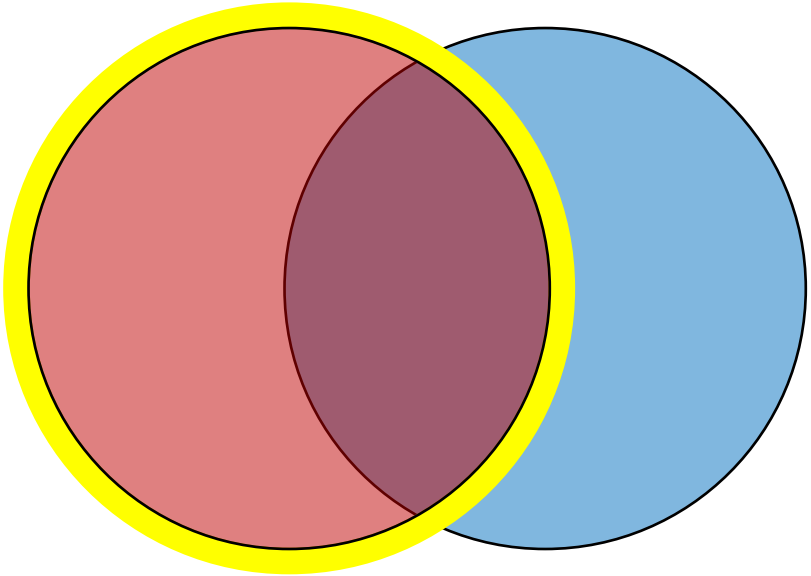
Essential Math for ML (Probability & Calculus)

- ✓ **Probability Theory for ML (Bayes Theorem)**
- ✓ **Common Probability Distributions**
- ✓ **Introduction to Calculus for ML**
- ✓ **Partial Derivatives and Chain Rule**
- ✓ **Optimization (Basics)**

Probability Basics

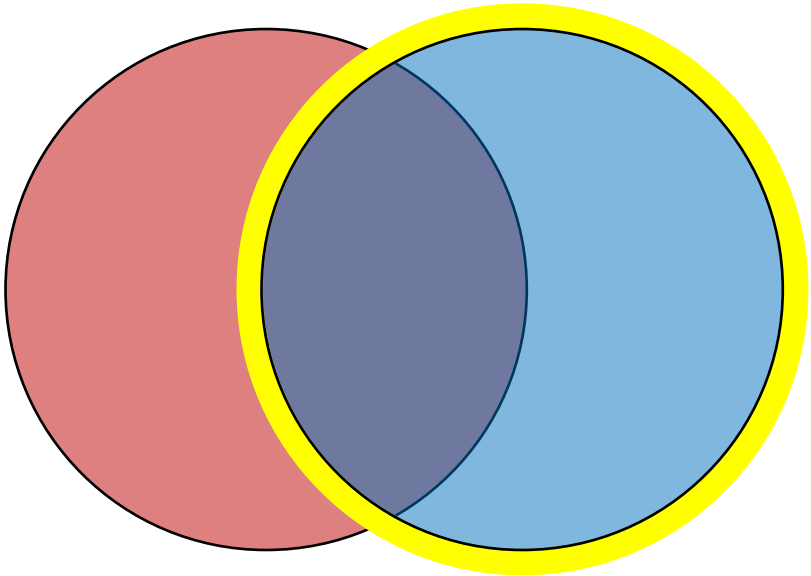


$P(X)$

- Single event probability:

$$P(X)$$

Probability Basics

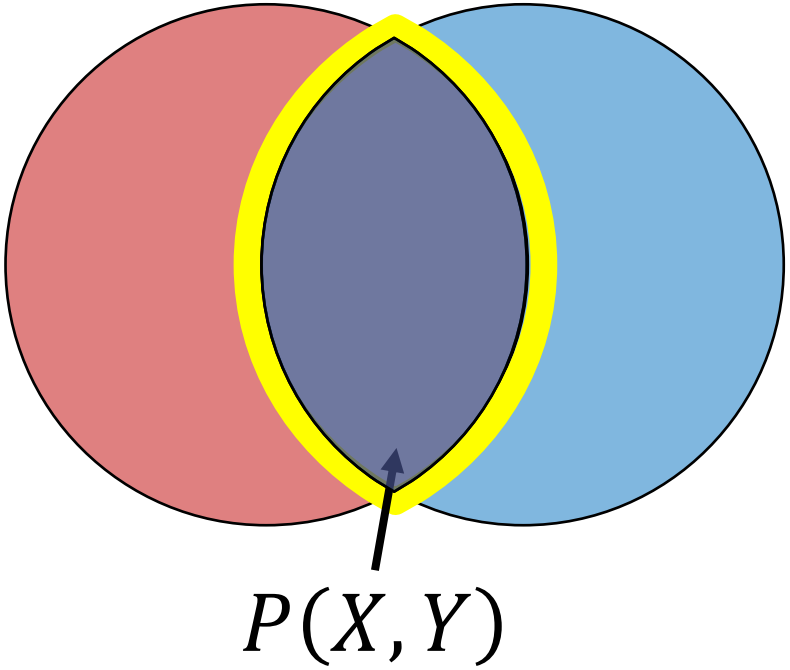


$P(X)$ $P(Y)$

- Single event probability:

