PAPER: Modeling and Simulation

TIME: 90 Minutes

INSTRUCTIONS: Copy all of the questions below, then click "Answer Sheet" button

Ongoing activities are currently being transmitted into your student database

TEN (10) QUESTIONS [100 Marks]

First enter your Full Names, follow by (Course Title) in bracket, into the Answer Sheet.

1. What ss a Least Squares Model?[10 Marks]
2. What is modeling?[10 Marks]
3. What is simulation?[10 Marks]
4. Mention types of simulation[10 Marks]
5. Why do we use models?[10 Marks]
6. How is simulation performed?[10 Marks]
7. What is optimization?[10 Marks]
8. What is independent replication?[10 Marks]
9. What is batch means?[10 Marks]
10. What is social simulation? [10 Marks]

1. What is a Least Squares Model?

The least squares method is a form of mathematical regression analysis used to determine the line of best fit for a set of data, providing a visual demonstration of the relationship between the data points.

The method of least squares is a standard approach in regression analysis to approximate the solution of overdetermined systems (sets of equations in which there are more equations than unknowns) by minimizing the sum of the squares of the residuals (a residual being the difference between an observed value and the fitted value provided by a model) made in the results of each individual equation.

The most important application is in data fitting. When the problem has substantial uncertainties in the independent variable (the x variable), then simple regression and least-squares methods have problems; in such cases, the methodology required for fitting errors-in-variables models may be considered instead of that for least squares.

Least squares problems fall into two categories: linear or ordinary least squares and nonlinear least squares, depending on whether or not the residuals are linear in all unknowns. The linear least-squares problem occurs in statistical regression analysis; it has a closed-form solution. The nonlinear problem is usually solved by iterative refinement; at each iteration the system is approximated by a linear one, and thus the core calculation is similar in both cases.

Polynomial least squares describes the variance in a prediction of the dependent variable as a function of the independent variable and the deviations from the fitted curve.

When the observations come from an exponential family with identity as its natural sufficient statistics and mild-conditions are satisfied (e.g. for normal, exponential, Poisson and binomial distributions), standardized least-squares estimates and maximum-likelihood estimates are identical.[1] The method of least squares can also be derived as a method of moments estimator.

The following discussion is mostly presented in terms of linear functions but the use of least squares is valid and practical for more general families of functions. Also, by iteratively applying local quadratic approximation to the likelihood (through the Fisher information), the least-squares method may be used to fit a generalized linear model.

The least-squares method was officially discovered and published by Adrien-Marie Legendre (1805),[2] though it is usually also co-credited to Carl Friedrich Gauss (1809),[3][4] who contributed significant theoretical advances to the method and may have also used it in his earlier work (1795).[5][6]

History

Founding

The method of least squares grew out of the fields of astronomy and geodesy, as scientists and mathematicians sought to provide solutions to the challenges of navigating the Earth's oceans during the Age of Discovery. The accurate description of the behavior of celestial bodies was the key to enabling ships to sail in open seas, where sailors could no longer rely on land sightings for navigation.

The method was the culmination of several advances that took place during the course of the eighteenth century:[7]

The combination of different observations as being the best estimate of the true value; errors decrease with aggregation rather than increase, perhaps first expressed by Roger Cotes in 1722.

The combination of different observations taken under the same conditions contrary to simply trying one's best to observe and record a single observation accurately. The approach was known as the method of averages. This approach was notably used by Tobias Mayer while studying the librations of the moon in 1750, and by Pierre-Simon Laplace in his work in explaining the differences in motion of Jupiter and Saturn in 1788.

The combination of different observations taken under different conditions. The method came to be known as the method of least absolute deviation. It was notably performed by Roger Joseph Boscovich in his work on the shape of the earth in 1757 and by Pierre-Simon Laplace for the same problem in 1799.

The development of a criterion that can be evaluated to determine when the solution with the minimum error has been achieved. Laplace tried to specify a mathematical form of the probability density for the errors and define a method of estimation that minimizes the error of estimation. For this purpose, Laplace used a symmetric two-sided exponential distribution we now call Laplace distribution to model the error distribution, and used the sum of absolute deviation as error of estimation. He felt these to be the simplest assumptions he could make, and he had hoped to obtain the arithmetic mean as the best estimate. Instead, his estimator was the posterior median.

The method

Carl Friedrich Gauss

The first clear and concise exposition of the method of least squares was published by Legendre in 1805.[8] The technique is described as an algebraic procedure for fitting linear equations to data and Legendre demonstrates the new method by analyzing the same data as Laplace for the shape of the earth. Within ten years after Legendre's publication, the method of least squares had been adopted as a standard tool in astronomy and geodesy in France, Italy, and Prussia, which constitutes an extraordinarily rapid acceptance of a scientific technique.[7]

In 1809 Carl Friedrich Gauss published his method of calculating the orbits of celestial bodies. In that work he claimed to have been in possession of the method of least squares since 1795.[9] This naturally led to a priority dispute with Legendre. However, to Gauss's credit, he went beyond Legendre and succeeded in connecting the method of least squares with the principles of probability and to the normal distribution. He had managed to complete Laplace's program of specifying a mathematical form of the probability density for the observations, depending on a finite number of unknown parameters, and define a method of estimation that minimizes the error of estimation. Gauss showed that the arithmetic mean is indeed the best estimate of the location parameter by changing both the probability density and the method of estimation. He then turned the problem around by asking what form the density should have and what method of estimation should be used to get the arithmetic mean as estimate of the location parameter. In this attempt, he invented the normal distribution.

An early demonstration of the strength of Gauss's method came when it was used to predict the future location of the newly discovered asteroid Ceres. On 1 January 1801, the Italian astronomer Giuseppe Piazzi discovered Ceres and was able to track its path for 40 days before it was lost in the glare of the sun. Based on these data, astronomers desired to determine the location of Ceres after it emerged from behind the sun without solving Kepler's complicated nonlinear equations of planetary motion. The only predictions that successfully allowed Hungarian astronomer Franz Xaver von Zach to relocate Ceres were those performed by the 24-year-old Gauss using least-squares analysis.

In 1810, after reading Gauss's work, Laplace, after proving the central limit theorem, used it to give a large sample justification for the method of least squares and the normal distribution. In 1822, Gauss was able to state that the least-squares approach to regression analysis is optimal in the sense that in a linear model where the errors have a mean of zero, are uncorrelated, and have equal variances, the best linear unbiased estimator of the coefficients is the least-squares estimator. This result is known as the Gauss–Markov theorem.

The idea of least-squares analysis was also independently formulated by the American Robert Adrain in 1808. In the next two centuries workers in the theory of errors and in statistics found many different ways of implementing least squares.[10]

2. What is modeling?

Modelling is about building representations of things in the ‘real world’ and allowing ideas to be investigated; it is central to all activities in the process for building or creating an artefact of some form or other. In effect, a model is a way of expressing a particular view of an identifiable system of some kind. Models are:

a means of understanding the problems involved in building something;

an aid to communication between those involved in the project, especially between the requirements analyst (a development role) and the user, as part of some deliverable;

a component of the methods used in development activities such as the analysis of the requirements for an artefact and the design of the artefact.

A model is an abstraction, which allows people to concentrate on the essentials of a (complex) problem by keeping out non-essential details. Since there is a limit to how much a person can understand at any one time, we build models to help in activities such as the development of large software systems. For example, developers build different models throughout the development process in order to verify that the eventual software system will meet the requirements.

Models are, in one respect, idealisations in the sense that they are less complicated than reality; they are simplifications of reality. The benefit arises from the fact that only the properties of the world relevant to the job in hand are represented. For example, a road map is a model of a particular part of the earth's surface. We do not show things like vegetation or birds' nests as they are not relevant to the map's purpose. We use a road map to plan our journeys from one place to another and so the map should only contain those aspects of the real world that serve the purpose of planning journeys.

A model and the real world are alike in some ways, but different in others. For example, road maps are helpful because they represent the distance between, and relative positions of, a set of places and the routes between them. They use the relevant properties of the real thing with just a change in scale; one centimetre on the road map, for instance, may be equivalent to one kilometre on the ground. A map may be unhelpful if it shows only major roads.

Quite often, a property of the real world may be represented by a different kind of property in a model. In the case of the road map, different colours are normally used to represent different classes or types of road. Such a road map should have a key or legend so that those who read the map can understand what the different coloured lines are intended to represent. In effect, an analogy is being used to exploit the similarity between two different properties: one in the real world and one in the model.

Models of a problem situation are only an approximate representation of that situation. The real world situation will have a complexity that tends to reduce your chances of achieving an exact representation. So, you need to find some way of achieving an acceptable balance between accuracy and manageability. As a project unfolds, there will be a number of practical considerations that result in some compromise. It is for this reason that several different models are built, each one representing different aspects (views) of the real world.

If a model is so complex that its author (or other team members) cannot use it, then that model is of little or no value. However, simplification is only of value if all simplifying assumptions, and their consequences, are made explicit. At some point in a project, any of these assumptions may need to be justified.

Models are subject to change. At the very least they require some form of testing so that a model can maintain its correspondence with reality. As towns and cities expand and contract, a road map must be changed to reflect the new situation. In the worst case, a change in scope necessitates a whole new model. For example, if there was a need to reflect the current status of roads and the traffic on them, a simple road map is inadequate since we would want to show, amongst other things, the changes in traffic density over time.

