Module 1: Introduction of processing and visualization

Hyoshin (John) Park[1]
Assistant Professor
Department of Computational Science & Engineering
North Carolina A&T State University
Email: hpark1@ncat.edu; Tel.: +1 3015310142; fax: +1 3014052585.

1 Introduction

Read Syllabus together for all modules.

2 Computational thinking

A set of problem-solving methods that involve expressing problems and their solutions in ways that a computer could also execute[2]. It involves the mental skills and practices for 1) designing computations that get computers to do jobs for us, and 2) explaining and interpreting the world as a complex of information processes. Those ideas range from basic computational thinking for beginners to advanced computational thinking for experts[3]. Possibilities are endless! Problem-solving effective and possible regardless of the problem's difficulty.

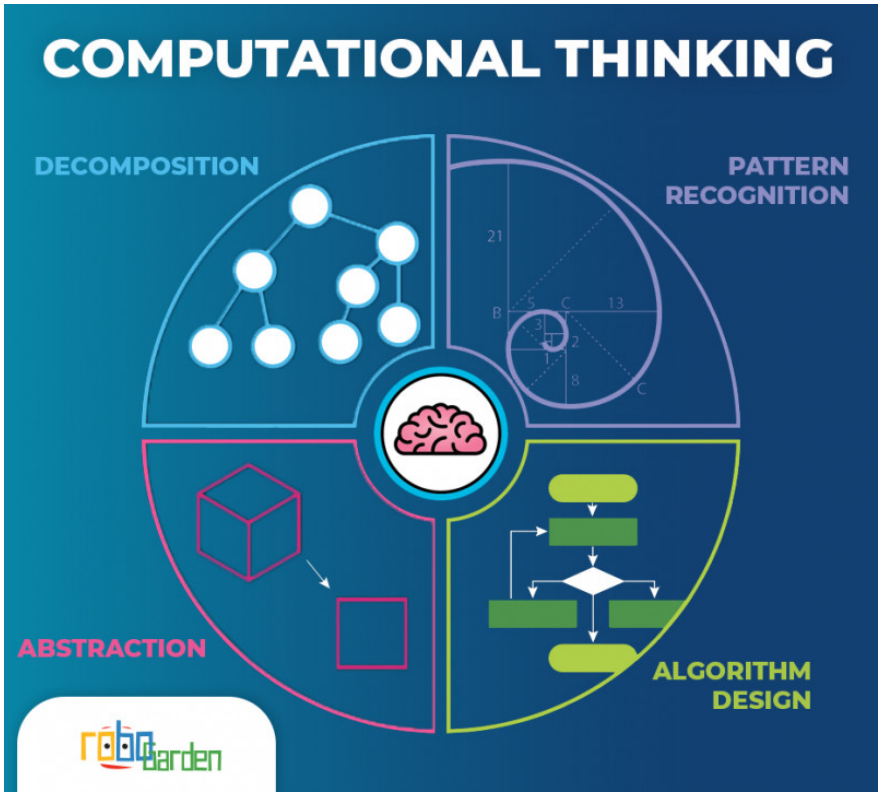

Figure 1: What is computational thinking from RoboGarden

2.1 Decomposition

You should never try to solve a complex problem immediately. First, divide it into small tasks that gradually increase in difficulty and begin with the solution to the simplest task.

[2]Wing, Jeannette (2014). "Computational Thinking Benefits Society". 40th Anniversary Blog of Social Issues in Computing
[3]Denning, P.J. and Tedre, M. Computational Thinking. The MIT Press, 2019.

2.2 Pattern Recognition

After we broke the complicated problem down into small tasks, it became easier to see that there were many similarities between those tasks. Once one task is solved, you can reuse the same solution and solve the whole problem faster.

2.3 Abstraction

Give attention to the major part of the problem and ignore the unnecessary details until you get a single solution that works easily and quickly for multiple problems.

2.4 Algorithm Design

Create solutions and instructions using a series of ordered steps to accomplish a specific task. These instructions will be used to solve repeated problems.

3 Data Life Cycle

3.1 Data Capture

The first experience that an item of data must have is to pass within the firewalls of the enterprise[4]. This is Data Capture, which can be defined as the act of creating data values that do not yet exist and have never existed within the enterprise.

There are three main ways that data can be captured. Data Acquisition: the ingestion of already existing data that has been produced by an organization outside the enterprise. Data Entry: the creation of new data values for the enterprise by human operators or devices that generate data for the enterprise. Signal Reception: the capture of data created by devices, typically important in control systems, but becoming more important for information systems with the Internet of Things. There may well be other ways, but the three identified above have significant Data Governance challenges. For instance, Data Acquisition often involves contracts that govern how the enterprise is allowed to use the data it obtains in this way.

3.2 Data Maintenance

Once data has been captured it usually encounters Data Maintenance. This can be defined as the supplying of data to points at which Data Synthesis and Data Usage occur, ideally in a form that is best suited for these purposes.

What Data Maintenance is about is processing the data without yet deriving any value from it for the enterprise. It often involves tasks such as movement, integration, cleansing, enrichment, changed data capture, as well as familiar extract-transform-load processes.

