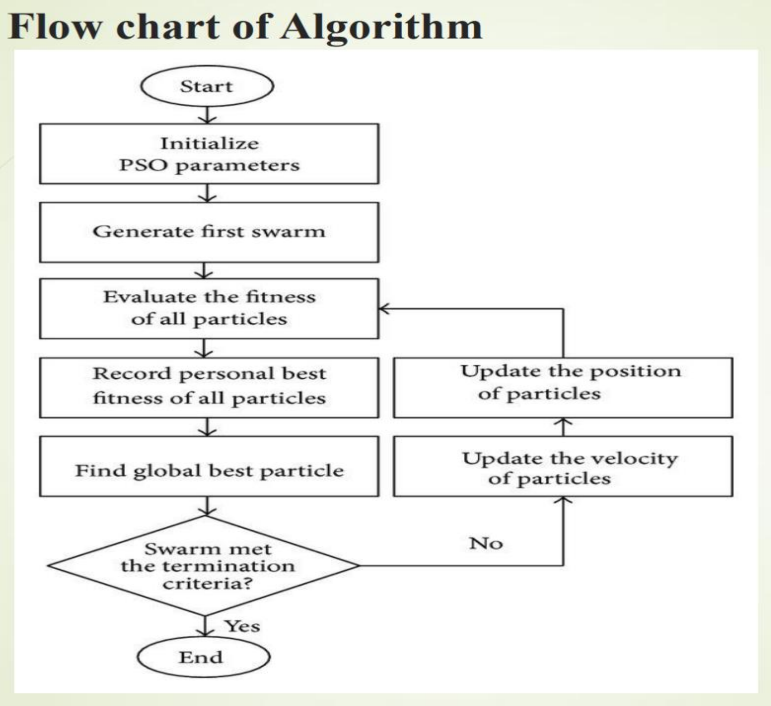
- When c1=c2, all particles are attracted towards the average of pbest and gbest.



Contd…

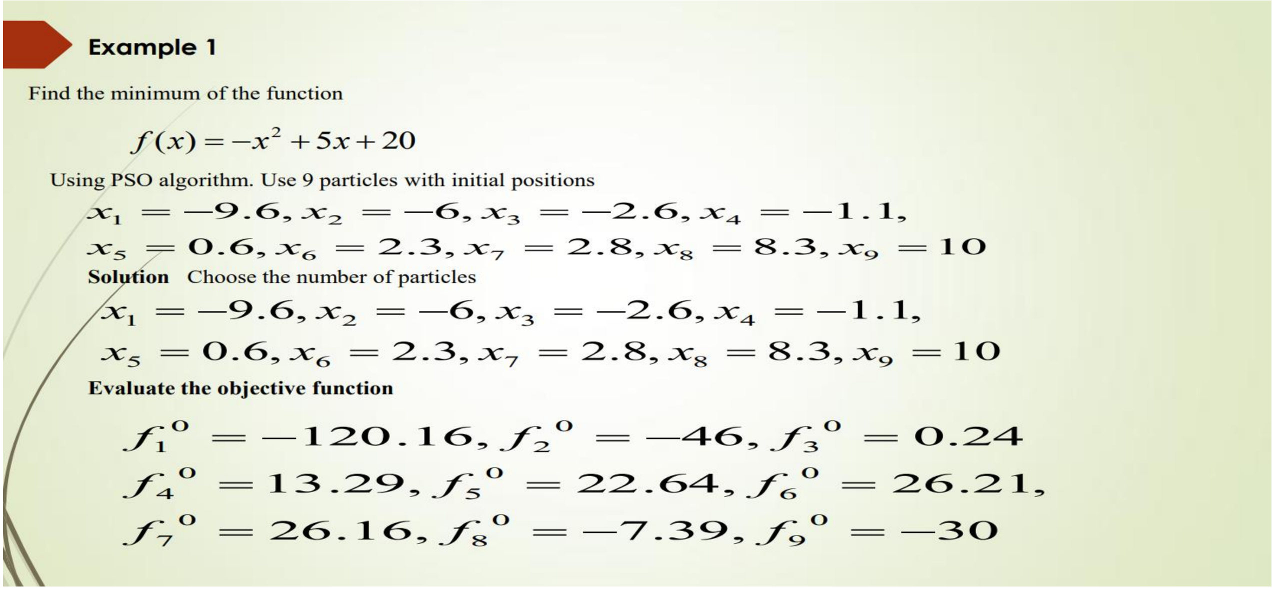
- ✓ **<u>Intensification</u>: explores the previous solutions, finds the best solution of a given region**
- ✓ **<u>Diversification</u>: searches new solutions, finds the regions with potentially the best solutions**
- ✓ **In PSO:**
- ✓

$$\boxed{\mathbf{v}_i^{t+1} = \underbrace{\mathbf{v}_i^{t}}_{\textbf{Diversification}} + \underbrace{c_1\mathbf{U}_1^{t}(\mathbf{pb}_i^{t} - \mathbf{p}_i^{t}) + c_2\mathbf{U}_2^{t}(\mathbf{gb}^{t} - \mathbf{p}_i^{t})}_{\textbf{Intensification}}}$$



Flow chart of Algorithm

Start
Initialize PSO parameters
Generate first swarm
Evaluate the fitness of all particles
Record personal best fitness of all particles
Find global best particle
Swarm met the termination criteria?
No
Update the velocity of particles
Update the position of particles
Yes
End



Example 1

Find the minimum of the function

$$f(x) = -x^2 + 5x + 20$$

Using PSO algorithm. Use 9 particles with initial positions

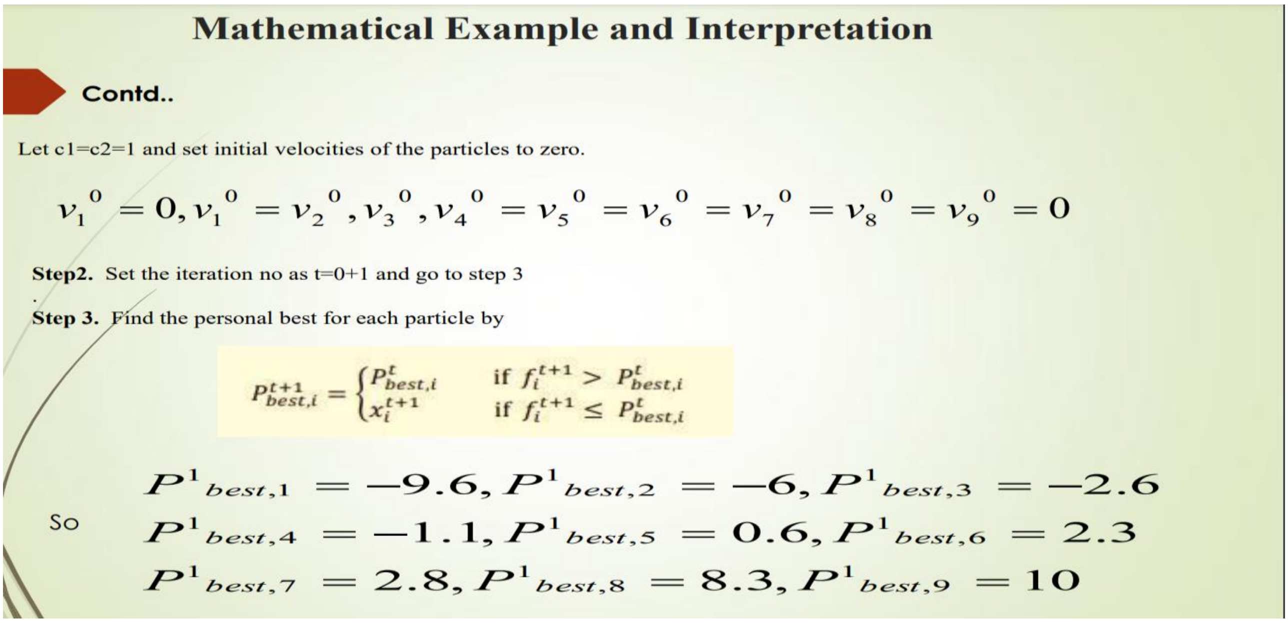
$$x_1 = -9.6, x_2 = -6, x_3 = -2.6, x_4 = -1.1,$$
$$x_5 = 0.6, x_6 = 2.3, x_7 = 2.8, x_8 = 8.3, x_9 = 10$$

Solution Choose the number of particles

$$x_1 = -9.6, x_2 = -6, x_3 = -2.6, x_4 = -1.1,$$
$$x_5 = 0.6, x_6 = 2.3, x_7 = 2.8, x_8 = 8.3, x_9 = 10$$

Evaluate the objective function

$$f_1{}^{o} = -120.16, f_2{}^{o} = -46, f_3{}^{o} = 0.24$$
$$f_4{}^{o} = 13.29, f_5{}^{o} = 22.64, f_6{}^{o} = 26.21,$$
$$f_7{}^{o} = 26.16, f_8{}^{o} = -7.39, f_9{}^{o} = -30$$



Mathematical Example and Interpretation

Contd..

Let c1=c2=1 and set initial velocities of the particles to zero.

$$v_1{}^{0} = 0, v_1{}^{0} = v_2{}^{0}, v_3{}^{0}, v_4{}^{0} = v_5{}^{0} = v_6{}^{0} = v_7{}^{0} = v_8{}^{0} = v_9{}^{0} = 0$$

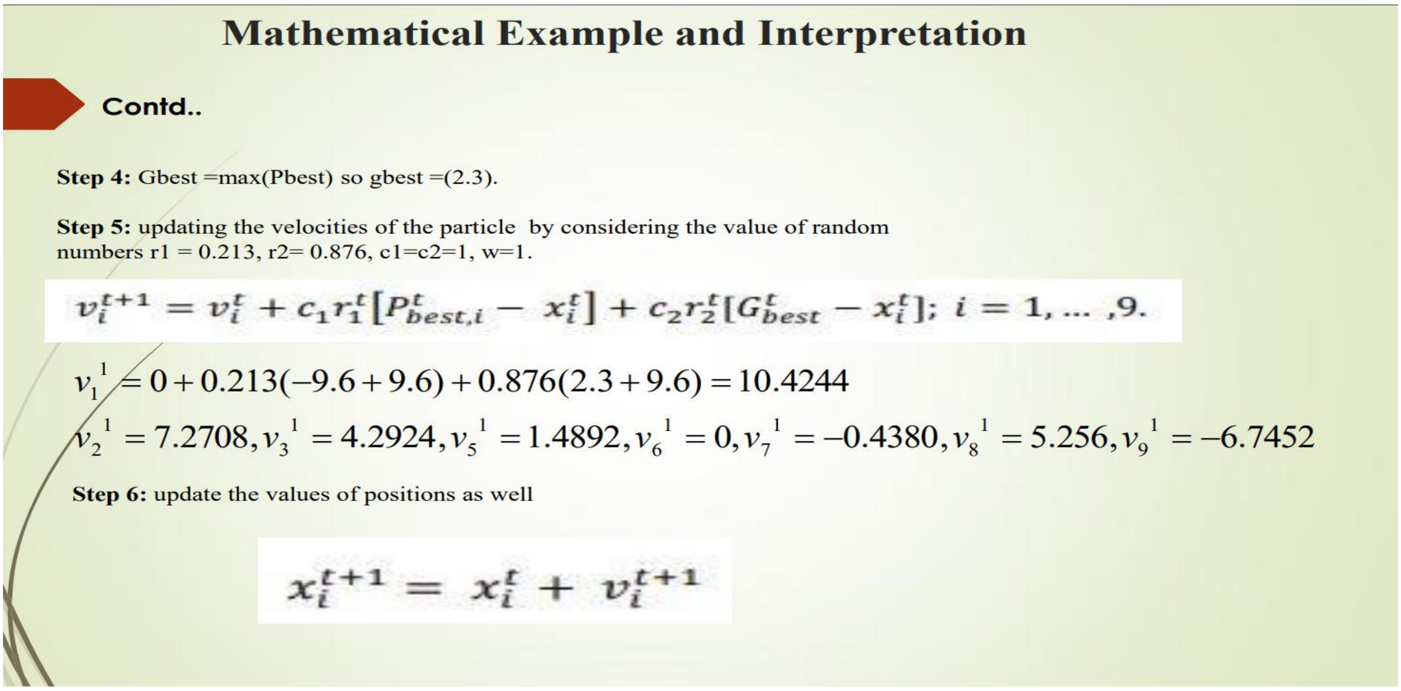
Step2. Set the iteration no as t=0+1 and go to step 3
.