Unfortunately, not all models of a particular type share the same notation, often because they originate from different sources. For example, different publishers will have different ways of constructing and presenting road maps.

When developing a product, a variety of models are likely to be constructed. It is unrealistic to expect to put everything into just one model. Too much detail in a model can only be a distraction. It would be hard to use such a model as an aid to communication.

Each model is used to illustrate a different point of view. For example, there are two different kinds of views that modellers often distinguish:

static models, which describe a set of elements and any relationships that exist between them;

dynamic models, which describe the behaviour of one or more elements over time.

There is a useful discussion in Michael Jackson's Software Requirements and Specifications (1995, p. 120) where he defines what he means by a model by distinguishing between a model and reality in relation to developing software. To Jackson, a model refers to the machine (what the software does when it executes on a computer), which embodies a simulation of the real thing; it is a description of a domain (that part of reality that you are interested in). A model and the domain are different so he draws attention to the difference between a description that is true in both the machine and the domain, a description that is true only of the domain, and another description that is true only of the machine. It is important to be clear whether you are talking about reality, the machine, or both.

3. What is simulation?

Simulation is the process of modeling a real phenomenon through a set of mathematical formulas or algorithms. Simulation and modeling tools allow you to virtually model a process in detail without having to spend the time, resources, or capital to physically test that design in a real-world environment.

A simulation is the imitation of the operation of a real-world process or system over time.[1] Simulations require the use of models; the model represents the key characteristics or behaviors of the selected system or process, whereas the simulation represents the evolution of the model over time. Often, computers are used to execute the simulation.

Simulation is used in many contexts, such as simulation of technology for performance tuning or optimizing, safety engineering, testing, training, education, and video games. Simulation is also used with scientific modelling of natural systems or human systems to gain insight into their functioning,[2] as in economics. Simulation can be used to show the eventual real effects of alternative conditions and courses of action. Simulation is also used when the real system cannot be engaged, because it may not be accessible, or it may be dangerous or unacceptable to engage, or it is being designed but not yet built, or it may simply not exist.[3]

Key issues in modeling and simulation include the acquisition of valid sources of information about the relevant selection of key characteristics and behaviors used to build the model, the use of simplifying approximations and assumptions within the model, and fidelity and validity of the simulation outcomes. Procedures and protocols for model verification and validation are an ongoing field of academic study, refinement, research and development in simulations technology or practice, particularly in the work of computer simulation.

Classification and terminology

Human-in-the-loop simulation of outer space

Visualization of a direct numerical simulation model.

Historically, simulations used in different fields developed largely independently, but 20th-century studies of systems theory and cybernetics combined with spreading use of computers across all those fields have led to some unification and a more systematic view of the concept.

Physical simulation refers to simulation in which physical objects are substituted for the real thing (some circles[4] use the term for computer simulations modelling selected laws of physics, but this article does not). These physical objects are often chosen because they are smaller or cheaper than the actual object or system.

Interactive simulation is a special kind of physical simulation, often referred to as a human-in-the-loop simulation, in which physical simulations include human operators, such as in a flight simulator, sailing simulator, or driving simulator.

Continuous simulation is a simulation based on continuous-time rather than discrete-time steps, using numerical integration of differential equations.[5]

Discrete-event simulation studies systems whose states change their values only at discrete times.[6] For example, a simulation of an epidemic could change the number of infected people at time instants when susceptible individuals get infected or when infected individuals recover.

Stochastic simulation is a simulation where some variable or process is subject to random variations and is projected using Monte Carlo techniques using pseudo-random numbers. Thus replicated runs with the same boundary conditions will each produce different results within a specific confidence band.[5]

Deterministic simulation is a simulation which is not stochastic: thus the variables are regulated by deterministic algorithms. So replicated runs from the same boundary conditions always produce identical results.

Hybrid simulation (or combined simulation) corresponds to a mix between continuous and discrete event simulation and results in integrating numerically the differential equations between two sequential events to reduce the number of discontinuities.[7]

A stand-alone simulation is a simulation running on a single workstation by itself.

A distributed simulation is one which uses more than one computer simultaneously, to guarantee access from/to different resources (e.g. multi-users operating different systems, or distributed data sets); a classical example is Distributed Interactive Simulation (DIS).[8]

Parallel simulation speeds up a simulation's execution by concurrently distributing its workload over multiple processors, as in High-Performance Computing.[9]

Interoperable simulation is where multiple models, simulators (often defined as federates) interoperate locally, distributed over a network; a classical example is High-Level Architecture.[10][11]

Modeling and simulation as a service is where simulation is accessed as a service over the web.[12]

Modeling, interoperable simulation and serious games is where serious game approaches (e.g. game engines and engagement methods) are integrated with interoperable simulation.[13]

Simulation fidelity is used to describe the accuracy of a simulation and how closely it imitates the real-life counterpart. Fidelity is broadly classified as one of three categories: low, medium, and high. Specific descriptions of fidelity levels are subject to interpretation, but the following generalizations can be made:

Low – the minimum simulation required for a system to respond to accept inputs and provide outputs

Medium – responds automatically to stimuli, with limited accuracy

High – nearly indistinguishable or as close as possible to the real system

A synthetic environment is a computer simulation that can be included in human-in-the-loop simulations.[16]

Simulation in failure analysis refers to simulation in which we create environment/conditions to identify the cause of equipment failure. This can be the best and fastest method to identify the failure cause.

Computer simulation

Main article: Computer simulation

A computer simulation (or "sim") is an attempt to model a real-life or hypothetical situation on a computer so that it can be studied to see how the system works. By changing variables in the simulation, predictions may be made about the behaviour of the system. It is a tool to virtually investigate the behaviour of the system under study.[1]

Computer simulation has become a useful part of modeling many natural systems in physics, chemistry and biology,[17] and human systems in economics and social science (e.g., computational sociology) as well as in engineering to gain insight into the operation of those systems. A good example of the usefulness of using computers to simulate can be found in the field of network traffic simulation. In such simulations, the model behaviour will change each simulation according to the set of initial parameters assumed for the environment.

Traditionally, the formal modeling of systems has been via a mathematical model, which attempts to find analytical solutions enabling the prediction of the behaviour of the system from a set of parameters and initial conditions. Computer simulation is often used as an adjunct to, or substitution for, modeling systems for which simple closed form analytic solutions are not possible. There are many different types of computer simulation, the common feature they all share is the attempt to generate a sample of representative scenarios for a model in which a complete enumeration of all possible states would be prohibitive or impossible.

Several software packages exist for running computer-based simulation modeling (e.g. Monte Carlo simulation, stochastic modeling, multimethod modeling) that makes all the modeling almost effortless.

Modern usage of the term "computer simulation" may encompass virtually any computer-based representation.

Computer science

In computer science, simulation has some specialized meanings: Alan Turing used the term simulation to refer to what happens when a universal machine executes a state transition table (in modern terminology, a computer runs a program) that describes the state transitions, inputs and outputs of a subject discrete-state machine.[18] The computer simulates the subject machine. Accordingly, in theoretical computer science the term simulation is a relation between state transition systems, useful in the study of operational semantics.

Less theoretically, an interesting application of computer simulation is to simulate computers using computers. In computer architecture, a type of simulator, typically called an emulator, is often used to execute a program that has to run on some inconvenient type of computer (for example, a newly designed computer that has not yet been built or an obsolete computer that is no longer available), or in a tightly controlled testing environment (see Computer architecture simulator and Platform virtualization). For example, simulators have been used to debug a microprogram or sometimes commercial application programs, before the program is downloaded to the target machine. Since the operation of the computer is simulated, all of the information about the computer's operation is directly available to the programmer, and the speed and execution of the simulation can be varied at will.

Simulators may also be used to interpret fault trees, or test VLSI logic designs before they are constructed. Symbolic simulation uses variables to stand for unknown values.

In the field of optimization, simulations of physical processes are often used in conjunction with evolutionary computation to optimize control strategies.

4. Mention types of simulation

Data analytics professionals should know these four types of simulation models:

Monte Carlo method.

Agent-based modeling.

Discrete event simulation.

System dynamic modeling.

What are 5 different types of simulations?

Simulation - Types, Benefits, Limitations, Models, Phases

Scope of Simulation in different area is as follows :

Healthcare (Clinical) Simulators : ...

Computer Simulators : ...

Military Simulations : ...

Finance Simulations : ...

Flight Simulators : ...

Engineering, Technology or Process Simulation

5. Why do we use models?

Models can help you visualize, or picture in your mind, something that is difficult to see or understand. Models can help scientists communicate their ideas, understand processes, and make predictions.

Models are useful tools in learning science which can be used to improve explanations, generate discussion, make predictions, provide visual representations of abstract concepts and generate mental models

When you hear the word model, you probably think of a toy-like car or airplane that is a smaller version of the real thing. Scientific models are representations of objects, systems or events and are used as tools for understanding the natural world. Models use familiar objects to represent unfamiliar things.

Models can help you visualize, or picture in your mind, something that is difficult to see or understand. Models can help scientists communicate their ideas, understand processes, and make predictions. The chart below shows examples of what models can represent.

6. How is simulation performed?

A simulation imitates the operation of real world processes or systems with the use of models. The model represents the key behaviours and characteristics of the selected process or system while the simulation represents how the model evolves under different conditions over time.

