UNIT 3

Probabilistic reasoning in Artificial intelligence

Uncertainty:

In AI and expert systems, uncertainty is **measured by using relative frequencies or by combining various statistical models based on data and information collected from various sources**. Some of these measures are objective in nature while others may be from domain experts

Causes of uncertainty:

Following are some leading causes of uncertainty to occur in the real world.

1. Information occurred from unreliable sources.
2. Experimental Errors
3. Equipment fault
4. Temperature variation
5. Climate change.

Probabilistic reasoning:

Probabilistic reasoning is a way of knowledge representation where we apply the concept of probability to indicate the uncertainty in knowledge. In probabilistic reasoning, we combine probability theory with logic to handle the uncertainty.

**Need of probabilistic reasoning in AI:**

* When there are unpredictable outcomes.
* When specifications or possibilities of predicates becomes too large to handle.
* When an unknown error occurs during an experiment.

In probabilistic reasoning, there are two ways to solve problems with uncertain knowledge:

* **Bayes' rule**
* **Bayesian Statistics**
* As probabilistic reasoning uses probability and related terms, so before understanding probabilistic reasoning, let's understand some common terms:
* **Probability:** Probability can be defined as a chance that an uncertain event will occur. It is the numerical measure of the likelihood that an event will occur. The value of probability always remains between 0 and 1 that represent ideal uncertainties.

1. 0 ≤ P(A) ≤ 1,   where P(A) is the probability of an event A.
2. P(A) = 0,  indicates total uncertainty in an event A.
3. P(A) =1, indicates total certainty in an event A.

We can find the probability of an uncertain event by using the below formula.

Probabilistic reasoning in Artificial intelligence

* P(¬A) = probability of a not happening event.
* P(¬A) + P(A) = 1.

**Event:** Each possible outcome of a variable is called an event.

**Sample space:** The collection of all possible events is called sample space.

**Random variables:** Random variables are used to represent the events and objects in the real world.

**Prior probability:** The prior probability of an event is probability computed before observing new information.

**Posterior Probability:** The probability that is calculated after all evidence or information has taken into account. It is a combination of prior probability and new information.

## Conditional probability:

Conditional probability is a probability of occurring an event when another event has already happened.

Let's suppose, we want to calculate the event A when event B has already occurred, "the probability of A under the conditions of B", it can be written as:

Probabilistic reasoning in Artificial intelligence

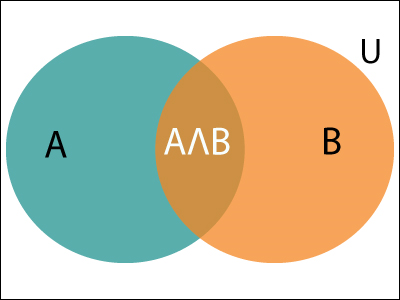
**Where P(*A*⋀*B*)= Joint probability of a and B**

**P(B)= Marginal probability of B.**

If the probability of A is given and we need to find the probability of B, then it will be given as:

Probabilistic reasoning in Artificial intelligence

It can be explained by using the below Venn diagram, where B is occurred event, so sample space will be reduced to set B, and now we can only calculate event A when event B is already occurred by dividing the probability of **P(A⋀*B*) by P( B )**.



**Example:**

In a class, there are 70% of the students who like English and 40% of the students who likes English and mathematics, and then what is the percent of students those who like English also like mathematics?

**Solution:**

Let, A is an event that a student likes Mathematics

B is an event that a student likes English.

Probabilistic reasoning in Artificial intelligence

**Hence, 57% are the students who like English also like Mathematics.**

Bayes' theorem in Artificial intelligence

Bayes' theorem:

Bayes' theorem is also known as **Bayes' rule, Bayes' law**, or **Bayesian reasoning**, which determines the probability of an event with uncertain knowledge.

In probability theory, it relates the conditional probability and marginal probabilities of two random events.

Bayes' theorem was named after the British mathematician **Thomas Bayes**. The **Bayesian inference** is an application of Bayes' theorem, which is fundamental to Bayesian statistics.

It is a way to calculate the value of P(B|A) with the knowledge of P(A|B).

Bayes' theorem allows updating the probability prediction of an event by observing new information of the real world.

**Example**: If cancer corresponds to one's age then by using Bayes' theorem, we can determine the probability of cancer more accurately with the help of age.

