OMIS 6000

Week 4:

- Linearization of nonlinear functions (absolute values, either-or, products)
- Lagrange multipliers and the Karush-Kuhn-Tucker (KKT) conditions.
- Utility maximization, price optimization, and Ridge regression.

Adam Diamant, Schulich School of Business, Winter 2025

Linear Programs

- If both the objective function and all the constraints are **linear** functions of the decision variables, the problem is a **linear program (LP).**
- Although it seems like this assumption is somewhat restrictive, there are a surprising number of problems that can be solved.
  - Absolute value, minimax or maximin, minimizing the sum of deviations, floor/ceiling constraints.
- **Key Insight:** Models must be *reformulated*!

Linear Reformulations of Nonlinear Programs

Craft Beer Distribution

Craft Beer Distribution

Church-Key Brewing Company® has two warehouses from which it distributes beer to seven bars. At the start of each week, the bars send their orders (in cases) to the brewery. They brewery then satisfies the orders by assigning cases from the two warehouses to each of the bars. The brewery would like to formulate an optimization problem to find an assignment that minimizes fuel costs.

For example, this week the brewery has 7000 cases at warehouse A and 8000 cases at warehouse B. The bars require 1000, 1800, 3600, 400, 1400, 2500, and 2000 cases, respectively. Which warehouse should supply each bar given that the number of cases supplied by one warehouse must be within 1200 cases of the other?

Craft Beer Distribution

The total cost per case (c_{ij}):

Bar/Warehouse	A	B
1	$2.00	$3.00
2	$4.00	$1.00
3	$5.00	$3.00
4	$2.00	$2.00
5	$1.00	$3.00
6	$2.50	$1.75
7	$1.90	$1.60

Craft Beer Distribution

Define the objective

Minimize the total cost

Define the decision variables

Craft Beer Distribution

Define the objective

Minimize the total cost

Define the decision variables

$\boldsymbol{x_{ij}}$ ***= the number of cases sent from warehouse*** $\boldsymbol{i = \{1,2\} = \{A,B\}}$ ***to bar j*** $\boldsymbol{= 1, \ldots, 7}$***.***

Craft Beer Distribution

Write the mathematical objective function

$$\textbf{Minimize } Z = \sum_{i=1}^{2} \sum_{j=1}^{7} c_{ij} x_{ij}$$

Let c_{ij} be the cost of stocking bar j from warehouse i (see table).

Craft Beer Distribution

Formulating the constraints

There are three types of constraints:

1. Demand constraints
2. Supply constraints
3. Absolute value constraint
4. Non-negativity and integer constraints

Craft Beer Distribution

Formulating the demand constraint

How do you ensure that each bar receives exactly the number of cases ordered?

$$\sum_{i=1}^{2} x_{ij} = d_j \quad for\ each\ bar\ j$$

Let d_j be the demand of bar j.

Craft Beer Distribution

Formulating the supply constraint

How do you ensure that the number of cases supplied by each warehouse does not exceed their inventory levels?

$$\sum_{j=1}^{7} x_{ij} \leq s_i \quad for\ each\ warehouse\ i$$

Let s_i be the inventory level of warehouse i.

Craft Beer Distribution

Formulating the $|\cdot|$ constraint

How do you ensure that the number of cases supplied by one warehouse is within 1200 cases of the other warehouse?

Craft Beer Distribution

Formulating the | · | constraint

How do you ensure that the number of cases supplied by one warehouse is within 1200 cases of the other warehouse?

$$\left| \sum_{j=1}^{7} x_{1j} - \sum_{j=1}^{7} x_{2j} \right| \leq 1200$$

Craft Beer Distribution

Formulating the $|\cdot|$ constraint

How do you ensure that the number of cases supplied by one warehouse is within 1200 cases of the other warehouse?

$$\sum_{j=1}^{7} x_{1j} - \sum_{j=1}^{7} x_{2j} \leq 1200$$

$$\sum_{j=1}^{7} x_{2j} - \sum_{j=1}^{7} x_{1j} \leq 1200$$

Craft Beer Distribution

Minimize $\boldsymbol{Z} = \sum_{i=1}^{2} \sum_{j=1}^{7} c_{ij} x_{ij}$

Subject to:

$\sum_{i=1}^{2} x_{ij} = d_j \quad for\ each\ bar\ j$ (Demand constraints)

$\sum_{j=1}^{7} x_{ij} \leq s_i \quad for\ each\ warehouse\ i$ (Supply constraints)

$\sum_{j=1}^{7} x_{1j} - \sum_{j=1}^{7} x_{2j} \leq 1200$ (Absolute value constraint #1)

$\sum_{j=1}^{7} x_{2j} - \sum_{j=1}^{7} x_{1j} \leq 1200$ (Absolute value constraint #2)

$x_{ij} \geq 0 \quad for\ all\ i\ and\ j$ (Non-negativity constraints)

$x_{ij} \in Integers \quad for\ all\ i\ and\ j$ (Integrality constraints)

Let d_j be the demand of bar j and s_i be the inventory level of warehouse i.

Craft Beer Distribution: Python Solution

- The absolute value constraint, in the original formulation, is nonlinear. Thus, the original optimization problem is **nonlinear**.
  - We **reformulate** the problem into a linear form by expressing the absolute value constraint as two separate constraints. This feasible region and objective function do not change. A similar approach was used when we **reformulated** ratio constraints.
  - What happens if you use np.absolute() or abs() instead?

What managerial intuition do you get from the Python solution?

Other Linearization Rules

Suppose that $x, y \in \{0,1\}$ and $u, w \in [0, A]$.

1. Let $z = xy \in \{0,1\}$. Add constraints:

$$z \leq x,\ \ z \leq y,\ \ z \geq x + y - 1$$

2. Let $z = wx \in [0, A]$. Add constraints:

$$z \leq w,\ \ z \leq Ax,\ \ z \geq y + A(x - 1)$$

3. Let $z = \max\{u, w\} \in [0, A]$. The problem is:

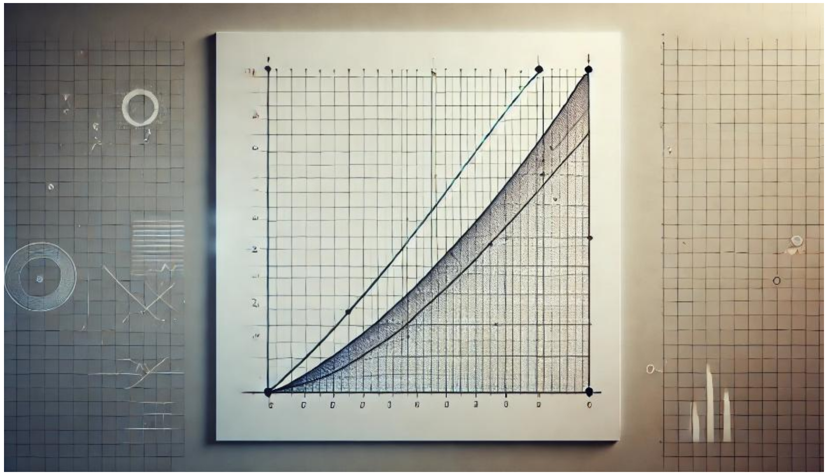
$$min\ z \text{ s.t. } z \geq u,\ \ z \geq w$$

4. Let $z = \min\{u, w\} \in [0, A]$. The problem is:

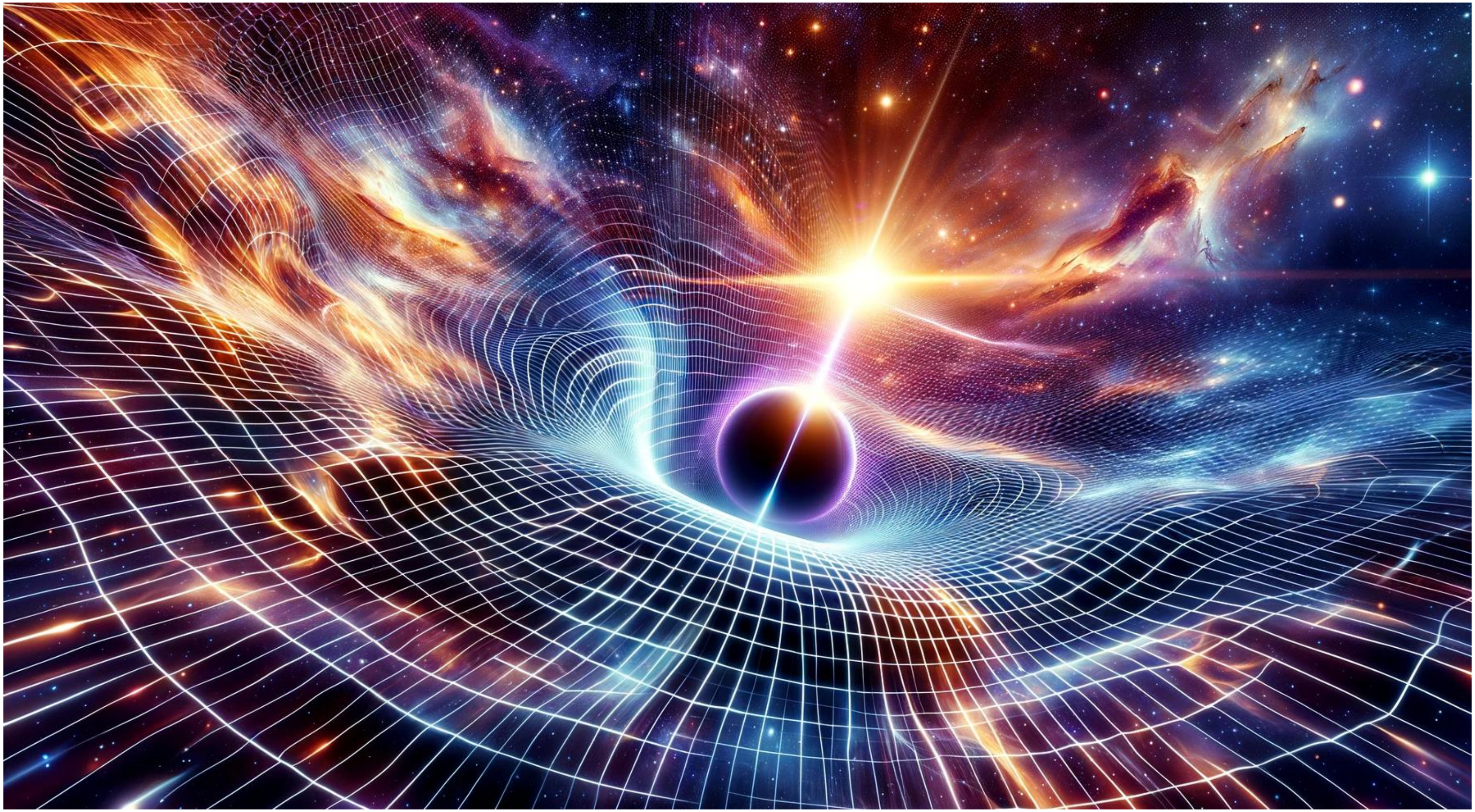
$$max\ z \text{ s.t. } z \leq u,\ \ z \leq w$$

Why Linear Programs?

- Typically, much faster and easier to solve.
- Provides a certificate of optimality, infeasibility, or unboundedness.
- All linear programs have global optima.
- Solutions are much more interpretable.



Nonlinear Optimization



What are *non*-linear functions?