Step 3. Find the personal best for each particle by

$$P^{t+1}_{best,i} = \begin{cases} P^{t}_{best,i} & \text{if } f_i^{t+1} > P^{t}_{best,i} \\ x_i^{t+1} & \text{if } f_i^{t+1} \le P^{t}_{best,i} \end{cases}$$

So

$$P^1{}_{best,1} = -9.6, P^1{}_{best,2} = -6, P^1{}_{best,3} = -2.6$$
$$P^1{}_{best,4} = -1.1, P^1{}_{best,5} = 0.6, P^1{}_{best,6} = 2.3$$
$$P^1{}_{best,7} = 2.8, P^1{}_{best,8} = 8.3, P^1{}_{best,9} = 10$$



Mathematical Example and Interpretation

Contd..

Step 4: Gbest =max(Pbest) so gbest =(2.3).

Step 5: updating the velocities of the particle by considering the value of random numbers r1 = 0.213, r2= 0.876, c1=c2=1, w=1.

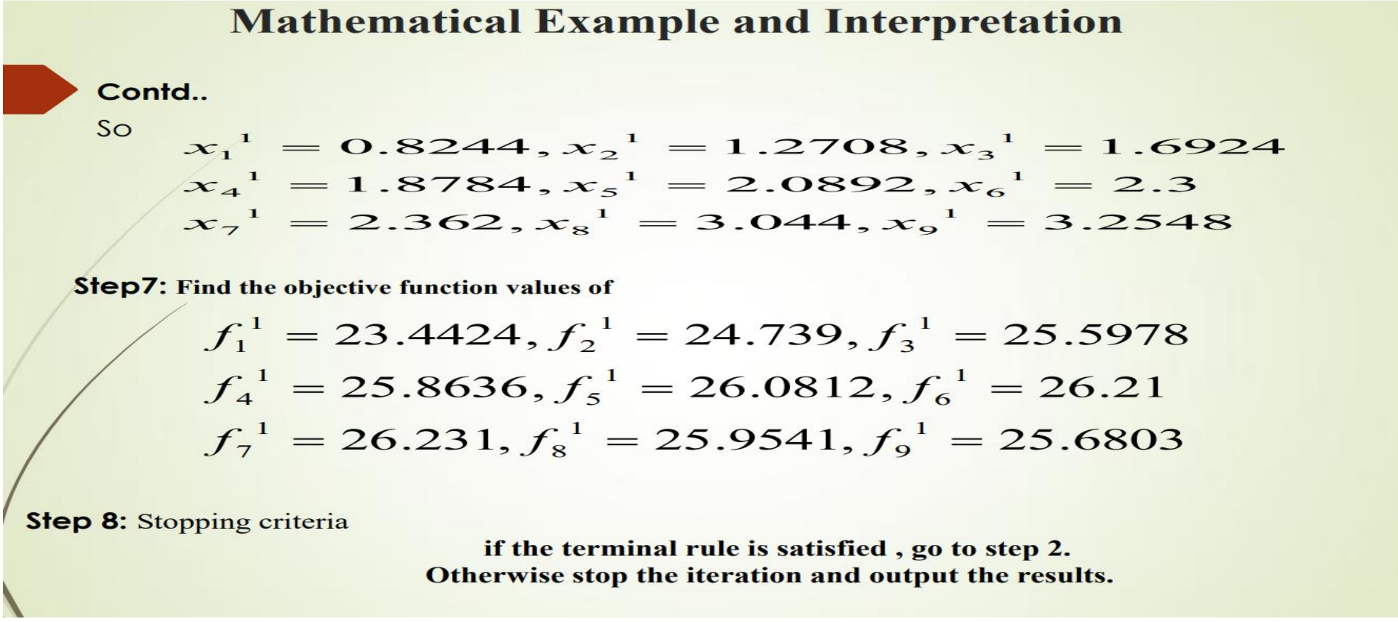
$$v_i^{t+1} = v_i^t + c_1 r_1^t [P_{best,i}^t - x_i^t] + c_2 r_2^t [G_{best}^t - x_i^t];\ i = 1, \dots ,9.$$

$$v_1^{\,1} = 0+0.213(-9.6+9.6)+0.876(2.3+9.6)=10.4244$$

$$v_2^{\,1} = 7.2708, v_3^{\,1} = 4.2924, v_5^{\,1} = 1.4892, v_6^{\,1} = 0, v_7^{\,1} = -0.4380, v_8^{\,1} = 5.256, v_9^{\,1} = -6.7452$$

Step 6: update the values of positions as well

$$x_i^{t+1} = x_i^t + v_i^{t+1}$$



Mathematical Example and Interpretation

Contd..

So

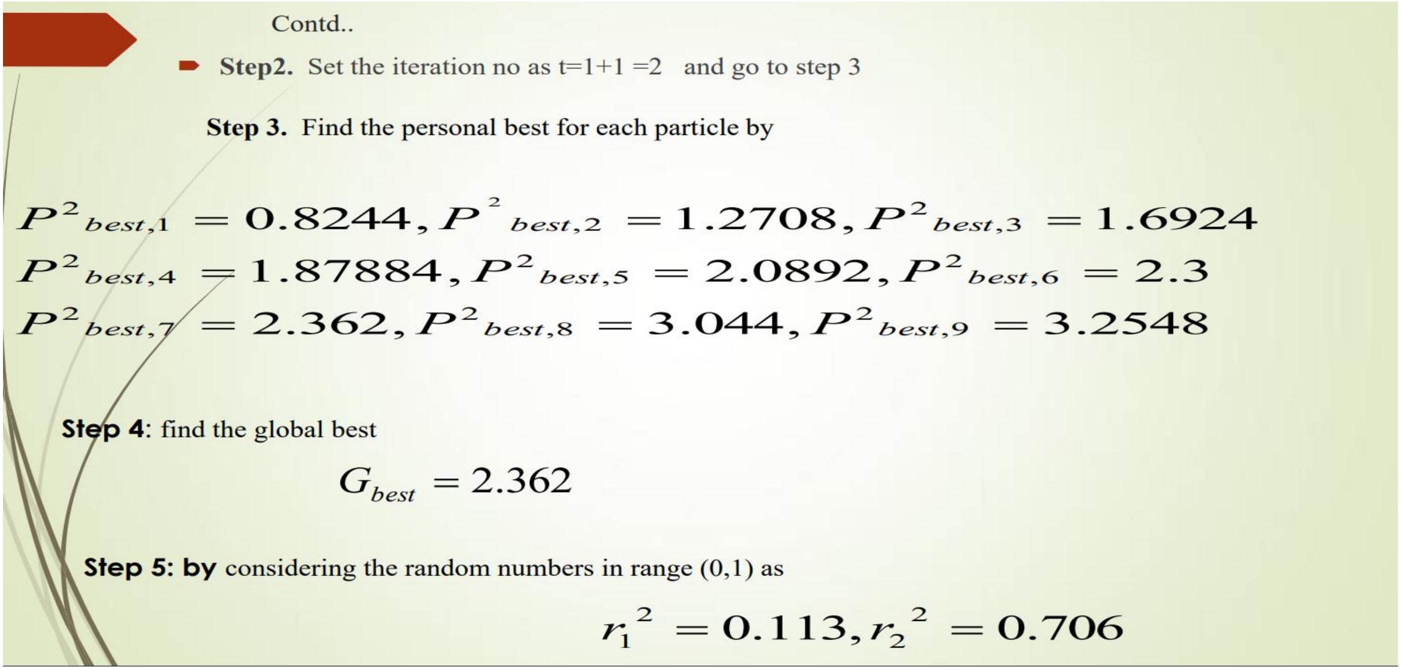
$$x_1^{\,1} = 0.8244, x_2^{\,1} = 1.2708, x_3^{\,1} = 1.6924$$
$$x_4^{\,1} = 1.8784, x_5^{\,1} = 2.0892, x_6^{\,1} = 2.3$$
$$x_7^{\,1} = 2.362, x_8^{\,1} = 3.044, x_9^{\,1} = 3.2548$$

Step7: Find the objective function values of

$$f_1^{\,1} = 23.4424, f_2^{\,1} = 24.739, f_3^{\,1} = 25.5978$$
$$f_4^{\,1} = 25.8636, f_5^{\,1} = 26.0812, f_6^{\,1} = 26.21$$
$$f_7^{\,1} = 26.231, f_8^{\,1} = 25.9541, f_9^{\,1} = 25.6803$$

Step 8: Stopping criteria

if the terminal rule is satisfied , go to step 2.
Otherwise stop the iteration and output the results.



Contd..

- **Step2.** Set the iteration no as t=1+1 =2 and go to step 3

Step 3. Find the personal best for each particle by

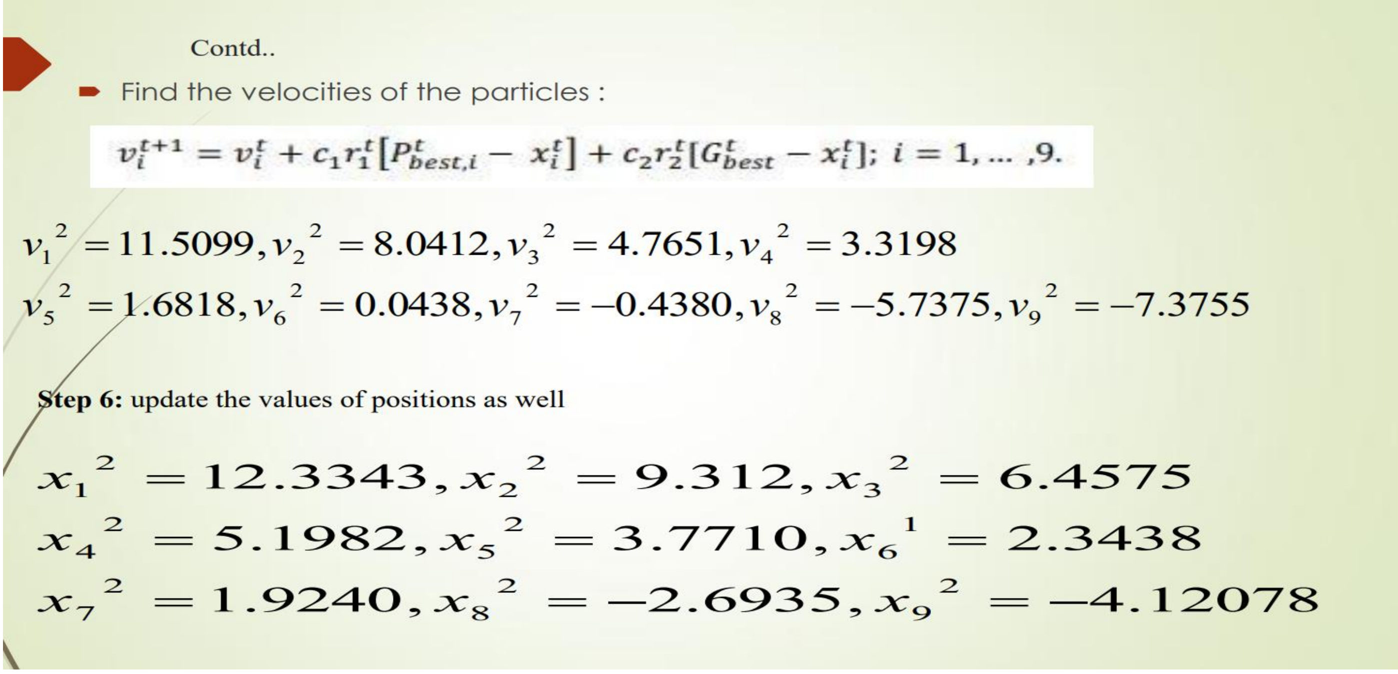
$$P^2{}_{best,1} = 0.8244, P^2{}_{best,2} = 1.2708, P^2{}_{best,3} = 1.6924$$
$$P^2{}_{best,4} = 1.87884, P^2{}_{best,5} = 2.0892, P^2{}_{best,6} = 2.3$$
$$P^2{}_{best,7} = 2.362, P^2{}_{best,8} = 3.044, P^2{}_{best,9} = 3.2548$$

Step 4: find the global best

$$G_{best} = 2.362$$

Step 5: by considering the random numbers in range (0,1) as

$$r_1^{\,2} = 0.113, r_2^{\,2} = 0.706$$



Contd..