$$P(X), P(Y)$$

Probability Basics



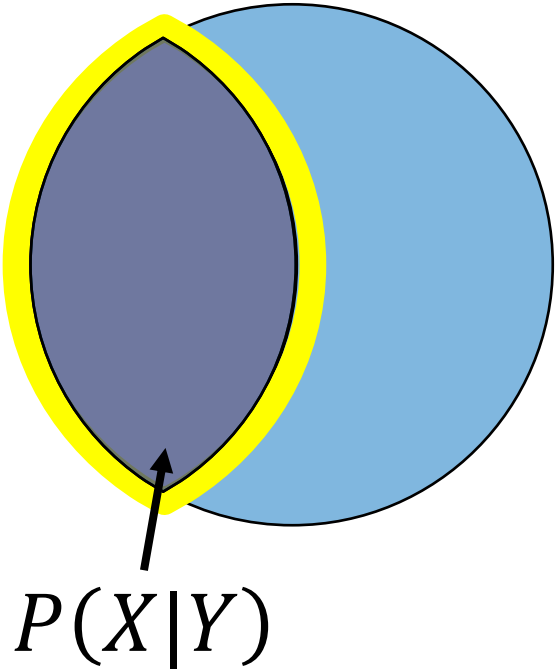
- Single event probability:

$$P(X), P(Y)$$

- Joint event probability:

$$P(X,Y)$$

Probability Basics



- Single event probability:

$$P(X), P(Y)$$

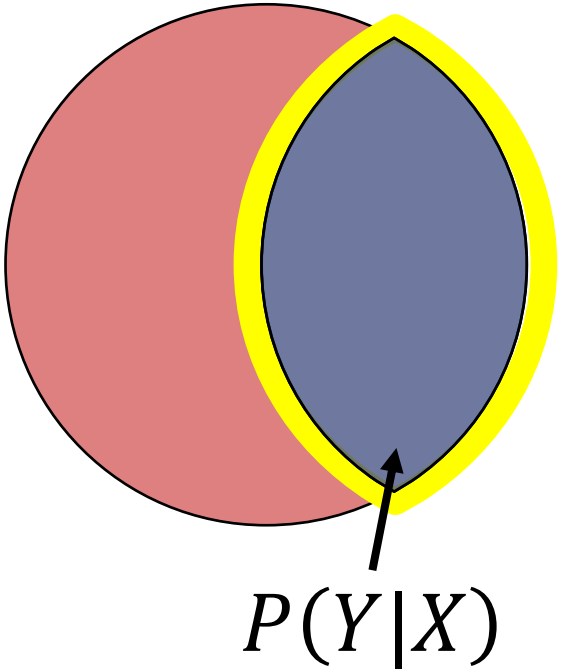
- Joint event probability:

$$P(X, Y)$$

- Conditional probability:

$$P(X|Y)$$

Probability Basics



- Single event probability:

$$P(X), P(Y)$$

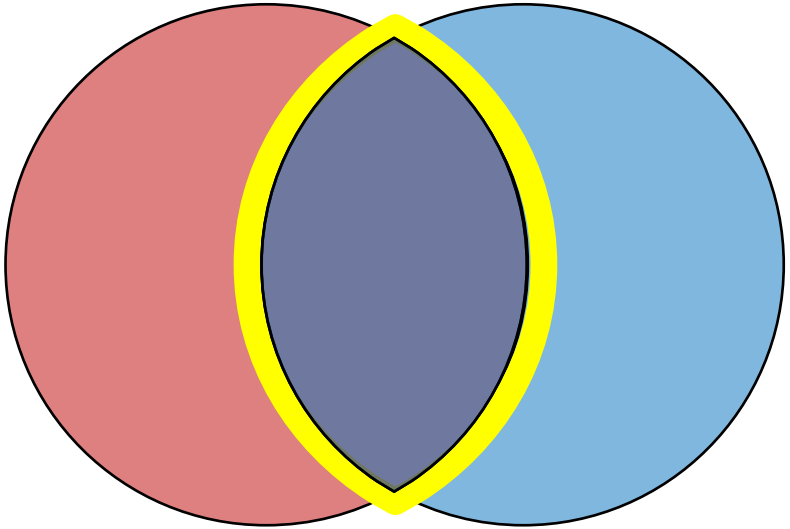
- Joint event probability:

$$P(X, Y)$$

- Conditional probability:

$$P(X|Y), P(Y|X)$$

Probability Basics



- Single event probability:

$$P(X), P(Y)$$

- Joint event probability:

$$P(X, Y)$$

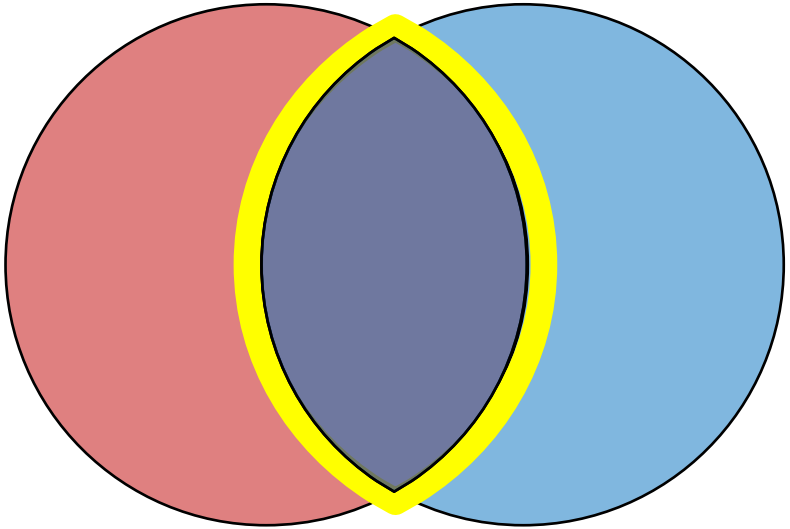
- Conditional probability:

$$P(X|Y), P(Y|X)$$

- Joint and conditional relationship:

$$P(X, Y) = P(Y|X) * P(X) = P(X|Y) * P(Y)$$

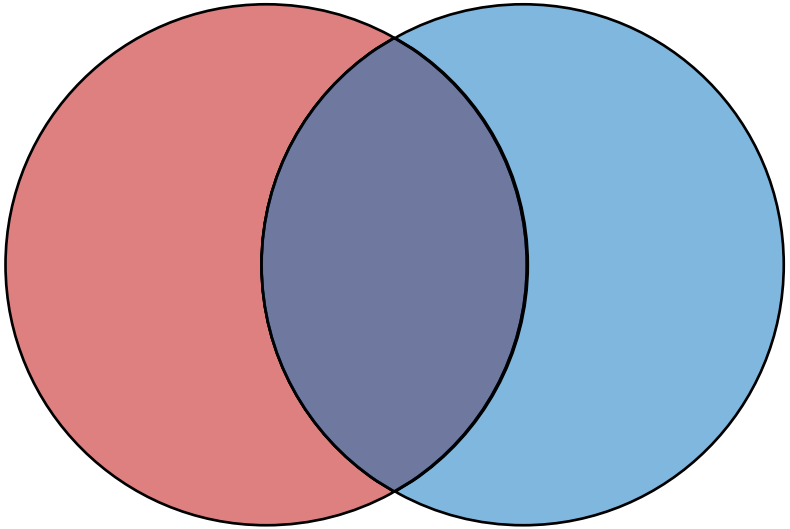
Bayes Theorem Derivation



- By conditional and joint relationship:

$$P(Y|X) * P(X) = P(X|Y) * P(Y)$$

Bayes Theorem Derivation



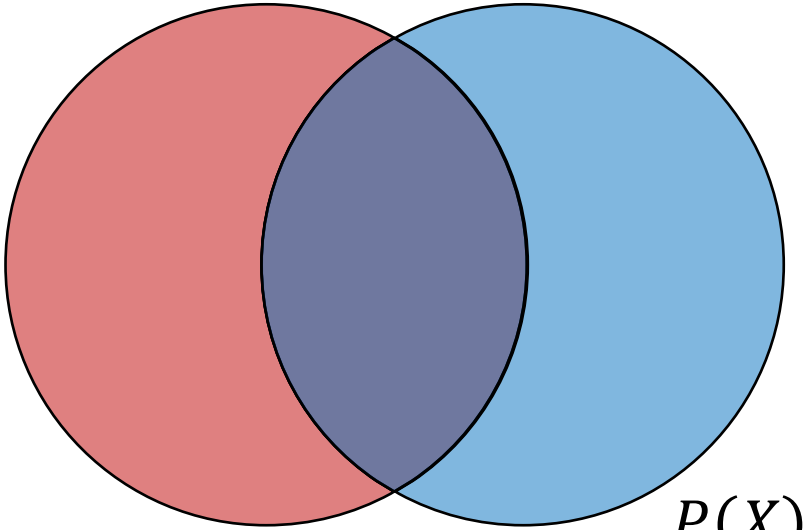
- Use conditional and joint relationship:

$$P(Y|X) * P(X) = P(X|Y) * P(Y)$$

- To invert conditional probability:

$$P(Y|X) = \frac{P(X|Y) * P(Y)}{P(X)}$$

Bayes Theorem Derivation



- Use conditional and joint relationship:

$$P(Y|X) * P(X) = P(X|Y) * P(Y)$$

- To invert conditional probability:

$$P(Y|X) = \frac{P(X|Y) * P(Y)}{P(X)}$$

$$P(X) = \sum_{Z} P(X, Z) = \sum_{Z} P(X|Z) * P(Z)$$

Bayes Theorem

$$P(Y|X) = \frac{P(X|Y) * P(Y)}{P(X)}$$

Bayes Theorem

$$P(Y|X) = \frac{P(X|Y) * P(Y)}{P(X)}$$

$$posterior = \frac{likelihood * prior}{evidence}$$

Common Probability Distributions

Important: We will use these extensively to model **data** as well as **parameters**

Some **discrete distributions** and what they can model:

- **Bernoulli:** Binary numbers, e.g., outcome (head/tail, 0/1) of a coin toss
- **Binomial:** Bounded non-negative integers, e.g., # of heads in n coin tosses
- **Multinomial:** One of K (>2) possibilities, e.g., outcome of a dice roll
- **Poisson:** Non-negative integers, e.g., # of words in a document
- .. and many others

Some **continuous distributions** and what they can model:

- **Uniform:** numbers defined over a fixed range
- **Beta:** numbers between 0 and 1, e.g., probability of head for a biased coin
- **Gamma:** Positive unbounded real numbers
- **Dirichlet:** vectors that sum of 1 (fraction of data points in different clusters)
- **Gaussian:** real-valued numbers or real-valued vectors
- .. and many others