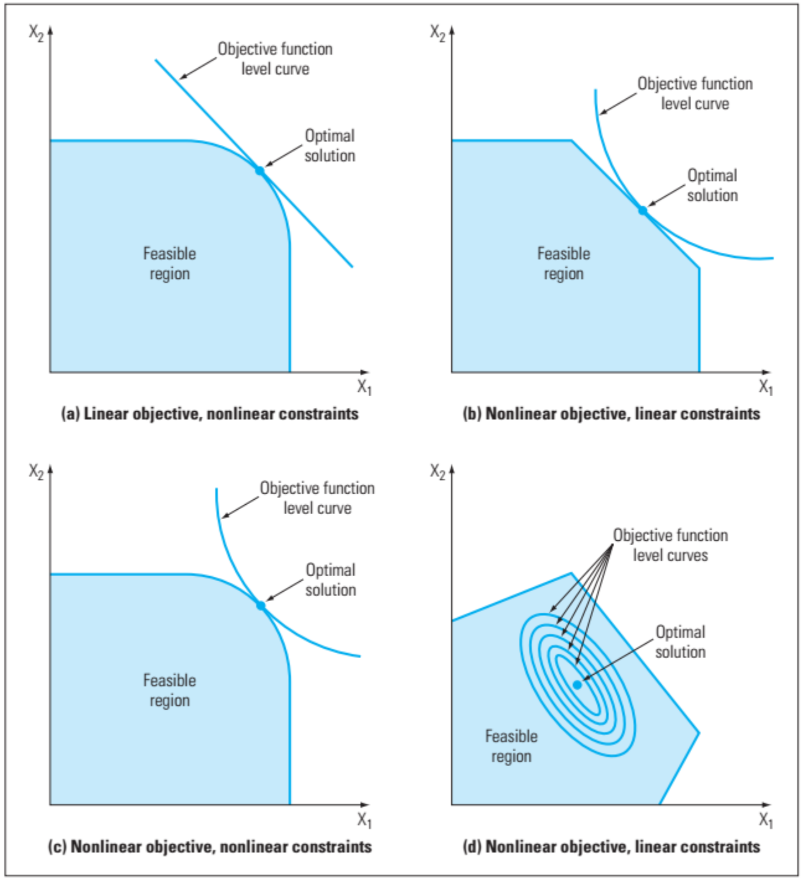
- $F(x) = ax^2 + bx + c$
- $F(x) = Max(x, 0)$
- $G(x, y) = xy$
- $G(x, y) = Max(x, 0) + Min(y + 300{,}0)$
- $G(x, y) = Min(x, 0)/Max(y - 150.46{,}1.29)$
- $G(x, y) = (x - a)^2 + (y - b)^2$
- $H(x, y, z) = xyz$
- $H(x, y, z) = \exp(x^2) + \sin(y) - \tan(z)$
- $H(x, y, z) = \exp(x^2)\sin(y) - az^4$

What is nonlinear programming (NLP)?

- Up to this point, we have assumed that the objective function and the constraints were all ***linear*** functions of the decision variables.
  - **Gurobi** is guaranteed to find the optimal solution. **Why?**

- What happens if either the objective function or the constraints are ***nonlinear***?
  - The problem formulation is almost identical.
  - The computational complexity is much harder!
  - **Gurobi** may return a *suboptimal* solution. **Why?**

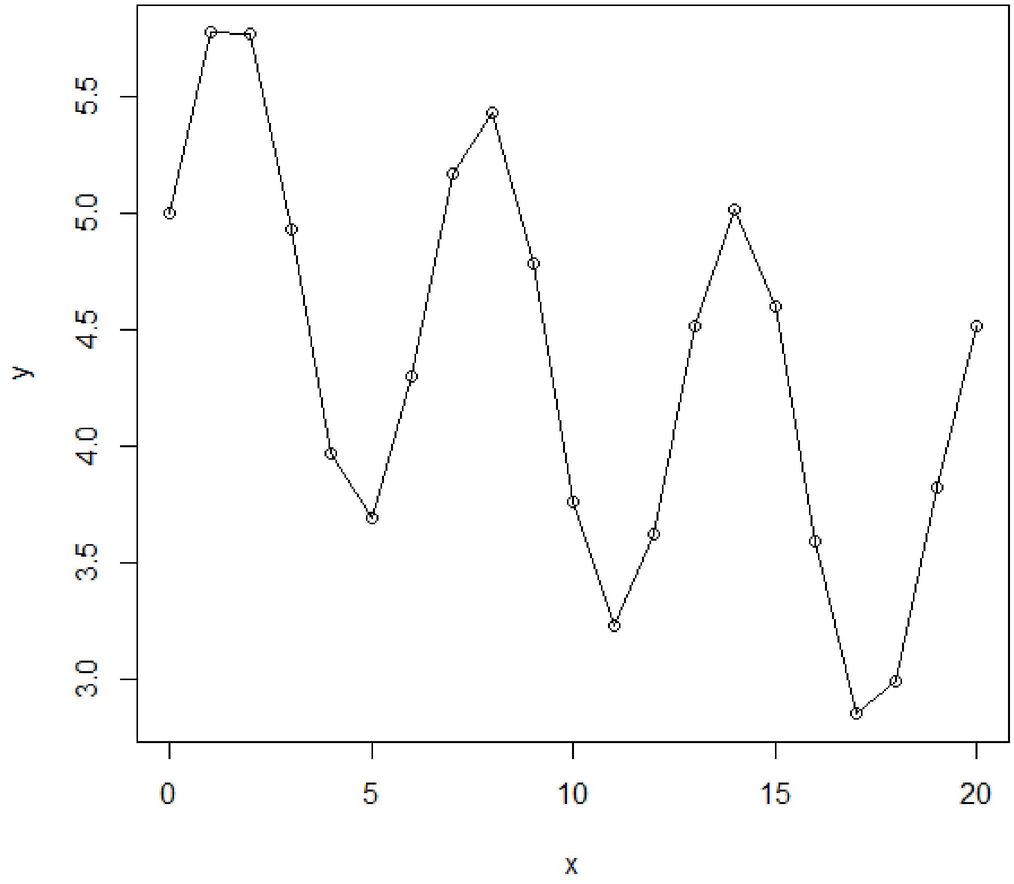
Why are NLPs difficult to solve?

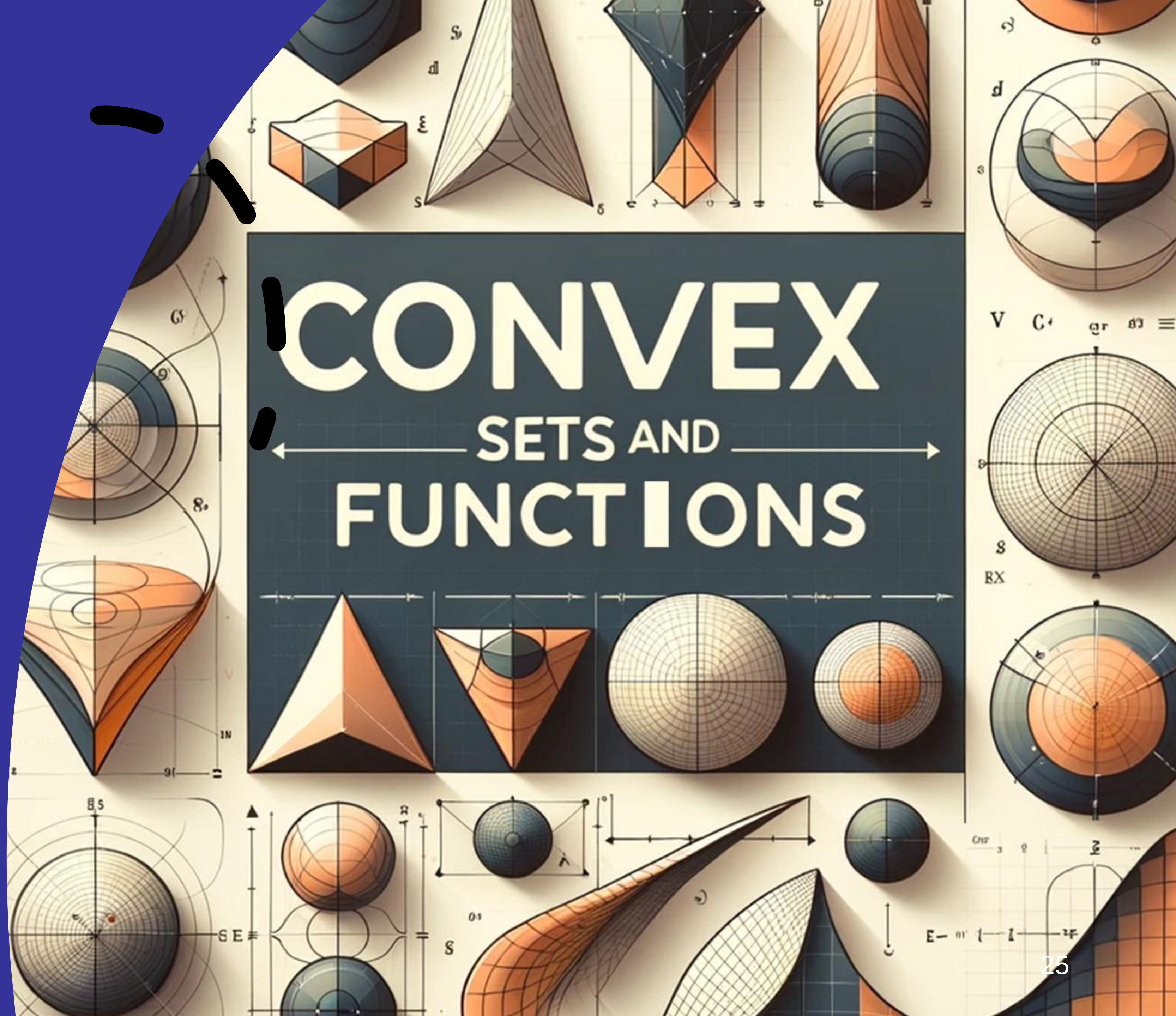


The optimal solution to some NLP problems might not occur on the boundary (i.e., at a **corner point**) of the feasible region like in an LP, but at some point in the *interior* of the feasible region.

Why are NLPs difficult to solve?

Gurobi may find a local optimum and not a global one. Local optima are *feasible solutions* but they may not be the best feasible solution.

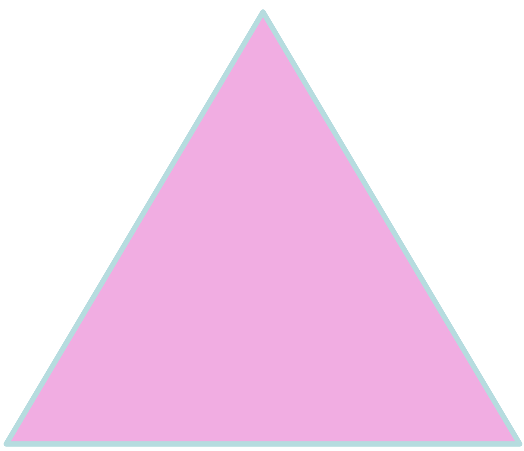



CONVEX
SETS AND
FUNCTIONS

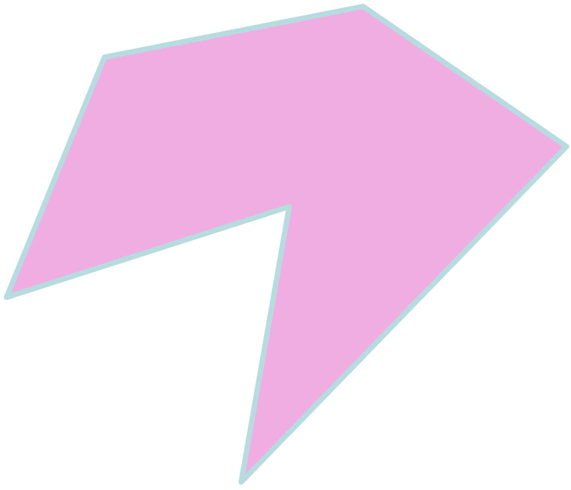
Convex Sets

A set of points $\mathcal{C} \subseteq \mathbb{R}^n$ is **convex** if, for every pair of points $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{C}$, any point on the line segment between $\boldsymbol{x}$ and $\boldsymbol{y}$ is also in $\mathcal{C}$.