Simulations are usually computer-based, using a software-generated model to provide support for the decisions of managers and engineers as well as for training purposes. Simulation techniques aid understanding and experimentation, as the models are both visual and interactive.

Simulation systems include discrete event simulation, process simulation and dynamic simulation. Businesses may use all of these systems across different levels of the organisation.

A simulation is a model that mimics the operation of an existing or proposed system, providing evidence for decision-making by being able to test different scenarios or process changes. This can be coupled with virtual reality technologies for a more immersive experience.

Simulations can be used to tune up performance, optimise a process, improve safety, testing theories, training staff and even for entertainment in video games! Scientifically modelling systems allows a user to gain an insight into the effects of different conditions and courses of action.

Simulation can also be used when the real system is inaccessible or too dangerous to assess or when a system is still in the design or theory stages.

Key to any simulation is the information that is used to build the simulation model and protocols for the verification and validation of models are still being researched and refined, particularly with regard to computer simulation.

How Simulation Works

Simulation works through the use of intuitive simulation software to create a visual mock-up of a process. This visual simulation should include details of timings, rules, resources and constraints, to accurately reflect the real-world process.

This can be applied to a range of scenarios, for example, you can model a supermarket and the likely behaviours of customers as they move around the shop as it becomes busier. This can inform decisions including staffing requirements, shop floor layout, and supply chain needs.

Another example would be a manufacturing environment where different parts of the line can be simulated to assess how their processes interact with those of others. This can provide an overview of how the entire system will perform in order to devise innovative methods to improve performance.

Advantages

There are a range of advantages to be gained through the use of simulation, including:

1. Less Financial Risk

Simulation is less expensive than real life experimentation. The potential costs of testing theories of real world systems can include those associated with changing to an untested process, hiring staff or even buying new equipment. Simulation allows you to test theories and avoid costly mistakes in real life.

2. Exact Repeated Testing

A simulation allows you to test different theories and innovations time after time against the exact same circumstances. This means you can thoroughly test and compare different ideas without deviation.

3. Examine Long-Term Impacts

A simulation can be created to let you see into the future by accurately modelling the impact of years of use in just a few seconds. This lets you see both short and long-term impacts so you can confidently make informed investment decisions now that can provide benefits years into the future.

4. Gain Insights for Process Improvement

The benefits of simulation are not only realised at the end of a project. Improvements can be integrated throughout an entire process by testing different theories.

5. Assess Random Events

A simulation can also be used to assess random events such as an unexpected staff absence or supply chain issues.

6. Test Non-Standard Distributions

A simulation can take account of changing and non-standard distributions, rather than having to repeat only set parameters. For example, when simulating a supermarket you can input different types of customer who will move through the shop at different speeds. A young businesswoman who is picking up a sandwich will move through the shop differently from an old couple or a mother doing a weekly shop with two children in tow. By taking such changing parameters into account, a simulation can more accurately mimic the real world.

7. Encourages In-Depth Thinking

Even the process of designing a simulation and determining the different parameters can offer solutions. By thinking in-depth about a process or procedure it is possible to come up with solutions or innovations without even using the final simulation.

8. Improve Stakeholder Buy-In

A visual simulation can also help improve buy-in from partners, associates and stakeholders. You can visually demonstrate the results of any process changes and how they were achieved, improving engagement with interested parties or even enabling a simulation based sales pitch.

Limitations

While there are a great many advantages to using simulation, there are still some limitations when compared to other similar techniques and technologies, such as digital twin.

A digital twin expands on simulation to incorporate real time feedback and a flow of information between the virtual simulation and a real life asset or assets. The difference being that while a simulation is theoretical, a digital twin is actual.

Due to this, simulations have limitations when it comes to assessing actual real-world situations as they occur.

Why is Simulation Used?

Simulation is used to evaluate the effect of process changes, new procedures and capital investment in equipment. Engineers can use simulation to assess the performance of an existing system or predict the performance of a planned system, comparing alternative solutions and designs.

Simulation is used as an alternative to testing theories and changes in the real world, which can be costly. Simulation can measure factors including system cycle times, throughput under different loads, resource utilisation, bottlenecks and choke points, storage needs, staffing requirements, effectiveness of scheduling and control systems.

What can be Simulated?

Any system or process that has a flow of events can be simulated. As a general rule, if you can draw a flowchart of the process, you can simulate it. However, simulation is most effective when applied to processes or equipment that change over time, have variable factors or random inputs. For example, our supermarket from earlier has variable and random factors due to customer use times, requirements and stocks.

Using simulation to model complex and changeable dynamic systems can offer insights that are difficult to gain using other methods.

While simulation can be used to manage processes, procedures and assets, Swedish philosopher Nick Bostrom took the notion of simulation further in his 2003 paper, 'Are You Living in a Computer Simulation?' He argues that by adding artificial consciousness to simulations, you can blur the lines between reality and simulation, making it difficult to tell if you are living in reality or if you are living in a simulation. This simulation hypothesis argues that, should you become aware that your 'reality' was not actually 'real,' your memories could be edited by the simulation to once again make you blissfully unaware that you are not actually a real person in the real world!

Moving away from the realms of post-human simulation, let's return to some 'real world' types of simulation...

Types of Simulation

Simulation can be broken down into three overarching types, as follows:

1. Discrete Event Simulation

Modelling a system as it progresses through time, for example;

factory operations (stamping, turning, milling)

traffic analysis (roads, networks, queues)

2. Dynamic Simulation

Modelling a system as it progresses through space, for example;

machine kinematics

human ergonomics

aerodynamic testing

virtual prototyping

3. Process Simulation

Modelling physical interactions between two or more systems, for example;

in-service product modelling

in-manufacture product modelling

weather forecasting

Examples

There are many examples of simulation across industry, entertainment, education, and more. Here are a few notable examples:

Automotive

Simulation allows the characteristics of a real vehicle to be replicated in a virtual environment, so that the driver feels as if they are sitting in a real car. Different scenarios can be mimicked so that the driver has a fully immersive experience. These type of simulators can help train both new and experienced drivers, offering a route to teach driving skills that can reduce maintenance and fuel costs and ensure the safety of the drivers themselves.

Biomechanics

Simulation can be applied to biomechanics to create models of human or animal anatomical structures in order to study their function and design medical treatments and devices. Biomechanics simulation can also be used to study sports performance, simulate surgical procedures, and assess joint loads. An additional example is neuromechanical simulation that unites neural network simulation with biomechanics to test hypotheses in a virtual environment.

City and Urban Planning

Simulation can be used to design new cities and urban environments as well as to test how existing urban areas can evolve as a result of policy decisions. This includes city infrastructure and traffic flow among other potential models.

Digital Lifecycle Design

Simulations can assist with product design, allowing digital prototyping and testing to create better performing products with a shorter time-to-market, while also assessing the lifecycle of the finished product.

Disaster Preparation

Simulations can replicate emergency situations, to help with disaster preparedness.This includes training and designing responses to events such as natural disasters, pandemics or terrorist attacks. Responses can be tracked and assessed through the simulation, highlighting potential problems and areas where more training may be required for responders, as well as ensuring any mistakes are made in a safe environment ahead of any real life event.

Economics and Finance

Economics, macroeconomics and finance also benefit from simulations. A mathematical model of the economy can, for example, be tested using historical data as a proxy for the actual economy. This can be used to assess inflation, unemployment, balance of trade and budgets.Elsewhere, simulations can replicate the stock exchange or be used to test financial models. Banks also use simulations to replicate payment and securities settlement systems.

Engineering Systems

Simulation is widely used for engineering systems to imitate operations and functions of equipment, processes and procedures. Engineering simulations can combine mathematical models and computer-assisted simulation for design or improvement of existing processes.

Ergonomics

Simulation can be used to analyse virtual products and working environments incorporating an anthropometric virtual representation of the human, also known as a mannequin or Digital Human Model (DHM). These DHMs can mimic the performance and capabilities of humans in simulated environments. This type of simulation has applications ranging from assembly lines to disaster management and video gaming to waste collection.

Flight Simulation

Flight simulators have been used for years to train new pilots in a safe environment. This not only allows pilots to be assessed safely, but can also test instrument failures and other problems without risking the pilot, the instructor or the aircraft. You can also easily repeat the exact same scenarios, such as approaching a runway to land, under different conditions, not to mention saving fuel and other costs compared to actual flying time.

Marine Craft Simulation

Much like flight simulation, it is also possible to simulate working in a ship or submarine. Simulators can include those that mimic the bridge, engine rooms, cargo handling bays, communications or remotely operated vehicles. These are used in training institutions, colleges and navies.

Military Applications

Sometimes referred to as 'war games,' military simulations can be used to test out military plans in a virtual environment using computer models. These can also incorporate social and political factors and are used by governments and military organisations around the world.

Network Systems

Simulations have been applied to network and distributed systems to test new algorithms and protocols before they are implemented in live systems. These can be applied to applications including content delivery networks, smart cities and the Internet of Things.

Project Management

Simulation can be used for project management analysis and training purposes. Whether training managers or analysing the outcomes of different decisions, simulation is frequently conducted with software tools.

Robotics

Robotics simulations are used to mimic situations that may not be possible to recreate and test in real life due to time, cost or other factors. The results of these tests can then be assessed and transferred to real life robots.