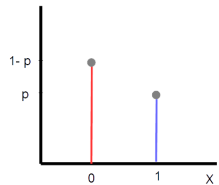
Discrete Distributions

Bernoulli Distribution

- Distribution over a binary r.v. $x \in \{0, 1\}$, like a coin-toss outcome
- Defined by a probability parameter $p \in (0, 1)$

$$P(x = 1) = p$$

- Distribution defined as: $\text{Bernoulli}(x; p) = p^x(1 - p)^{1-x}$

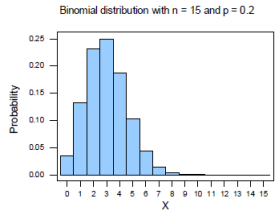


- Mean: $\mathbb{E}[x] = p$
- Variance: $\text{var}[x] = p(1 - p)$

Binomial Distribution

- Distribution over number of successes m (an r.v.) in a number of trials
- Defined by two parameters: total number of trials (N) and probability of each success $p \in (0, 1)$
- Can think of Binomial as multiple independent Bernoulli trials
- Distribution defined as

$$\text{Binomial}(m; N, p) = \binom{N}{m} p^m (1-p)^{N-m}$$



- Mean: $\mathbb{E}[m] = Np$
- Variance: $\text{var}[m] = Np(1-p)$

Multinoulli Distribution

- Also known as the **categorical distribution** (models categorical variables)
- Think of a random assignment of an item to one of K bins - a K dim. binary r.v. $\boldsymbol{x}$ with single 1 (i.e., $\sum_{k=1}^{K} x_k = 1$): **Modeled by a multinoulli**

$$\underbrace{[0 \quad 0 \quad 0 \quad \ldots 0 \quad 1 \quad 0 \quad 0]}_{\text{length} = K}$$

- Let vector $\boldsymbol{p} = [p_1, p_2, \ldots, p_K]$ define the probability of going to each bin
  - $p_k \in (0, 1)$ is the probability that $x_k = 1$ (assigned to bin k)
  - $\sum_{k=1}^{K} p_k = 1$
- The multinoulli is defined as: $\text{Multinoulli}(\boldsymbol{x}; \boldsymbol{p}) = \prod_{k=1}^{K} p_k^{x_k}$
- Mean: $\mathbb{E}[x_k] = p_k$
- Variance: $\text{var}[x_k] = p_k(1 - p_k)$

Multinomial Distribution

- Think of repeating the Multinoulli N times
- Like distributing N items to K bins. Suppose x_k is count in bin k

$$0 \leq x_k \leq N \quad \forall\ k = 1, \ldots, K, \qquad \sum_{k=1}^{K} x_k = N$$

- Assume probability of going to each bin: $\boldsymbol{p} = [p_1, p_2, \ldots, p_K]$
- Multonomial models the bin allocations via a discrete vector $\boldsymbol{x}$ of size K

$$[x_1 \quad x_2 \quad \ldots x_{k-1} \quad x_k \quad x_{k-1} \ldots \quad x_K]$$

- Distribution defined as

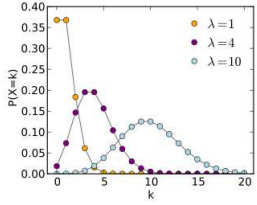
$$\text{Multinomial}(\boldsymbol{x}; N, \boldsymbol{p}) = \binom{N}{x_1 x_2 \ldots x_K} \prod_{k=1}^{K} p_k^{x_k}$$

- Mean: $\mathbb{E}[x_k] = Np_k$
- Variance: $\text{var}[x_k] = Np_k(1 - p_k)$
- Note: For $N = 1$, multinomial is the same as multinoulli

Poisson Distribution

- Used to model a non-negative integer (count) r.v. k
- Examples: number of words in a document, number of events in a fixed interval of time, etc.
- Defined by a positive rate parameter λ
- Distribution defined as

$$\text{Poisson}(k; \lambda) = \frac{\lambda^k e^{-\lambda}}{k!} \qquad k = 0, 1, 2, \ldots$$



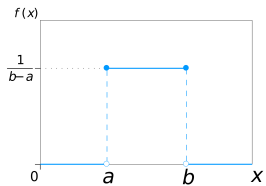
- Mean: $\mathbb{E}[k] = \lambda$
- Variance: $\text{var}[k] = \lambda$

Continuous Distributions

Uniform Distribution

- Models a continuous r.v. x distributed uniformly over a finite interval $[a, b]$

$$\text{Uniform}(x; a, b) = \frac{1}{b-a}$$

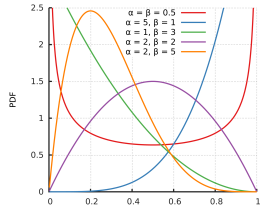


- Mean: $\mathbb{E}[x] = \frac{(b+a)}{2}$
- Variance: $\text{var}[x] = \frac{(b-a)^2}{12}$

Beta Distribution

- Used to model an r.v. p between 0 and 1 (e.g., a probability)
- Defined by two **shape parameters** α and β

$$\text{Beta}(p; \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1}$$

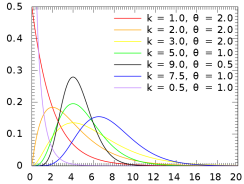


- Mean: $\mathbb{E}[p] = \frac{\alpha}{\alpha+\beta}$
- Variance: $\text{var}[p] = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$
- Often used to model the probability parameter of a Bernoulli or Binomial (also **conjugate** to these distributions)