- If $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{C}$, $t\boldsymbol{x} + (1-t)\boldsymbol{y} \in \mathcal{C}$ for all $t \in [0,1]$.



Convex Set



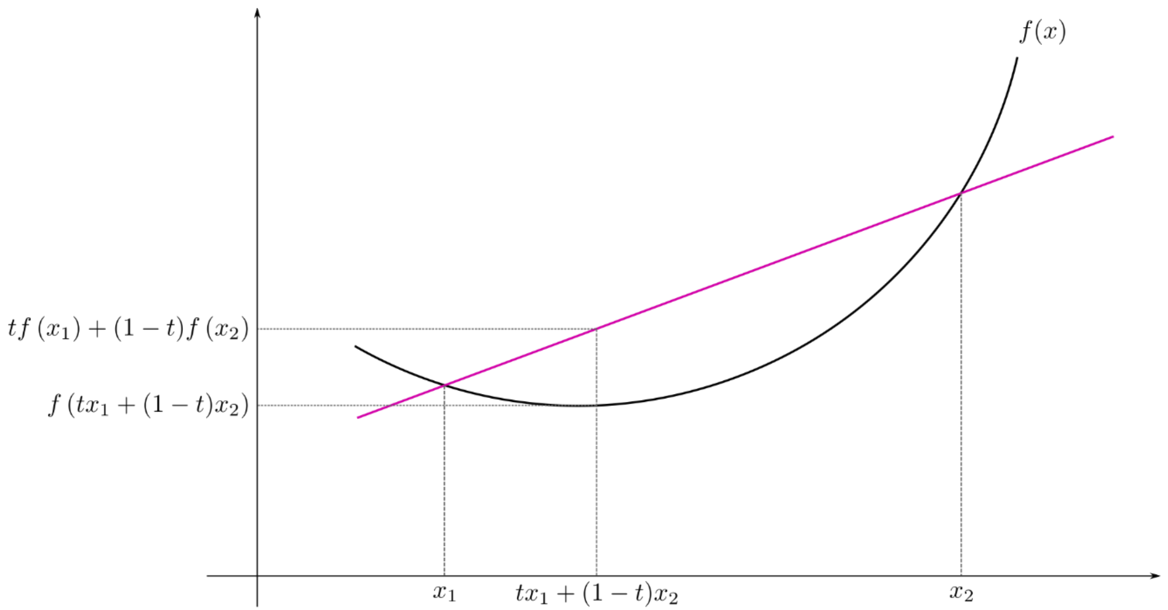
Not Convex

Convex Functions

A function $f(\boldsymbol{x})$ is **convex** if $\mathcal{C}$ is a convex set and

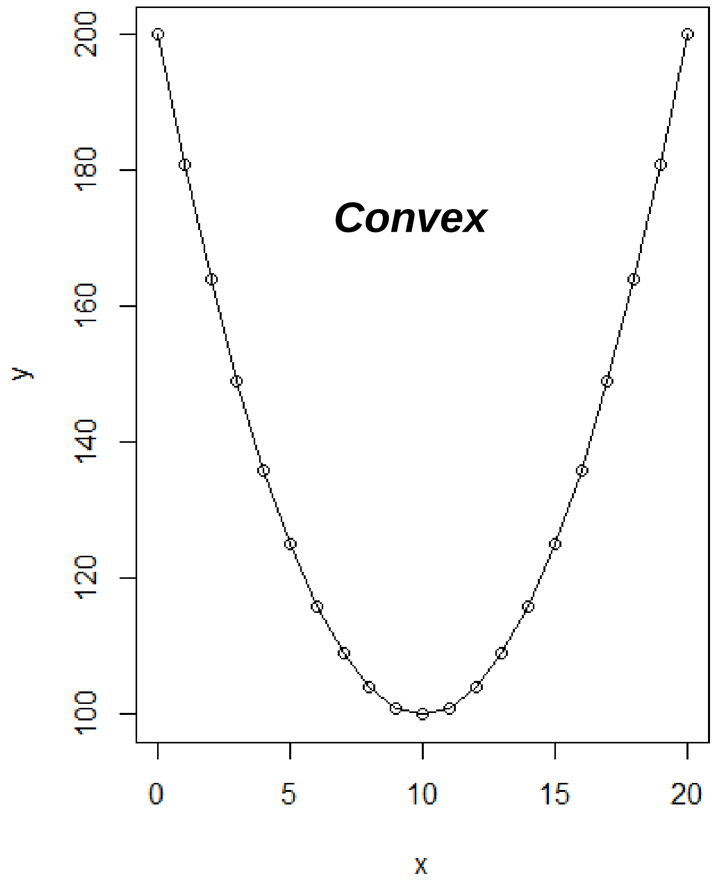
$$f(t\boldsymbol{x_1} + (1-t)\boldsymbol{x_2}) \leq tf(\boldsymbol{x_1}) + (1-t)f(\boldsymbol{x_2})$$

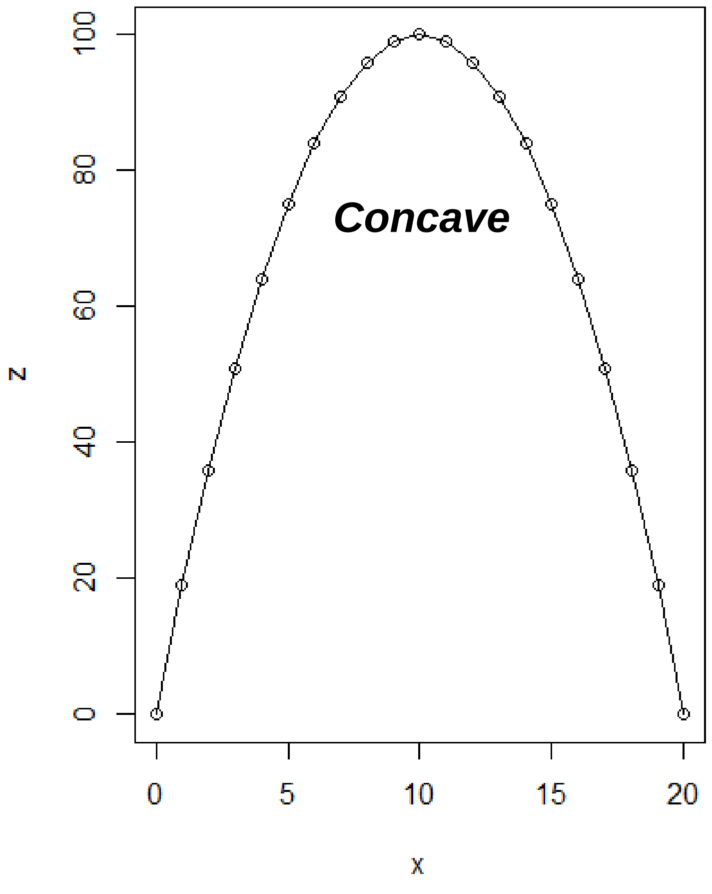
for all vectors $\boldsymbol{x_1}, \boldsymbol{x_2} \in \mathcal{C}$ and $t \in [0,1]$.



Convex Functions

For convex and concave functions, local extrema are global extrema. This ensures that optimal solutions can be found.





When can you guarantee that an NLP solution is optimal?

- For maximization problems:
  - The objective function is **concave**.
- For minimization problems:
  - The objective function is **convex**.
- The constraint set is convex:
  - A set of linear inequalities $\boldsymbol{Ax} \leq \boldsymbol{b}$.
  - Quadratic constraints such as $\boldsymbol{x^T A x} + \boldsymbol{Bx} \leq \boldsymbol{b}$.

Lagrangian Duality

Lagrangian Duality

Consider the following, possibly nonlinear, optimization problem where $\boldsymbol{g}: \mathbb{R}^n \to \mathbb{R}^m$, $\boldsymbol{h}: \mathbb{R}^n \to \mathbb{R}^k$ and $f: \mathbb{R}^n \to \mathbb{R}$ for $\boldsymbol{x} \in \mathbb{R}^n$.

$$z^* = \min_{\boldsymbol{x} \geq \boldsymbol{0}} f(\boldsymbol{x}) \;\; s.t.\, \boldsymbol{g}(\boldsymbol{x}) \leq \boldsymbol{0} \; and \; \boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{0}\,.$$

We call this model formulation **standard form**. All constrained optimization problems (linear/nonlinear) can be written like this.

Lagrangian Duality

Consider the following, possibly nonlinear, optimization problem where $\boldsymbol{g}: \mathbb{R}^n \to \mathbb{R}^m$, $\boldsymbol{h}: \mathbb{R}^n \to \mathbb{R}^k$ and $f: \mathbb{R}^n \to \mathbb{R}$ for $\boldsymbol{x} \in \mathbb{R}^n$. Introduce the variables $\boldsymbol{\lambda} \in \mathbb{R}_+^m$ and $\boldsymbol{\mu} \in \mathbb{R}^k$.

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\boldsymbol{x}) + \boldsymbol{\lambda}^T \boldsymbol{g}(\boldsymbol{x}) + \boldsymbol{\mu}^T \boldsymbol{h}(\boldsymbol{x})$$

We call $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu})$ the **Lagrangian** and the variables $(\boldsymbol{\lambda}, \boldsymbol{\mu})$ **Dual**/**Lagrange Multipliers**.

Lagrangian Duality

Consider the following, possibly nonlinear, optimization problem where $\boldsymbol{g}: \mathbb{R}^n \to \mathbb{R}^m$, $\boldsymbol{h}: \mathbb{R}^n \to \mathbb{R}^k$ and $f: \mathbb{R}^n \to \mathbb{R}$ for $\boldsymbol{x} \in \mathbb{R}^n$. Introduce the variables $\boldsymbol{\lambda} \in \mathbb{R}^m$ and $\boldsymbol{\mu} \in \mathbb{R}^k$.

$$\mathfrak{D}(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \min_{\boldsymbol{x} \geq \boldsymbol{0}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu})$$

This is the **Lagrangian dual function** which is an unconstrained optimization problem.

Weak Lagrangian Duality

Theorem: For any feasible solution $\boldsymbol{x}$, a solution to the **Lagrangian dual function** for any $\boldsymbol{\lambda} \geq \mathbf{0}$ and $\boldsymbol{\mu} \in \mathbb{R}^k$ gives the bound:

$$z^* \geq \mathfrak{D}(\boldsymbol{\lambda}, \boldsymbol{\mu})$$

Proof: In-Class

Weak Lagrangian Duality

Theorem: For any feasible solution $\boldsymbol{x}$, a solution to the **Lagrangian dual function** for any $\boldsymbol{\lambda} \geq \mathbf{0}$ and $\boldsymbol{\mu} \in \mathbb{R}^k$ gives the bound:

$$z^* \geq \mathfrak{D}(\boldsymbol{\lambda}, \boldsymbol{\mu})$$

This relationship holds for all nonlinear optimization problems (e.g., non-convex).