Production Systems

Production systems can be simulated using methods such as discrete event simulation to assess manufacturing processes, assembly times, machine set-up, and more.

Sales

Sales can be simulated to examine the flow of transactions and customer orders as well as costs, labour times and more.

Satellites and Space

The Kennedy Space Centre used simulation to train space shuttle engineers for launch operations. This would see people interact with a simulated shuttle and ground support equipment. Simulation is also used for satellite navigation tests.

Sport

Statistics are widely used as part of sport simulation to predict the outcome of events and the performance of individual sportspeople. Sports simulation can also be used to predict the outcome of games and events as well as for fantasy sports leagues. Biomechanics models can also be used to assist training, assess fatigue levels and their effect on performance and more.

Weather

Weather forecasting uses simulations based on past data to predict extreme weather conditions such as hurricanes or cyclones.

Conclusion

Simulations are used for a range of applications across industry, saving time and expense while being able to test theories and ideas before implementing them in the real world. Although related techniques such as digital twin may provide added benefits due to the two-way flow of information this allows, simulations still have a great many uses.

Whether testing theories, assessing procedural performance or determining the lifecycle of an asset simulation is a useful tool for many businesses and organisations.

7. What is optimization?

optimization, also known as mathematical programming, collection of mathematical principles and methods used for solving quantitative problems in many disciplines, including physics, biology, engineering, economics, and business. The subject grew from a realization that quantitative problems in manifestly different disciplines have important mathematical elements in common. Because of this commonality, many problems can be formulated and solved by using the unified set of ideas and methods that make up the field of optimization.

The historical term mathematical programming, broadly synonymous with optimization, was coined in the 1940s before programming became equated with computer programming. Mathematical programming includes the study of the mathematical structure of optimization problems, the invention of methods for solving these problems, the study of the mathematical properties of these methods, and the implementation of these methods on computers. Faster computers have greatly expanded the size and complexity of optimization problems that can be solved. The development of optimization techniques has paralleled advances not only in computer science but also in operations research, numerical analysis, game theory, mathematical economics, control theory, and combinatorics.

Optimization problems typically have three fundamental elements. The first is a single numerical quantity, or objective function, that is to be maximized or minimized. The objective may be the expected return on a stock portfolio, a company's production costs or profits, the time of arrival of a vehicle at a specified destination, or the vote share of a political candidate. The second element is a collection of variables, which are quantities whose values can be manipulated in order to optimize the objective. Examples include the quantities of stock to be bought or sold, the amounts of various resources to be allocated to different production activities, the route to be followed by a vehicle through a traffic network, or the policies to be advocated by a candidate. The third element of an optimization problem is a set of constraints, which are restrictions on the values that the variables can take. For instance, a manufacturing process cannot require more resources than are available, nor can it employ less than zero resources. Within this broad framework, optimization problems can have different mathematical properties. Problems in which the variables are continuous quantities (as in the resource allocation example) require a different approach from problems in which the variables are discrete or combinatorial quantities (as in the selection of a vehicle route from among a predefined set of possibilities).

An important class of optimization is known as linear programming. Linear indicates that no variables are raised to higher powers, such as squares. For this class, the problems involve minimizing (or maximizing) a linear objective function whose variables are real numbers that are constrained to satisfy a system of linear equalities and inequalities. Another important class of optimization is known as nonlinear programming. In nonlinear programming the variables are real numbers, and the objective or some of the constraints are nonlinear functions (possibly involving squares, square roots, trigonometric functions, or products of the variables). Both linear and nonlinear programming are discussed in this article. Other important classes of optimization problems not covered in this article include stochastic programming, in which the objective function or the constraints depend on random variables, so that the optimum is found in some "expected," or probabilistic, sense; network optimization, which involves optimization of some property of a flow through a network, such as the maximization of the amount of material that can be transported between two given locations in the network; and combinatorial optimization, in which the solution must be found among a finite but very large set of possible values, such as the many possible ways to assign 20 manufacturing plants to 20 locations.

Linear programming

Origins and influences

Although widely used now to solve everyday decision problems, linear programming was comparatively unknown before 1947. No work of any significance was carried out before this date, even though the French mathematician Joseph Fourier seemed to be aware of the subject's potential as early as 1823. In 1939 a Russian mathematician, Leonid Vitalyevich Kantorovich, published an extensive monograph, Matematicheskie metody organizatsi i planirovaniya proizvodstva ("Mathematical Methods for Organization and Planning of Production"), which is now credited with being the first treatise to recognize that certain important broad classes of scheduling problems had well-defined mathematical structures. Unfortunately, Kantorovich's proposals remained mostly unknown both in the Soviet Union and elsewhere for nearly two decades. Meanwhile, linear programming had developed considerably in the United States and Western Europe. In the period following World War II, officials in the United States government came to believe that efficient coordination of the energies and resources of a whole nation in the event of nuclear war would require the use of scientific planning techniques. The advent of the computer made such an approach feasible.

Intensive work began in 1947 in the U.S. Air Force. The linear programming model was proposed because it was relatively simple from a mathematical viewpoint, and yet it provided a sufficiently general and practical framework for representing interdependent activities that share scarce resources. In the linear programming model, the modeler views the system to be optimized as being made up of various activities that are assumed to require a flow of inputs (e.g., labour and raw materials) and outputs (e.g., finished goods and services) of various types proportional to the level of the activity. Activity levels are assumed to be representable by nonnegative numbers. The revolutionary feature of the approach lies in expressing the goal of the decision process in terms of minimizing or maximizing a linear objective function—for example, maximizing possible sorties in the case of the air force, or maximizing profits in industry. Before 1947 all practical planning was characterized by a series of authoritatively imposed rules of procedure and priorities. General objectives were never stated, probably because of the impossibility of performing the calculations necessary to minimize an objective function under constraints. In 1947 a method (described in the section The simplex method) was introduced that turned out to solve practical problems efficiently. Interest in linear programming grew rapidly, and by 1951 its use spread to industry. Today it is almost impossible to name an industry that is not using mathematical programming in some form, although the applications and the extent to which it is used vary greatly, even within the same industry.

Interest in linear programming has also extended to economics. In 1937 the Hungarian-born mathematician John von Neumann analyzed a steadily expanding economy based on alternative methods of production and fixed technological coefficients. As far as mathematical history is concerned, the study of linear inequality systems excited virtually no interest before 1936. In 1911 a vertex-to-vertex movement along edges of a polyhedron (as is done in the simplex method) was suggested as a way to solve a problem that involved optimization, and in 1941 movement along edges was proposed for a problem involving transportation. Credit for laying much of the mathematical foundations should probably go to von Neumann. In 1928 he published his famous paper on game theory, and his work culminated in 1944 with the publication, in collaboration with the Austrian economist Oskar Morgenstern, of the classic Theory of Games and Economic Behaviour. In 1947 von Neumann conjectured the equivalence of linear programs and matrix games, introduced the important concept of duality, and made several proposals for the numerical solution of linear programming and game problems. Serious interest by other mathematicians began in 1948 with the rigorous development of duality and related matters.

The general simplex method was first programmed in 1951 for the United States Bureau of Standards SEAC computer. Starting in 1952, the simplex method was programmed for use on various IBM computers and later for those of other companies. As a result, commercial applications of linear programs in industry and government grew rapidly. New computational techniques and variations of older techniques continued to be developed.

More recently there has been much interest in solving large linear problems with special structures—for example, corporate models and national planning models that are multistaged, are dynamic, and exhibit a hierarchical structure. It is estimated that certain developing countries will have the potential of increasing their gross national product (GNP) by 10 to 15 percent per year if detailed growth models of the economy can be constructed, optimized, and implemented.

Theory

Basic ideas

A simple problem in linear programming is one in which it is necessary to find the maximum (or minimum) value of a simple function subject to certain constraints. An example might be that of a factory producing two commodities. In any production run, the factory produces x1 of the first type and x2 of the second. If the profit on the second type is twice that on the first, then x1 + 2x2 represents the total profit. The function x1 + 2x2 is known as the objective function.

Clearly the profit will be highest if the factory devotes its entire production capacity to making the second type of commodity. In a practical situation, however, this may not be possible; a set of constraints is introduced by such factors as availability of machine time, labour, and raw materials. For example, if the second type of commodity requires a raw material that is limited so that no more than five can be made in any batch, then x2 must be less than or equal to five; i.e., x2 ? 5. If the first commodity requires another type of material limiting it to eight per batch, then x1 ? 8. If x1 and x2 take equal time to make and the machine time available allows a maximum of 10 to be made in a batch, then x1 + x2 must be less than or equal to 10; i.e., x1 + x2 ? 10.

optimization problem

optimization problem

Two other constraints are that x1 and x2 must each be greater than or equal to zero, because it is impossible to make a negative number of either; i.e., x1 ? 0 and x2 ? 0. The problem is to find the values of x1 and x2 for which the profit is a maximum. Any solution can be denoted by a pair of numbers (x1, x2); for example, if x1 = 3 and x2 = 6, the solution is (3, 6). These numbers can be represented by points plotted on two axes, as shown in the figure. On this graph the distance along the horizontal axis represents x1 and that along the vertical represents x2. Because of the constraints given above, the feasible solutions must lie within a certain well-defined region of the graph. For example, the constraint x1 ? 0 means that points representing feasible solutions lie on or to the right of the x2 axis. Similarly, the constraint x2 ? 0 means that they also lie on or above the x1 axis. Application of the entire set of constraints gives the feasible solution set, which is bounded by a polygon formed by the intersection of the lines x1 = 0, x2 = 0, x1 = 8, x2 = 5, and x1 + x2 = 10. For example, production of three items of commodity x1 and four of x2 is a feasible solution since the point (3, 4) lies in this region. To find the best solution, however, the objective function x1 + 2x2 = k is plotted on the graph for some value of k, say k = 4. This value is indicated by the broken line in the figure. As k is increased, a family of parallel lines are produced and the line for k = 15 just touches the constraint set at the point (5, 5). If k is increased further, the values of x1 and x2 will lie outside the set of feasible solutions. Thus, the best solution is that in which equal quantities of each commodity are made. It is no coincidence that an optimal solution occurs at a vertex, or “extreme point,” of the region. This will always be true for linear problems, although an optimal solution may not be unique. Thus, the solution of such problems reduces to finding which extreme point (or points) yields the largest value for the objective function.