Gamma Distribution

- Used to model positive real-valued r.v. x
- Defined by a **shape parameters** k and a **scale parameter** θ

$$\text{Gamma}(x; k, \theta) = \frac{x^{k-1} e^{-\frac{x}{\theta}}}{\theta^k \Gamma(k)}$$

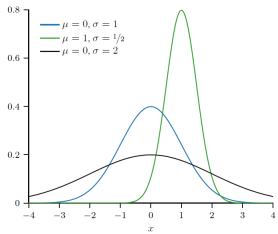


- Mean: $\mathbb{E}[x] = k\theta$
- Variance: $\text{var}[x] = k\theta^2$
- Often used to model the rate parameter of Poisson or exponential distribution (conjugate to both), or to model the inverse variance (precision) of a Gaussian (conjuate to Gaussian if mean known)

Univariate Gaussian Distribution

- Distribution over real-valued scalar r.v. x
- Defined by a scalar **mean** μ and a scalar **variance** σ^2
- Distribution defined as

$$\mathcal{N}(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

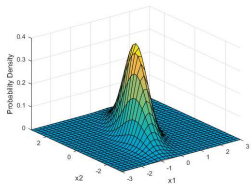


- Mean: $\mathbb{E}[x] = \mu$
- Variance: $\text{var}[x] = \sigma^2$
- Precision (inverse variance) $\beta = 1/\sigma^2$

Multivariate Gaussian Distribution

- Distribution over a multivariate r.v. vector $\boldsymbol{x} \in \mathbb{R}^D$ of real numbers
- Defined by a **mean vector** $\boldsymbol{\mu} \in \mathbb{R}^D$ and a $D \times D$ **covariance matrix** $\boldsymbol{\Sigma}$

$$\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^D |\boldsymbol{\Sigma}|}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}$$



- The covariance matrix $\boldsymbol{\Sigma}$ must be symmetric and positive definite
  - All eigenvalues are positive
  - $\boldsymbol{z}^\top \boldsymbol{\Sigma} \boldsymbol{z} > 0$ for any real vector $\boldsymbol{z}$
- Often we parameterize a multivariate Gaussian using the inverse of the covariance matrix, i.e., the **precision matrix** $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$

Understanding Calculus – Limits, Continuity, and Derivatives

- **Limit:**

  > Measures how a function behaves as input approaches a certain value.
  > *Example:* $\lim_{x \to 2}(3x + 1) = 7$

- **Continuity:**

  > A function is continuous if there are no breaks, jumps, or holes in its graph.
  > *Think:* You can draw it without lifting your pen.

- **Derivative:**

  > Measures how much a function changes as its input changes. It's the *slope* of the function.
  > $f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$

Derivatives and Gradients in Machine Learning

- **Gradient = Derivative for Multi-Dimensional Input:**

  > In ML, we often have many variables. The **gradient** is a vector of partial derivatives:
  > $\nabla f(\mathbf{x}) = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots\right]$

- **Gradient Descent Algorithm:**

  > Uses the gradient to move "downhill" on the loss function.
  > $\theta \leftarrow \theta - \eta \nabla J(\theta)$

- **Why It Matters:**

  > Gradients tell models *how to update parameters* to minimize error.

Partial Derivatives & Chain Rule

- **Partial Derivative:**

  > Derivative with respect to one variable while keeping others constant.
  > $\frac{\partial f(x,y)}{\partial x} = \text{"How f changes as x changes, y fixed"}$

- **Chain Rule:**

  > Used when functions are nested.
  > $\frac{dy}{dx} = \frac{dy}{du} \cdot \frac{du}{dx}$
  > Applies in neural networks when outputs depend on intermediate functions.

- **Intuition:**

  > Each layer's output depends on previous layer → apply chain rule through layers.

Using Chain Rule in Backpropagation

- **Neural Network Layers:**

  > Each layer transforms inputs. To compute loss gradient w.r.t. weights, use chain rule through layers.

- **Backpropagation Algorithm:**

  > Systematic application of the chain rule to compute gradients efficiently.

$$\frac{\partial L}{\partial w} = \frac{\partial L}{\partial a} \cdot \frac{\partial a}{\partial z} \cdot \frac{\partial z}{\partial w}$$

- **Why It Works:**

  > Allows gradient descent to update weights throughout deep networks.

Optimization

Are you using optimization?

The word “optimization” may be very familiar or may be quite new to you.

……. but whether you know about optimization or not, you are using optimization in many occasions of your day to day life …….

…….Examples……

Optimization in real life

Newspaper hawker

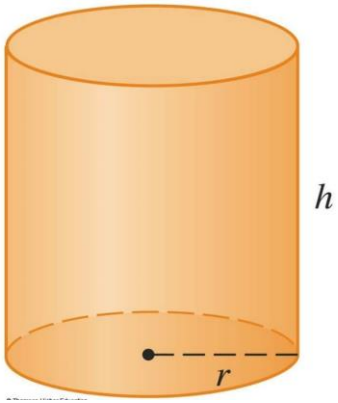
Cooking

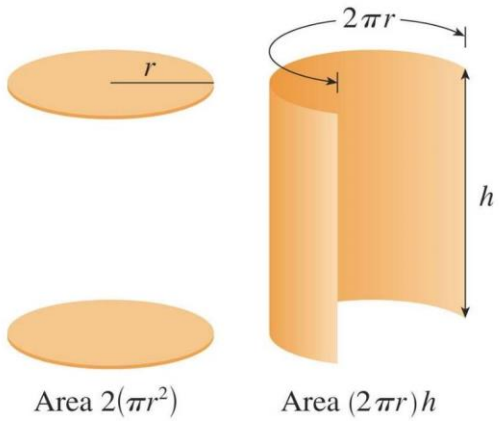
Forensic artist

Ant colony

Example

A manufacturer needs to make a cylindrical can that will hold 1.5 liters of liquid. Determine the dimensions of the can that will minimize the amount of material used in its construction.