Data Maintenance is the focus of a broad range of data management activities. Because of this, Data Governance faces a lot of challenges in this area. Perhaps one of the most important is rationalizing how data is supplied to the end points for Data Synthesis and Data Usage, e.g. preventing proliferation of point-to-point transfers.

[4]Bloomberg data governance

3.3 Data Synthesis

This is comparatively new, and perhaps still not a very common phase in the Data Life Cycle. It can be defined as the creation of data values via inductive logic, using other data as input.

It is the arena of analytics that uses modeling, such as is found in risk modeling, actuarial modeling, and modeling for investment decisions. Derivation by deductive logic is not part of this – that occurs in Data Maintenance. An example of deductive logic is Net Sales = Gross Sales – Taxes. If I know Gross Sales and Taxes, and I know the simple equation just outlined, then I can calculate Net Sales.

Inductive logic requires some kind of expert experience, judgement, and/or opinion as a part of the logic, e.g. the way in which credit scores are created.

3.4 Data Usage

So far we have seen how our in single data value has entered the enterprise via Data Capture, and has been moved around the enterprise, perhaps being transformed and enriched in Data Maintenance, and possibly being an input to Data Synthesis. Next, it reaches a point where it is used in support of the enterprise. This is Data Usage, which can be defined as the application of data as information to tasks that the enterprise needs to run and manage itself.

This would normally be tasks outside the data life cycle itself. However, data is becoming more central to business models in many enterprises. For instance, data may itself be a product or service (or part of a product or service) that the enterprise offers. This too is Data Usage, even if it is part of the Data Life Cycle, because it is part of the business model of the enterprise.

Data usage has special Data Governance challenges. One of them is whether it is legal to use the data in the ways which business people want. This is referred to as "permitted use of data". There may be regulatory or contractual constraints on how data may actually be used, and part of the role of Data Governance is to ensure that these constraints are observed.

3.5 Data Publication

In being used, it is possible that our single data value may be sent outside of the enterprise. This is Data Publication, which can be defined as the sending of data to a location outside of the enterprise.

An example would be a brokerage that sends monthly statements to its clients. Once data has been sent outside the enterprise it is de facto impossible to recall it. Data values that are wrong cannot be corrected as they are beyond the reach of the enterprise. Data Governance may be need to assist in deciding how incorrect data that has been sent out of the enterprise will be dealt with. Unhappily, data breaches also fall under Data Publication.

3.6 Data Archival

Our single data value may experience many rounds of usage and publication, but eventually the end of its life begins to loom large. The first part of this is to archive the data value. Data Archival is the copying of data to an environment where it is stored in case it is needed again in an active production environment, and the removal of this data from all active production environments.

A data archive is simply a place where data is stored, but where no maintenance, usage, or publication occurs. If necessary the data can be restored to an environment where one or more of these occur.

3.7 Data Purging

We now come to the actual end of life of our single data value. Data Purging is the removal of every copy of a data item from the enterprise.

Ideally, this will be done from an archive. A Data Governance challenge in this phase of the data life cycle is proving that the purge has actually been done properly.

4 Data Ingestion Challenges

4.1 Process challenges

Speed can be a challenge for both the ingestion process and the data pipeline. As data grows more complex, it's more time-consuming to develop and maintain data ingestion pipelines, particularly when it comes to "real-time" data processing, which depending on the application can be fairly slow (updating every 10 minutes) or incredibly current (think stock ticker applications during trading hours).

4.2 Pipeline challenges

Businesses make decisions based on the data in their analytics infrastructure, and the value of that data depends on their ability to ingest and integrate it. If the initial ingestion of data is problematic, every stage down the line will suffer, so holistic planning is essential for a performant pipeline.

5 Data Management

The big opportunity of the Information Age[5]: extract useful findings from the immense wealth of data and information acquired, computed, and stored by modern information systems.

However, there are many obstacles, which impede the effective exploitation of such an opportunity: users and analysts may get overwhelmed by irrelevant, or inappropriately processed or presented information – the information overload problem.

- Heterogeneity of data sources.
- Different data types.
- Data streams.
- Working under pressure.
- Time consuming activities.

Data management ensures data consistency and standards.

Figure 2 shows the high-level framework for analyzing vis use according to three questions[6]: what data the user sees, why the user intends to use a vis tool, and how the visual encoding and interaction idioms are constructed in terms of design choices. Each three-fold what–why–how question has a corresponding data–taskidiom answer trio. One of these analysis trios is called an instance[7].

[5]Book, Solving problems with visual analytics
[6]Book: Visualization Analysis and Design
[7]Book: Visualization Analysis and Design

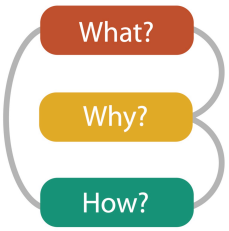

Figure 2: Three-part analysis framework for a vis instance: why is the task being performed, what data is shown in the views, and how is the vis idiom constructed in terms of design choices.

