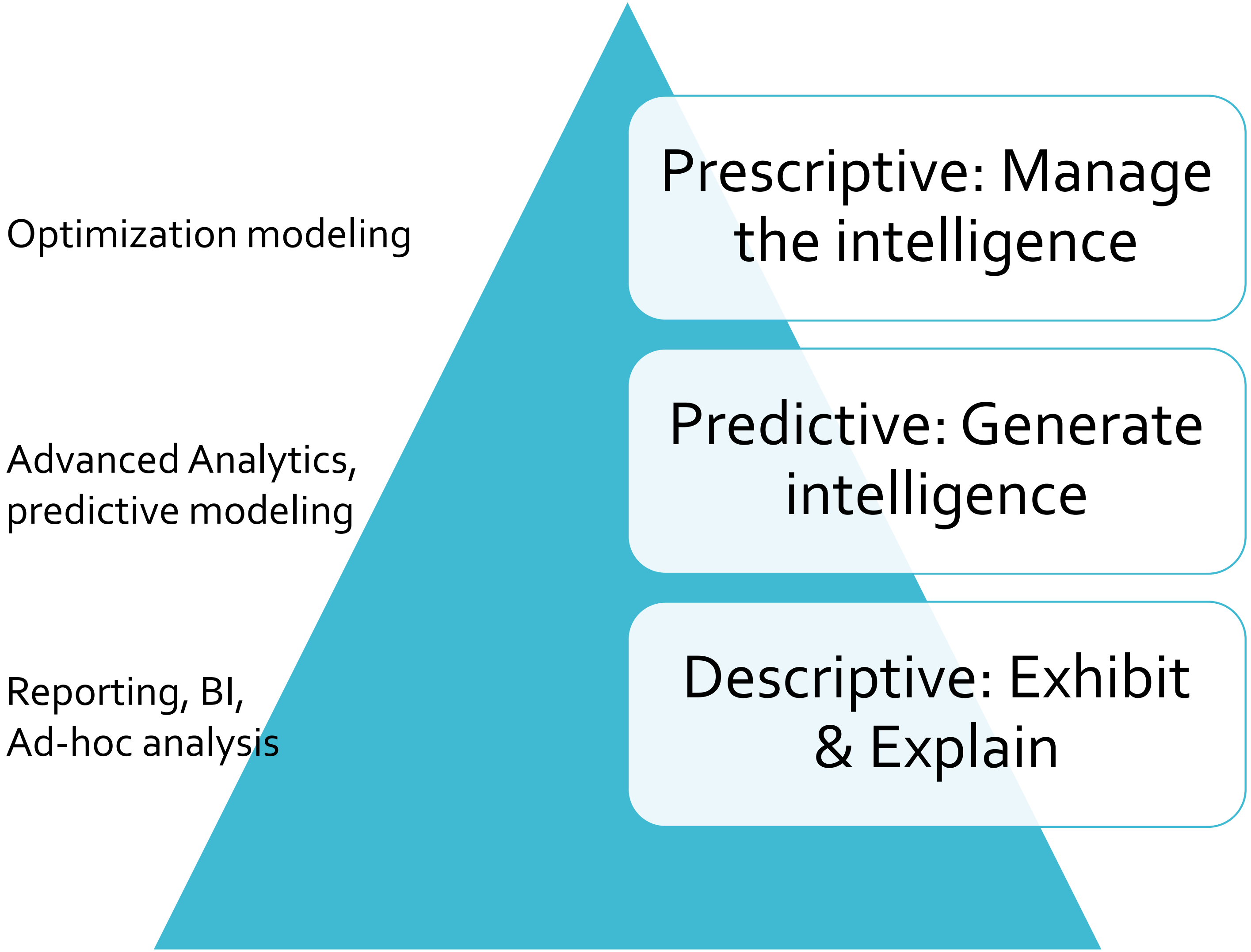
Mathematical Programming

DSA 6000: Data Science and Analytics, Fall 2019

Wayne State University

Levels of Analytics



Optimal Decision-making

- Predictive analytics tells you what the future will look like (with certain confidence).
- However, if knowing an additional piece of information does not change your decision, then this piece of information is of no use for you.
- Ultimately, you want to make better decisions.
- One machinery for formulating complex decision-making is Mathematical Programming (or Mathematical Optimization, or more broadly Operations Research).