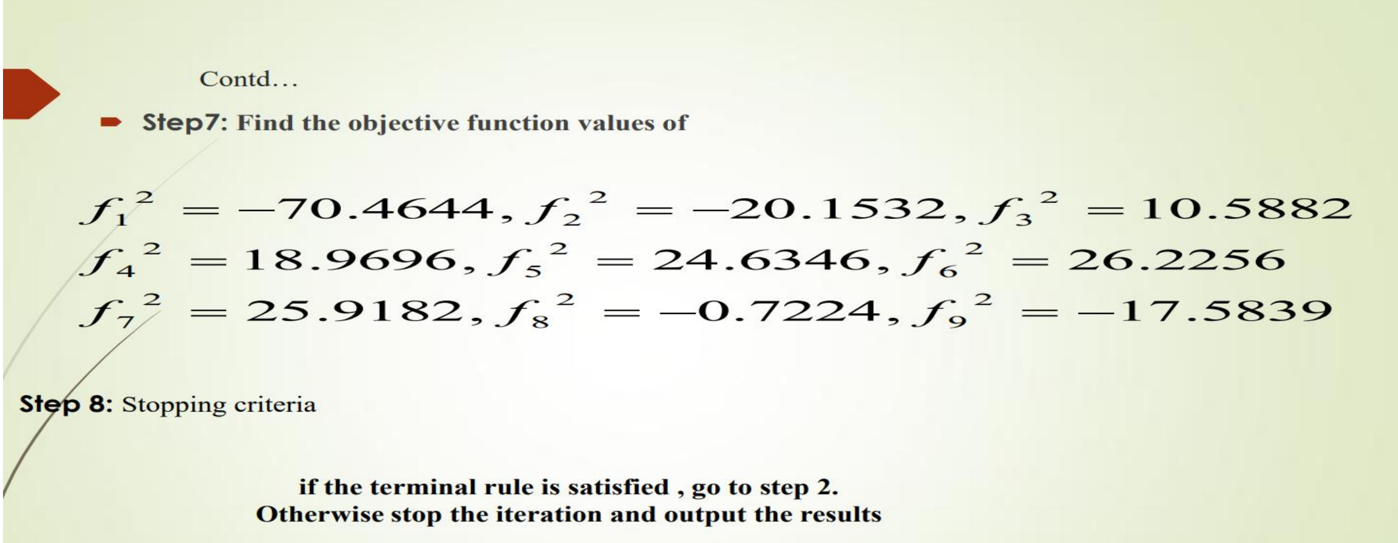
- Find the velocities of the particles :

$$v_i^{t+1} = v_i^t + c_1 r_1^t [P_{best,i}^t - x_i^t] + c_2 r_2^t [G_{best}^t - x_i^t];\ i = 1, \dots ,9.$$

$$v_1^{\ 2} = 11.5099, v_2^{\ 2} = 8.0412, v_3^{\ 2} = 4.7651, v_4^{\ 2} = 3.3198$$
$$v_5^{\ 2} = 1.6818, v_6^{\ 2} = 0.0438, v_7^{\ 2} = -0.4380, v_8^{\ 2} = -5.7375, v_9^{\ 2} = -7.3755$$

Step 6: update the values of positions as well

$$x_1^{\ 2} = 12.3343, x_2^{\ 2} = 9.312, x_3^{\ 2} = 6.4575$$
$$x_4^{\ 2} = 5.1982, x_5^{\ 2} = 3.7710, x_6^{\ 1} = 2.3438$$
$$x_7^{\ 2} = 1.9240, x_8^{\ 2} = -2.6935, x_9^{\ 2} = -4.12078$$



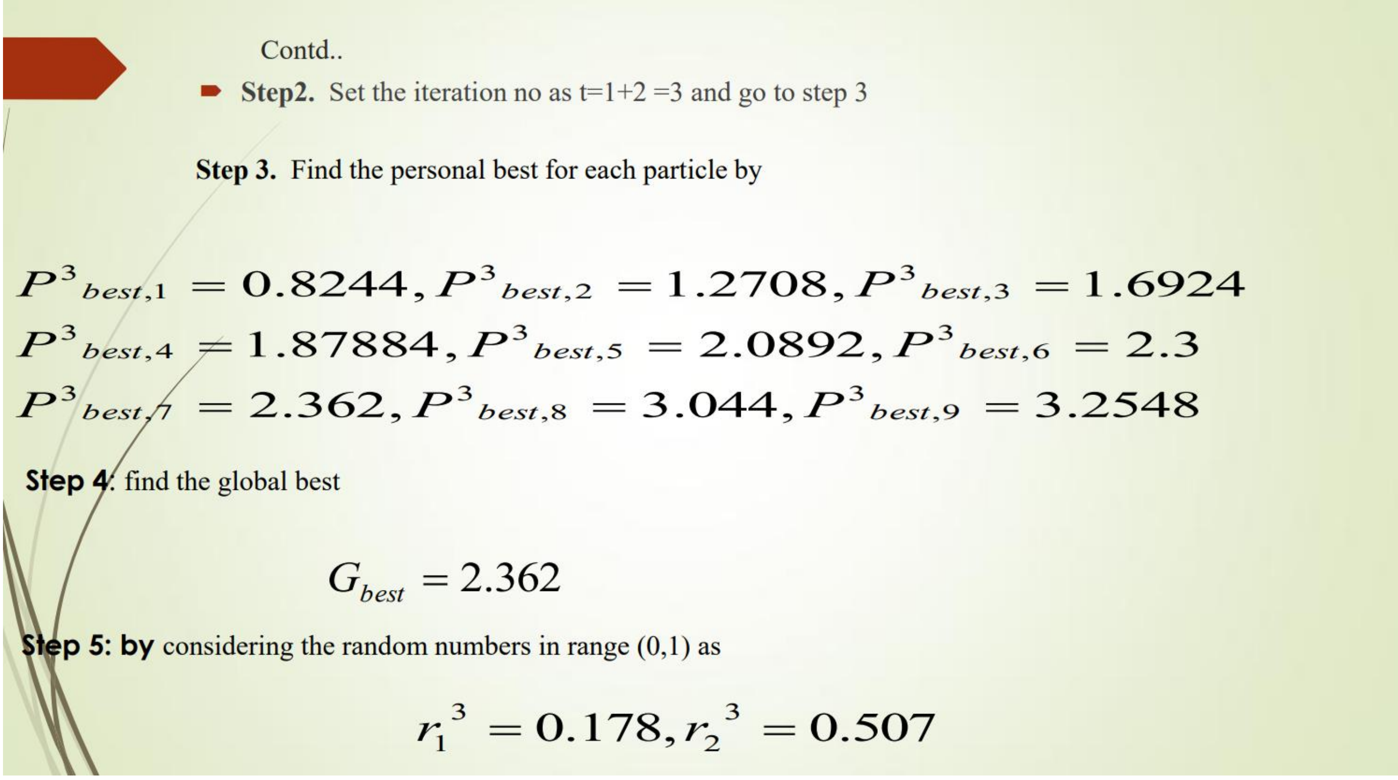
Contd…

- **Step7: Find the objective function values of**

$$f_1^{\ 2} = -70.4644, f_2^{\ 2} = -20.1532, f_3^{\ 2} = 10.5882$$
$$f_4^{\ 2} = 18.9696, f_5^{\ 2} = 24.6346, f_6^{\ 2} = 26.2256$$
$$f_7^{\ 2} = 25.9182, f_8^{\ 2} = -0.7224, f_9^{\ 2} = -17.5839$$

Step 8: Stopping criteria

if the terminal rule is satisfied , go to step 2.
Otherwise stop the iteration and output the results



Contd..

- **Step2.** Set the iteration no as t=1+2 =3 and go to step 3

Step 3. Find the personal best for each particle by

$$P^3{}_{best,1} = 0.8244, P^3{}_{best,2} = 1.2708, P^3{}_{best,3} = 1.6924$$
$$P^3{}_{best,4} = 1.87884, P^3{}_{best,5} = 2.0892, P^3{}_{best,6} = 2.3$$
$$P^3{}_{best,7} = 2.362, P^3{}_{best,8} = 3.044, P^3{}_{best,9} = 3.2548$$

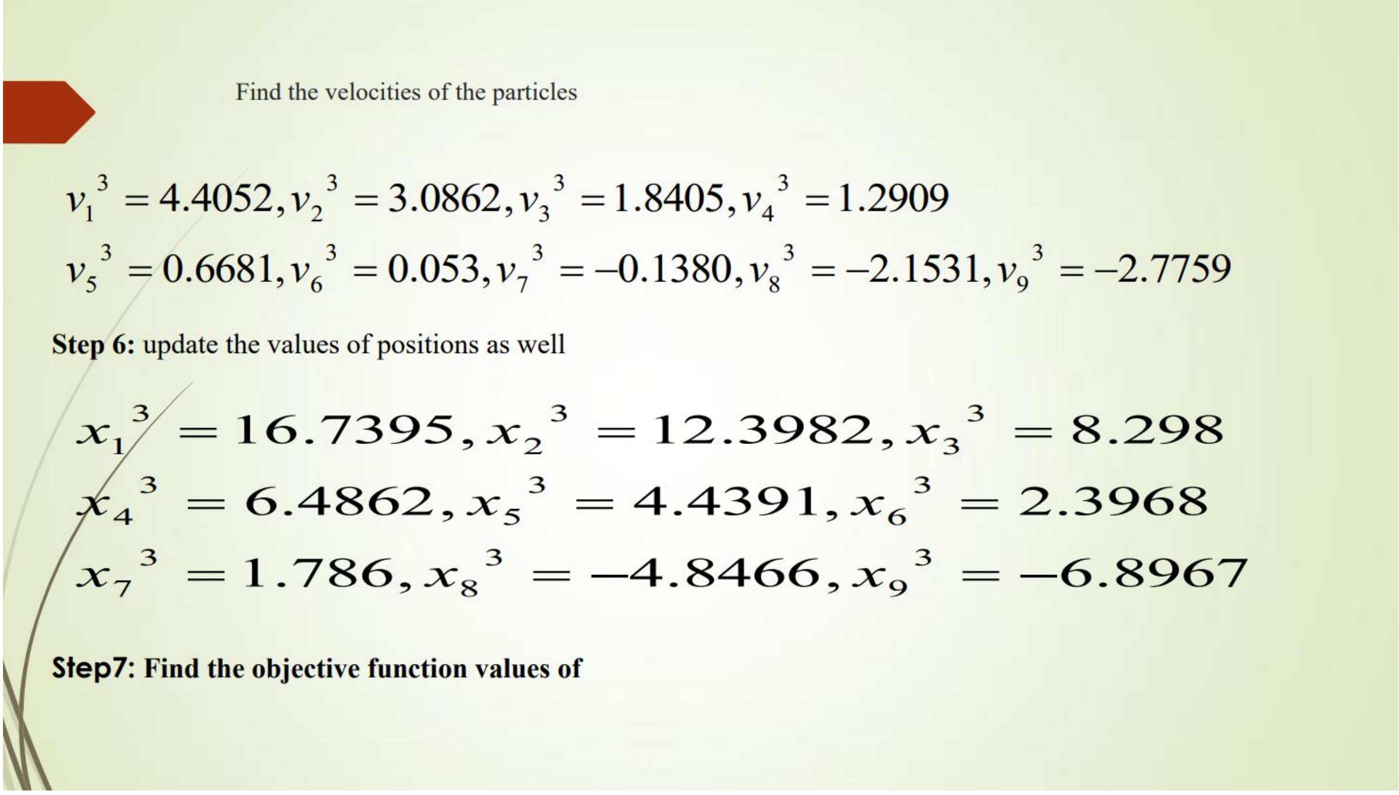
Step 4: find the global best

$$G_{best} = 2.362$$

Step 5: by considering the random numbers in range (0,1) as

$$r_1^{\ 3} = 0.178, r_2^{\ 3} = 0.507$$

By: Adham Magdy



Find the velocities of the particles

$$v_1^{\ 3} = 4.4052, v_2^{\ 3} = 3.0862, v_3^{\ 3} = 1.8405, v_4^{\ 3} = 1.2909$$

$$v_5^{\ 3} = 0.6681, v_6^{\ 3} = 0.053, v_7^{\ 3} = -0.1380, v_8^{\ 3} = -2.1531, v_9^{\ 3} = -2.7759$$