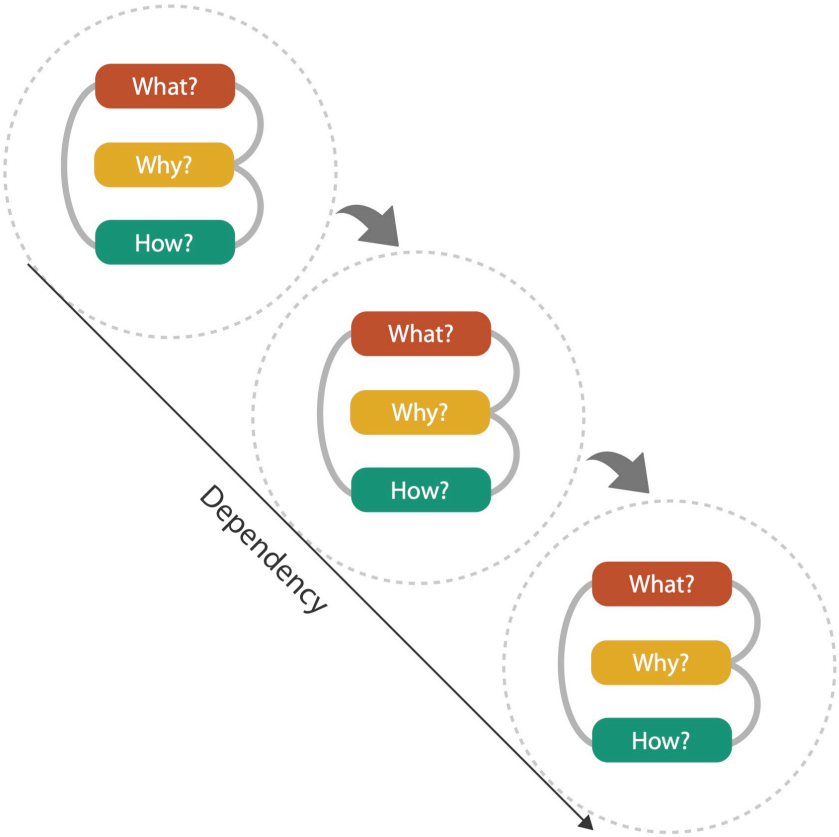

Figure 3: Analyzing vis usage as chained sequences of instances, where the output of one instance is the input to another.

Simple vis tools can be fully described as an isolated analysis instance, but complex vis tool usage often requires analysis in terms of a sequence of instances that are chained together. In these cases, the chained sequences are a way to express dependencies. All analysis instances have the input of what data is shown; in some cases, output data is produced as a result of using the vis tool. An abstract example of a chained sequence, where the output of a prior instance serves as the input to a subsequent one.

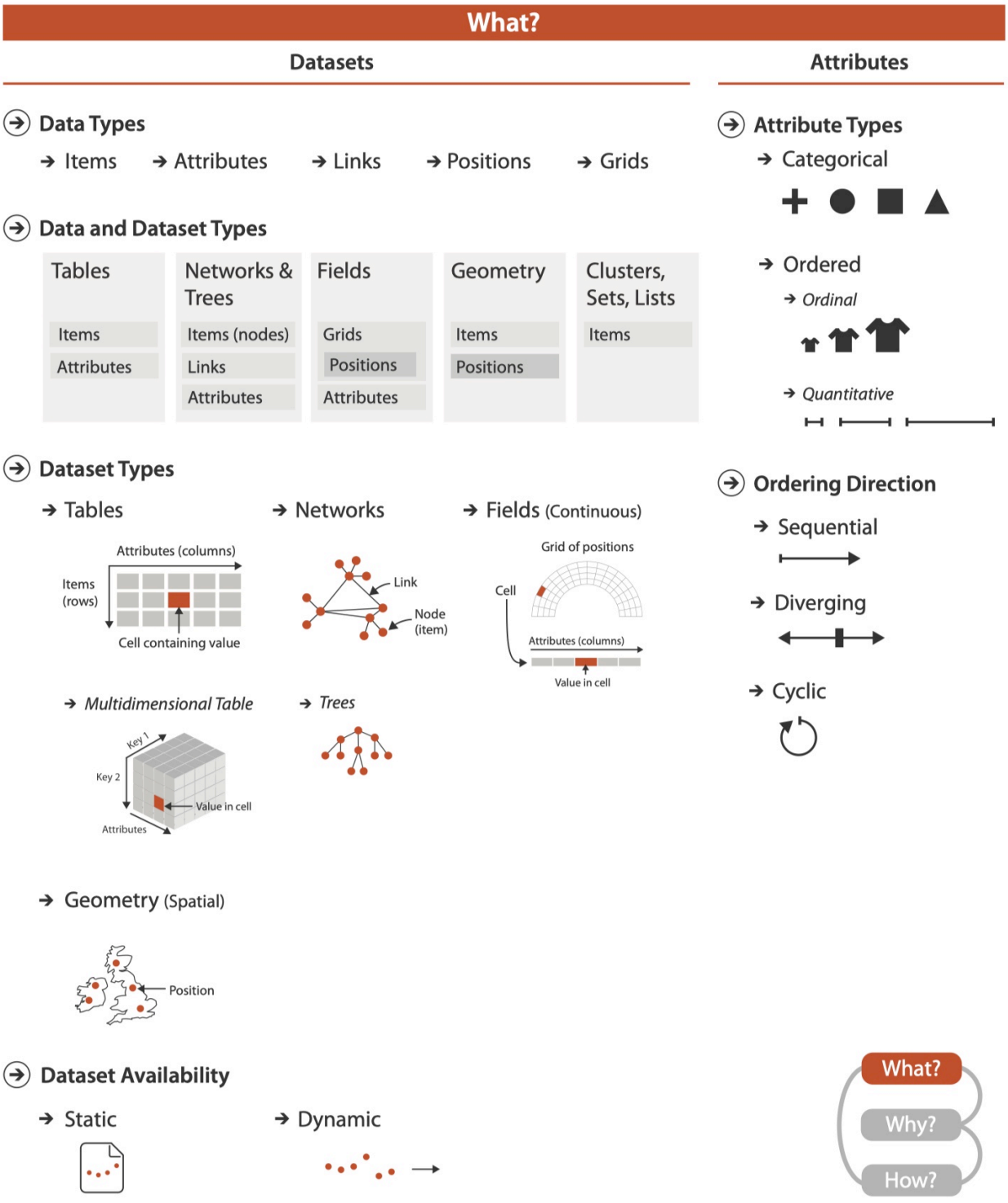
What?
Datasets
Attributes
Data Types
→ Items
→ Attributes
→ Links
→ Positions
→ Grids
Data and Dataset Types
Tables
Items
Attributes
Networks & Trees
Items (nodes)
Links
Attributes
Fields
Grids
Positions
Attributes
Geometry
Items
Positions
Clusters, Sets, Lists
Items
Dataset Types
→ Tables
Attributes (columns)
Items (rows)
Cell containing value
→ Networks
Link
Node (item)
→ Fields (Continuous)
Grid of positions
Cell
Attributes (columns)
Value in cell
→ Multidimensional Table
Key 1
Key 2
Value in cell
Attributes
→ Trees
→ Geometry (Spatial)
Position
Dataset Availability
→ Static
→ Dynamic
Attribute Types
→ Categorical
→ Ordered
→ Ordinal
→ Quantitative
Ordering Direction
→ Sequential
→ Diverging
→ Cyclic
What?
Why?
How?