Weak Lagrangian Duality

To find the best lower bound $z^* \geq d^*$, we can solve the following maximin problem:

$$d^* = \max_{\boldsymbol{\lambda} \geq 0,\, \boldsymbol{\mu} \in \mathbb{R}^k} \mathfrak{D}(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \max_{\boldsymbol{\lambda} \geq 0,\, \boldsymbol{\mu} \in \mathbb{R}^k} \min_{\boldsymbol{x} \geq \boldsymbol{0}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu})$$

The difference, $z^* - d^*$, is known as the **duality gap** which always exist unless the original optimization problem is **convex** and certain **regularity conditions** are satisfied.

Strong Lagrangian Duality

Theorem: If the objective and constraints of an optimization problem are convex and $\boldsymbol{\lambda}^T \boldsymbol{g}(\boldsymbol{x}) = \mathbf{0}$, then the **duality gap** is zero.

$$z^* = d^*$$

Conditions guaranteeing that strong duality holds are called constraint qualifications. In this case, we enforce $\boldsymbol{\lambda}^T \boldsymbol{g}(\boldsymbol{x}) = \mathbf{0}$ which ensures the complementary slackness condition holds. The proof of this result relies on analysing the max-min inequality.

Applying Strong Duality

Applying Strong Duality

First Derivative: The slope at a point.

$$\left.\frac{\partial f(x)}{\partial x}\right|_{x_1} = f'(x_1) = \lim_{h \to 0} \frac{f(x_1 + h) - f(x_1)}{h}$$

First Derivative (Gradient): The slope at a point.

$$\nabla f(\boldsymbol{x}) = \left(\frac{\partial f(\boldsymbol{x})}{\partial x_1}, \frac{\partial f(\boldsymbol{x})}{\partial x_2}, \dots, \frac{\partial f(\boldsymbol{x})}{\partial x_m}\right)$$

Applying Strong Duality

When solving convex optimization problems, how can we determine whether the optimal solution that is obtained is globally optimal?

Property 1: For convex functions, a global optimum occurs when $\nabla f(\boldsymbol{x}) = \boldsymbol{0}$.

Proof: This follows from multivariable calculus.

Applying Strong Duality

When solving convex optimization problems, how can we determine whether the optimal solution that is obtained is globally optimal?

Implication: For constrained optimization problems, if the problem is classified as convex, then a global optimum occurs when:

$$\nabla f(\boldsymbol{x}) + \boldsymbol{\lambda}^T \nabla \boldsymbol{g}(\boldsymbol{x}) + \boldsymbol{\mu}^T \nabla \boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{0}$$

Applying Strong Duality

When solving convex optimization problems, how can we determine whether the optimal solution that is obtained is globally optimal?

Property 2: At the optimal solution $(\boldsymbol{x}^*)$, $z^* = f(\boldsymbol{x}^*) = \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$, implying ${\boldsymbol{\lambda}^*}^T \boldsymbol{g}(\boldsymbol{x}^*) = \boldsymbol{0}$. Thus, nonbinding constraints must have their **shadow prices** $(\boldsymbol{\lambda}^*)$ be equal zero.

Proof: $\mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = f(\boldsymbol{x}^*) + {\boldsymbol{\lambda}^*}^T \boldsymbol{g}(\boldsymbol{x}^*) + {\boldsymbol{\mu}^*}^T \boldsymbol{h}(\boldsymbol{x}^*)$

Applying Strong Duality

When solving convex optimization problems, how can we determine whether the optimal solution that is obtained is globally optimal?

Implication: By enforcing that the constraint $\boldsymbol{\lambda}^T \boldsymbol{g}(\boldsymbol{x}) = \mathbf{0}$ holds with equality and noting that feasible solutions should also have $\boldsymbol{\mu}^T \boldsymbol{h}(\boldsymbol{x}) = \mathbf{0}$, we are guaranteed to find a solution (if one exists) that is *feasible* to the original problem.

KKT Conditions

Optimality condition (stationarity):

$$\nabla f(\boldsymbol{x}) + \boldsymbol{\lambda}^T \nabla \boldsymbol{g}(\boldsymbol{x}) + \boldsymbol{\mu}^T \nabla \boldsymbol{h}(\boldsymbol{x}) = \mathbf{0}$$

Complementary Slackness:

$$\boldsymbol{\lambda}^T \boldsymbol{g}(\boldsymbol{x}) = \mathbf{0}$$

Primal Feasibility:

$$\boldsymbol{g}(\boldsymbol{x}) \leq \mathbf{0}, \boldsymbol{h}(\boldsymbol{x}) = \mathbf{0}$$

Dual Feasibility:

$$\boldsymbol{\lambda} \geq \mathbf{0}, \boldsymbol{\mu} \in \mathbb{R}^k$$

These conditions are necessary and sufficient for optimality.

KKT Conditions

Optimality condition (stationarity):

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{x}} = \nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = 0$$

Complementary Slackness:

$$\boldsymbol{\lambda}^T \frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}} = \nabla_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \mathbf{0}$$

Primal Feasibility:

$$\boldsymbol{g}(\boldsymbol{x}) \leq \mathbf{0}, \boldsymbol{h}(\boldsymbol{x}) = \mathbf{0}$$

Dual Feasibility:

$$\boldsymbol{\lambda} \geq \mathbf{0}, \boldsymbol{\mu} \in \mathbb{R}^k$$

KKT Conditions

Utility Maximization

Price Optimization

Regularized Regression

Utility Maximization

Intertemporal Consumption

Rohit just turned 16 and inherited a substantial sum of money. In particular, he will get an endowment W_1 immediately and will get another endowment W_2 after turning 35. Since many big purchases occur before this age (e.g., house purchase, wedding, childcare), Rohit can borrow against the second endowment or lend using the first endowment, at yearly interest rate of r. Rohit is risk-averse and therefore, his marginal utility decreases the more he uses his endowment (e.g., assume logarithmic). His objective is to maximize the weighted sum ($\alpha_1 + \alpha_2 = 1$) of his overall utility over his lifetime while adhering to an intertemporal budget constraint, which guides his spending and investment decisions x_1 and x_2 across the two periods.

Intertemporal Consumption

Define the objective

Maximize the total utility

Define the decision variables

$\boldsymbol{x_t}$ ***= the amount spent in period*** $\boldsymbol{t = \{1, 2\}}$

Intertemporal Consumption

Write the mathematical objective function

$$\textbf{Maximize } \boldsymbol{Z} = \alpha_1 \ln x_1 + \alpha_2 \ln x_2$$

where we assume $\alpha_1 + \alpha_2 = 1$ *represents the intertemporal weights.*

Intertemporal Consumption

Write the mathematical objective function

$$\textbf{Maximize } \boldsymbol{Z} = \alpha_1 \ln x_1 + (1 - \alpha_1) \ln x_2$$

where we assume $\alpha_1 + \alpha_2 = 1$ *represents the intertemporal weights.*

Intertemporal Consumption

Formulating the constraints

There are two types of constraints:

1. Budget constraint
2. Non-negativity constraints

Intertemporal Consumption

Formulating the budget constraint

How do you ensure that the budget constraint is satisfied given Rohit can borrow against his future endowment?

$$x_1 + x_2\left(\frac{1}{1+r}\right)^{19} = W_1 + W_2\left(\frac{1}{1+r}\right)^{19}$$

Intertemporal Consumption

$$\textbf{Maximize } \boldsymbol{Z} = \alpha_1 \ln x_1 + (1-\alpha_1)\ln x_2 \quad s.t.$$

$$x_1 + x_2\left(\frac{1}{1+r}\right)^{19} = W_1 + W_2\left(\frac{1}{1+r}\right)^{19}$$

$$x_1 \geq 0, \quad x_2 \geq 0$$

Step 1: Write down the Lagrangian Function

$$\mathcal{L}(\boldsymbol{x},\lambda) = \alpha_1 \ln x_1 + (1-\alpha_1)\ln x_2 + \lambda\left(W_1 + W_2\left(\frac{1}{1+r}\right)^{19} - x_1 - x_2\left(\frac{1}{1+r}\right)^{19}\right)$$

Note that for simplicity, <u>in this case only</u>, I am omitting the non-negativity constraints as they *do not* affect the optimal solution.

Intertemporal Consumption

$$\textbf{Maximize } \boldsymbol{Z} = \alpha_1 \ln x_1 + (1 - \alpha_1) \ln x_2 \quad s.t.$$

$$x_1 + x_2 \left(\frac{1}{1+r}\right)^{19} = W_1 + W_2 \left(\frac{1}{1+r}\right)^{19}$$

$$x_1 \geq 0, \quad x_2 \geq 0$$

Step 2: Derive the optimality conditions

$$\frac{\partial \mathcal{L}(\boldsymbol{x}, \lambda)}{\partial x_1} = \frac{\alpha_1}{x_1} - \lambda = 0$$

$$\frac{\partial \mathcal{L}(\boldsymbol{x}, \lambda)}{\partial x_2} = \frac{1 - \alpha_1}{x_2} - \lambda \left(\frac{1}{1+r}\right)^{19} = 0$$

$$h(x_1, x_2) = x_1 + x_2 \left(\frac{1}{1+r}\right)^{19} - W_1 - W_2 \left(\frac{1}{1+r}\right)^{19} = 0$$

Intertemporal Consumption

$$\textbf{Maximize } \boldsymbol{Z} = \alpha_1 \ln x_1 + (1-\alpha_1)\ln x_2 \quad s.t.$$

$$x_1 + x_2\left(\frac{1}{1+r}\right)^{19} = W_1 + W_2\left(\frac{1}{1+r}\right)^{19}$$

$$x_1 \geq 0, \quad x_2 \geq 0$$

Step 3: Solve the system (e.g., Wolfram Alpha).

$$x_1 = \left(W_1 + W_2\left(\frac{1}{1+r}\right)^{19}\right)\alpha_1$$

$$x_2 = \left((1+r)^{19}W_1 + W_2\right)(1-\alpha_1)$$

$$\lambda = \frac{1}{W_1 + W_2\left(\frac{1}{1+r}\right)^{19}}$$

Intertemporal Consumption

$$\textbf{Maximize } \boldsymbol{Z} = \alpha_1 \ln x_1 + (1-\alpha_1)\ln x_2 \quad s.t.$$

$$x_1 + x_2\left(\frac{1}{1+r}\right)^{19} = W_1 + W_2\left(\frac{1}{1+r}\right)^{19}$$

$$x_1 \geq 0, \quad x_2 \geq 0$$

Step 4: Ensure primal/dual feasibility is satisfied.