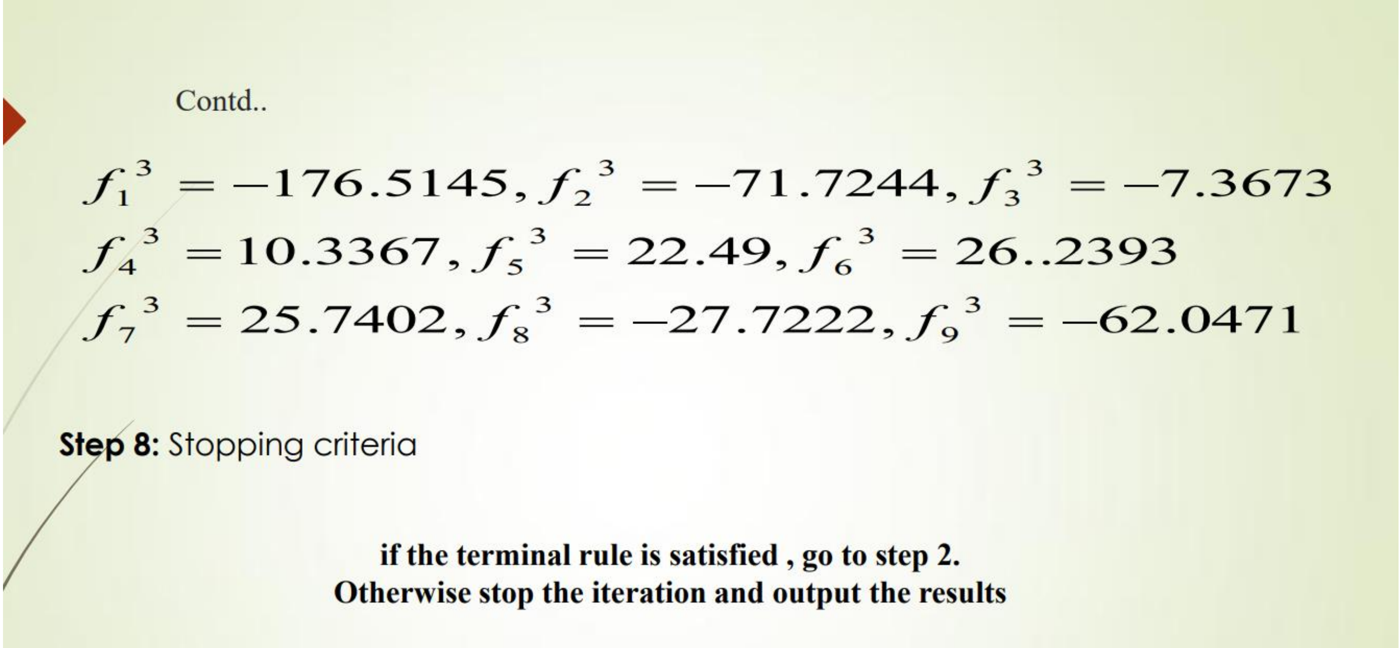
Step 6: update the values of positions as well

$$x_1^{\ 3} = 16.7395, x_2^{\ 3} = 12.3982, x_3^{\ 3} = 8.298$$

$$x_4^{\ 3} = 6.4862, x_5^{\ 3} = 4.4391, x_6^{\ 3} = 2.3968$$

$$x_7^{\ 3} = 1.786, x_8^{\ 3} = -4.8466, x_9^{\ 3} = -6.8967$$

Step7: Find the objective function values of



Contd..

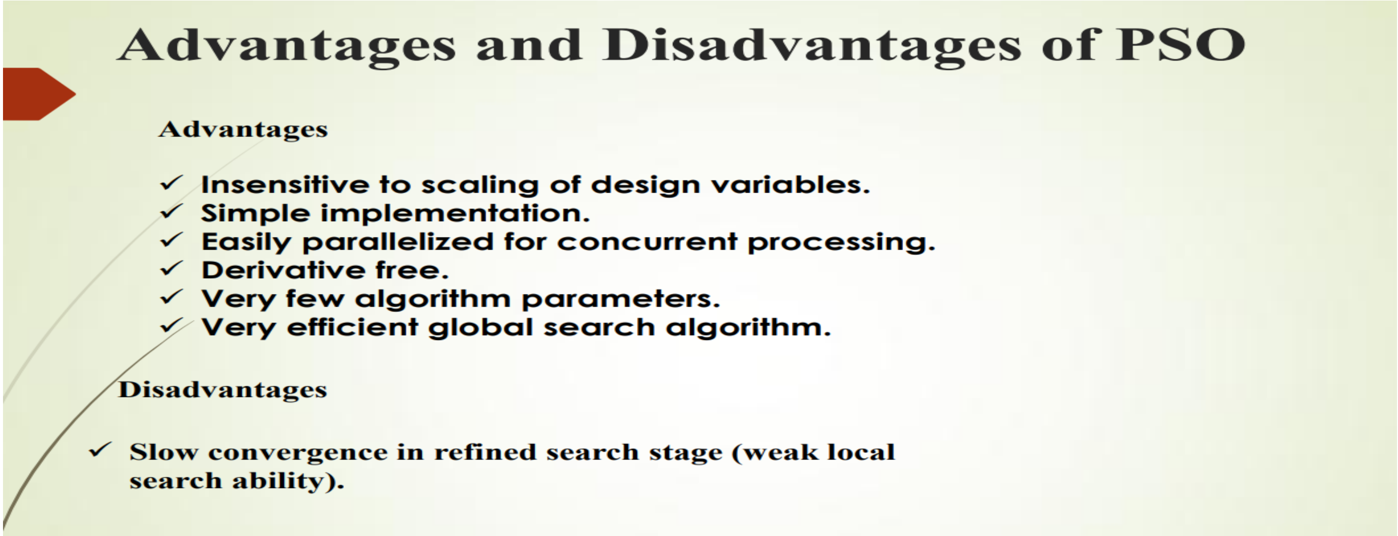
$$f_1^{\ 3} = -176.5145, f_2^{\ 3} = -71.7244, f_3^{\ 3} = -7.3673$$

$$f_4^{\ 3} = 10.3367, f_5^{\ 3} = 22.49, f_6^{\ 3} = 26..2393$$

$$f_7^{\ 3} = 25.7402, f_8^{\ 3} = -27.7222, f_9^{\ 3} = -62.0471$$

Step 8: Stopping criteria

if the terminal rule is satisfied , go to step 2.
Otherwise stop the iteration and output the results



Advantages and Disadvantages of PSO

Advantages

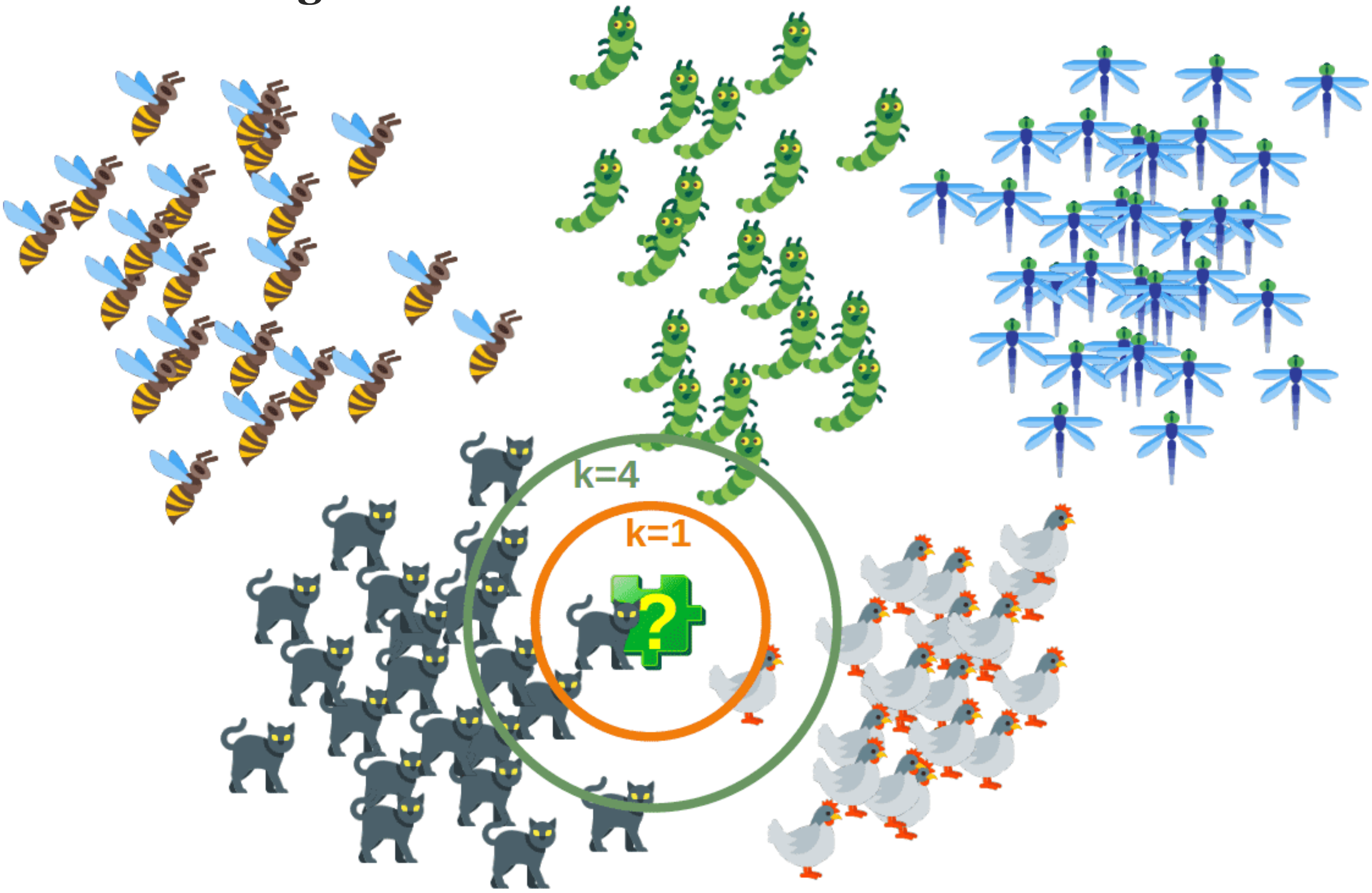
- ✓ **Insensitive to scaling of design variables.**
- ✓ **Simple implementation.**
- ✓ **Easily parallelized for concurrent processing.**
- ✓ **Derivative free.**
- ✓ **Very few algorithm parameters.**
- ✓ **Very efficient global search algorithm.**