Minimize: $A = 2\pi r^2 + 2\pi rh$

Constraint: $\pi r^2 h = 1500$

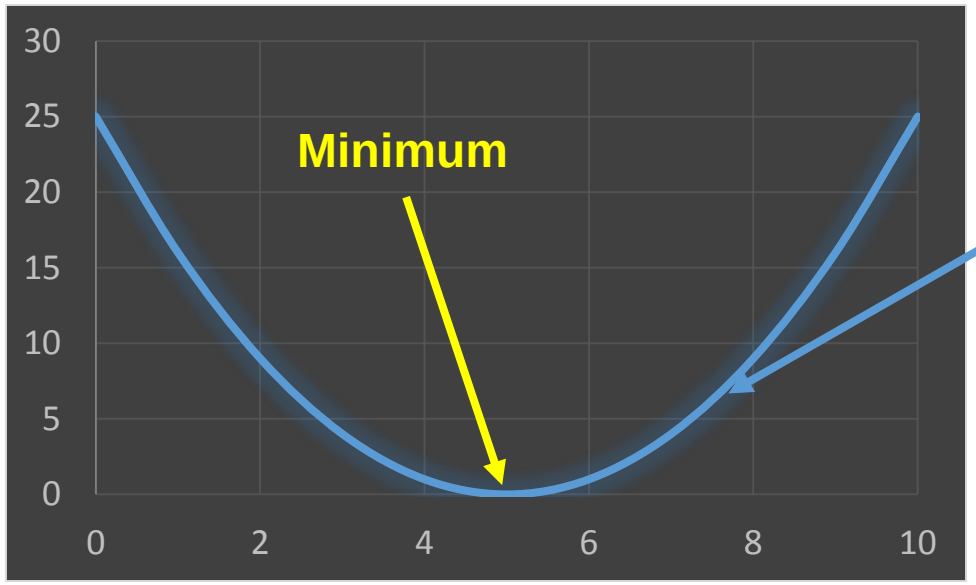
Dimension is in cm

What is Optimization?

- Optimization is the act of obtaining the best result under a given circumstances.

- Optimization is the mathematical discipline which is concerned with finding the maxima and minima of functions, possibly subject to constraints.