More data ≠ better decisions

- Herbert Simon (1971, pp. 40–41): "In an information rich world, the wealth of information means a dearth of something else: a scarcity of whatever it is that information consumes. What information consumes is rather obvious: it consumes the attention of its recipients. Hence a wealth of information creates a poverty of attention and a need to allocate that attention efficiently among the overabundance of information sources that might consume it."

Basic formulation

- Building blocks in a Mathematical Program:
  - A **SET** of objects of interest or of relevance to the problem
    - Product = {Desk, Chair}, Resource = {Carpentry, Painting}
  - Quantitative attributes of objects and their relations, called **PARAMETERS**
    - Price of a desk is $100
    - There are 50 carpentry hours available per week
    - It takes 2 hours of carpentry to make a desk
  - Quantitative **VARIABLES** representing the decisions or actions, to be obtained via solving the mathematical program
    - How many desks shall we make per week?
- A mathematical program is consists of
  - An objective function
  - A set of constraints

Objective function

- Objective function – an algebraic function expressed in terms of the parameters and variables that captures the decision consequences to be minimized or maximized
  - Maximize: profit, rate of success, social welfare, ROI, etc.
  - Minimize: cost, risk, loss, pollution
- The objective must be describable by decision variables and input parameters
  - You cannot optimize something that you cannot describe/measure
  - The objective value must depend on the decision variables

Constraints

- All decisions are made in the confines of some constraints
  - Physical constraints
  - Resource constraints
  - Regulatory constraints
- Constraints model the relation between data and decision and the acceptable range of decisions
- Constraints are in the form of equalities or inequalities

Standard format

- Once the variables are defined, constraints detailed, and objectives quantified, the mathematical model is complete.
- A standard statement of an optimization model has the form:
  - Minimize (or maximize) [Objective function]
  - Subject to (s.t.) [Constraints]

Example (fictitious)

- You want to find a least expensive diet plan to meet nutrient requirements
- x_1 = servings of yogut to eat per day
- x_2 = servings of carrots to eat per day

$$
\begin{aligned}
\min\ \ & 2.0x_1 + 1.5x_2 \ \ \text{(cost)} \\
s.t.\ \ & 0.3x_1 + 0.4x_2 \geq 2 \ \ \text{(protein)} \\
& 0.4x_1 + 0.2x_2 \geq 1.5 \ \ \text{(calorie)} \\
& 0.2x_1 + 0.3x_2 \geq 0.5 \ \ \text{(vitamin)} \\
& x_1 \leq 9 \ \ \text{(yogut availability)} \\
& x_2 \leq 6 \ \ \text{(carrot availability)} \\
& x_1, x_2 \geq 0
\end{aligned}
$$

Feasible solution

- $$\begin{aligned} \min\ & 2.0x_1 + 1.5x_2 \\ s.t.\ & 0.3x_1 + 0.4x_2 \geq 2 \ \text{(protein)} \\ & 0.4x_1 + 0.2x_2 \geq 1.5 \ \text{(calorie)} \\ & 0.2x_1 + 0.3x_2 \geq 0.5 \ \text{(vitamin)} \\ & x_1 \leq 9 \ \text{(yogut availability)} \\ & x_2 \leq 6 \ \text{(carrot availability)} \\ & x_1, x_2 \geq 0 \end{aligned}$$
- A **feasible set** (or region) of an optimization model is a collection of choices for decision variables satisfying all model constraints.
- Each element in the feasible set is called a **feasible solution**.
- Is $(x_1, x_2) = (3, 4)$ a feasible solution?
- An optimization model is **infeasible** if no choice of decision variables satisfy all constraints.

Optimal solutions

- An **optimal solution** is a feasible solution with objective function value at least equal to (if not better than) that of any other solution satisfying all constraints
- The **optimal value** in an optimization model is the objective function value of any optimal solutions.
- An optimization model may have a **unique optimal solution** or several **alternative optimal solutions**.
- An optimization model is **unbounded** when feasible choices of the decision variables can produce arbitrarily good objective function values.
  - Unbounded models have no optimal solutions because any possibility can be improved.