Disadvantages

- ✓ **Slow convergence in refined search stage (weak local search ability).**

6)KNN classifier

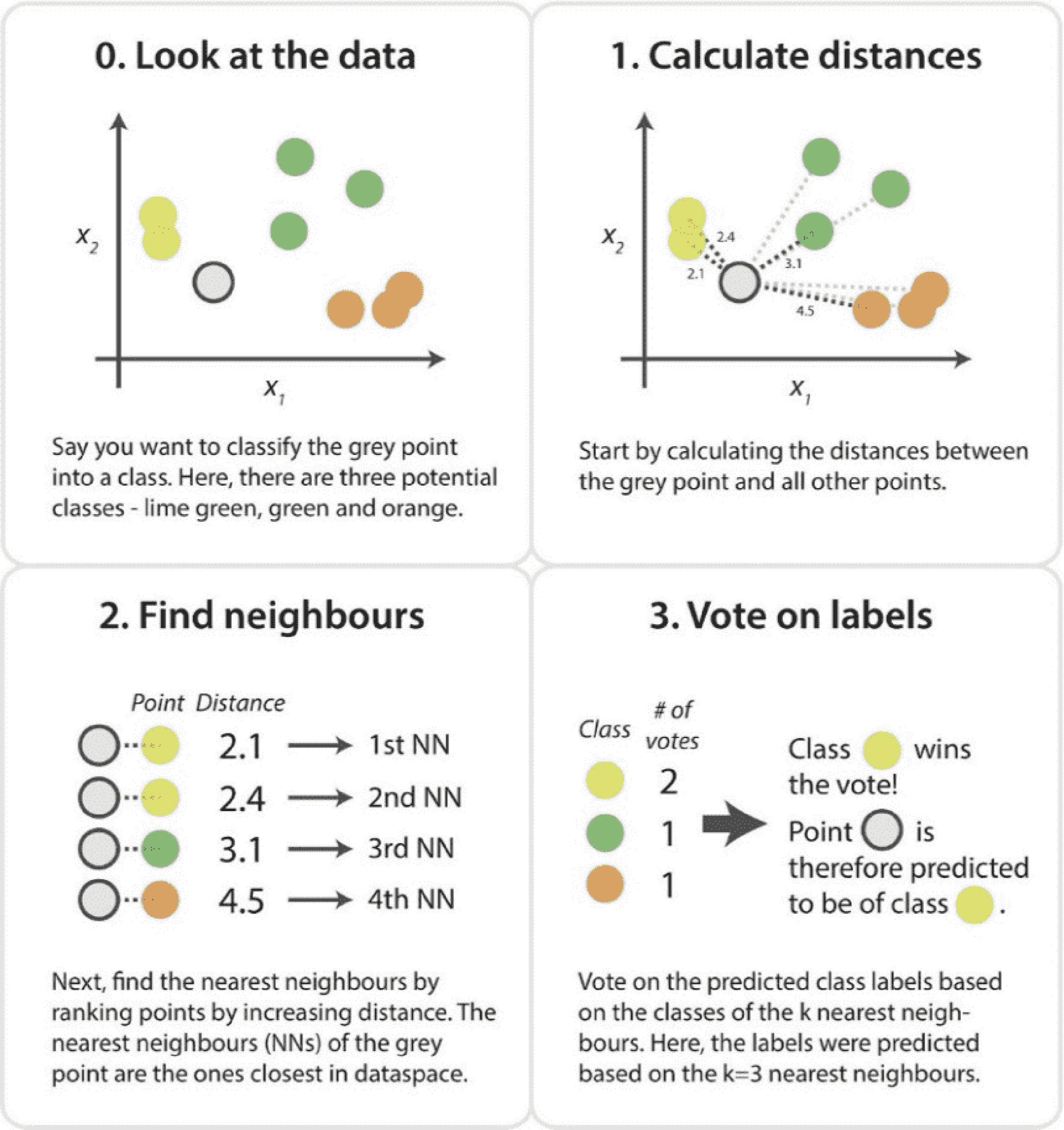
K-Nearest Neighbor



Introduction

K-nearest neighbors (KNN) is a type of supervised learning algorithm used for both regression and classification. KNN tries to predict the correct class for the test data by calculating the distance between the test data and all the training points. Then select the K number of points which is closet to the test data. The KNN algorithm calculates the probability of the test data belonging to the classes of 'K' training data and class holds the highest probability will be selected. In the case of regression, the value is the mean of the 'K' selected training points.

Let see the below example to make it a better understanding



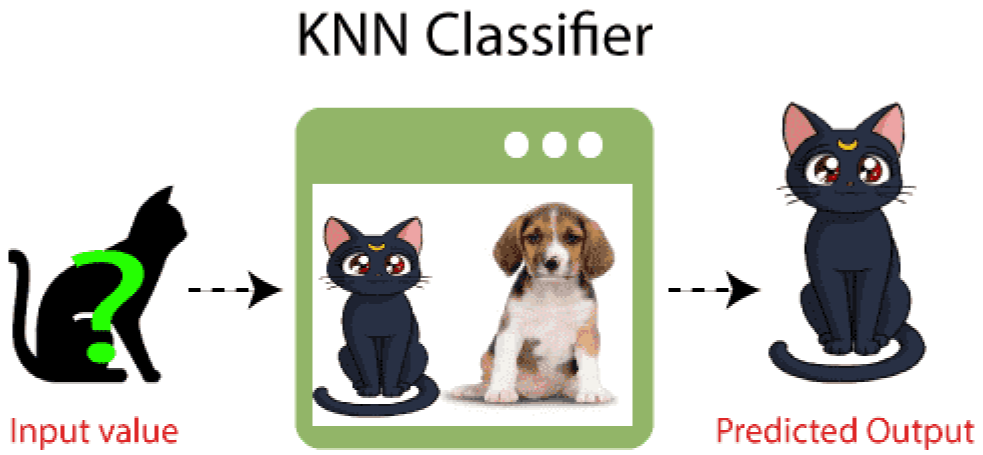
Say you want to classify the grey point into a class. Here, there are three potential classes - lime green, green and orange.

Start by calculating the distances between the grey point and all other points.

Next, find the nearest neighbours by ranking points by increasing distance. The nearest neighbours (NNs) of the grey point are the ones closest in dataspace.

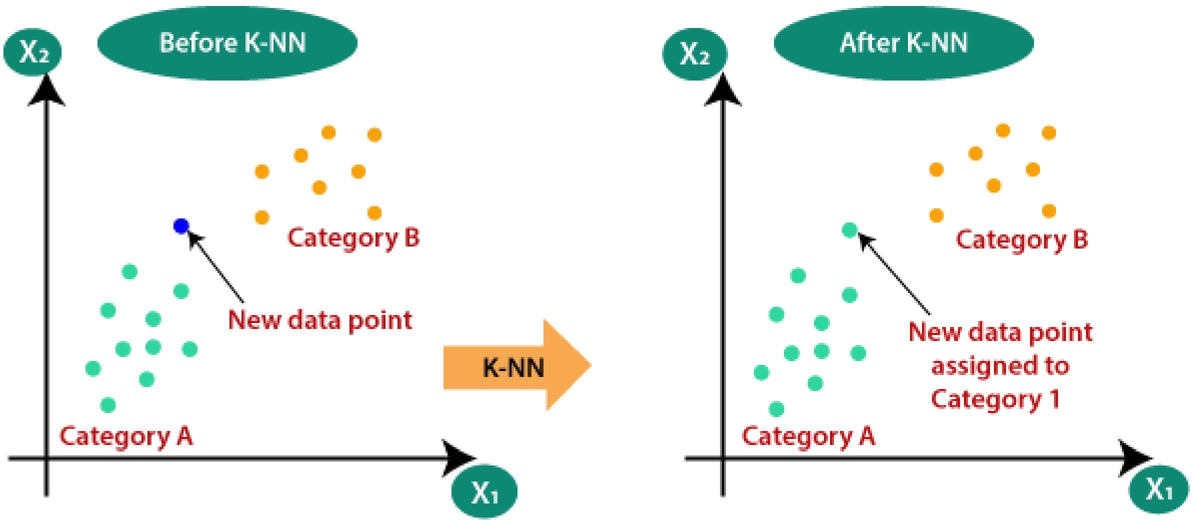
Vote on the predicted class labels based on the classes of the k nearest neighbours. Here, the labels were predicted based on the k=3 nearest neighbours.

Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.



Why do we need a K-NN Algorithm?

Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x1, so this data point will lie in which of these categories. To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset. Consider the below diagram:



How does K-NN work?

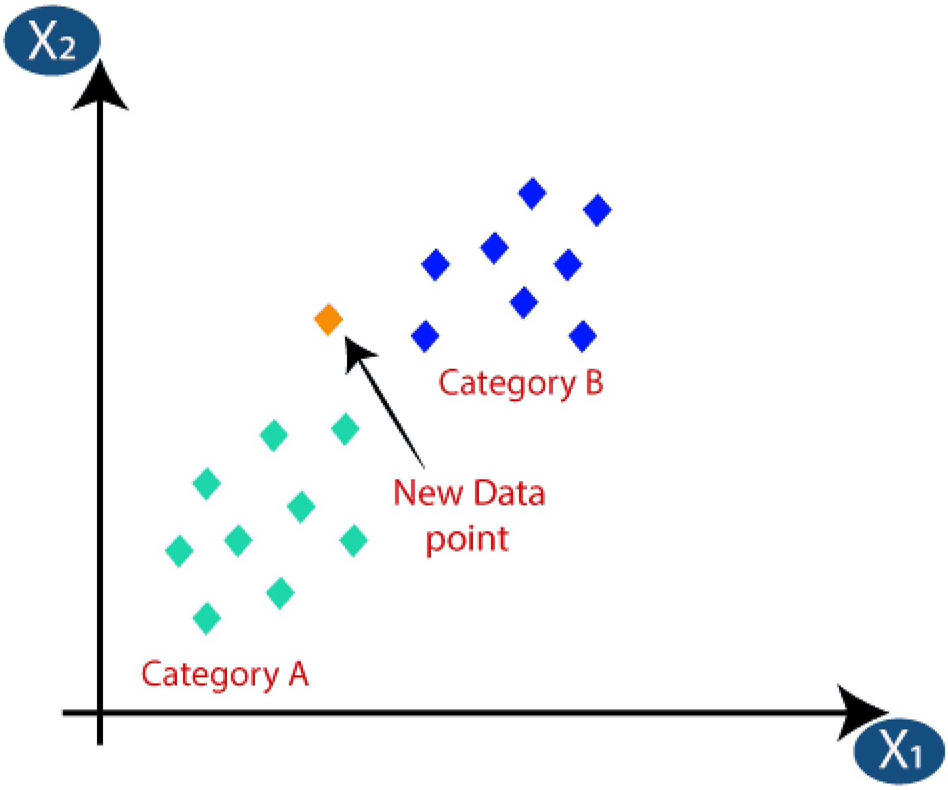
The K-NN working can be explained on the basis of the below algorithm:

- Step-1: Select the number K of the neighbors

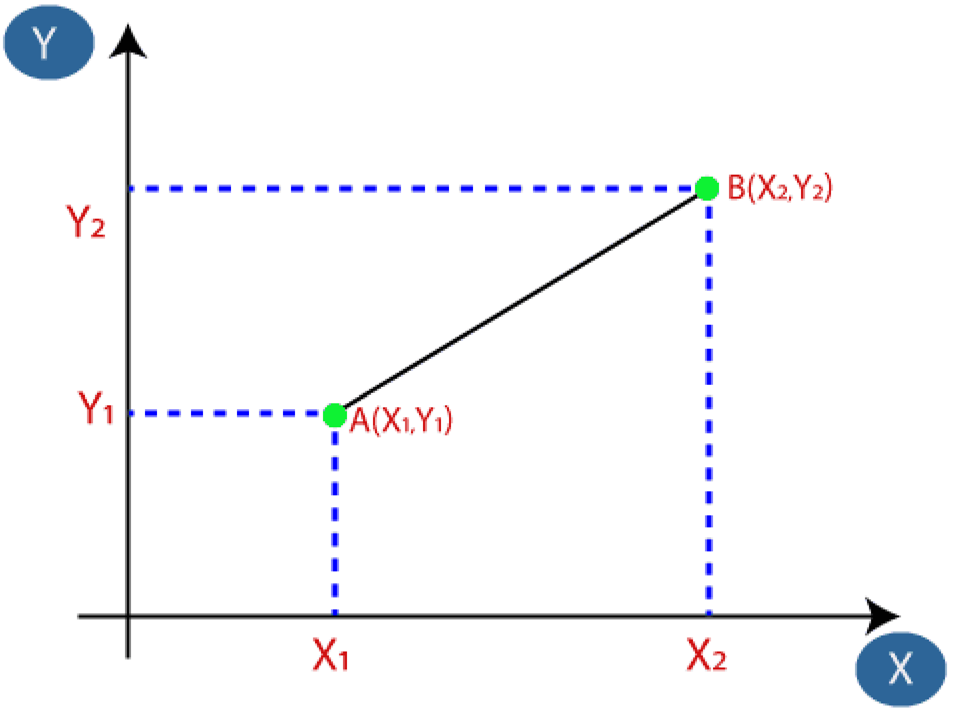
By: Adham Magdy

- Step-2: Calculate the Euclidean distance of K number of neighbors
- Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.
- Step-4: Among these k neighbors, count the number of the data points in each category.
- Step-5: Assign the new data points to that category for which the number of the neighbor is maximum.
- Step-6: Our model is ready.