Introduction to optimization

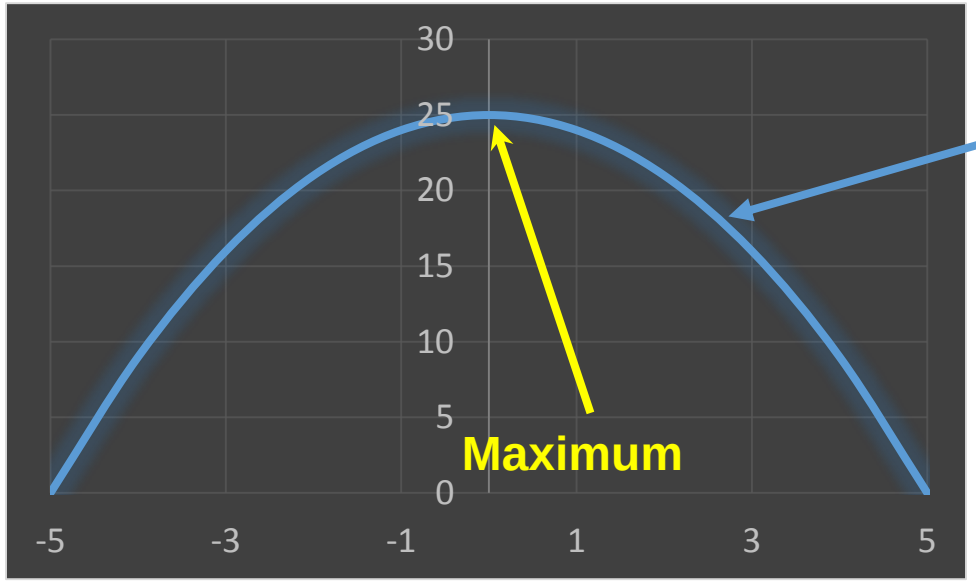


$f = (x-5)^2$ Equation of the line

How to find out the minimum of the function

$f' = 2 \times (x-5) = 0$

$x^* = 5$ Optimal solution

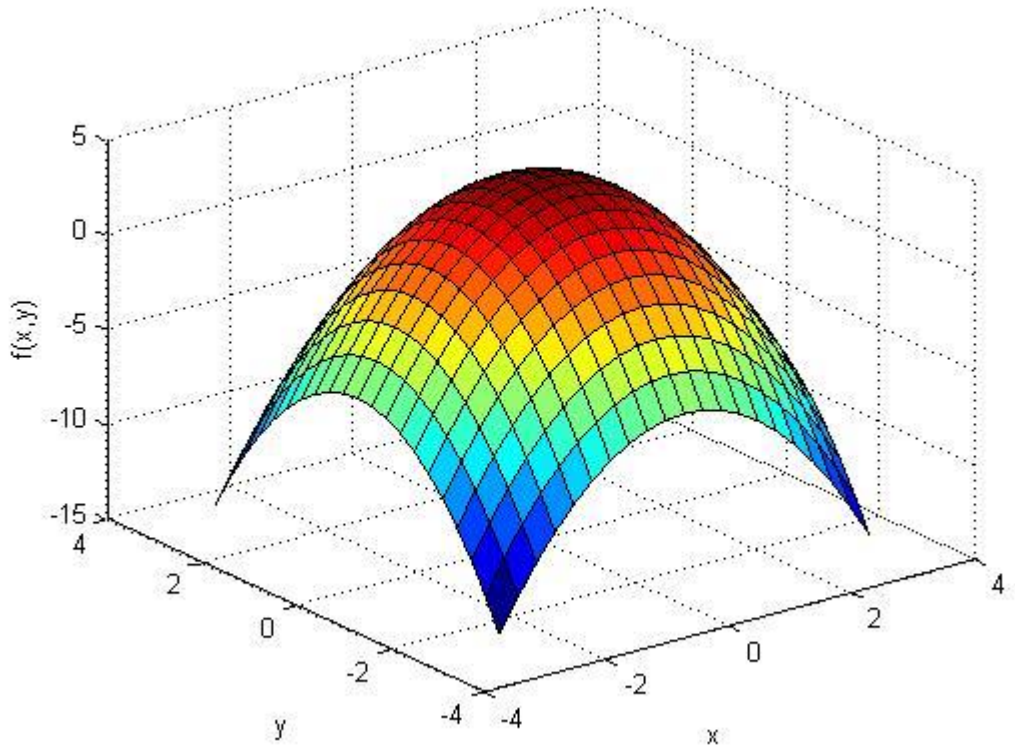


$f = 25 + x^2$ Equation of the line

$f' = 2x = 0$

$x^* = 0$ Optimal solution

Introduction to optimization



Optimal solution is $(0,0)$

Equation of the surface

$$f(x,y) = -(x^2 + y^2) + 4$$

In this case, we can obtain the optimal solution by taking derivatives with respect to variable x and y and equating them to zero

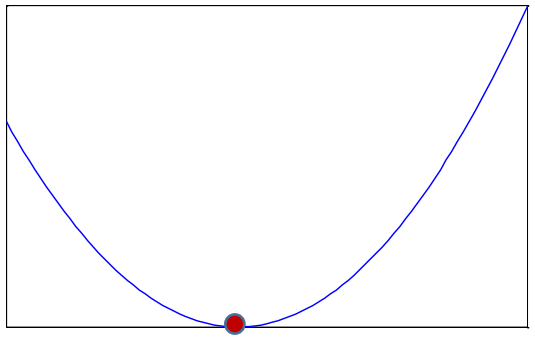
$$\frac{\partial f}{\partial x} = -2x = 0 \qquad \Rightarrow x^* = 0$$

$$\frac{\partial f}{\partial y} = -2y = 0 \qquad \Rightarrow y^* = 0$$

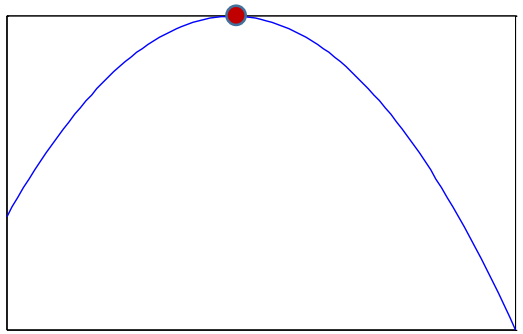
Single variable optimization

Stationary points

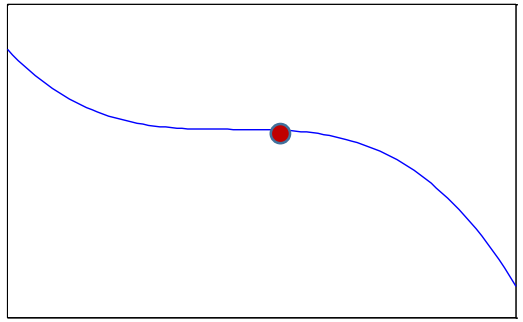
For a continuous and differentiable function *f(x), a stationary point x* is a point at which the* slope of the function is zero, *i.e.* $f'(x) = 0$ *at* $x = x^*$,