Linear and Nonlinear programs

- The general form of a mathematical program:
  - $\min or \max f(x_1, \dots, x_n)$
  - s.t. $g_i(x_1, \dots, x_n) \begin{Bmatrix} \le \\ = \\ \ge \end{Bmatrix} b_i, \quad i = 1, \dots, m$
  - where $f, g_1, \dots, g_m$ are given functions of decision variables $x_1, \dots x_n$, and $b_1, \dots, b_m$ are specified constant parameters, called the Right-hand Sides (RHS).
- A **Linear Program (LP)** is a mathematical program in which f and all g_i's are linear in the decision variables.
- If f or any g_i is nonlinear in the decision variables, the mathematical program is called a **Nonlinear Program (NLP).**
- When there is an option, linear constraints and objective functions are preferred to nonlinear ones. Nonlinearity usually reduces the *tractability* of a mathematical program as compared to linear forms.

Linear vs nonlinear functions

- A function is linear if it is a constant-weighted sum of decision variables. Otherwise, it is nonlinear.
- Are these functions linear or nonlinear (Note: x_1, x_2, x_3 are variables and all other symbols are constants)?
  - $f(x_1, x_2, x_3) = 9x_1 - 17x_3$
  - $f(x_1, x_2, x_3) = \sum_{j=1}^{3} c_j x_j$
  - $f(x_1, x_2, x_3) = \frac{5}{x_1} + 3x_2 - 6x_3$
  - $f(x_1, x_2, x_3) = x_1 x_2 + (x_2)^3 + \ln(x_3)$
  - $f(x_1, x_2, x_3) = e^{\alpha} x_1 + \ln(\beta)\, x_3$
  - $f(x_1, x_2, x_3) = \frac{x_1 + x_2}{x_2 - x_3}$

Discrete (or Integer) Programs

- A variable is **discrete** if it is limited to a fixed or countable set of values.
  - $x =$ install solar panels or not, 1 yes, 0 no
  - $x_1 =$ how many male passengers to board on Bus #1
- A variable is **continuous** if it can take on any value in a specified interval.
- When there is an option, such as when optimal variable magnitudes are likely to be large enough that fractions have no practical importance, modeling with continuous variables is preferred to discrete.
  - Example: barrels of crude oil to import, when we are talking about thousands or more, fraction of a barrel is not important.

Integer and Mixed Integer Programs

- An optimization model is an integer program (IP) if any one of its decision variables is discrete. If all variables are discrete, the model is a pure integer program; otherwise, it is a mixed integer program (MIP).

Choosing discrete vs continuous variables (exercise)

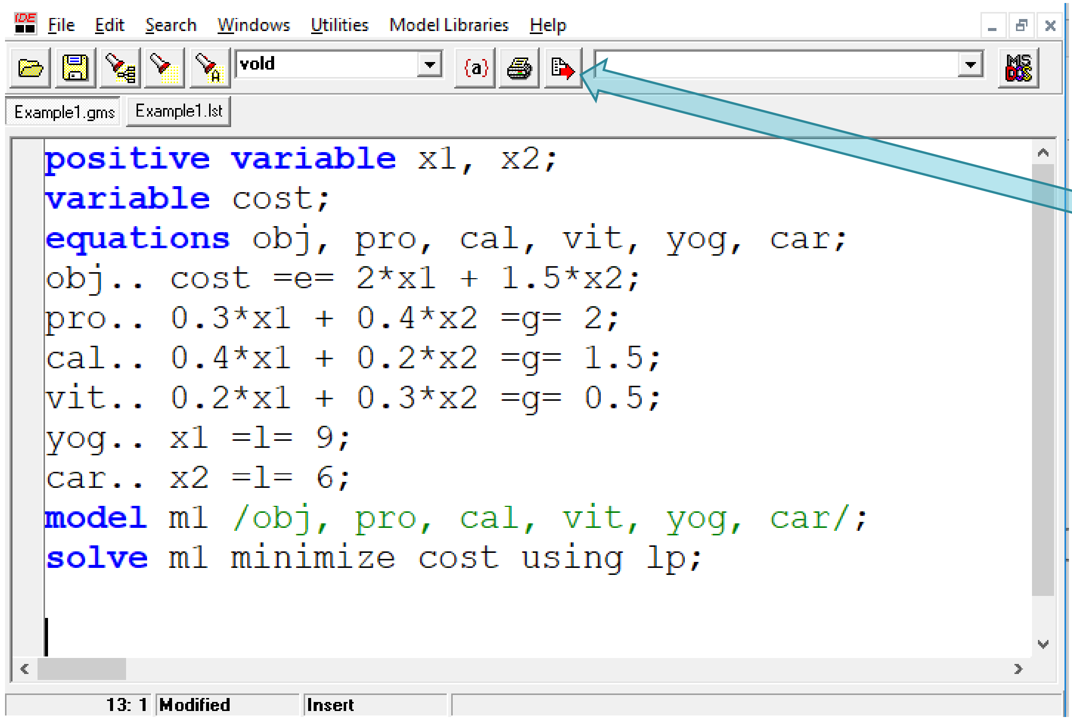
- Decide whether a discrete or a continuous variable would be best employed to model each of the following quantities.
  a) The operating temperature of a chemical process
  b) The warehouse slot assigned for a particular product
  c) Whether a capital project is selected for investment
  d) The amount of money converted from yuan to dollars
  e) The number of aircraft produced on a defense contract