Suppose we have a new data point and we need to put it in the required category. Consider the below image:

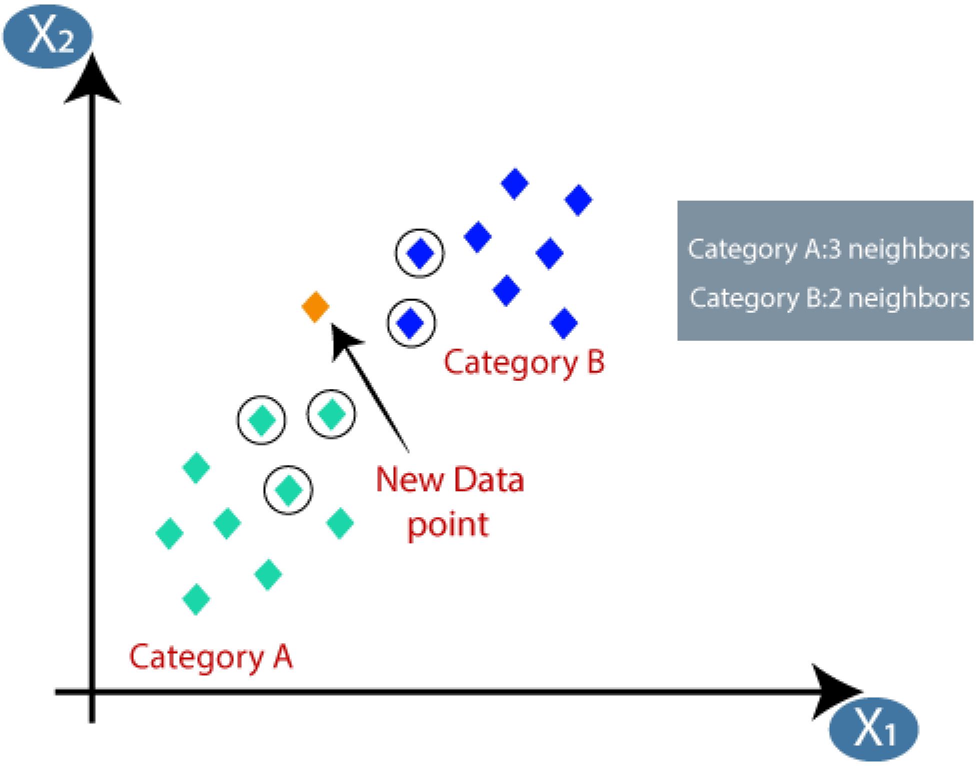


- Firstly, we will choose the number of neighbors, so we will choose the k=5.
- Next, we will calculate the Euclidean distance between the data points. The Euclidean distance is the distance between two points, which we have already studied in geometry. It can be calculated as:



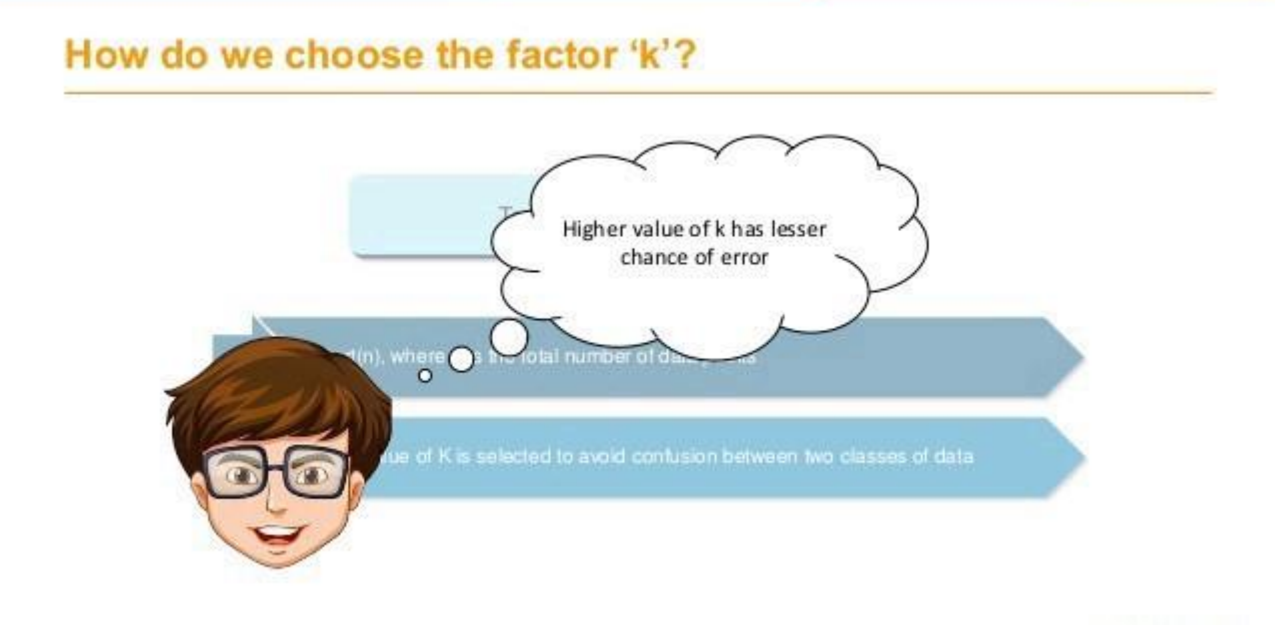
Euclidean Distance between A_1 and B_2 = $\sqrt{(X_2-X_1)^2+(Y_2-Y_1)^2}$

- By calculating the Euclidean distance we got the nearest neighbors, as three nearest neighbors in category A and two nearest neighbors in category B. Consider the below image:



- As we can see the 3 nearest neighbors are from category A, hence this new data point must belong to category A.

How to choose a K value?

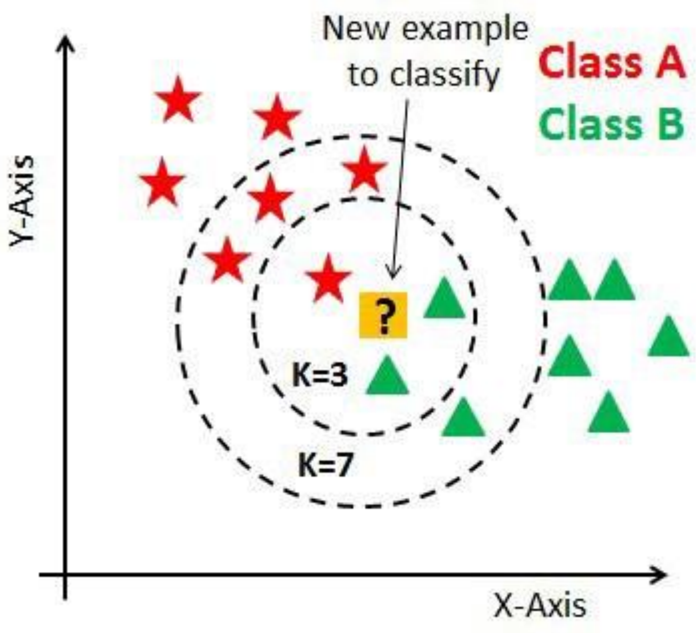


How do we choose the factor ‘k’?

(n), where s the total number of da

ue of K is selected to avoid confusion between two classes of data

Kvalue indicates the count of the nearest neighbors. We have to compute distances between test points and trained labels points. Updating distance metrics with every iteration is computationally expensive, and that’s why KNN is a lazy learning algorithm.



- As you can verify from the above image, if we proceed with K=3, then we predict that test input belongs to class B, and if we continue with K=7, then we predict that test input belongs to class A.

- That’s how you can imagine that the K value has a powerful effect on KNN performance.

Then how to select the optimal K value?

- There are no pre-defined statistical methods to find the most favorable value of K.
- Initialize a random K value and start computing.
- Choosing a small value of K leads to unstable decision boundaries.
- The substantial K value is better for classification as it leads to smoothening the decision boundaries.
- Derive a plot between error rate and K denoting values in a defined range. Then choose the K value as having a minimum error rate.

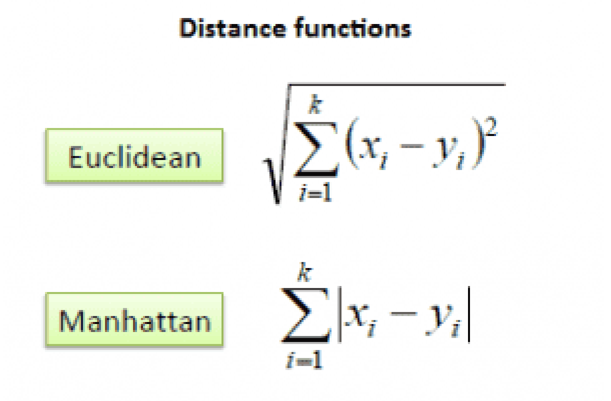
Now you will get the idea of choosing the optimal K value by implementing the model.

Calculating distance:

The first step is to calculate the distance between the new point and each training point. There are various methods for calculating this distance, of which the most commonly known methods are — Euclidian, Manhattan (for continuous) and Hamming distance (for categorical).

Euclidean Distance: Euclidean distance is calculated as the square root of the sum of the squared differences between a new point (x) and an existing point (y).

Manhattan Distance: This is the distance between real vectors using the sum of their absolute difference.



Distance functions

Euclidean $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$

Manhattan $\sum_{i=1}^{k}|x_i - y_i|$

Hamming Distance: It is used for categorical variables. If the value (x) and the value (y) are the same, the distance D will be equal to 0 . Otherwise D=1.

$$D_H = \sum_{i=1}^{k}|x_i - y_i|$$

$$x = y \Rightarrow D = 0$$
$$x \neq y \Rightarrow D = 1$$

7)KMeans clustering

Introduction to K-Means Clustering

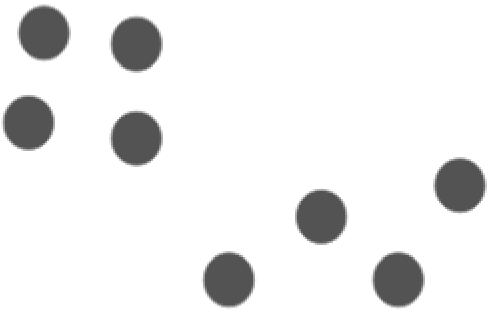
Recall the first property of clusters – it states that the points within a cluster should be similar to each other. So, **our aim here is to minimize the distance between the points within a cluster.**

There is an algorithm that tries to minimize the distance of the points in a cluster with their centroid – the k-means clustering technique.

K-means is a centroid-based algorithm, or a distance-based algorithm, where we calculate the distances to assign a point to a cluster. In K-Means, each cluster is associated with a centroid.

The main objective of the K-Means algorithm is to minimize the sum of distances between the points and their respective cluster centroid.

Let’s now take an example to understand how K-Means actually works:



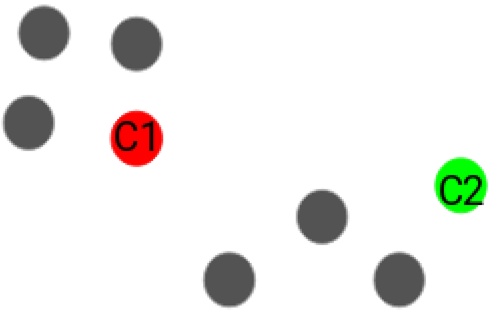
We have these 8 points and we want to apply k-means to create clusters for these points. Here’s how we can do it.