The simplex method

The graphical method of solution illustrated by the example in the preceding section is useful only for systems of inequalities involving two variables. In practice, problems often involve hundreds of equations with thousands of variables, which can result in an astronomical number of extreme points. In 1947 George Dantzig, a mathematical adviser for the U.S. Air Force, devised the simplex method to restrict the number of extreme points that have to be examined. The simplex method is one of the most useful and efficient algorithms ever invented, and it is still the standard method employed on computers to solve optimization problems. First, the method assumes that an extreme point is known. (If no extreme point is given, a variant of the simplex method, called Phase I, is used to find one or to determine that there are no feasible solutions.) Next, using an algebraic specification of the problem, a test determines whether that extreme point is optimal. If the test for optimality is not passed, an adjacent extreme point is sought along an edge in the direction for which the value of the objective function increases at the fastest rate. Sometimes one can move along an edge and make the objective function value increase without bound. If this occurs, the procedure terminates with a prescription of the edge along which the objective goes to positive infinity. If not, a new extreme point is reached having at least as high an objective function value as its predecessor. The sequence described is then repeated. Termination occurs when an optimal extreme point is found or the unbounded case occurs. Although in principle the necessary steps may grow exponentially with the number of extreme points, in practice the method typically converges on the optimal solution in a number of steps that is only a small multiple of the number of extreme points.

To illustrate the simplex method, the example from the preceding section will be solved again. The problem is first put into canonical form by converting the linear inequalities into equalities by introducing “slack variables” x3 ? 0 (so that x1 + x3 = 8), x4 ? 0 (so that x2 + x4 = 5), x5 ? 0 (so that x1 + x2 + x5 = 10), and the variable x0 for the value of the objective function (so that x1 + 2x2 - x0 = 0). The problem may then be restated as that of finding nonnegative quantities x1, ..., x5 and the largest possible x0 satisfying the resulting equations. One obvious solution is to set the objective variables x1 = x2 = 0, which corresponds to the extreme point at the origin. If one of the objective variables is increased from zero while the other one is fixed at zero, the objective value x0 will increase as desired (subject to the slack variables satisfying the equality constraints). The variable x2 produces the largest increase of x0 per unit change; so it is used first. Its increase is limited by the nonnegativity requirement on the variables. In particular, if x2 is increased beyond 5, x4 becomes negative.

At x2 = 5, this situation produces a new solution—(x0, x1, x2, x3, x4, x5) = (10, 0, 5, 8, 0, 5)—that corresponds to the extreme point (0, 5) in the figure. The system of equations is put into an equivalent form by solving for the nonzero variables x0, x2, x3, x5 in terms of those variables now at zero; i.e., x1 and x4. Thus, the new objective function is x1 - 2x4 = -10, while the constraints are x1 + x3 = 8, x2 + x4 = 5, and x1 - x4 + x5 = 5. It is now apparent that an increase of x1 while holding x4 equal to zero will produce a further increase in x0. The nonnegativity restriction on x3 prevents x1 from going beyond 5. The new solution—(x0, x1, x2, x3, x4, x5) = (15, 5, 5, 3, 0, 0)—corresponds to the extreme point (5, 5) in the figure. Finally, since solving for x0 in terms of the variables x4 and x5 (which are currently at zero value) yields x0 = 15 - x4 - x5, it can be seen that any further change in these slack variables will decrease the objective value. Hence, an optimal solution exists at the extreme point (5, 5).

Standard formulation

In practice, optimization problems are formulated in terms of matrices—a compact symbolism for manipulating the constraints and testing the objective function algebraically. The original (or “primal”) optimization problem was given its standard formulation by von Neumann in 1947. In the primal problem the objective is replaced by the product (px) of a vector x = (x1, x2, x3, ..., xn)T, whose components are the objective variables and where the superscript “transpose” symbol indicates that the vector should be written vertically, and another vector p = (p1, p2, p3, ..., pn), whose components are the coefficients of each of the objective variables. In addition, the system of inequality constraints is replaced by Ax ? b, where the m by n matrix A replaces the m constraints on the n objective variables, and b = (b1, b2, b3, ..., bm)T is a vector whose components are the inequality bounds.

Nonlinear programming

Origins

Although the linear programming model works fine for many situations, some problems cannot be modeled accurately without including nonlinear components. One example would be the isoperimetric problem: determine the shape of the closed plane curve having a given length and enclosing the maximum area. The solution, but not a proof, was known by Pappus of Alexandria c. 340 CE:

Bees, then, know just this fact which is useful to them, that the hexagon is greater than the square and the triangle and will hold more honey for the same expenditure of material in constructing each. But we, claiming a greater share of wisdom than the bees, will investigate a somewhat wider problem, namely that, of all equilateral and equiangular plane figures having the same perimeter, that which has the greater number of angles is always greater, and the greatest of them all is the circle having its perimeter equal to them.

The branch of mathematics known as the calculus of variations began with efforts to prove this solution, together with the challenge in 1696 by the Swiss mathematician Johann Bernoulli to find the curve that minimizes the time it takes an object to slide, under only the force of gravity, between two nonvertical points. (The solution is the brachistochrone.) In addition to Johann Bernoulli, his brother Jakob Bernoulli, the German Gottfried Wilhelm Leibniz, and the Englishman Isaac Newton all supplied correct solutions. In particular, Newton's approach to the solution plays a fundamental role in many nonlinear algorithms. Other influences on the development of nonlinear programming, such as convex analysis, duality theory, and control theory, developed largely after 1940. For problems that include constraints as well as an objective function, the optimality conditions discovered by the American mathematician William Karush and others in the late 1940s became an essential tool for recognizing solutions and for driving the behaviour of algorithms.

An important early algorithm for solving nonlinear programs was given by the Nobel Prize-winning Norwegian economist Ragnar Frisch in the mid-1950s. Curiously, his approach fell out of favour for some decades, reemerging as a viable and competitive approach only in the 1990s. Other important algorithmic approaches include sequential quadratic programming, in which an approximate problem with a quadratic objective and linear constraints is solved to obtain each search step; and penalty methods, including the "method of multipliers," in which points that do not satisfy the constraints incur penalty terms in the objective to discourage algorithms from visiting them.

More From Britannica

systems engineering: Modeling and optimization

The Nobel Prize-winning American economist Harry M. Markowitz provided a boost for nonlinear optimization in 1958 when he formulated the problem of finding an efficient investment portfolio as a nonlinear optimization problem with a quadratic objective function. Nonlinear optimization techniques are now widely used in finance, economics, manufacturing, control, weather modeling, and all branches of engineering.

Theory

An optimization problem is nonlinear if the objective function f(x) or any of the inequality constraints ci(x) ? 0, i = 1, 2, ..., m, or equality constraints dj(x) = 0, j = 1, 2, ..., n, are nonlinear functions of the vector of variables x. For example, if x contains the components x1 and x2, then the function 3 + 2x1 - 7x2 is linear, whereas the functions (x1)3 + 2x2 and 3x1 + 2x1x2 + x2 are nonlinear.

8. What is independent replication?

Independent replication. An experiment that addresses the same questions and/or hypotheses as a previous study, but is conducted without knowledge of, or deference to, that prior study either because the researchers are unaware of the prior work, or because they want to avoid bias.

Replicating the experiment by independent researchers. Repeating the whole experiment by researchers that were not part of the initial experiment. This occurs when a paper is published and others try to obtain the same results.

Independent replication typically can boost more confidence than the one that is not independent. This is because bias is more likely to occur in a replication that is not independent.

9. What is batch means?

In the batch mean method, only one simulation run is executed. After deleting the warm up period, the remainder of the run is divided into k batches, with each batch average representing a single observation as illustrated in Figure 5.24. In fact, this is essentially the same concept as previously discussed when processing the data using the Welch Plot Analyzer.

Illustration of the batch means concept

Figure 5.24: Illustration of the batch means concept

The advantages of the batch means method are that it entails a long simulation run, thus dampening the effect of the initial conditions. The disadvantage is that the within replication data are correlated and unless properly formed the batches may also exhibit a strong degree of correlation.