Figure 4

6 Interaction analysis

6.1 Two -way interactions

We interpret two significant interactions between driver character predictors in Figure 2. Te first interaction (two-way) occurs between gender and age groups ($p < 0.001$). To be more specific, young and middle-age females stopped farther away from the stop line as compared to males in the same age groups, who stopped closer to the stop line. On the contrary, females in the old age group stopped closer to the stop line, while in the old age group males stopped farther away. Te older age group presents a smaller discrepancy between females and males as compared to those in the young and middle-age groups. This finding of a significant difference in gender depending on different age groups indicates that there is indeed significant interaction. Different patterns of slope make each age group intersect with the others and present a crossover interaction[8].

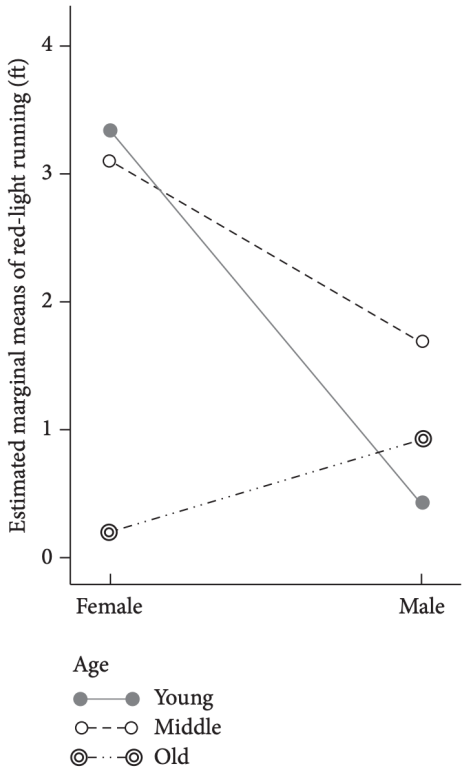

Figure 5: Two-way interaction between gender and age infuencing red-light running

6.2 Three -way interactions

A three-way interaction occurs between cell phone usage, interface, and age groups. To portray this three-way interaction, we plot two-way interactions separately. When a driver has an HF phone(a), the shapes of lines for young, middle, and old age groups have similarities that do not indicate any significant difference between no, incoming, and outgoing cell phone use groups. Te similarity between age groups does not

[8]Park and Pugh, Generalized Estimating Equation Model Based Recursive Partitioning: Application to Distracted Driving, Journal of Advanced Transportation, Volume 2018 |Article ID 3245864 | 11 pages

present interaction with cell phone use groups. However, there is a three-way interaction, because two-way interaction plots (age∗cell phone use) corresponding to different levels of the cell phone interface present significant differences. Age group lines are not parallel; instead, they cross each other one time when a driver has an HS phone (b), and they cross two times when a driver has an HH phone (c). We find that this difference in two-way interactions indicates that there is a statistically significant three-way interaction. These values are based on covariates evaluated at value of speed (62.20 fs) and yellow interval (4.04 s).