Bayes' theorem can be derived using product rule and conditional probability of event A with known event B:

As from product rule we can write:

1. P(A ⋀ B)= P(A|B) P(B) or

Similarly, the probability of event B with known event A:

1. P(A ⋀ B)= P(B|A) P(A)

Equating right hand side of both the equations, we will get:

Bayes theorem in Artificial intelligence

The above equation (a) is called as **Bayes' rule** or**Bayes' theorem**. This equation is basic of most modern AI systems for **probabilistic inference**.

**Question: From a standard deck of playing cards, a single card is drawn. The probability that the card is king is 4/52, then calculate posterior probability P(King|Face), which means the drawn face card is a king card.**

**Solution:**

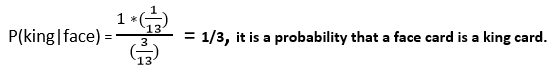
Bayes theorem in Artificial intelligence

P(king): probability that the card is King= 4/52= 1/13

P(face): probability that a card is a face card= 3/13

P(Face|King): probability of face card when we assume it is a king = 1

Putting all values in equation (i) we will get:



Application of Bayes' theorem in Artificial intelligence:

**Following are some applications of Bayes' theorem:**

* It is used to calculate the next step of the robot when the already executed step is given.
* Bayes' theorem is helpful in weather forecasting.
* It can solve the Monty Hall problem.
* Bayesian Belief Network in artificial intelligence
* Bayesian belief network is key computer technology for dealing with probabilistic events and to solve a problem which has uncertainty. We can define a Bayesian network as:
* "A Bayesian network is a probabilistic graphical model which represents a set of variables and their conditional dependencies using a directed acyclic graph."
* It is also called a **Bayes network, belief network, decision network**, or **Bayesian model**.
* Bayesian networks are probabilistic, because these networks are built from a **probability distribution**, and also use probability theory for prediction and anomaly detection.

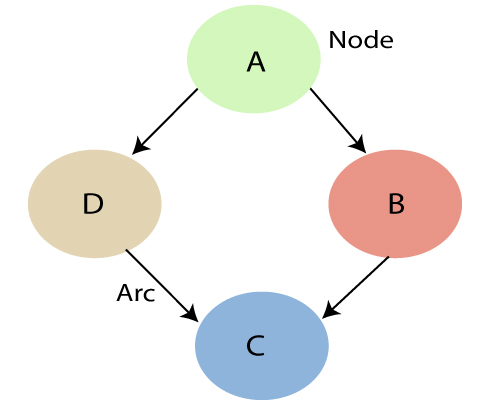
Real world applications are probabilistic in nature, and to represent the relationship between multiple events, we need a Bayesian network. It can also be used in various tasks including **prediction, anomaly detection, diagnostics, automated insight, reasoning, time series prediction**, and **decision making under uncertainty**.

Bayesian Network can be used for building models from data and experts opinions, and it consists of two parts:

* **Directed Acyclic Graph**
* **Table of conditional probabilities.**

The generalized form of Bayesian network that represents and solve decision problems under uncertain knowledge is known as an **Influence diagram**.

**A Bayesian network graph is made up of nodes and Arcs (directed links), where:**



* Each **node** corresponds to the random variables, and a variable can be **continuous** or **discrete**.
* **Arc or directed arrows** represent the causal relationship or conditional probabilities between random variables. These directed links or arrows connect the pair of nodes in the graph.  
  These links represent that one node directly influence the other node, and if there is no directed link that means that nodes are independent with each other
  + **In the above diagram, A, B, C, and D are random variables represented by the nodes of the network graph.**
  + **If we are considering node B, which is connected with node A by a directed arrow, then node A is called the parent of Node B.**
  + **Node C is independent of node A.**

The Bayesian network has mainly two components:

* **Causal Component**
* **Actual numbers**

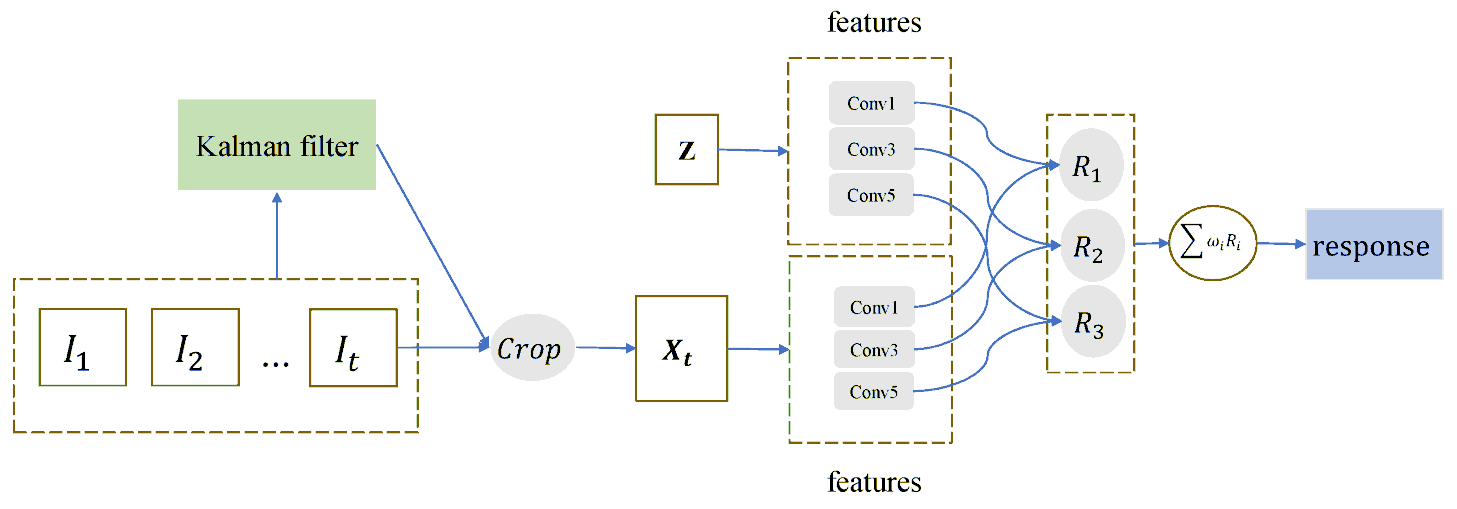
Each node in the Bayesian network has condition probability distribution **P(Xi |Parent(Xi) )**, which determines the effect of the parent on that node.

Bayesian network is based on Joint probability distribution and conditional probability. So let's first understand the joint probability distribution:

**Kalman Filter algorithm**

A Kalman Filter is an algorithm that takes data inputs from multiple sources and estimates unknown variables, despite a potentially high level of signal noise. Often used in navigation and control technology, the Kalman Filter has the advantage of being able to predict unknown values more accurately than if individual predictions are made using singular methods of measurement.

Kalman Filters use a two-step process for estimating unknown variables. The algorithm works by first estimating the current state variables, and measures their uncertainties. Then, the algorithm updates the estimates using a weighted average, wherein more weight is attributed to estimates with higher levels of uncertainty. Because the filter takes in information from multiple sources, both current state and predicted state, the filter is able to provide a level of accuracy higher than if estimates were made given only one of the multiple sources.



### Kalman Filter and Machine Learning

One of the most common uses for the Kalman Filter is in navigation and positioning technology. Imagine a car with a GPS transmitter is traveling down a mountain road. A Kalman Filter can be applied to take in the GPS data from the car, however GPS devices are not always entirely accurate. So, the Kalman Filter can take in speed and velocity data to adjust the rate of change in the cars position over time. Naturally, given the laws of physics, the level of variable uncertainty is lower when the car is traveling faster, and vice versa. All of this information is used to predict where the car will be, and then the process is repeated with updated information as the car travels down the road. Because the Kalman Filter is recursive, it doesn't need to know the entirety of the cars position and speed data, but rather just the last known position and speed. The underlying model of updating information is similar to that of a Hidden Markov model.

# Rules of Inference in Artificial intelligence

## Inference:

In artificial intelligence, we need intelligent computers which can create new logic from old logic or by evidence, **so generating the conclusions from evidence and facts is termed as Inference**.

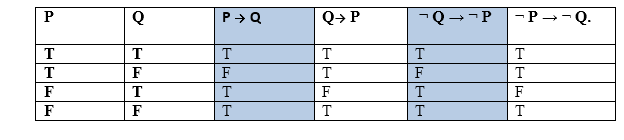
## Inference rules:

Inference rules are the templates for generating valid arguments. Inference rules are applied to derive proofs in artificial intelligence, and the proof is a sequence of the conclusion that leads to the desired goal.