The following presentation assumes that a warm up analysis has already been performed and that the data that has been collected occurs after the warm up period. For simplicity, the presentation assumes observation based data. The discussion also applies to time-based data that has been cut into discrete equally spaced intervals of time as described in Section 5.3.1. Therefore, assume that a series of observations,

$(X_1, X_2, X_3, \ldots, X_n)$

, is available from within the one long replication after the warm up period. As shown earlier at the beginning of Section 5.1, the within replication data can be highly correlated. In that section, it was mentioned that standard confidence intervals based on the regular formula for the sample variance

$S^2($

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

are not appropriate for this type of data. Suppose you were to ignore the correlation, what would be the harm? In essence, a confidence interval implies a certain level of confidence in the decisions based on the confidence interval. When you use $S^2(n)$ as defined above, you will not achieve the desired level of confidence because

$S^2(n)$ is a biased estimator for the variance of $\bar{X}$ when the data are correlated. Under the assumption that the data are covariance stationary, an assessment of the harm in ignoring the correlation can be made. For a series that is covariance stationary, one can show that

$$Var(\bar{X}) = \frac{?_0}{n}[1 + 2 \, n-1 \, ?_k$$

=
1
(
1
-
k
n
)
?
k
]

where
?
0
=
V
a
r
(
X
i
)
,
?
k
=
C
o

$v(X_i, X_{i+k})$, and $?_k = ?_k / ?_0$ for $k = 1, 2, \ldots, n-1$

.

When the data are correlated, S^2/n is a biased estimator of $Var(\bar{X})$. To show this, you need to compute the expected value of S^2/n as follows:

$E[S^2/n] = ?$

0

n

[

1

-

2

R

n

-

1

]

where

R

=

n

-

1

?

k

=

1

(

1

-

k

n

)

?

k

Bias is defined as the difference between the expected value of the estimator and the quantity being estimated. In this case, the bias can be computed with some algebra as:

Bias

=

E

[

S

2

/

n

]

-

V

a

r

(

—

Y

)

=

-

2

?

0

R

n

-

1

Since

?

0

>

0

and

n

>

1

the sign of the bias depends on the quantity R and thus on the correlation. There are three cases to consider: zero correlation, negative correlation, and positive correlation. Since

-

1

?

?

k

?

1

, examining the limiting values for the correlation will determine the range of the bias.

For positive correlation,

0

?

?

k

?

1

, the bias will be negative, (

-

?

0

?

B

i

a

s

?

0

). Thus, the bias is negative if the correlation is positive, and the bias is positive if the correlation is negative. In the case of positive correlation,

S

2

/

n

underestimates the

V

a

r

(

—

X

)

. Thus, using

S

2

/

n

to form confidence intervals will make the confidence intervals too short. You will have unjustified confidence in the point estimate in this case. The true confidence will not be the desired

1

-

?

. Decisions based on positively correlated data will have a higher than planned risk of making an error based on the confidence interval.

One can easily show that for negative correlation,

-

1

?

?

k

?

0

, the bias will be positive (

0

?

B

i

a

s

?

?

0

). In the case of negatively correlated data,

S

2

/

n

over estimates the $Var(\overline{X})$. A confidence interval based on S^2/n will be too wide and the true quality of the estimate will be better than indicated. The true confidence coefficient will not be the desired $1 - ?$; it will be greater than $1 - ?$.

Of the two cases, the positively correlated case is the more severe in terms of its effect on the decision making process; however, both are problems. Thus, the naive use of S^2/n for dependent data is highly unwarranted. If you want to build confidence intervals on

—

X you need to find an unbiased estimator of the $Var(\bar{X})$.

The method of batch means provides a way to develop (at least approximately) an unbiased estimator for $Var(\bar{X})$. Assuming that you have a series of data point, the method of batch means method divides the data into subsequences of contiguous batches:

$X_1, X_2, \ldots, X$

b
??????????????????

b
a
t
c
h
1

?
X
b
+
1
,
X
b
+
2
,
...
,
X
2
b
??????????????????????????

b
a
t

c

h

2

?

X

(

j

-

1

)

b

+

1

,

X

(

j

-

1

)

b

+

2

,

…

,

X

j

b

?????????????????????????????????????

b

a

t

c

h

j

?

X

(

k

-

1

)

b

+

1

,

X

(

k

-

1

)

b

+

2

,

...

,

X

k

b

????????????????????????????????????

b

a

t

c

h

k

and computes the sample average of the batches. Let

k

be the number of batches each consisting of

b

observations, so that

k

=

?

n

/

b

?

. If

b

is not a divisor of

n

then the last

$(n-kb)$ data points will not be used. Define $\bar{X}_j(b)$ as the j^{th} batch mean for $j=1,2,\ldots,k$, where,

$\bar{X}$

$$ {}_j(b) = \frac{1}{b} \sum_{i=1}^{b} X_{(j-1)b+i} $$

Each of the batch means are treated like observations in the batch means series. For example, if the batch means are re-labeled as

$$ Y_j = \bar{X} $$

$j(b)$, the batching process simply produces another series of data, $(Y_1, Y_2, Y_3, \ldots, Y_k)$ which may be more like a random sample. To form a $1 - ?$ % confidence interval, you simply treat this new series like a random sample and compute approximate confidence intervals using the sample average and sample variance of the batch means series:

$$\bar{Y}(k) =$$

1
k
k
?
j
=
1

Y
j

S
2
b
(
k
)
=
1
k
-
1
k
?
j
=
1

(
Y

j

-

—

Y

)

2

—

Y

(

k

)

±

t

?

/

2

,

k

-

1

S

b

(

k

)

?

k

Since the original X's are covariance stationary, it follows that the resulting batch means are also covariance stationary. One can show, see (Alexopoulos and Seila 1998), that the correlation in the batch means reduces as both the size of the batches, b and the number of data points, n increases. In addition, one can show that $S_b^2(k)/k$ approximates $\text{Var}(\bar{X})$ with error that reduces as both b and n increase towards infinity.

The basic difficulty with the batch means method is determining the batch size or alternatively the number of batches. Larger batch sizes are good for independence but reduce the number of batches, resulting in higher variance for the estimator. (Schmeiser 1982) performed an analysis that suggests that there is little benefit if the number of batches is larger than 30 and recommended that the number of batches remain in the range from 10 to 30. However, when trying to access whether or not the batches are independent it is better to have a large number of batches (

>

100) so that tests on the lag-k correlation have better statistical properties.

There are a variety of procedures that have been developed that will automatically batch the data as it is collected, see for example (Fishman and Yarberry 1997), (Steiger and Wilson 2002), and Banks et al. (2005). has its own batching algorithm. The batching algorithm is described in Kelton, Sadowski, and Sturrock (2004) page 311. See also (Fishman 2001) page 254 for an analysis of the effectiveness of the algorithm.

The discussion here is based on the description in Kelton, Sadowski, and Sturrock (2004). When the algorithm has recorded a sufficient amount of data, it begins by forming k = 20 batches. As more data is collected, additional batches are formed until k = 40 batches are collected. When 40 batches are formed, the algorithm collapses the number of batches back to 20, by averaging each pair of batches. This has the net effect of doubling the batch size. This process is repeated as more data is collected, thereby ensuring that the number of batches is between 20 and 39. The algorithm begins the formation of batches when it has at least 320 observations of tally-based data.

For time-persistent data, The algorithm requires that there were at least 5 time units during which the time-based variable changed 320 times. If there are not enough observations within a run then Insufficient is reported for the half-width value on the output reports. In addition, the algorithm also tests to see if the lag-1 correlation is significant by testing the hypothesis that the batch means are uncorrelated using the following test statistic, see (Alexopoulos and Seila 1998):

C

=

?

k

2

-

1

k

-

2

[

^

?

1

+

[

Y

1

-

—

Y

]

2

+

[

Y

k

-

—

Y

]

2

2

?

k

j

=

1

(

Y

j

-

—

Y
)
2
]

^
?
1
=
?
k
-
1
j
=
1
(
Y
j
-
—
Y
)
(
Y
j
+
1
-
—

Y

)

?

k

j

=

1

(

Y

j

-

—

Y

)

2

The hypothesis is rejected if

C

>

z

?

for a given confidence level

?

. If the batch means do not pass the test, Correlated is reported for the half-width on the statistical reports.

5.4.1 Performing the Method of Batch Means

Performing the method of batch means in is relatively straight forward. The following assumes that a warm up period analysis has already been performed. Since batches are formed during the simulation run and the confidence intervals are based on the batches, the primary concern will be to determine the run length that will ensure a desired half-width on the confidence intervals. A fixed sampling based method and a sequential sampling method will be illustrated.

The analysis performed to determine the warm up period should give you some information concerning how long to make this single run and how long to set it’s warm up period. Assume that a warm up analysis has been performed using n_0 replications of length T_e and that the analysis has indicated a warm up period of length T_w.

As previously discussed, the method of replication deletion spreads the risk of choosing a bad initial condition across multiple replications. The method of batch means relies on only one replication. If you were satisfied with the warm up period analysis based on n_0 replications and you were going to perform replication deletion, then you are willing to throw away the observations contained in at least $n_0 \times T_w$ time units and you are willing to use the data collected over $n_0 \times (T_e$

$- T_w)$ time units. Therefore, the warm up period for the single replication can be set at $n_0 \times T_w$ and the run length can be set at $n_0 \times T_e$.