Step 1: Choose the number of clusters *k*

The first step in k-means is to pick the number of clusters, k.

Step 2: Select k random points from the data as centroids

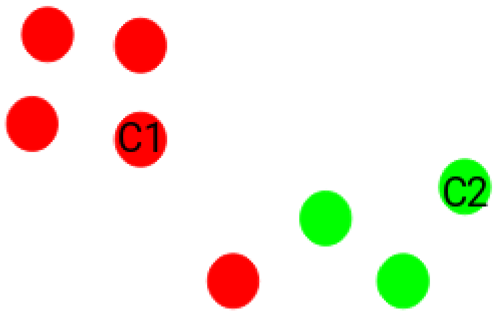
Next, we randomly select the centroid for each cluster. Let’s say we want to have 2 clusters, so k is equal to 2 here. We then randomly select the centroid:



Here, the red and green circles represent the centroid for these clusters.

Step 3: Assign all the points to the closest cluster centroid

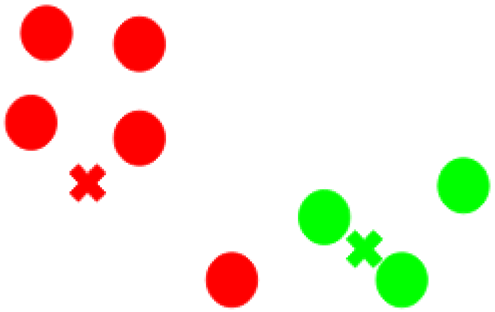
Once we have initialized the centroids, we assign each point to the closest cluster centroid:



Here you can see that the points which are closer to the red point are assigned to the red cluster whereas the points which are closer to the green point are assigned to the green cluster.

Step 4: Recompute the centroids of newly formed clusters

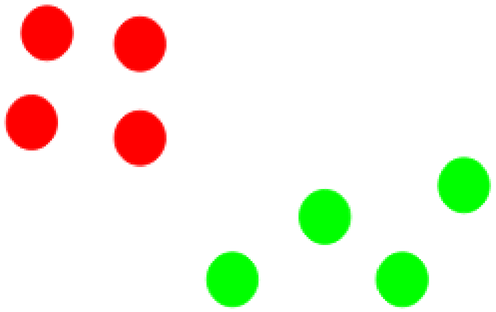
Now, once we have assigned all of the points to either cluster, the next step is to compute the centroids of newly formed clusters:



Here, the red and green crosses are the new centroids.

Step 5: Repeat steps 3 and 4

We then repeat steps 3 and 4:



The step of computing the centroid and assigning all the points to the cluster based on their distance from the centroid is a single iteration. But wait – when should we stop this process? It can't run till eternity, right?

Stopping Criteria for K-Means Clustering

There are essentially three stopping criteria that can be adopted to stop the K-means algorithm:

1. Centroids of newly formed clusters do not change
2. Points remain in the same cluster
3. Maximum number of iterations are reached

We can stop the algorithm if the centroids of newly formed clusters are not changing. Even after multiple iterations, if we are getting the same centroids for all the clusters, we can say that the algorithm is not learning any new pattern and it is a sign to stop the training.

Another clear sign that we should stop the training process if the points remain in the same cluster even after training the algorithm for multiple iterations.

Finally, we can stop the training if the maximum number of iterations is reached. Suppose if we have set the number of iterations as 100. The process will repeat for 100 iterations before stopping.

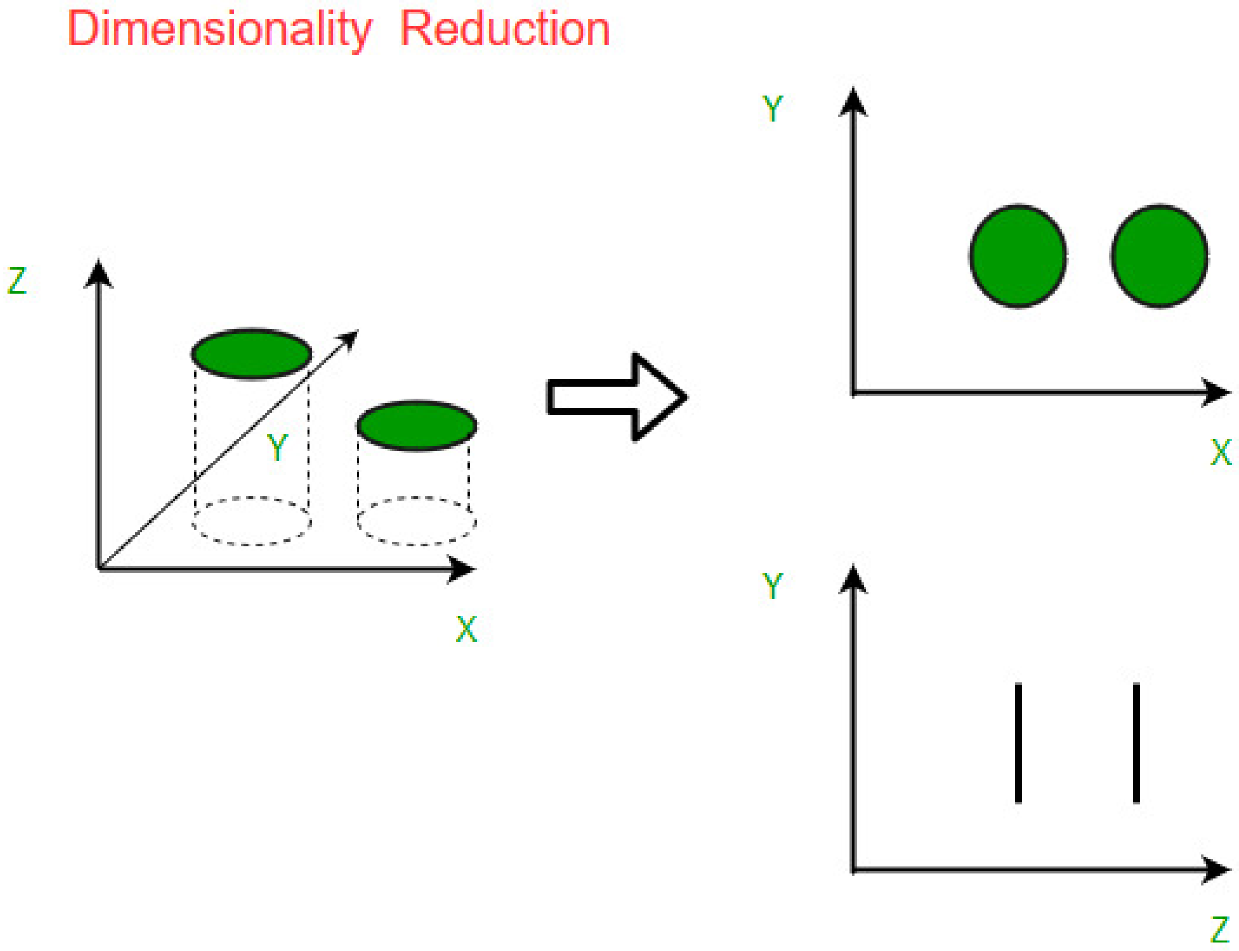
Dimensionality Reduction

What is Dimensionality Reduction?

In machine learning classification problems, there are often too many factors on the basis of which the final classification is done. These factors are basically variables called features. The higher the number of features, the harder it gets to visualize the training set and then work on it. Sometimes, most of these features are correlated, and hence redundant. This is where dimensionality reduction algorithms come into play. 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.

Why is Dimensionality Reduction important in Machine Learning and Predictive Modeling?

An intuitive example of dimensionality reduction can be discussed through a simple e-mail classification problem, where we need to classify whether the e-mail is spam or not. This can involve a large number of features, such as whether or not the e-mail has a generic title, the content of the e-mail, whether the e-mail uses a template, etc. However, some of these features may overlap. In another condition, a classification problem that relies on both humidity and rainfall can be collapsed into just one underlying feature, since both of the aforementioned are correlated to a high degree. Hence, we can reduce the number of features in such problems. A 3-D classification problem can be hard to visualize, whereas a 2-D one can be mapped to a simple 2 dimensional space, and a 1-D problem to a simple line. The below figure illustrates this concept, where a 3-D feature space is split into two 1-D feature spaces, and later, if found to be correlated, the number of features can be reduced even further.



By: Adham Magdy

Components of Dimensionality Reduction

There are two components of dimensionality reduction:

- **Feature selection:** In this, we 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:
    0. Filter
    1. Wrapper
    2. Embedded
- **Feature extraction:** This reduces the data in a high dimensional space to a lower dimension space, i.e. a space with lesser no. of dimensions.

Methods of Dimensionality Reduction

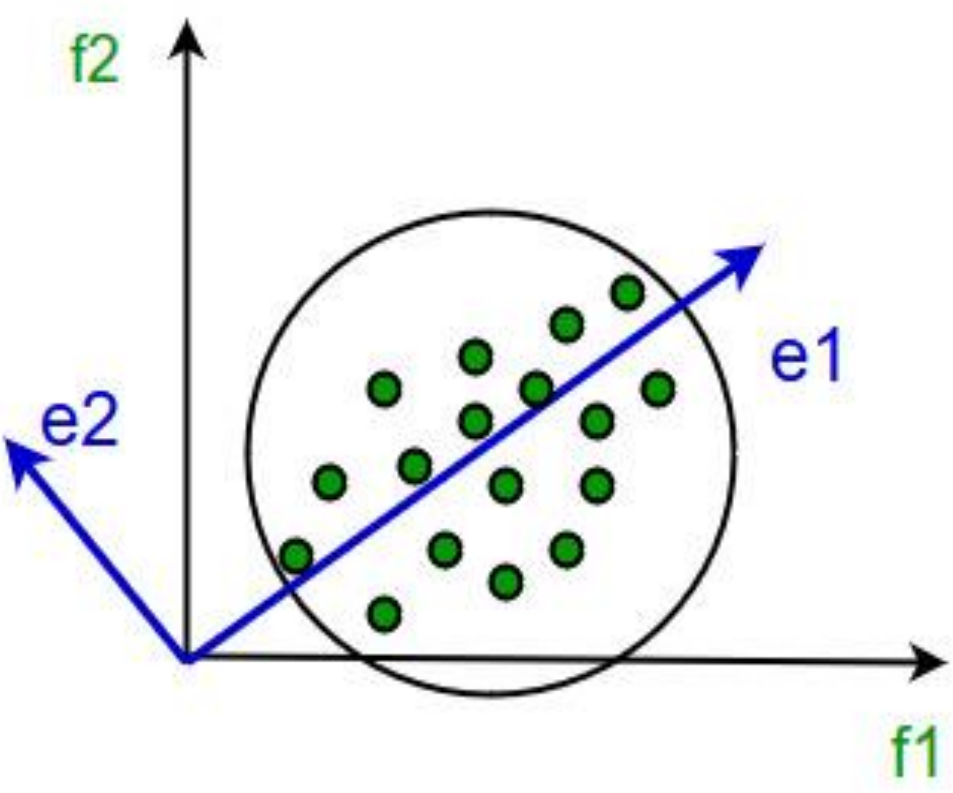
The various methods used for dimensionality reduction include:

- Principal Component Analysis (PCA)
- Linear Discriminant Analysis (LDA)
- Generalized Discriminant Analysis (GDA)

Dimensionality reduction may be both linear or non-linear, depending upon the method used. The prime linear method, called Principal Component Analysis, or PCA, is discussed below.

Principal Component Analysis

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.



It involves the following steps:

- Construct the covariance matrix of the data.
- Compute the eigenvectors of this matrix.
- Eigenvectors corresponding to the largest eigenvalues are used to reconstruct a large fraction of variance of the original data.

Hence, we are left with a lesser number of eigenvectors, and there might have been some data loss in the process. But, the most important variances should be retained by the remaining eigenvectors.

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.