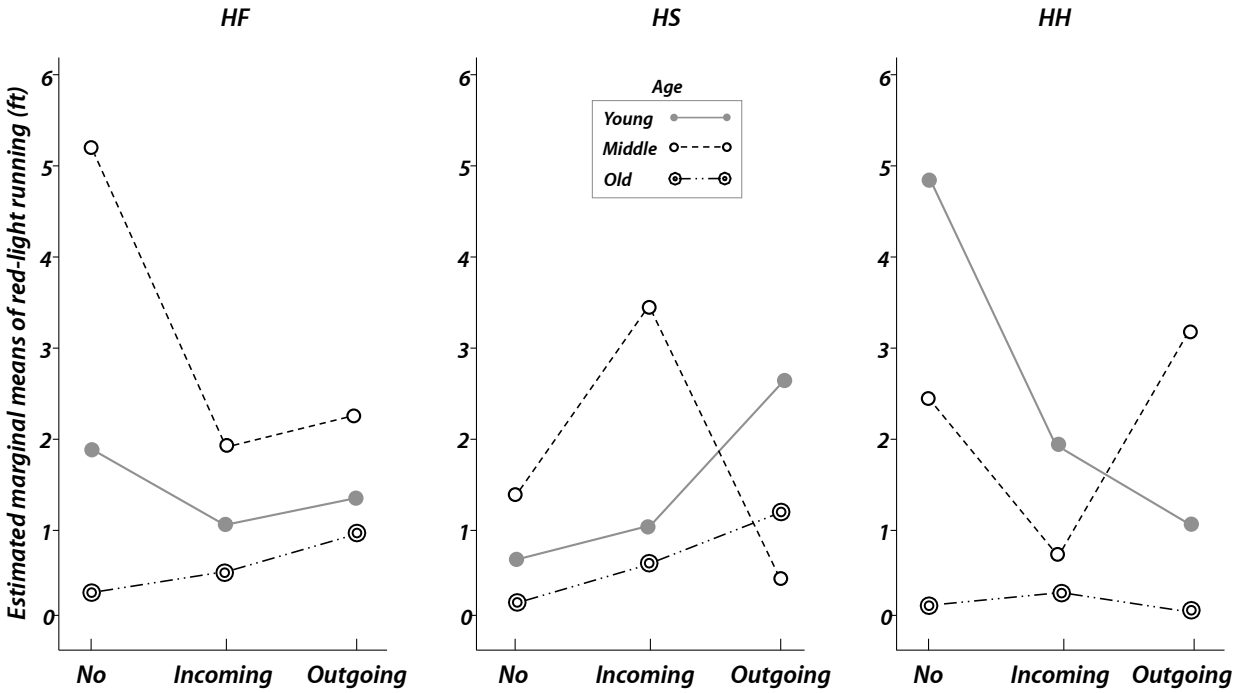

Figure 6: Tree-way interaction between cell phone usage, interface, and age infuencing red-light running.

Appendix

7 Modeling Process

The first step is to analyze the problem and define the objectives of the model. This step should include a review of the literature to uncover previous research on the topic, experimental or field-measured data showing various states of the system and the measured outcomes, mathematical representations of the system derived from theories, and previous modeling efforts.

As that information is being gathered, it is also important to define the objectives of the fgf modeling effort. There are several questions that should be addressed while considering the model objectives: What are the outcomes that we would like the model to predict? Are we interested in every possible outcome or is there a subset of conditions that would satisfy our model objectives? For example, we could be interested in just the average or normal state of affairs associated with a phenomenon or potential extreme events may be critical for our analysis. What level of accuracy is required for the predicted outcomes? This will impact the nature of the simplifying assumptions, input data, and computing algorithms that are required to build the model.

The second step in the process is to create a conceptual model of the system based on the analysis in the first step. A conceptual model will begin to specify all of the cause and effect relationships in the system, information on the data required and available to implement a model, and references to documents that were found in the initial analysis. The conceptual model should include a concept map showing the cause and effect relationships associated with the model and tables showing the different variables, data sources, and references. This can be done on a whiteboard, pencil and paper, or using a formal flowcharting or concept-mapping tool. There are also a number of commercial packages.