$$x_1 = \left(W_1 + W_2\left(\frac{1}{1+r}\right)^{19}\right)\alpha_1 \geq 0$$

$$x_2 = \left((1+r)^{19}W_1 + W_2\right)(1-\alpha_1) \geq 0$$

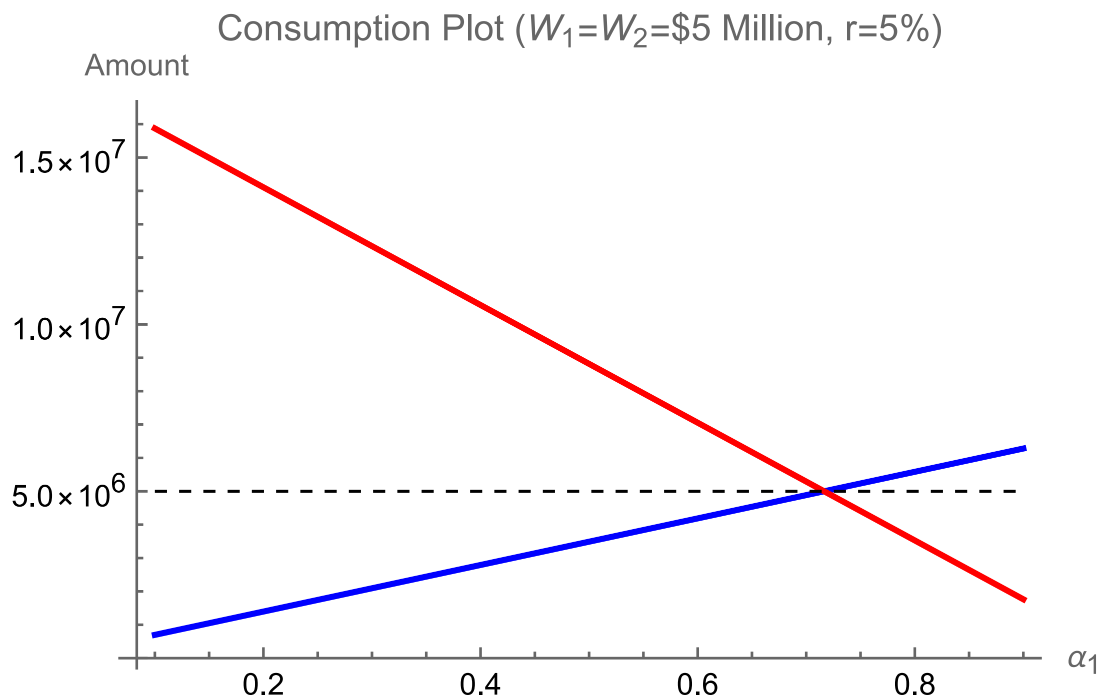
$$\lambda = \frac{1}{W_1 + W_2\left(\frac{1}{1+r}\right)^{19}} \geq 0$$

for any choice of parameters $W_1 \geq 0, W_2 \geq 0$, and $r \geq 0$.

Intertemporal Consumption

Should Rohit borrow/lend across the two periods?

– If $x_1 \geq W_1$, he is borrowing from the second endowment to fund his life before he turns 35. If $x_2 \geq W_2$, he is investing some of his first endowment to fund his life after 35.



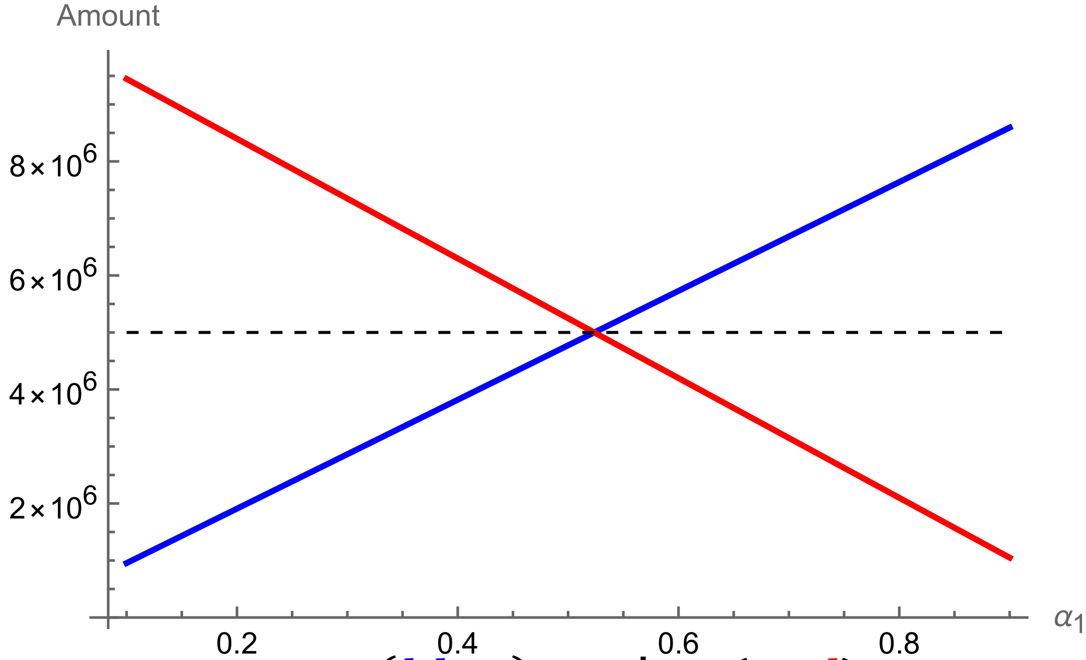
$x_1(\color{blue}{blue})$ and $x_2(\color{red}{red})$

Intertemporal Consumption

Should Rohit borrow/lend across the two periods?

- If $x_1 \geq W_1$, he is borrowing from the second endowment to fund his life before he turns 35. If $x_2 \geq W_2$, he is investing some of his first endowment to fund his life after 35.

Consumption Plot (W_1=W_2=\$5 Million, r=0.5%)

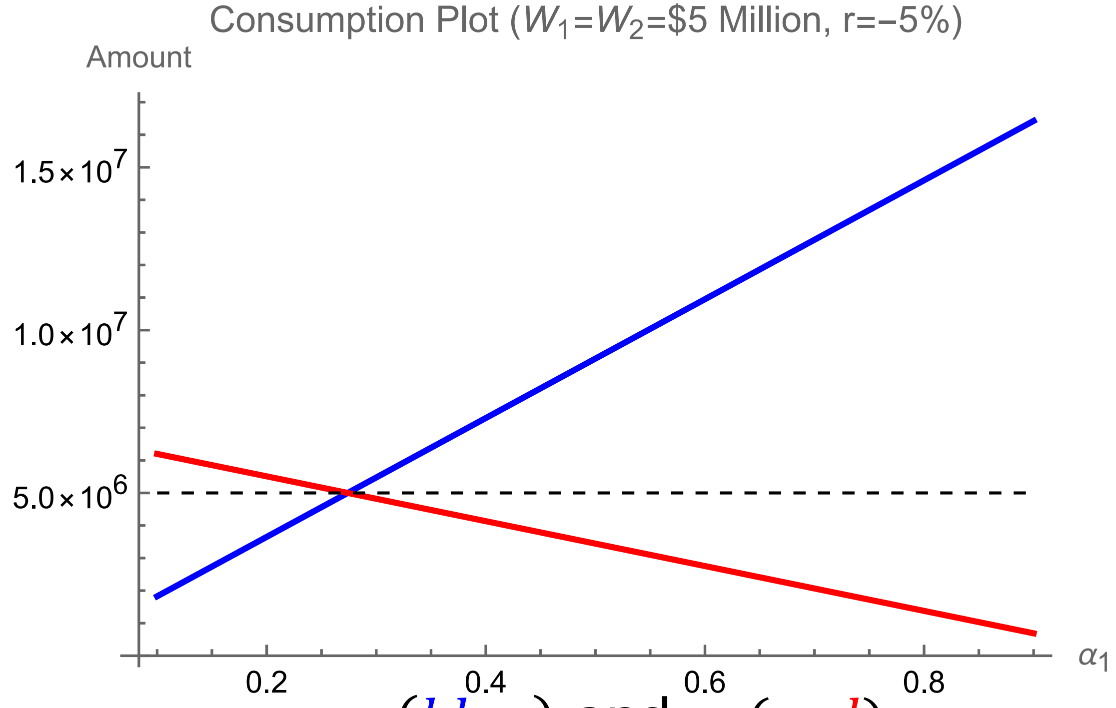


$x_1(\color{blue}{blue})$ and $x_2(\color{red}{red})$

Intertemporal Consumption

Should Rohit borrow/lend across the two periods?

- If $x_1 \geq W_1$, he is borrowing from the second endowment to fund his life before he turns 35. If $x_2 \geq W_2$, he is investing some of his first endowment to fund his life after 35.



x_1(*blue*) and x_2(*red*)

Price Optimization

Price Response Functions

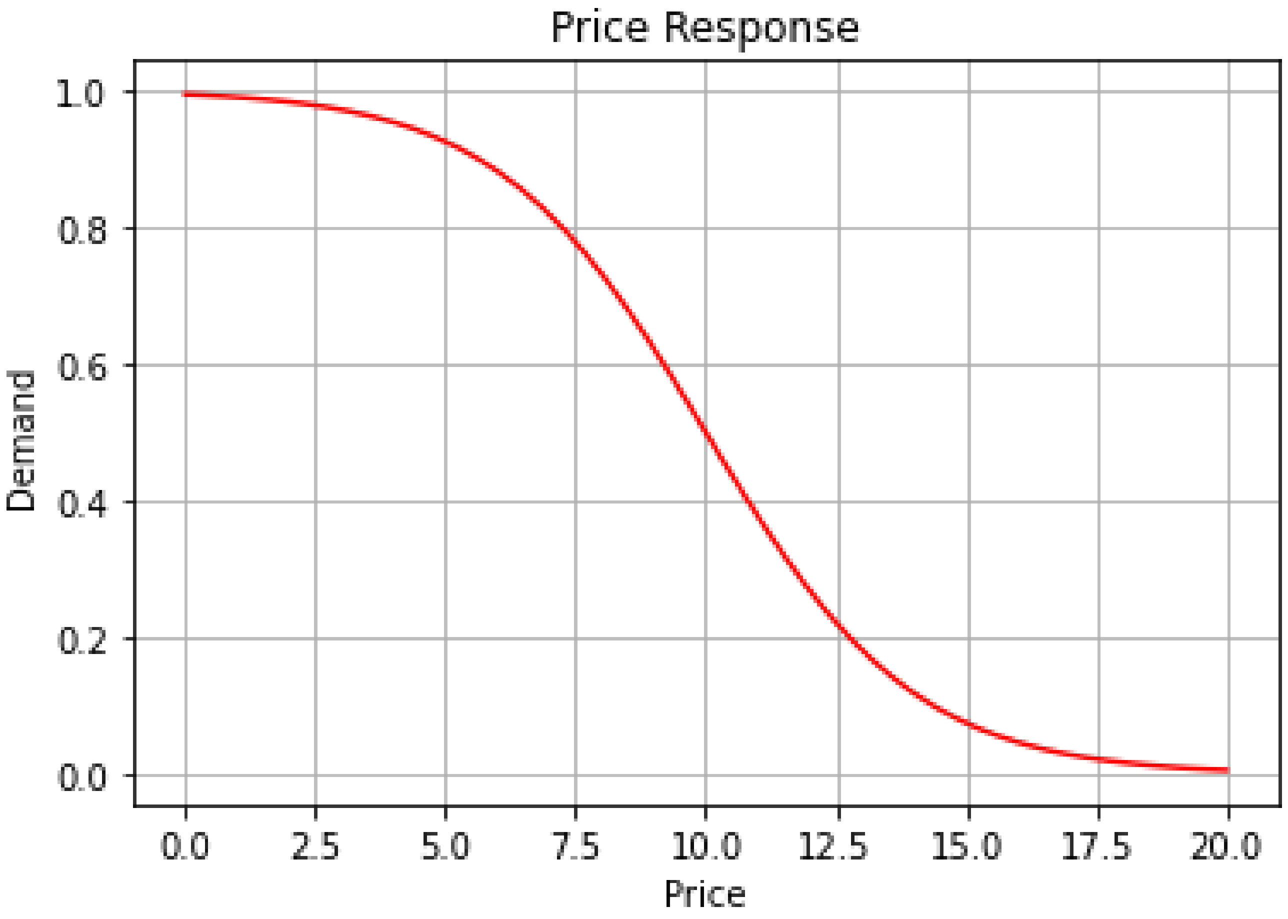
Describes the demand for a product $\boldsymbol{d(p)}$ as a function of a price vector $\boldsymbol{p} \geq \boldsymbol{0}$.