Feature selection vs. Feature Extraction

Feature Selection Concepts & Techniques

Simply speaking, feature selection is about **selecting a subset of features** out of the original features in order to reduce model complexity, enhance the computational efficiency of the models and reduce generalization error introduced due to noise by irrelevant features. The following represents some of the important feature selection techniques:

- **Regularization techniques** such as L1 norm regularisation which results in most features' weight to turn to zero
- **Feature importance techniques** such as using **estimator** such as **Random Forest algorithm** to fit a model and select features based on the value of attribute such as **feature_importances_** .
- **Greedy search algorithms** such as some of the following which are useful for algorithms (such as K-nearest neighbours, K-NN) where regularization techniques are not supported.
  - Sequential forward selection
  - Sequential floating forward selection
  - Sequential backward selection
  - Sequential floating backward selection

According to the utilized training data (labeled, unlabeled, or partially labeled), feature selection methods can be divided into supervised, unsupervised, and semi-supervised models. According to their relationship with learning methods, feature selection methods can be classified into the following:

- **Filter methods:** The filter model only considers the association between the feature and the class label
- Wrapper methods
- **Embedded methods**: In embedded method, the features are selected in the training process of learning model, and the feature selection result outputs automatically while the training process is finished. Training the Lasso regression model is a classic example of embedded method for feature selection.

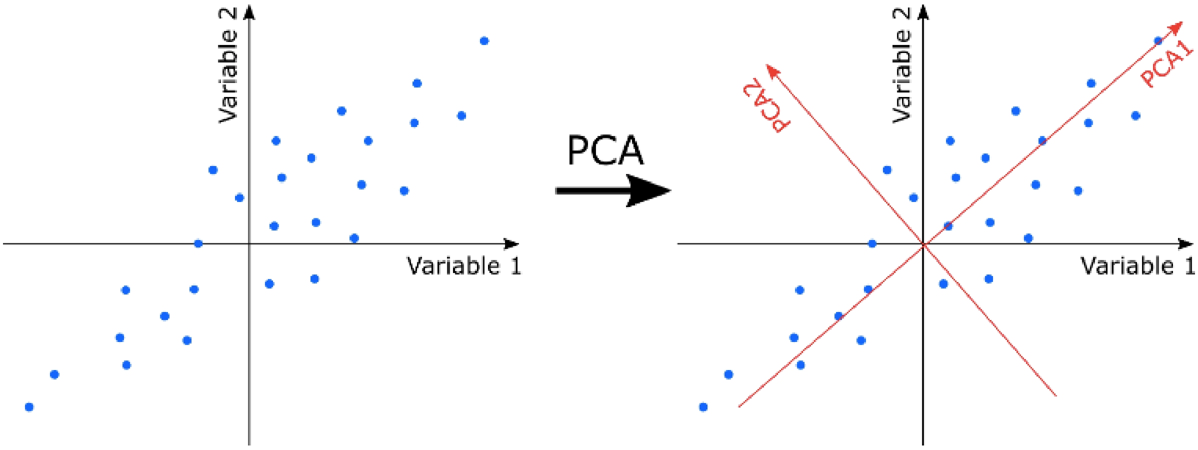
According to the evaluation criterion, feature selection methods can be derived from correlation, Euclidean distance, consistency, dependence and information measures.

According to the type of output, feature selection methods can be divided into feature rank (weighting) and subset selection models.

Feature Extraction Concepts & Techniques

Feature extraction is about **extracting/deriving** information from the original features set to create a new features subspace. The primary idea behind feature extraction is to compress the data with the goal of maintaining most of the relevant information. As with feature selection techniques, these techniques are also used for reducing the number of features from the original features set to reduce model complexity, model overfitting, enhance model computation efficiency and reduce generalization error. The following are different types of feature extraction techniques:

- **Principal component analysis** (**PCA**) for unsupervised data compression. Here is a detailed post on feature extraction using PCA with Python example. You will get a good understanding of how PCA can help with finding the directions of maximum variance in high-dimensional data and projects the data onto a new subspace with equal or fewer dimensions than the original one**.** This is explained with example of identifying **Taj Mahal (7th wonder of world)** from top view or side view based on **dimensions** in which there is **maximum variance**. The diagram below shows the dimensions of maximum variance (PCA1 and PCA2) as a result of PCA.



- **Linear discriminant analysis** (**LDA**) as a supervised dimensionality reduction technique for maximizing class separability
- Nonlinear dimensionality reduction via **kernel principal component analysis** (**KPCA**)

When to use Feature Selection & Feature Extraction

The **key difference** between feature selection and feature extraction techniques used for dimensionality reduction is that while the **original features are maintained** in the case of feature selection algorithms, the feature extraction algorithms **transform the data onto a new feature space**.

Feature selection techniques can be used if the requirement is to **maintain the original features,** unlike the feature extraction techniques which derive useful information from data to construct a new feature subspace. Feature selection techniques are used when model explainability is a key requirement.

Feature extraction techniques can be used to improve the predictive performance of the models, especially, in the case of **algorithms that don't support regularization**.

Unlike feature selection, feature extraction usually needs to transform the original data to features with strong pattern recognition ability, where the original data can be regarded as features with weak recognition ability.

Quiz – Test your knowledge

Here is a quick quiz you can use to check your knowledge on feature selection vs feature extraction.

Feature selection and feature extraction methods are one and same.

True

[✓ False]

Correct!

Which of the following can be used for feature extraction?

PCA

LDA

KPCA

[✓ All of the above]

Correct!

Which of the following technique is used for feature extraction?

Grid search algorithms

[✓ PCA]

Correct!

Which of the following can be used for feature selection?

[✓ Random forest]

LDA

PCA

Correct!

By: Adham Magdy

Which of the following can be used for feature selection?

Regularization techniques

Grid search algorithms

Random forest

All of the above

Correct!

In which of the following techniques, the original features set are maintained?

Feature extraction

Feature selection

Correct!

Which of the following techniques is recommended when original feature set is required to be maintained?

Feature selection

Feature extraction

Correct!

Which of the following technique is recommended when the model interpretability is key requirement?

Feature extraction

Feature selection

Correct!

By: Adham Magdy

Eager classifiers vs. Lazy classifiers

1. Lazy learners

Lazy learners simply store the training data and wait until a testing data appear. When it does, classification is conducted based on the most related data in the stored training data. Compared to eager learners, lazy learners have less training time but more time in predicting.

Ex. k-nearest neighbor, Case-based reasoning

2. Eager learners

Eager learners construct a classification model based on the given training data before receiving data for classification. It must be able to commit to a single hypothesis that covers the entire instance space. Due to the model construction, eager learners take a long time for train and less time to predict.

Ex. Decision Tree, Naive Bayes, Artificial Neural Networks

Deep Learning

Deep Learning

Deep learning attempts to mimic the human brain—albeit far from matching its ability—enabling systems to cluster data and make predictions with incredible accuracy.

What is deep learning?

Deep learning is a subset of machine learning, which is essentially a neural network with three or more layers. These neural networks attempt to simulate the behavior of the human brain—albeit far from matching its ability—allowing it to "learn" from large amounts of data. While a neural network with a single layer can still make approximate predictions, additional hidden layers can help to optimize and refine for accuracy.

Deep learning drives many artificial intelligence (AI) applications and services that improve automation, performing analytical and physical tasks without human intervention. Deep learning technology lies behind everyday products and services (such as digital assistants, voice-enabled TV remotes, and credit card fraud detection) as well as emerging technologies (such as self-driving cars).

Deep learning vs. machine learning

If deep learning is a subset of machine learning, how do they differ? Deep learning distinguishes itself from classical machine learning by the type of data that it works with and the methods in which it learns.

Machine learning algorithms leverage structured, labeled data to make predictions—meaning that specific features are defined from the input data for the model and organized into tables. This doesn't necessarily mean that it doesn't use unstructured data; it just means that if it does, it generally goes through some pre-processing to organize it into a structured format.

Deep learning eliminates some of data pre-processing that is typically involved with machine learning. These algorithms can ingest and process unstructured data, like text and images, and it automates feature extraction, removing some of the dependency on human experts. For example, let's say that we had a set of photos of different pets, and we wanted to categorize by "cat", "dog", "hamster", et cetera. Deep learning algorithms can determine which features (e.g. ears) are most important to distinguish each animal from another. In machine learning, this hierarchy of features is established manually by a human expert.

Then, through the processes of gradient descent and backpropagation, the deep learning algorithm adjusts and fits itself for accuracy, allowing it to make predictions about a new photo of an animal with increased precision.

Machine learning and deep learning models are capable of different types of learning as well, which are usually categorized as supervised learning, unsupervised learning, and reinforcement learning. Supervised learning utilizes labeled datasets to categorize or make predictions; this requires some kind of human intervention to label input data correctly. In contrast, unsupervised learning doesn't require labeled datasets, and instead, it detects patterns in the data, clustering them by any distinguishing characteristics. Reinforcement learning is a process in which a model learns to become more accurate for performing an action in an environment based on feedback in order to maximize the reward.

For a deeper dive on the nuanced differences between the different technologies, see "AI vs. Machine Learning vs. Deep Learning vs. Neural Networks: What's the Difference?"

For a closer look at the specific differences between supervised and unsupervised learning, see "Supervised vs. Unsupervised Learning: What's the Difference?"

How deep learning works

Deep learning neural networks, or artificial neural networks, attempts to mimic the human brain through a combination of data inputs, weights, and bias. These elements work together to accurately recognize, classify, and describe objects within the data.

Deep neural networks consist of multiple layers of interconnected nodes, each building upon the previous layer to refine and optimize the prediction or categorization. This progression of computations through the network is called forward propagation. The input and output layers of a deep neural network are called *visible* layers. The input layer is where the deep learning model ingests the data for processing, and the output layer is where the final prediction or classification is made.

Another process called backpropagation uses algorithms, like gradient descent, to calculate errors in predictions and then adjusts the weights and biases of the function by moving backwards through the layers in an effort to train the model. Together, forward propagation and backpropagation allow a neural network to make predictions and correct for any errors accordingly. Over time, the algorithm becomes gradually more accurate.

The above describes the simplest type of deep neural network in the simplest terms. However, deep learning algorithms are incredibly complex, and there are different types of neural networks to address specific problems or datasets. For example,

- *Convolutional neural networks (CNNs),* used primarily in computer vision and image classification applications, can detect features and patterns within an image, enabling tasks, like object detection or recognition. In 2015, a CNN bested a human in an object recognition challenge for the first time.
- *Recurrent neural network (RNNs)* are typically used in natural language and speech recognition applications as it leverages sequential or times series data.

Deep learning applications

Real-world deep learning applications are a part of our daily lives, but in most cases, they are so well-integrated into products and services that users are unaware of the complex data processing that is taking place in the background. Some of these examples include the following:

Law enforcement

Deep learning algorithms can analyze and learn from transactional data to identify dangerous patterns that indicate possible fraudulent or criminal activity. Speech recognition, computer vision, and other deep learning applications can improve the efficiency and effectiveness of investigative analysis by extracting patterns and evidence from sound and video recordings, images, and documents, which helps law enforcement analyze large amounts of data more quickly and accurately.

Financial services

Financial institutions regularly use predictive analytics to drive algorithmic trading of stocks, assess business risks for loan approvals, detect fraud, and help manage credit and investment portfolios for clients.

Customer service

Many organizations incorporate deep learning technology into their customer service processes. Chatbots—used in a variety of applications, services, and customer service portals—are a straightforward form of AI. Traditional chatbots use natural language and even visual recognition, commonly found in call center-like menus. However, more sophisticated chatbot solutions attempt to determine, through learning, if there are multiple responses to ambiguous questions. Based on the responses it receives, the chatbot then tries to answer these questions directly or route the conversation to a human user.

Virtual assistants like Apple's Siri, Amazon Alexa, or Google Assistant extends the idea of a chatbot by enabling speech recognition functionality. This creates a new method to engage users in a personalized way.

Healthcare

The healthcare industry has benefited greatly from deep learning capabilities ever since the digitization of hospital records and images. Image recognition applications can support medical imaging specialists and radiologists, helping them analyze and assess more images in less time.