In inference rules, the implication among all the connectives plays an important role. Following are some terminologies related to inference rules:

* **Implication:** It is one of the logical connectives which can be represented as P → Q. It is a Boolean expression.
* **Converse:** The converse of implication, which means the right-hand side proposition goes to the left-hand side and vice-versa. It can be written as Q → P.
* **Contrapositive:** The negation of converse is termed as contrapositive, and it can be represented as ¬ Q → ¬ P.
* **Inverse:** The negation of implication is called inverse. It can be represented as ¬ P → ¬ Q.

From the above term some of the compound statements are equivalent to each other, which we can prove using truth table:



Hence from the above truth table, we can prove that P → Q is equivalent to ¬ Q → ¬ P, and Q→ P is equivalent to ¬ P → ¬ Q.

## Types of Inference rules:

### 1. Modus Ponens:

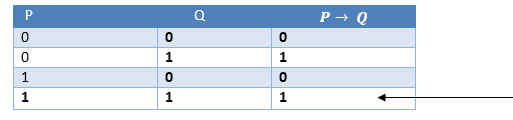
The Modus Ponens rule is one of the most important rules of inference, and it states that if P and P → Q is true, then we can infer that Q will be true. It can be represented as:

Rules of Inference in Artificial intelligence

**Example:**

Statement-1: "If I am sleepy then I go to bed" ==> P→ Q  
Statement-2: "I am sleepy" ==> P  
Conclusion: "I go to bed." ==> Q.  
Hence, we can say that, if P→ Q is true and P is true then Q will be true.

**Proof by Truth table:**



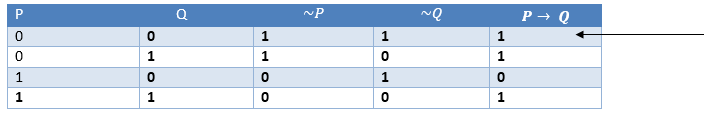
### 2. Modus Tollens:

The Modus Tollens rule state that if P→ Q is true and **¬ Q is true, then ¬ P** will also true. It can be represented as:

Rules of Inference in Artificial intelligence

**Statement-1:** "If I am sleepy then I go to bed" ==> P→ Q  
**Statement-2:** "I do not go to the bed."==> ~Q  
**Statement-3:** Which infers that "**I am not sleepy**" => ~P

**Proof by Truth table:**



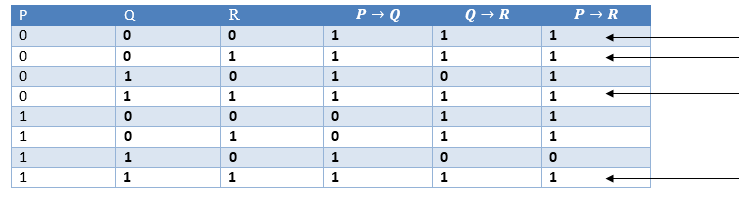
### 3. Hypothetical Syllogism:

The Hypothetical Syllogism rule state that if P→R is true whenever P→Q is true, and Q→R is true. It can be represented as the following notation:

**Example:**

**Statement-1:** If you have my home key then you can unlock my home. **P→Q**  
**Statement-2:** If you can unlock my home then you can take my money. **Q→R**  
**Conclusion:** If you have my home key then you can take my money. **P→R**

**Proof by truth table:**



### 4. Disjunctive Syllogism:

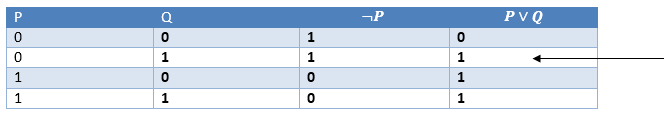
The Disjunctive syllogism rule state that if P∨Q is true, and ¬P is true, then Q will be true. It can be represented as:

Rules of Inference in Artificial intelligence

**Example:**

**Statement-1:** Today is Sunday or Monday. ==>P∨Q  
**Statement-2:** Today is not Sunday. ==> ¬P  
**Conclusion:** Today is Monday. ==> Q

**Proof by truth-table:**



### 5. Addition:

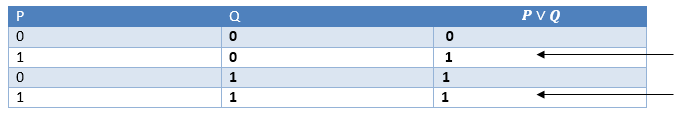
The Addition rule is one the common inference rule, and it states that If P is true, then P∨Q will be true.