– Specifies how many more (less) customers would buy if prices were lowered or how many customers would leave (arrive) if prices were raised.

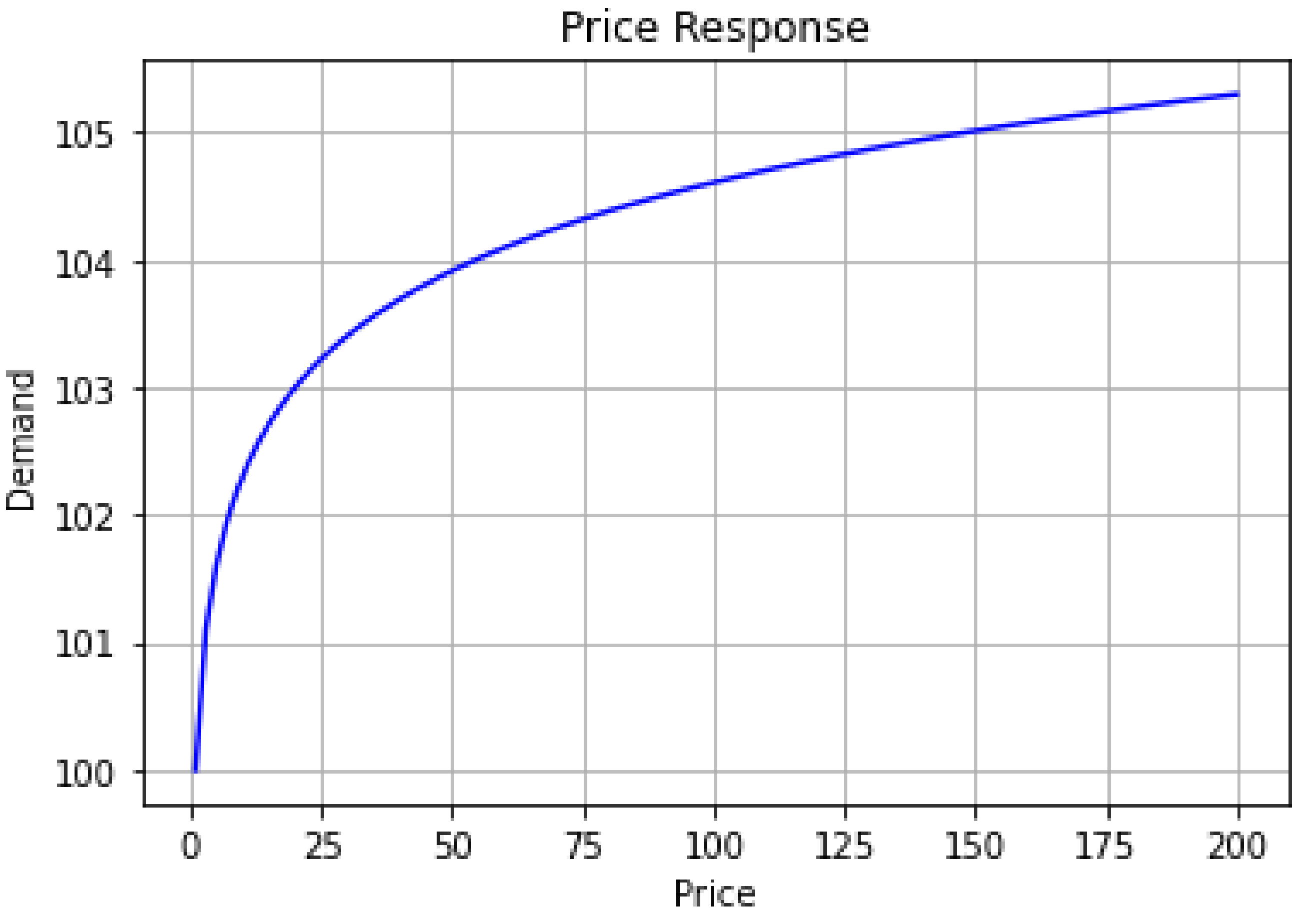
– Differs from the traditional market supply-demand curve from economics, which predicts how an entire industry/market will respond to price changes.

– Can account for contextual information such as the prices of other brand lines, competing products, and such as time, weather, promotions, etc.

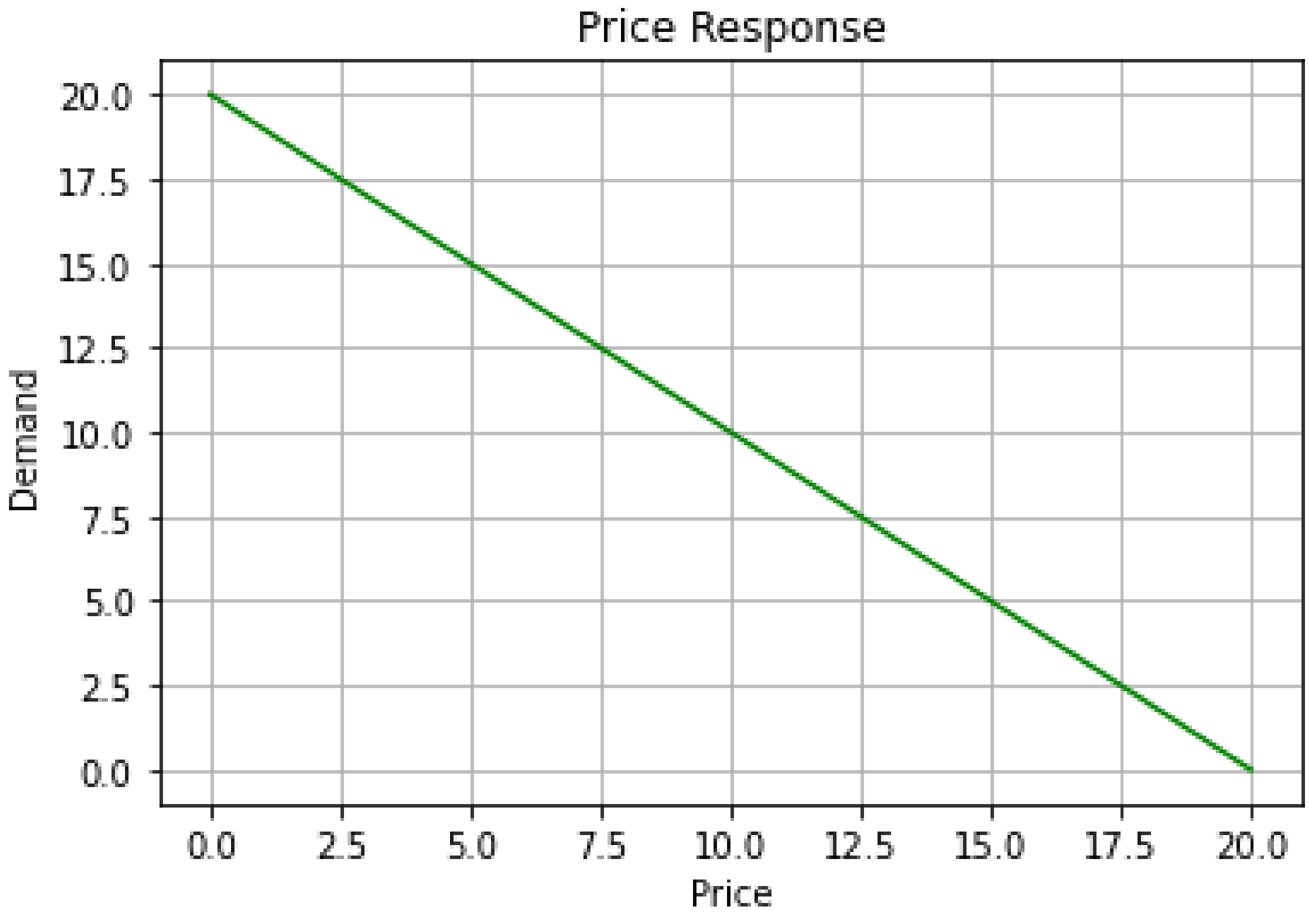
Price Response Functions



Price Response Functions



Price Response Functions



Price Optimization

TechEssentials Inc. is a rising star in the technology industry, known for its innovative but cost-conscious cell phone designs. The company offers two models: the InfiniteEdge and the Fusion Elite. TechEssentials' primary goal is to strategically price these products to maximize revenue. Both models have a linear relationship between price and demand – higher prices result in lower demand, and lower prices increase demand. Additionally, because the Fusion Elite is a more advanced model, its price must be at least $550 higher than that of the InfiniteEdge.

	Max Demand	**Slope**
InfiniteEdge	35234	26
Fusion Elite	27790	9

Price Optimization

There are three ways we can price these products:

1. Solve via graphical method:
   - The constraint set is linear and there are only two decision variables: p_1 the price for a InfiniteEdge cell phone and p_2 the price for a Fusion Elite cell phone. We would need new tools to deal with the objective.
2. Solve via Gurobi:
   - This is a quadratic programming problem.
3. Solve using the KKT conditions.
   - Let's do it!

	Max Demand	**Slope**
InfiniteEdge	35234	26
Fusion Elite	27790	9

Price Optimization

Define the objective

Maximize revenue

Define the decision variables

p_1: the price for a InfiniteEdge cell phone

p_2: the price for a Fusion Elite cell phone

	Max Demand	Slope
InfiniteEdge	35234	26
Fusion Elite	27790	9

Price Optimization

Define the objective

Maximize $p_1 d_1(p_1) + p_2 d_2(p_2)$

Define the decision variables

p_1: the price for a InfiniteEdge cell phone

p_2: the price for a Fusion Elite cell phone

	Max Demand	Slope
InfiniteEdge	35234	26
Fusion Elite	27790	9

Price Optimization

Define the objective

Maximize $p_1(35234 - 26p_1) + p_2(27790 - 9p_2)$

Define the decision variables

p_1: the price for a InfiniteEdge cell phone

p_2: the price for a Fusion Elite cell phone

	Max Demand	Slope
InfiniteEdge	35234	26
Fusion Elite	27790	9

Price Optimization

Formulating the constraints

There are three types of constraints:

1. Demand constraints
2. Pricing constraint
3. Non-negativity constraints

Price Optimization

Formulating the demand constraints

Demand cannot be negative for any price.

Price Optimization

Formulating the demand constraints

Demand cannot be negative for any price.

$$d_1(p_1) = 35234 - 26p_1 \geq 0$$

$$d_2(p_2) = 27790 - 9p_2 \geq 0$$

Price Optimization

Formulating the pricing constraint

Because the Fusion Elite is a more advanced model, its price must be at least $550 higher than that of the InfiniteEdge.