Constraints with Discrete Variables (exercise)

- In choosing among a collection of 16 investment projects, variables
  - $w_j = \begin{cases} 1, & \text{if project } j \text{ is selected} \\ 0, & otherwise \end{cases}$
- Express each of the following constraints in terms of these variables.
  a) At least one of the first eight projects must be selected.
  b) At most three of the last eight projects can be selected.
  c) Either project 4 or project 9 must be selected, but not both.
  d) Project 11 can be selected only if project 2 is also selected.

Solving a mathematical program

- $$\begin{aligned} \min \quad & 2.0x_1 + 1.5x_2 \quad \text{(cost)} \\ s.t. \quad & 0.3x_1 + 0.4x_2 \geq 2 \quad \text{(protein)} \\ & 0.4x_1 + 0.2x_2 \geq 1.5 \quad \text{(calorie)} \\ & 0.2x_1 + 0.3x_2 \geq 0.5 \quad \text{(vitamin)} \\ & x_1 \leq 9 \quad \text{(yogut availability)} \\ & x_2 \leq 6 \quad \text{(carrot availability)} \\ & x_1, x_2 \geq 0 \end{aligned}$$
- Enter the model into a modeling system, e.g., GAMS, AMPL, OPL



Then, run the program.

Solving an optimization model

The diet planning problem is an instance of a class of problems called **production planning problem**.

- Make a set of products using a set of resources
- Maximize utility or minimize production cost
- Subject to resource availability constraints and product demand requirements

		Product			Availability	Cost	Resource Consumed
		Protein	Calerie	Vitamin			
Resource	Yogurt	0.3	0.4	0.2	9	2	
	Carrot	0.4	0.2	0.3	6	1.5	
Demand		2	1.5	0.5			
Revenue		0	0	0			
Products Made							

In this example, resources are foods, products are nutrients.
Many other scenarios fall into this paradigm.

Formulate a problem in GAMS

		Product			Availability	Cost	Resource Consumed
		Protein	Calerie	Vitamin			
Resource	Yogurt	0.3	0.4	0.2	9	2	
	Carrot	0.4	0.2	0.3	6	1.5	
Demand		2	1.5	0.5			
Revenue		0	0	0			
Products Made							

```
set food /yogurt, carrot/;
set nutrient /protein, calerie, vitamin/;
alias(food,i);
alias(nutrient,j);
parameter
    price(i) /yogurt 2, carrot 1.5/
    demand(j) /protein 2, calerie 1.5, vitamin 0.5/
    available(i) /yogurt 9, carrot 6/;
table body(i,j)
        protein     calerie     vitamin
yogurt  0.3         0.4         0.2
carrot  0.4         0.2         0.3
;
positive variable eat(i);
variable cost;
equation obj, meetDemand(j), obeyResource(i);
obj.. cost =e= sum(i, price(i)*eat(i));
meetDemand(j).. sum(i, body(i,j)*eat(i)) =g= demand(j);
obeyResource(i).. sum(j, body(i,j)*eat(i)) =l= available(i);
model m2 /obj, meetDemand, obeyResource/;
solve m2 min cost using lp;
```