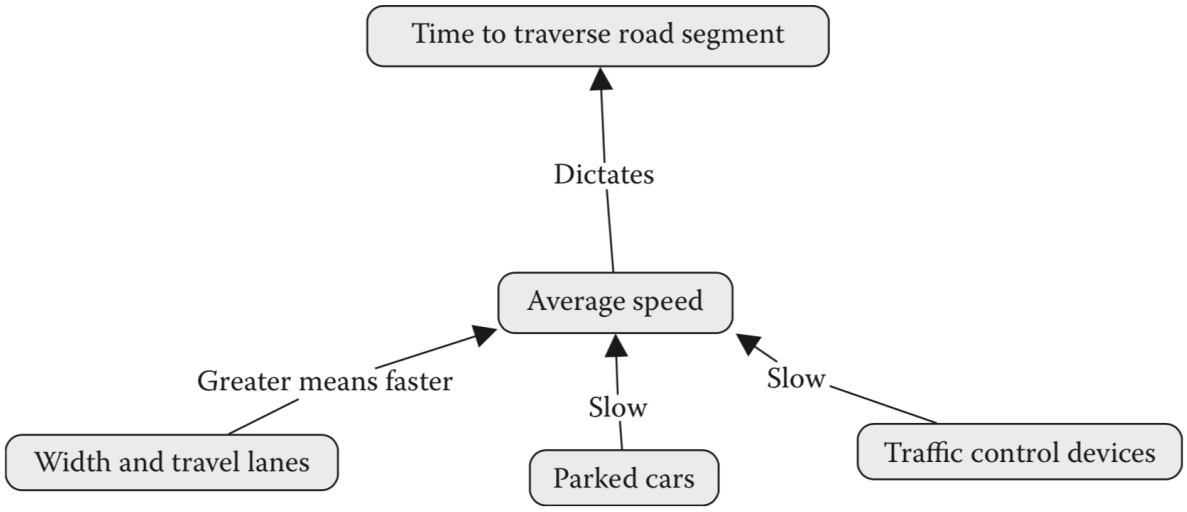

Figure 7: Partial concept map of model to calculate travel time using Cmap..

The average speed across a road segment is slowed by parked cars and traffic control devices while wider lanes and higher speed limits take less time. The total time for a trip would need to add the average times associated with traversing each road segment. Thus, data on each segment will be needed as input to the model. Simple versions of such estimates are provided by global positioning satellite (GPS) equipment or the Internet mapping services that are available online. There are many other conditions that would

impact this system. Modeling traffic conditions are a topic of one of the exercises at the end of the chapter.

7.1 Modeling Princples

A system has inputs and outputs. The standard system class (causal, continuous-time, finite-dimensional, continuous-valued, and linear). Transfer functions are not well defined for time varying OR nonlinear systems.: Suppose I derive a model for the system with states x_1 and x_2: Then there is a model with states $x_1 + ax_2$ and $x_1 + bx_2$ (provided that $a \neq b$).

There are several major approaches to modeling[9]:

7.1.1 Modeling from first principles and elements

(a) The first principles themselves such as Newton's Laws, etc...

(b) Constitutional relations or dependencies.

(c) A "connectivity" description such as a circuit schematic. This is usually described in graph-theoretic terms.

7.1.2 Modeling as black boxes

Using input and output data, a model form is postulated/assumed and the data/first principles are used to verify and justify the model. Actually this is how Mr. Ohm and Mr. Hooke found their laws. Fancy modern-age examples include: neural networks, fuzzy approximators/models, linear regression, etc. This is deeply related to the so-called system identification problems. This often uses a connectionist approach.

7.1.3 Extracting black boxes

Express them as rules: visualization.

Train the neural network, extract the rules, and build a decision tree.

Will be discussed more in the Topic 2.

7.1.4 Modeling as grey boxes

These are hybrid models where insight, understanding and first principles can be used to build the model, but only partially. There are pieces and parameters in the model that still need to be estimated/learned. A classical example is what is called a human-in-the-loop system or model.

A piloted airplane example. The airplane model can be assumed to be known and "exact" and may be constructed from first principles. The pilot model is typically not known and cannot be built from "first principle" because such first principle do not exist in general. One approach is to augment the plane model with a connectionist model which is trained using instrumentation of the plane that can model the pilot reaction time, percentage of correct decisions, adherence to protocol.

8 Modeling and Simulation

A mathematical model is a representation of a phenomenon or system that is used to provide insights and predictions about system behavior. There are several different ways to classify models.