Price Optimization

Formulating the pricing constraint

Because the Fusion Elite is a more advanced model, its price must be at least \$550 higher than that of the InfiniteEdge.

$$p_2 \geq 550 + p_1$$

Price Optimization

Maximize $\mathbf{Z} = p_1(35234 - 26p_1) + p_2(27790 - 9p_2)$

Subject to:

$35234 - 26p_1 \geq 0$	(Demand constraint #1)
$27790 - 9p_2 \geq 0$	(Demand constraint #2)
$p_2 \geq 550 + p_1$	(Pricing constraint)
$p_1, p_2 \geq 0$	(Nonnegativity constraints)

If we solve using Gurobi, the optimal solution is:

$$p_1 = \$677.57, p_2 = \$1543.88$$

Let's confirm using the KKT conditions that this is true.

Price Optimization

Step 1: Write down the Lagrangian Function

$$\mathcal{L}(p_1, p_2, \lambda, \mu_1, \mu_2)$$

Price Optimization

Step 1: Write down the Lagrangian Function

$$\mathcal{L}(p_1, p_2, \lambda, \mu_1, \mu_2)$$
$$= p_1 (35234 - 26\, p_1) + p_2 (27790 - 9p_2)$$

Price Optimization

Step 1: Write down the Lagrangian Function

$$\mathcal{L}(p_1, p_2, \lambda, \mu_1, \mu_2)$$
$$= p_1(35234 - 26\,p_1) + p_2(27790 - 9p_2)$$
$$-\mu_1(26\,p_1 - 35234) - \mu_2(9p_2 - 27790)$$

Price Optimization

Step 1: Write down the Lagrangian Function

$$\mathcal{L}(p_1, p_2, \lambda, \mu_1, \mu_2)$$
$$= p_1(35234 - 26\,p_1) + p_2(27790 - 9p_2)$$
$$-\mu_1(26\,p_1 - 35234) - \mu_2(9p_2 - 27790)$$
$$-\lambda(p_1 - p_2 + 550)$$

Price Optimization

Step 2: Derive the optimality conditions

$$\frac{\partial \mathcal{L}}{\partial p_1} = 0 \rightarrow$$

$$\frac{\partial \mathcal{L}}{\partial p_2} = 0 \rightarrow$$

Price Optimization

Step 2: Derive the optimality conditions

$$\frac{\partial \mathcal{L}}{\partial p_1} = 0 \rightarrow p_1 = \frac{1}{52}\left(-\lambda - 26\mu_1 + 35234\right)$$

$$\frac{\partial \mathcal{L}}{\partial p_2} = 0 \rightarrow p_2 = \frac{1}{18}\left(\lambda - 9\mu_2 + 27790\right)$$

Price Optimization

Step 2: Derive the optimality conditions

$$\frac{\partial \mathcal{L}}{\partial p_1} = 0 \rightarrow p_1 = \frac{1}{52}\,(-\lambda - 26\mu_1 + 35234)$$

$$\frac{\partial \mathcal{L}}{\partial p_2} = 0 \rightarrow p_2 = \frac{1}{18}\,(\lambda - 9\mu_2 + 27790)$$

$$\mu_1(26\,p_1 - 35234) = 0$$

$$\mu_2(9p_2 - 27790) = 0$$

$$\lambda(\,p_1 - \,p_2 + 550) = 0$$

Price Optimization

Step 3: Solve the system

In this case, there are multiple solutions that satisfy this system of equations! They may not all be feasible…

$$z = \$20.27\ million$$
$$p_1 = \$1355.15$$
$$p_2 = \$1905.15$$
$$\lambda = 6502.77$$
$$\mu_1 = -1605.26$$
$$\mu_2 = 0$$

Price Optimization

Step 3: Solve the system

In this case, there are multiple solutions that satisfy this system of equations! They may not all be feasible…

$$z = \$32.72\ milion$$
$$p_1 = \$758.91$$
$$p_2 = \$1308.91$$
$$\lambda = -4229.54$$
$$\mu_1 = \mu_2 = 0$$

Price Optimization

Step 3: Solve the system

In this case, there are multiple solutions that satisfy this system of equations! They may not all be feasible…

$$z = \$33.39\ milion$$
$$p_1 = \$677.58$$
$$p_2 = \$1543.89$$
$$\lambda = \mu_1 = \mu_2 = 0$$

Price Optimization

Step 3: Solve the system

In this case, there are multiple solutions that satisfy this system of equations! They may not all be feasible…

$$z = \$21.45\ million$$
$$p_1 = \$1355.15$$
$$p_2 = \$1543.89$$
$$\mu_1 = -1355.15$$
$$\lambda = \mu_2 = 0$$

Price Optimization

Step 4: Ensure primal/dual feasibility.

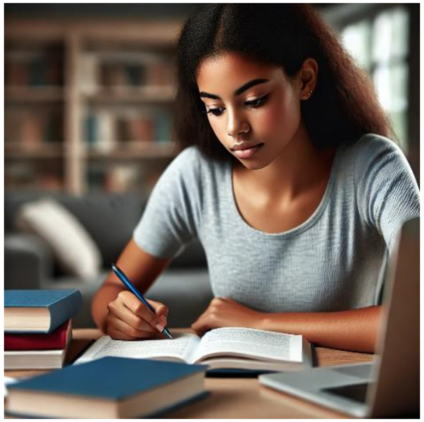
Not all solutions we obtain using this approach satisfy the above conditions. For instance…

$$z = \$21.45\,million$$
$$p_1 = \$1355.15$$
$$p_2 = \$1543.89$$
$$\mu_1 = -1355.15$$
$$\lambda = \mu_2 = 0$$

Inadmissible! Does not satisfy the primal feasibility constraint $p_2 \geq 550 + p_1$.

Price Optimization

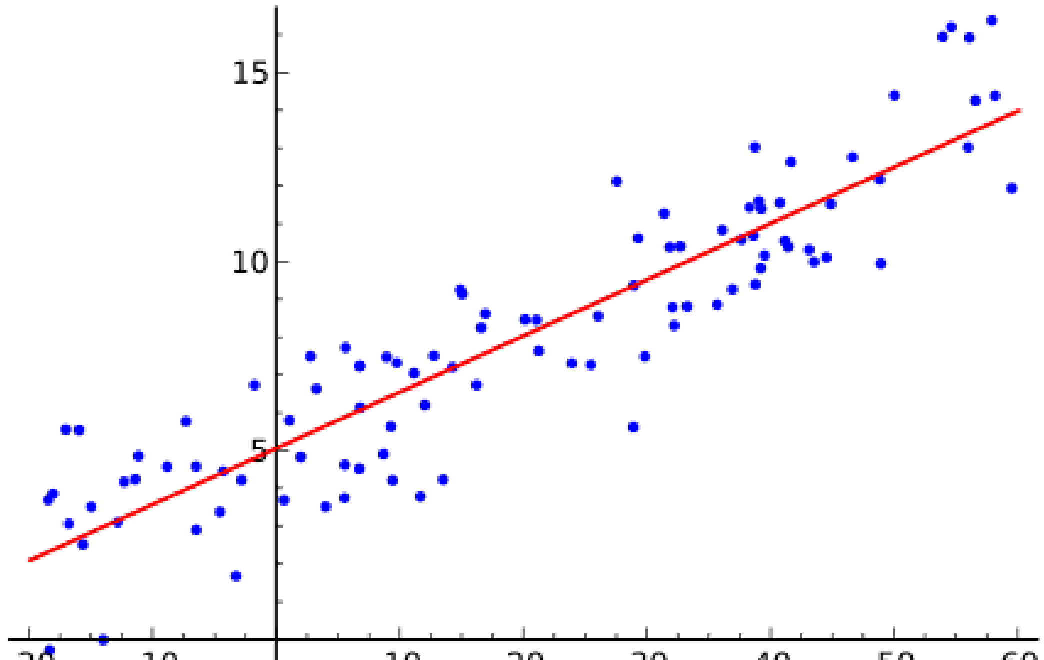
- While the KKT conditions give us a way to figure out the optimal solution to a nonlinear system in closed-form, they:
  1. Are generally still pretty difficult to solve.
  2. Require you to check many points unless the number of feasible solutions is small.



Regularized Regression

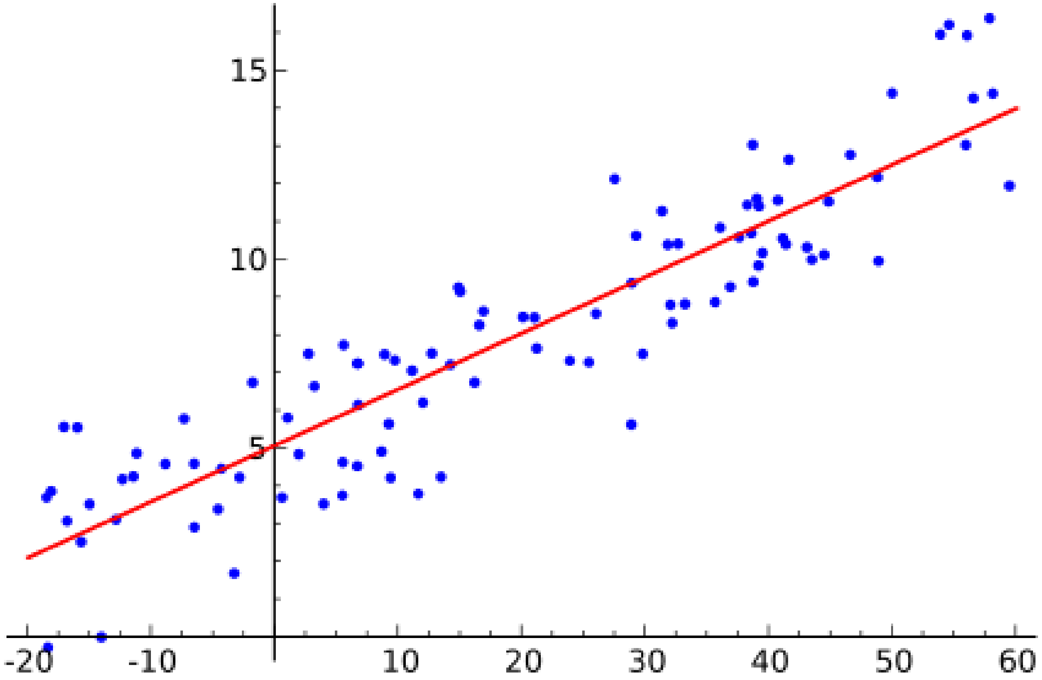
Did you know?

Linear regression is a nonlinear (quadratic) optimization problem!