For example, suppose your warm up analysis was based on the initial results shown in Section 5.1, i.e. $n_0 = 10$, $T_e = 30000$, $T_w = 3000$. Thus, your starting run length would be n_0

$\times T_e = 10 \times 30,000 = 300,000$ and the warm period will be $n_0 \times T_w = 30,000$. For these setting, the results shown in Figure 5.25 are very close to the results for the replication-deletion example.

Initial batch means results.

Figure 5.25: Initial batch means results.

Suppose now you want to ensure that the half-widths from a single replication are less than a given error bound. The half-widths reported by the simulation for a single replication are based on the batch means. You can get an approximate idea of how much to increase the length of the replication by using the functions: TNUMBAT(Tally ID) and TBATSIZ(Tally ID) for observation based statistics or DNUMBAT(DSTAT ID) and DBATSIZ(DSTAT ID) in conjunction with the half-width sample size determination formula.

$$n ? n_0 \left(\frac{h_0}{h}\right)^2$$

In this case, you interpret n and n_0 as the number of batches. OUTPUT statistics can be added to the model to observe the number of batches for the waiting time in queue and for the size of each batch, as shown in Figure 5.26. The resulting values for the number of batches formed for the waiting times and the size of the batches are given in Figure 5.27 Using this information in the half-width based sample size formula with $n_0 = 32$, h_0

$= 0.06$, and $h = 0.02$, yields:

$$n ? n_0 \frac{h_0^2}{h^2} = 32 \times \frac{(0.06)^2}{(0.02)^2} = 288$$

batches

OUTPUT Statistics to get number of batches and batch size.

Figure 5.26: OUTPUT Statistics to get number of batches and batch size.

Results for number of batches and batch size.

Figure 5.27: Results for number of batches and batch size.

Since each batch in the run had 8192 observations, this yields the need for additional observations for the waiting time in the queue. Since, in this model, customers arrive at a mean rate of 1 per minute, this requires about 2,359,296 additional time units of simulation. Because of the warm up period, you therefore need to set

T

e

equal to (2,359,296 + 30,000 = 2389296). Re-running the simulation yields the results shown in Figure 5.28. The results show that the half-width meets the desired criteria. This approach is approximate since you do not know how the observations will be batched when making the final run.

Batch means results for fixed sample size.

Figure 5.28: Batch means results for fixed sample size.

Rather than trying to fix the amount of sampling, you might instead try to use a sequential sampling technique that is based on the half-width computed during the simulation run. This is easy to do by supplying the appropriate expression within the Terminating Condition field on the Run Setup

>

Replication Parameters dialog.

Figure 5.29 illustrates that you can use a Boolean expression within the Terminating Condition field. In this case, the THALF(Tally ID) function is used to specify that the simulation should terminate when the half-width criteria is met. The batching algorithm computes the value of THALF(Tally ID) after sufficient data has been observed. This expression can be expanded to include other performance measures in a compound Boolean statement.

Sequential sampling using terminating condition.

Figure 5.29: Sequential sampling using terminating condition.

The results of running the simulation based on the sequential method are given in Figure 5.30. In this case, the simulation run ended at approximately time 1,928,385. This is lower than the time specified for the fixed sampling procedure (but the difference is not excessive).

Results for infinite horizon sequential sampling method.

Figure 5.30: Results for infinite horizon sequential sampling method.

Once the warm up period has been analyzed, performing infinite horizon simulations using the batch means method is relatively straight forward. A disadvantage of this method is that it will be more difficult to use the statistical methods available within the Process Analyzer or within OptQuest because they assume a replication-deletion approach.

If you are faced with an infinite horizon simulation, then you can use either the replication-deletion approach or the batch means method. In either case, you should investigate if there may be any problems related to initialization bias. If you use the replication-deletion approach, you should play it safe when specifying the warm up period. Making the warm up period longer than you think it should be is better than replicating a poor choice. When performing an infinite horizon simulation based on one long run, you should make sure that your run length is long enough. A long run length can help to “wash out” the effects of initial condition bias.

Ideally, in the situation where you have to make many simulation experiments using different parameter settings of the same model, you should perform a warm up analysis for each design configuration. In practice, this is not readily feasible when there are a large number of experiments. In this situation, you should use your common sense to pick the design configurations (and performance measures) that you feel will most likely suffer from initialization bias. If you can determine long enough warm up periods for these configurations, the other configurations should be relatively safe from the problem by using the longest warm up period found.

There are a number of other techniques that have been developed for the analysis of infinite horizon simulations including the standardized time series method, the regenerative method, and spectral methods. An overview of these methods and others can be found in (Alexopoulos and Seila 1998) and in (Law 2007).

When performing an infinite horizon simulation analysis, we are most interested in the estimation of long-run (steady state) performance measures. In this situation, it can be useful to apply analytical techniques such as queueing theory to assist with determining whether or not the model is producing credible results. Even in the case of a finite horizon simulation, the steady state performance results from analytical models of queueing and inventory systems can be very helpful in understanding if the results produced by the simulation model make sense. In the next section, we apply the results of analytical queueing models from Appendix C to the simulation of a small manufacturing system in order to check the results of a simulation model. Being able to verify and validate a simulation model is a crucial skill to get your simulation models used in practice.

10. What is social simulation?

Social simulation is a research field that applies computational methods to study issues in the social sciences.

Social simulation is a research field that applies computational methods to study issues in the social sciences. The issues explored include problems in computational law, psychology,[1] organizational behavior,[2] sociology, political science, economics, anthropology, geography, engineering,[2] archaeology and linguistics (Takahashi, Sallach & Rouchier 2007).

Social simulation aims to cross the gap between the descriptive approach used in the social sciences and the formal approach used in the natural sciences, by moving the focus on the processes/mechanisms/behaviors that build the social reality.

In social simulation, computers support human reasoning activities by executing these mechanisms. This field explores the simulation of societies as complex non-linear systems, which are difficult to study with classical mathematical equation-based models. Robert Axelrod regards social simulation as a third way of doing science, differing from both the deductive and inductive approach; generating data that can be analysed inductively, but coming from a rigorously specified set of rules rather than from direct measurement of the real world. Thus, simulating a phenomenon is akin to generating it—constructing artificial societies. These ambitious aims have encountered several criticisms.

The social simulation approach to the social sciences is promoted and coordinated by three regional associations, ESSA for Europe, North America (reorganizing under the new CSSS name), and PAAA Pacific Asia.

History and development

The history of the agent-based model can be traced back to the Von Neumann machine, a theoretical machine capable of reproducing itself. The device von Neumann proposed would follow precisely detailed instructions to fashion a copy of itself. The concept was then improved by von Neumann's friend Stanislaw Ulam, also a mathematician; Ulam suggested that the machine be built on paper, as a collection of cells on a grid. The idea intrigued von Neumann, who drew it up—creating the first of devices later termed cellular automata.

Another improvement was brought by mathematician, John Conway. He constructed the well-known Game of Life. Unlike the von Neumann's machine, Conway's Game of Life operated by simple rules in a virtual world in the form of a 2-dimensional checkerboard.

The birth of the agent-based model as a model for social systems was primarily brought about by a computer scientist, Craig Reynolds. He tried to model the reality of lively biological agents, known as the artificial life, a term coined by Christopher Langton.

Joshua M. Epstein and Robert Axtell developed the first large scale agent model, the Sugarscape, to simulate and explore the role of social phenomena such as seasonal migrations, pollution, sexual reproduction, combat, transmission of disease, and even culture.

Kathleen M. Carley published "Computational Organizational Science and Organizational Engineering" defining the movement of simulation into organizations, established a journal for social simulation applied to organizations and complex socio-technical systems: Computational and Mathematical Organization Theory, and was the founding president of the North American Association of Computational Social and Organizational Systems that morphed into the current CSSSA.

Nigel Gilbert published with Klaus G. Troitzsch the first textbook on social simulation: "Simulation for the Social Scientist" (1999) and established its most relevant journal: the Journal of Artificial Societies and Social Simulation.

More recently, Ron Sun developed methods for basing agent-based simulation on models of human cognition, known as cognitive social simulation (see (Sun 2006))

Topics

Here are some sample topics that have been explored with social simulation:

Social norms: Robert Axelrod has used simulations to investigate the foundation of morality;[3] others have modeled the emergence of norms using memes,[4] or how social norms and emotions can regulate each other.[5][6]

Institutions: by investigating under what conditions agents manage to coordinate,[7] or by modeling the works of Robert Putnam on civic traditions[8]

Reputation, for example by making agents with a model of reputation from Pierre Bourdieu (image, social esteem, and prestige) and observing their behavior in a virtual marketplace.[9]

Knowledge transmission and the social process of science: there is a special section on that topic in the Journal of Artificial Societies and Social Simulation[10]

Elections: Kim (2011) has modeled a psychological model of judgement from previous research (notably featuring motivated reasoning), and compared the statistical regularities of the simulation with empirical observations of voter behavior;[11] others have compared delegation methods.[12][13]

Economics: see computational economics and agent-based computational economics.

Types of simulation and modeling

Social simulation can refer to a general class of strategies for understanding social dynamics using computers to simulate social systems. Social simulation allows for a more systematic way of viewing the possibilities of outcomes.

There are four major types of social simulation:

System level simulation.

System level modeling.

Agent-based simulation.

Agent-based modeling.