[9] Extended from Dr. Bikdash's System Theory Note and Dr. Hyoshin Park's paper

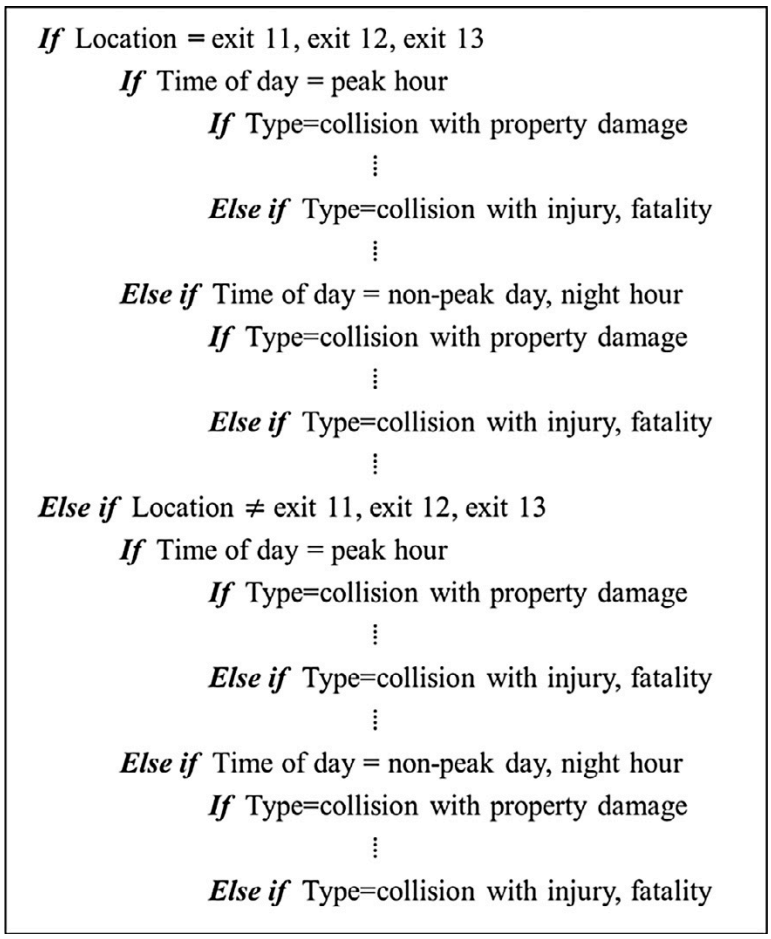

Figure 8: Extracted If-Then-Else rules for second split from decision tree (H. Park, A. Haghani, Transportation Research Part C 70 (2016) 69–85)

Deterministic vs Probabilistic. A deterministic model applies a set of inputs or initial conditions and uses one or more equations to produce model outputs. The outputs of a deterministic model will be the same for each execution of the computer code with the same inputs.

Another term for probabilistic is stochastic meaning a random process or a process, which occurs by chance. A probabilistic model includes one or more elements that might occur by chance or at random while a deterministic model does not. A probabilistic model will exhibit random effects that will produce different outputs for each model run.

Static vs Dynamic. A dynamic model considers the state of a system over time while a static model does not. For example, one could have a model of a material like a steel beam that considered its ability to bear weight without bending under a set of standard environmental conditions. This would be considered to be a static model of that system. A dynamic model of the same structure would simulate how the bearing strength and possible deformation of the beam would change under stresses over time such as under high temperatures, vibration, and chemical corrosion.

Simulation is the application of a model to imitate the behavior of the system under a variety of circumstances.

8.1 State-Space Model

A time derivative is a derivative of a function with respect to time, usually interpreted as the rate of change of the value of the function

Input-output description of the plant, usually expressed as a transfer function. These methods do not use any knowledge of the interior structure of the plant, and limit us to single-input single-output (SISO) systems, and as we have seen allows only limited control of the closed-loop behavior when feedback control is used[10].

Modern control theory solves many of the limitations by using a much "richer" description of the plant dynamics. The so-called state-space description provide the dynamics as a set of coupled first-order differential equations in a set of internal variables known as state variables, together with a set of algebraic equations that combine the state variables into physical output variables.

In the standard form the mathematical description of the system is expressed as a set of n coupled first-order ordinary differential equations, known as the state equations, in which the time derivative of each state variable is expressed in terms of the state variables $x_1(t), \ldots, x_n(t)$ and the system inputs $u_1(t), \ldots, u_n(t)$. In the general case the form of the state equations is:

$$\dot{\mathbf{x}}(t) \equiv A(t)\mathbf{x}(t) + B(t)\mathbf{u}(t) \tag{1}$$
$$\mathbf{y}(t) \equiv C(t)\mathbf{x}(t) + D(t)\mathbf{u}(t) \tag{2}$$

$\mathbf{x}(\cdot)$ is called the"state vector", $\mathbf{x}(t) \in \mathbb{R}^n$;
$\mathbf{y}(\cdot)$ is called the"output vector", $\mathbf{y}(t) \in \mathbb{R}^q$;
$\mathbf{u}(\cdot)$ is called the"input vector", $\mathbf{u}(t) \in \mathbb{R}^p$;
$A(\cdot)$ is the"state (or system) matrix", $dim\big[A(\cdot)\big] = n \times n$;
$B(\cdot)$ is the"input matrix", $dim\big[B(\cdot)\big] = n \times p$;
$C(\cdot)$ is the"output matrix", $dim\big[C(\cdot)\big] = q \times n$;
$D(\cdot)$ is the"feedthrough matrix", $dim\big[D(\cdot)\big] = q \times p$;

$$\dot{\mathbf{x}}(t) := \frac{d}{dt}\mathbf{x}. \tag{3}$$

In this general formulation, all matrices are allowed to be time-variant (i.e. their elements can depend on time); however, in the common case, matrices will be time invariant. The time variable t can be continuous or discrete. In the latter case, the time variable k is usually used instead of t. Hybrid systems allow for time domains that have both continuous and discrete parts. Depending on the assumptions taken, the state-space model representation can assume different forms: Continuous time-invariant, Continuous time-variant, Explicit discrete time-invariant, Explicit discrete time-variant, and etc.

A steady-state model is a model that has gone through a transient state such as a start-up or warm-up period and arrived at an observed behavior that remains constant.

An example of the steady-state model is the flow of fluid through a pipe. In the initial, transient state period, the pipe is empty and will fill with fluid under pressure until the capacity of the pipe is reached. This will be its steady-state condition. In economics, a steady-state economy is one that has reached a relatively stable size.