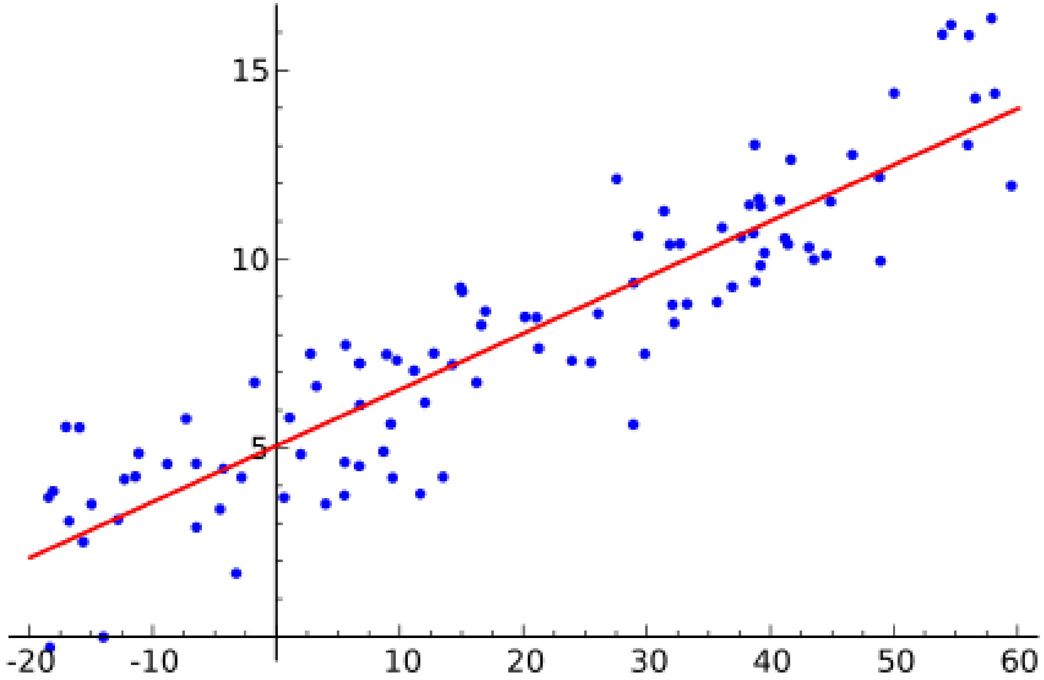
Linear Regression

Suppose you observe N values of feature x_i and its outcome y_i where $i = 1, \ldots, N$.



Linear Regression

The objective is to find the parameters of a straight line so as to minimize the distance between each point (y_i, x_i) and the line.



Linear Regression

The objective is to find the parameters of a straight line so as to minimize the distance between each point (y_i, x_i) and the line.

$$\boldsymbol{Minimize}\; Z = \frac{1}{N}\sum_{i=1}^{N}(y_i - \alpha - \beta x_i)^2$$

where α and β are the decision variables of the problem. Notice that they are *unconstrainted*; they can take both positive and negative values.

Linear Regression

Suppose you now observe N values of J **features** x_{ij} for every outcome y_i where $i = 1, \dots, N$. The nonlinear optimization problem now becomes

$$\boldsymbol{Minimize}\; Z = \frac{1}{N} \sum_{i=1}^{N} \left(y_i - \alpha - \sum_{j=1}^{J} \beta_j x_{ij} \right)^2$$

where α and β_j for $j = 1, \dots, J$ are decision variables. All variables are *unconstrained* as they can take both positive and negative values.

Linear Regression

Suppose you now observe N values of J **features** x_{ij} for every outcome y_i where $i = 1, \dots, N$. The nonlinear optimization problem now becomes

$$\boldsymbol{Minimize}\ Z = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

where α and β_j for $j = 1, \dots, J$ are decision variables and $\hat{y}_i = \alpha + \sum_{j=1}^{J}\beta_j x_{ij}$ is the predicted value. All variables are *unconstrained* as they can take both positive and negative values.

Linear Regression

- This framework is called **OLS regression**; we choose the parameters of a linear function given a set of **feature** variables by minimizing the sum of the squared differences between the observed ***dependent*** variable in the data and the value ***predicted*** by the linear function.
- There are other ways to perform regression:
  - Maximum likelihood
  - Generalized method of moments
  - Bayesian approaches
- Assumption of normality can also be imposed.

Regularized Regression

A regression model that either includes constraints that limit the size of the regression coefficients or a term in the objective that acts to shrink their size.

- It is useful technique to mitigate the issue of multicollinearity (i.e., many model parameters).
- It also prevents overfitting a regression model to a data set by encouraging the optimization procedure to favor less complex formulations (i.e., those with fewer parameters).
- Helps to identify features that exhibit the strongest predictive effects (feature selection).

Unconstrained Ridge Regression

Dataset: You have a data set of $i = 1, \dots, N$ instances with $j = 1, \dots, J$ features x_{ij} for every outcome y_i.

$$\textbf{Minimize } Z = \frac{1}{N} \sum_{i=1}^{N} \left(y_i - \alpha - \sum_{j=1}^{J} \beta_j x_{ij} \right)^2 + \lambda \left(\sum_{j=1}^{J} \beta_j^2 \right)$$

where α and β_j for $j = 1, \dots, J$ are decision variables and $\lambda \geq 0$ is a complexity parameter that controls the amount of shrinkage towards zero (this is also called weight decay in neural networks).

Constrained Ridge Regression

Dataset: You have a data set of $i = 1, \ldots, N$ instances with $j = 1, \ldots, J$ features x_{ij} for every outcome y_i.

Minimize $Z = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \alpha - \sum_{j=1}^{J}\beta_j x_{ij}\right)^2$ **subject to:**

$$\sum_{j=1}^{J}\beta_j^2 \leq t$$

for decision variables α and β_j for $j = 1, \ldots, J$.

Ridge Regression

The unconstrained and constrained version of ridge regression are equivalent!

How? Use the KKT conditions!

Ridge Regression

$$\mathcal{L}(\boldsymbol{\beta},\lambda) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \alpha - \sum_{j=1}^{J}\beta_j x_{ij}\right)^2 + \lambda\left(\sum_{j=1}^{J}\beta_j^2 - t\right)$$

$$= \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \alpha - \sum_{j=1}^{J}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{J}\beta_j^2 - \lambda t$$

Optimality condition (stationarity):

$$\frac{\partial\mathcal{L}}{\partial\beta_j} = -\frac{1}{N}\sum_{i=1}^{N}2\,x_{ij}\left(y_i - \alpha - \sum_{j=1}^{J}\beta_j x_{ij}\right) + 2\lambda\beta_j = 0$$

Ridge Regression

$$\mathcal{L}(\boldsymbol{\beta}, \lambda) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \alpha - \sum_{j=1}^{J}\beta_j x_{ij}\right)^2 + \lambda\left(\sum_{j=1}^{J}\beta_j^2 - t\right)$$

$$= \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \alpha - \sum_{j=1}^{J}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{J}\beta_j^2 - \lambda t$$

Complementary Slackness:

$$\lambda\frac{\partial\mathcal{L}}{\partial\lambda} = \lambda\left(\sum_{j=1}^{J}\beta_j^2 - t\right) = 0$$

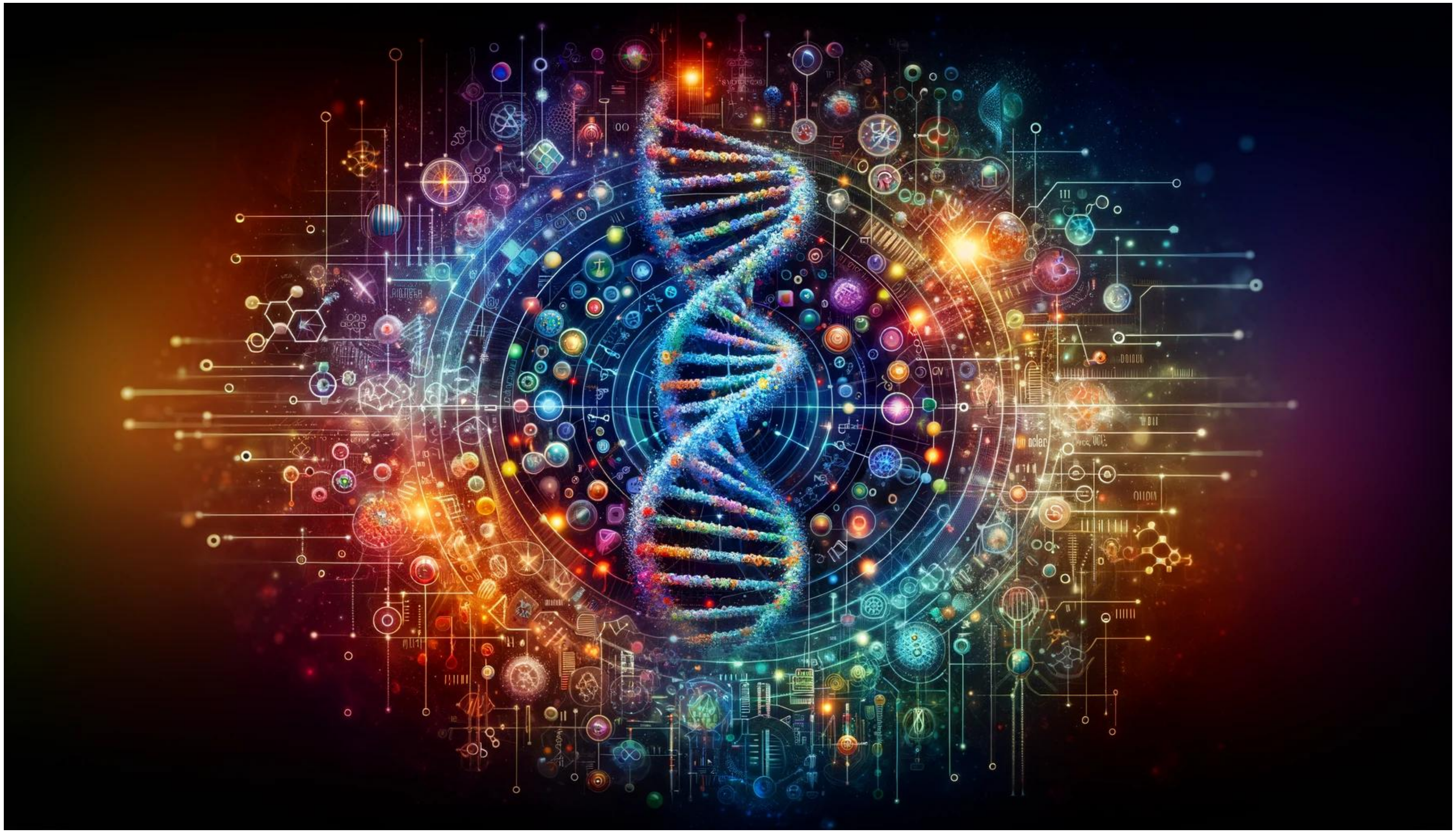
Ridge Regression

Due to the KKT conditions we see that:

1. **From the optimality condition:** The first-order conditions used to solve for the optimal regression coefficients $(\alpha, \boldsymbol{\beta})$ are <u>*identical*</u> for both regression formulations.
2. **From complementary slackness:** We either have that $\lambda = 0$ or $\lambda > 0$.
   - $\lambda = 0$: Trivially equivalent, no regularization term.
   - $\lambda > 0$: Implies that at $\boldsymbol{\beta}^*$, $t = \sum_{j=1}^{J} \beta_j^2$ and feasible $\boldsymbol{\beta}$ values are constrained by $\sum_{j=1}^{J} \beta_j^2 \leq t$.

Ridge Regression

Ridge regression is a commonly applied method when attempting to identify genes associated with a disease.



Ridge Regression

The goal is to extract the main phenotypes given that the number of features (e.g., genes) is much larger than the number of observations (e.g., patients).

Next Class: Nonlinear Models and Optimization Algorithms

- Algorithms for solving constrained nonlinear optimization models differ from solving a linear program (i.e., using the simplex method).
  - Gradient-descent and projected gradient descent,
  - Sequential quadratic programming.
  - Penalty and interior-point methods.
  - **Examples:** Quadratic optimization problems.
- It's important to understand that the difficultly in solving these problems is much greater than LPs. Gurobi can only solve quadratic programs.