Rules of Inference in Artificial intelligence

**Example:**

**Statement:** I have a vanilla ice-cream. ==> P  
**Statement-2:** I have Chocolate ice-cream.  
**Conclusion:** I have vanilla or chocolate ice-cream. ==> (P∨Q)

**Proof by Truth-Table:**

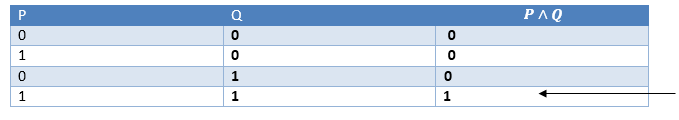


### 6. Simplification:

The simplification rule state that if **P∧ Q** is true, then **Q or P** will also be true. It can be represented as:

Rules of Inference in Artificial intelligence

**Proof by Truth-Table:**

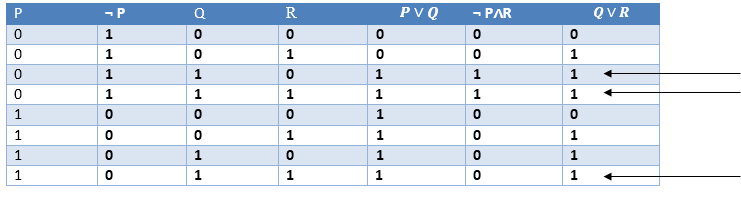


### 7. Resolution:

The Resolution rule state that if P∨Q and ¬ P∧R is true, then Q∨R will also be true. **It can be represented as**

Rules of Inference in Artificial intelligence

**Proof by Truth-Table:**



**Hidden Markov Model** **he**

A Hidden Markov Model (HMM) is a statistical model which is also used in machine learning. It can be used to describe the evolution of observable events that depend on internal factors, which are not directly observable. These are a class of probabilistic graphical models that allow us to predict a sequence of unknown variables from a set of observed variables. In this article, we will discuss the Hidden Markov Models in detail. We will understand the contexts where it can be used and we will also discuss its different applications. We will also discuss the use of HMM for PoS tagging with python implementation .

**Hidden Markov Model**

The Hidden Markov model is a probabilistic model which is used to explain or derive the probabilistic characteristic of any random process. It basically says that an observed event will not be corresponding to its step-by-step status but related to a set of probability distributions. Let’s assume a system that is being modelled is assumed to be a Markov chain and in the process, there are some hidden states. In that case, we can say that hidden states are a process that depends on the main Markov process/chain.

**Hidden Markov Model With an Example**

To explain it more we can take the example of two friends, Rahul and Ashok. Now Rahul completes his daily life works according to the weather conditions. Major three activities completed by Rahul are- go jogging, go to the office, and cleaning his residence. What Rahul is doing today depends on whether and whatever Rahul does he tells Ashok and Ashok has no proper information about the weather But Ashok can assume the weather condition according to Rahul work.

Ashok believes that the weather operates as a discrete Markov chain, wherein the chain there are only two states whether the weather is Rainy or it is sunny. The condition of the weather cannot be observed by Ashok, here the conditions of the weather are hidden from Ashok. On each day, there is a certain chance that Bob will perform one activity from the set of the following activities {“jog”, “work”,” clean”}, which are depending on the weather. Since Rahul tells Ashok that what he has done, those are the observations. The entire system is that of a hidden Markov model (HMM).

Here we can say that the parameter of HMM is known to Ashok because he has general information about the weather and he also knows what Rahul likes to do on average.

So let’s consider a day where Rahul called Ashok and told him that he has cleaned his residence. In that scenario, Ashok will have a belief that there are more chances of a rainy day and we can say that belief Ashok has is the start probability of HMM let’s say which is like the following.