A social simulation may fall within the rubric of computational sociology which is a recently developed branch of sociology that uses computation to analyze social phenomena. The basic premise of computational sociology is to take advantage of computer simulations (Polhill & Edmonds 2007) in the construction of social theories. It involves the understanding of social agents, the interaction among these agents, and the effect of these interactions on the social aggregate. Although the subject matter and methodologies in social science differ from those in natural science or computer science, several of the approaches used in contemporary social simulation originated from fields such as physics and artificial intelligence.

System level simulation

System Level Simulation (SLS) is the oldest level of social simulation. System level simulation looks at the situation as a whole. This theoretical outlook on social situations uses a wide range of information to determine what should happen to society and its members if certain variables are present. Therefore, with specific variables presented, society and its members should have a certain response to the new situation. Navigating through this theoretical simulation will allow researchers to develop educated ideas of what will happen under some specific variables.

For example, if NASA were to conduct a system level simulation it would benefit the organization by providing a cost-effective research method to navigate through the simulation. This allows the researcher to steer through the virtual possibilities of the given simulation and develop safety procedures, and to produce proven facts about how a certain situation will play out. (National Research 2006)

System level modeling

System level modeling (SLM) aims to specifically predict (unlike system level simulation's generalization in prediction) and convey any number of actions, behaviors, or other theoretical possibilities of nearly any person, object, construct et cetera within a system using a large set of mathematical equations and computer programming in the form of models.

A model is a representation of a specific thing ranging from objects and people to structures and products created through mathematical equations and are designed, using computers, in such a way that they are able to stand-in as the aforementioned things in a study. Models can be either simplistic or complex, depending on the need for either; however, models are intended to be simpler than what they are representing while remaining realistically similar in order to be used accurately. They are built using a collection of data that is translated into computing languages that allow them to represent the system in question. These models, much like simulations, are used to help us better understand specific roles and actions of different things so as to predict behavior and the like.

Agent-based simulation

Agent-based social simulation (ABSS) consists of modeling different societies after artificial agents, (varying on scale) and placing them in a computer simulated society to observe the behaviors of the agents. From this data it is possible to learn about the reactions of the artificial agents and translate them into the results of non-artificial agents and simulations. Three main fields in ABSS are agent-based computing, social science, and computer simulation.

Agent-based computing is the design of the model and agents, while the computer simulation is the part of the simulation of the agents in the model and the outcomes. The social science is a mixture of sciences and social part of the model. It is where the social phenomena is developed and theorized. The main purpose of ABSS is to provide models and tools for agent-based simulation of social phenomena. With ABSS we can explore different outcomes for phenomena where we might not be able to view the outcome in real life. It can provide us valuable information on society and the outcomes of social events or phenomena.

Agent-based modeling

Agent-based modeling (ABM) is a system in which a collection of agents independently interact on networks. Each individual agent is responsible for different behaviors that result in collective behaviors. These behaviors as a whole help to define the workings of the network. ABM focuses on human social interactions and how people work together and communicate with one another without having one, single "group mind". This essentially means that it tends to focus on the consequences of interactions between people (the agents) in a population. Researchers are better able to understand this type of modeling by modeling these dynamics on a smaller, more localized level. Essentially, ABM helps to better understand interactions between people (agents) who, in turn, influence one another (in response to these influences). Simple individual rules or actions can result in coherent group behavior. Changes in these individual acts can affect the collective group in any given population.

Agent-based modeling is an experimental tool for theoretical research. It enables one to deal with more complex individual behaviors, such as adaptation. Overall, through this type of modeling, the creator, or researcher, aims to model behavior of agents and the communication between them in order to better understand how these individual interactions impact an entire population. In essence, ABM is a way of modeling and understanding different global patterns.

Current research

There are several current research projects that relate directly to modeling and agent-based simulation the following are listed below with a brief overview.

"Generative e-Social Science for Socio-Spatial Simulation" or (GENESIS) is a research node of the UK National Centre for e-Social Science funded by the UK research council ESRC.

"National e-Infrastructure for Social Simulation" or (NeISS) is a UK-based project funded by JISC.

"Network Models Governance and R&D collaboration networks" or (N.E.M.O) is a research centre whose main focus is to identify ways to create and to assess desirable network structures for typical functions; (e.g. knowledge, creation, transfer, and distribution.) This research will ultimately aid policy-makers at all political levels in improving the effectiveness and efficiency of network-based policy instruments at promoting the knowledge economy in Europe.

"Agent-based Simulations of Market and Consumer Behavior" is another research group that is funded by the Unilever Corporate Research. The current research that is being conducted is investigating the usefulness of agent-based simulations for modeling consumer behavior and to show the potential value and insights it can add to long-established marketing methods.

"New and Emergent World Models Through Individual, Evolutionary and Social Learning" or (New Ties) is a three-year project that will ultimately create a virtual society developed by agent-based simulation. The project will develop a simulated society capable of exploring the environment and developing its own image of this environment and the society through interaction. The goal of the research project is for the simulated society to exhibit individual learning, evolutionary learning and social learning.

Bruch and Mare's project on neighborhood segregation: The purpose of the study is to figure out the reasoning for neighborhood segregation based on race, and to figure out the tipping point or when people become uncomfortable with the integration levels into their neighborhood, and decide to flee from the neighborhood. They set up a model using flash cards, and put the agent's house in the middle and put houses of different races surrounding the agent's house. They asked people how comfortable they would feel with different situations; if they were okay with one situation, they asked another until the neighborhood was fully integrated. Bruch and Mare's results showed that the tipping point was at 50%. When a neighborhood became 50% minority and 50% white, people of both races began to become uncomfortable and white flight began to rise. The use of agent-based modeling showed how useful it can be in the world of sociology, people did not have to answer why they would become uncomfortable, just which situation they were uncomfortable with.

The MAELIA Program (Multi-Agent Emergent Norms Assessment) is a project dealing with the relationships between the users and managers of a natural resource, in that case water, and the related norms and laws that are to be built within them (conventions) or are imposed to them by other actors (institutions). The purpose of the project is to build a generic multiscale platform which is planned to deal with water conflict-related issues.

The Mosi-Agil project is a four-year program funded by the Autonomous Region of Madrid through the program MOSI-AGIL-CM (grant S2013/ICE-3019, co-funded by EU Structural Funds FSE and FEDER). It aims at creating a body of knowledge and practical tools which are necessary to handle more effectively the behavior of occupants of large facilities. Therefore, the project studies the development of ambient intelligence and intelligent environments supported by the use of Agent-Based Social Simulation.

Agent-based modeling is most useful in providing a bridge between micro and macro levels, which is a large part of what sociology studies. Agent-based models are most appropriate for studying processes that lack central coordination, including the emergence of institutions that, once established, impose order from the top down. The models focus on how simple and predictable local interactions generate familiar but highly detailed global patterns, such as emergence of norms and participation of collective action. Michael W. Macy and Robert Willer researched a recent survey of applications and found that there were two main problems with agent-based modeling the self-organization of social structure and the emergence of social order (Macy & Willer 2002). Below is a brief description of each problem Macy and Willer believe there to be;

"Emergent structure. In these models, agents change location or behavior in response to social influences or selection pressures. Agents may start out undifferentiated and then change location or behavior so as to avoid becoming different or isolated (or in some cases, overcrowded). Rather than producing homogeneity, however, these conformist decisions aggregate to produce global patterns of cultural differentiation, stratification, and homophilic clustering in local networks. Other studies reverse the process, starting with a heterogeneous population and ending in convergence: the coordination, diffusion, and sudden collapse of norms, conventions, innovations, and technological standards."

"Emergent social order. These studies show how egoistic adaptation can lead to successful collective action without either altruism or global (top down) imposition of control. A key finding across numerous studies is that the viability of trust, cooperation, and collective action depends decisively on the embeddedness of interaction."

These examples simply show the complexity of our environment and that agent-based models are designed to explore the minimal conditions, the simplest set of assumptions about human behavior, required for a given social phenomenon to emerge at a higher level of organization.

Criticisms

Since its creation, computerized social simulation has been the target of some criticism in regard to its practicality and accuracy. Social simulation's simplification of the complex to form models from which we can better understand the latter is sometimes seen as a draw back, as using fairly simple models to simulate real life with computers is not always the best way to predict behavior.

Most of the criticism seems to be aimed at agent-based models and simulation and how they work:

Simulations, being man-made from mathematical interfaces, predict human behavior in a far too simple manner in regard to the complexities of humanity and our actions.

Simulations cannot enlighten researchers as to how people interact or behave in ways not programmed into their models. For this reason, the scope of simulations are limited in that the researchers must already know what they are going to find (to a degree, for they cannot find anything they themselves did not place in the model) at least vaguely, possibly skewing the results.

Due to the complexities of what is being measured, simulations must be analyzed in unbiased ways; however, with the model running on a pre-made set of instructions coded into it by a modeler, biases exist almost universally.

It is highly difficult and often impractical to attempt to link the findings from the abstract world the simulation creates and our complex society and all of its variation.

Researchers working in social simulation might respond that the competing theories from the social sciences are far simpler than those achieved through simulation and therefore suffer the aforementioned drawbacks much more strongly. Theories in some social science tend to be linear models that are not dynamic, and are generally inferred from small laboratory experiments (laboratory tests are most common in psychology but rare in sociology, political science, economics and geography). The behavior of populations of agents under these models is rarely tested or verified against empirical observation.