The internal state variables are the smallest possible subset of system variables that can represent the entire state of the system at any given time.The minimum number of state variables required to represent a given system, n, is usually equal to the order of the system's defining differential equation. If the system is represented in transfer function form, the minimum number of state variables is equal to the order of the transfer function's denominator after it has been reduced to a proper fraction. It is important to understand that converting a state-space realization to a transfer function form may lose some internal

[10] Derek Rowell, 2002, State-Space Representation of LTI Systems

information about the system, and may provide a description of a system which is stable, when the state-space realization is unstable at certain points. In electric circuits, the number of state variables is often, though not always, the same as the number of energy storage elements in the circuit such as capacitors and inductors. The state variables defined must be linearly independent, i.e., no state variable can be written as a linear combination of the other state variables or the system will not be able to be solved.

Perhaps making things more confusing, a dynamic model can have deterministic components. Such a model would track the state of a system over time and/or space. Given a current state, a deterministic function may be used to predict the future state of the system. Alternatively, the future state may be stochastic, which is impacted by random events.

Finally, dynamic models may be characterized as being discrete or continuous. A continuous model would represent time as a continuous function, whereas a discrete model divides time into small increments and calculates its state for each time period. In computer modeling, most (all?) dynamic models divide time into discrete increments to facilitate rapid calculations that mimic continuous systems.

In order to gain insights into system behavior, simulations are used to ask what if questions about how the system changes under different circumstances. How these questions are addressed depends in part on the typeof model and its underlying mathematical structure. Solving those mathematical equations on a computer also leads to differences in programming logic or the algorithms that are used to calculate the most accurate answer most efficiently. We will discuss some of those algorithms as we go through the rest of this class.

8.2 Different Simulation Approaches

8.2.1 Deterministic models

Deterministic models consist of one or more equations that characterize the behavior of a system. Most such models simplify the system by assuming that one or more causal variables or parameters are constant for a single calculation of the model outcomes.

Example of travelers choice behavior: the willingness to make a trip is inversely proportional to the trip distance. That is, people are more likely to make a trip from home to get to a destination that is closer than the one that is far away. Empirical studies have shown that this friction of distance changes depending on the nature of the trip. People are much more willing to make a longer trip to get to work than they are to do a convenience shopping trip. To simplify the system, these models assume a constant value of this friction of distance factor for each type of trip. When such a model is applied to a new urban area, there is some uncertainty that the constants found in previous studies in different places match the area where the model is being applied. Thus, a study is done where the model is run with different but reasonable variations in the constants to ascertain the impact of those changes on the predicted trips. Those can then be compared with a sample of real data to calibrate and validate the model.

Example of structures models different environmental conditions will alter system behavior, air and water pollution models where assumptions are made about the rate of dispersion of contaminants, and models of drug absorption into the blood stream where assumptions are made about absorption rates and excretion rates of the drug within the body. Many models include components that are both stochastic and deterministic where parametric studies are done on the deterministic components.

8.2.2 Dynamic Models

The state of the system at any time period is dependent, in part, on the state of the system at the previous time period. Simulations calculate the changes in the state of the system over time. An example is a model of ball being dropped from a bridge. As it is dropped the ball accelerates due to the force of gravity. At each time increment, the model will calculate the velocity of the ball and its position in space. That position will depend on where it was in the previous time period and how far it was dropped related to its velocity during that time period. The model will then predict when the ball will hit the water and at what velocity.

8.2.3 Stochastic Models

Typically will have characteristics in common with dynamic models. The difference is that one or more of the governing parameters are probabilistic or could happen by random chance. One example is a model of the spread of a disease that is passed by human contact. A susceptible person may make contact with an infected person but will not necessarily become infected. There is a probability of being infected that is related to the virility of the disease, the state of health of the susceptible person, and the nature of the contact. A model of this system would simulate those probabilities to project the potential spread of a disease outbreak.

As we go through the rest of this semester, we will describe the mathematical representation of each of these types of models and the programming steps needed to implement them on the computer.

8.2.4 Static vs Dynamic Linear Models

Static Linear models describe a continuous response variable as a function of one or more predictor variables. They can help you understand and predict the behavior of complex systems or analyze experimental, financial, and biological data. Linear regression is a statistical method used to create a linear model.

Dynamic Linear models A static linear regression has the parameter is allowed to vary over time in a dynamic regression while it is fixed for all time in static regression. In terms of the generative process, for the static model, we would place a distribution on whose parameters are fixed for all time. We could then generate data by drawing from this distribution and then generating.

For the dynamic model, we could place a distribution on that depends only on data up through time. Thus, the key difference between the two models is that the parameters of in the static model are fixed for all time while they can change in the dynamic model.

8.2.5 Deterministic vs Dynamic Linear Models

Deterministic vs. probabilistic (stochastic): A deterministic model is one in which every set of variable states is uniquely determined by parameters in the model and by sets of previous states of these variables; therefore, a deterministic model always performs the same way for a given set of initial conditions. Conversely, in a stochastic model—usually called a "statistical model"—randomness is present, and variable states are not described by unique values, but rather by probability distributions.

8.2.6 Discrete vs Continuous Models

Discrete vs. continuous: A discrete model treats objects as discrete, such as the particles in a molecular model or the states in a statistical model; while a continuous model represents the objects in a continuous manner, such as the velocity field of fluid in pipe flows, temperatures and stresses in a solid, and electric field that applies continuously over the entire model due to a point charge.