The states and observation are:

states = ('Rainy', 'Sunny')

observations = ('walk', 'shop', 'clean')

And the start probability is:

start\_probability = {'Rainy': 0.6, 'Sunny': 0.4}

Now the distribution of the probability has the weightage more on the rainy day stateside so we can say there will be more chances for a day to being rainy again and the probabilities for next day weather states are as following

transition\_probability = {

   'Rainy' : {'Rainy': 0.7, 'Sunny': 0.3},

   'Sunny' : {'Rainy': 0.4, 'Sunny': 0.6},

   }

From the above we can say the changes in the probability for a day is transition probabilities and according to the transition probability the emitted results for the probability of work that Rahul will perform is

emission\_probability = {

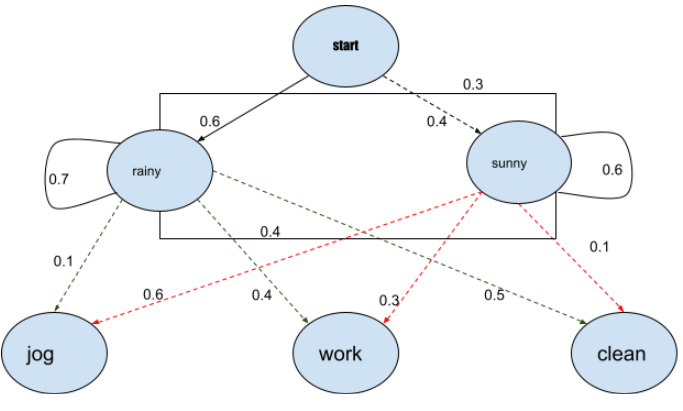
   'Rainy' : {'jog': 0.1, 'work': 0.4, 'clean': 0.5},

   'Sunny' : {'jog': 0.6, 'work: 0.3, 'clean': 0.1},

   }

This probability can be considered as the emission probability. Using the emission probability Ashok can predict the states of the weather or using the transition probabilities Ashok can predict the work which Rahul is going to perform the next day.

Below image shown the HMM process for making probabilities



So here from the above intuition and the example we can understand how we can use this probabilistic model to make a prediction. Now let’s just discuss the applications where it can be used.

**Application of Hidden Markov Model**

An application, where HMM is used, aims to recover the data sequence where the next sequence of the data can not be observed immediately but the next data depends on the old sequences. Taking the above intuition into account the HMM can be used in the following applications:

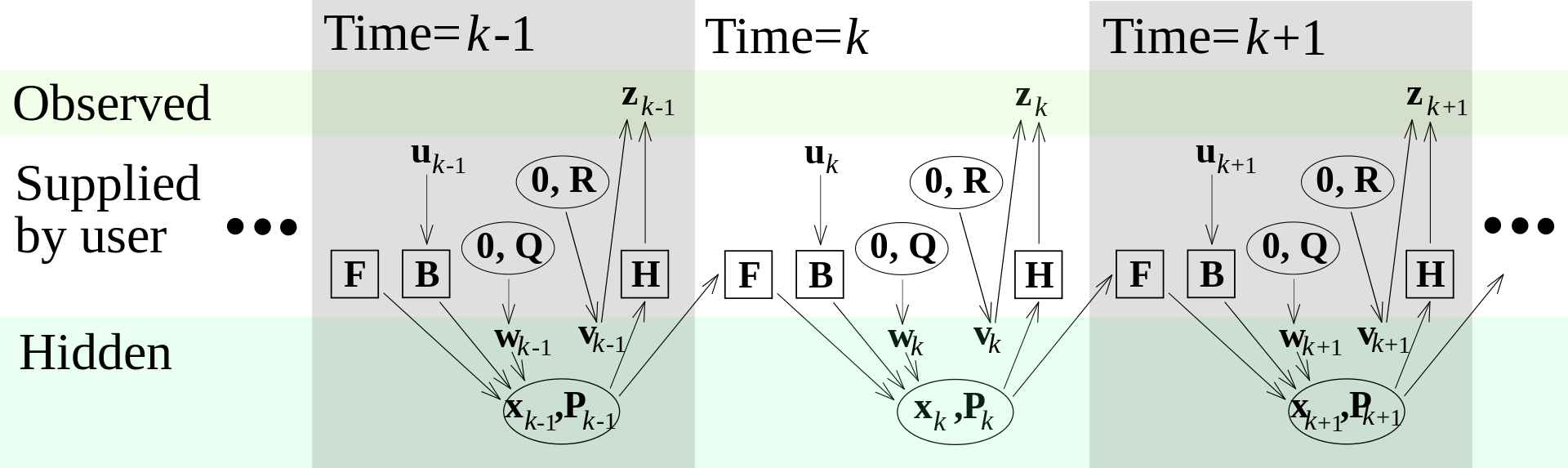
* Computational finance
* speed analysis
* Speech recognition
* Speech synthesis
* Part-of-speech tagging
* Document separation in scanning solutions
* Machine translation
* Handwriting recognition
* Time series analysis
* Activity recognition
* Sequence classification
* Transportation forecasting

## Kalman Filter

A Kalman Filter is an algorithm that takes data inputs from multiple sources and estimates unknown variables, despite a potentially high level of signal noise. Often used in navigation and control technology, the Kalman Filter has the advantage of being able to predict unknown values more accurately than if individual predictions are made using singular methods of measurement.

## How does a Kalman Filter work

Kalman Filters use a two-step process for estimating unknown variables. The algorithm works by first estimating the current state variables, and measures their uncertainties. Then, the algorithm updates the estimates using a weighted average, wherein more weight is attributed to estimates with higher levels of uncertainty. Because the filter takes in information from multiple sources, both current state and predicted state, the filter is able to provide a level of accuracy higher than if estimates were made given only one of the multiple sources.



### Kalman Filter and Machine Learning

One of the most common uses for the Kalman Filter is in navigation and positioning technology. Imagine a car with a GPS transmitter is traveling down a mountain road. A Kalman Filter can be applied to take in the GPS data from the car; however GPS devices are not always entirely accurate. So, the Kalman Filter can take in speed and velocity data to adjust the rate of change in the cars position over time. Naturally, given the laws of physics, the level of variable uncertainty is lower when the car is traveling faster, and vice versa. All of this information is used to predict where the car will be, and then the process is repeated with updated information as the car travels down the road. Because the Kalman Filter is recursive, it doesn't need to know the entirety of the cars position and speed data, but rather just the last known position and speed. The underlying model of updating information is similar to that of a Hidden Markov model.

### Speech recognition

Speech recognition, or speech-to-text, is the ability of a machine or program to identify words spoken aloud and convert them into readable text. Rudimentary speech recognition software has a limited vocabulary and may only identify words and phrases when spoken clearly. More sophisticated software can handle natural speech, different accents and various languages.

Speech recognition uses a broad array of research in computer science, linguistics and computer engineering. Many modern devices and text-focused programs have speech recognition functions in them to allow for easier or hands-free use of a device.

Speech recognition and voice recognition are two different technologies and should not be confused:

* **Speech recognition** is used to identify words in spoken language.
* **Voice recognition** is a biometric technology for identifying an individual's voice.

### How does speech recognition work?

Speech recognition systems use computer algorithms to process and interpret spoken words and convert them into text. A software program turns the sound a microphone records into written language that computers and humans can understand, following these four steps:

1. analyze the audio;
2. break it into parts;
3. digitize it into a computer-readable format; and
4. Use an algorithm to match it to the most suitable text representation.

Speech recognition software must adapt to the highly variable and context-specific nature of human speech. The software algorithms that process and organize audio into text are trained on different speech patterns, speaking styles, languages, dialects, accents and phrasings. The software also separates spoken audio from background noise that often accompanies the signal.

To meet these requirements, speech recognition systems use two types of models:

* **Acoustic models.** These represent the relationship between linguistic units of speech and audio signals.
* **Language models.** Here, sounds are matched with word sequences to distinguish between words that sound similar.

### What applications is speech recognition used for

Speech recognition systems have quite a few applications. Here is a sampling of them.

**Mobile devices.** Smartphone’s use voice commands for call routing, speech-to-text processing, voice dialing and voice search. Users can respond to a text without looking at their devices. On Apple phones, speech recognition powers the keyboard and Siri, the virtual assistant. Functionality is available in secondary languages, too. Speech recognition can also be found in word processing applications like Microsoft Word, where users can dictate words to be turned into text.

**orld's most comprehensive  
data science & artificial intelligence  
glossary**

VIEW GLOSSARY

Top of Form

**Get the week's most  
popular data science  
research in your inbox -  
every Saturday**