Minima



Maxima

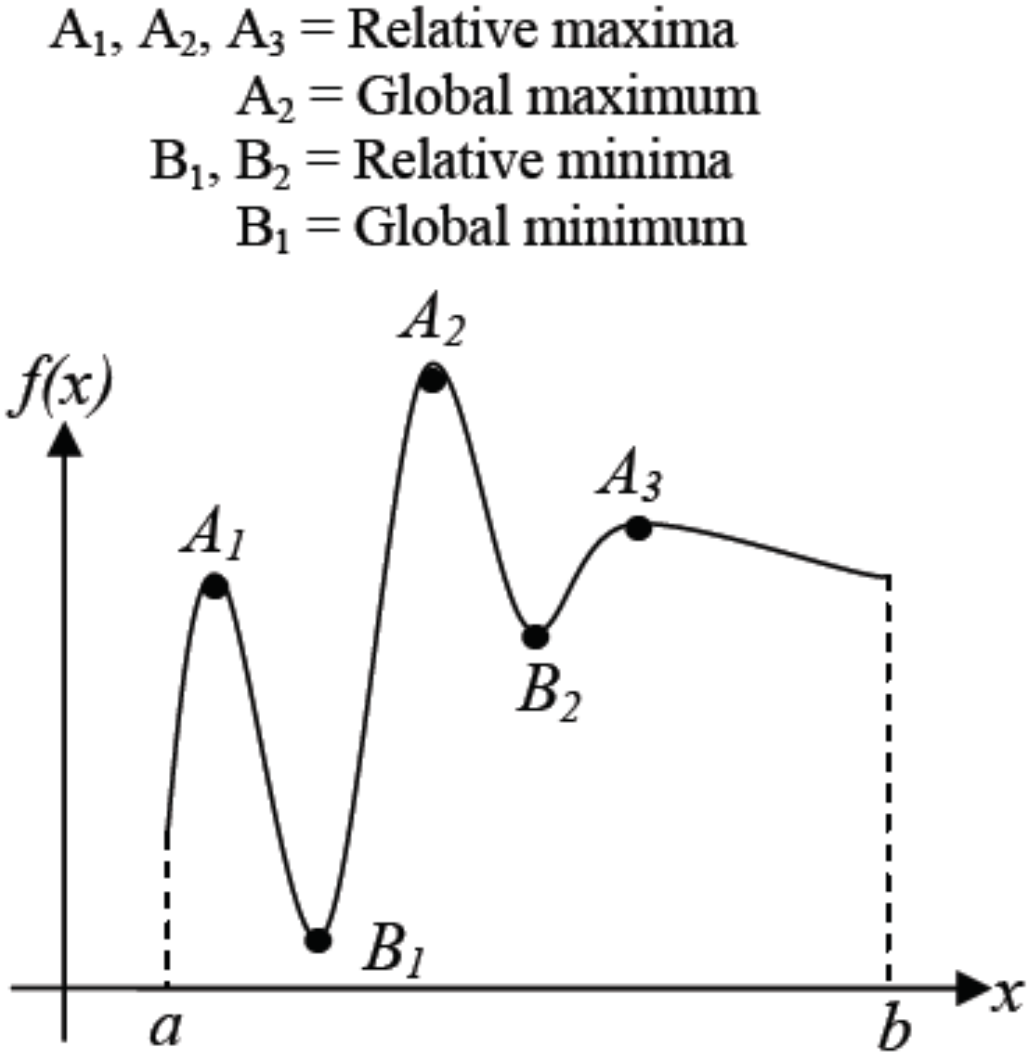


Inflection point

Global minimum and maximum

A function is said to have a *global or absolute minimum at* $x = x^*$ *if* $f(x^*) \leq f(x)$ *for all x in the domain over which f(x) is defined.*

A function is said to have a *global or absolute maximum at* $x = x^*$ *if* $f(x^*) \geq f(x)$ *for all x in the domain over which f(x) is defined.*



Necessary and sufficient conditions for optimality

Necessary condition

If a function $f(x)$ is defined in the interval $a \leq x \leq b$ and has a relative minimum at $x = x^*$, Where $a \leq x^* \leq b$ and if $f'(x)$ exists as a finite number at $x = x^*$, then $f'(x^*) = 0$

Proof

$$f'(x^*) = \lim_{h \to 0} \frac{f(x^* + h) - f(x^*)}{h}$$

Since x^* is a relative minimum $f(x^*) \leq f(x^* + h)$

For all values of h sufficiently close to zero, hence

$$\frac{f(x^* + h) - f(x^*)}{h} \geq 0 \qquad \text{if } h \geq 0$$

$$\frac{f(x^* + h) - f(x^*)}{h} \leq 0 \qquad \text{if } h \leq 0$$

Necessary and sufficient conditions for optimality

Thus

$f^{/}(x^{*}) \geq 0$ If h tends to zero through +ve value

$f^{/}(x^{*}) \leq 0$ If h tends to zero through -ve value

Thus only way to satisfy both the conditions is to have

Sufficient conditions for optimality

Sufficient condition

Suppose at point x^* , the first derivative is zero and first nonzero higher derivative is denoted by n, then

1. *If n is odd, x* is an inflection point*
2. *If n is even, x* is a local optimum*
    1. *If the derivative is positive, x* is a local minimum*
    2. *If the derivative is negative, x* is a local maximum*

Sufficient conditions for optimality

Proof

Apply Taylor's series

$$f(x^* + h) = f(x^*) + hf'(x^*) + \frac{h^2}{2!}f''(x^*) + \cdots + \frac{h^{n-1}}{(n-1)!}f^{n-1}(x^*) + \frac{h^n}{n!}f^n(x^*)$$

Since $f'(x^*) = f''(x^*) = \cdots = f^{n-1}(x^*) = 0$

$$f(x^* + h) - f(x^*) = \frac{h^n}{n!}f^n(x^*)$$

When n is even $\frac{h^n}{n!} \geq 0$

Thus if $f'(x^*)$ is positive $f(x^* + h) - f(x^*)$ is positive Hence it is local minimum

Thus if $f'(x^*)$ negative $f(x^* + h) - f(x^*)$ is negative Hence it is local maximum

When n is odd $\frac{h^n}{n!}$ changes sign with the change in the sign of h.
Hence it is an inflection point

Sufficient conditions for optimality

Take an example

$$f(x) = x^3 - 10x - 2x^2 - 10$$

Apply necessary condition $f'(x) = 3x^2 - 10 - 4x = 0$

Solving for x $x^* = 2.61\ and\ -1.28$ These two points are stationary points

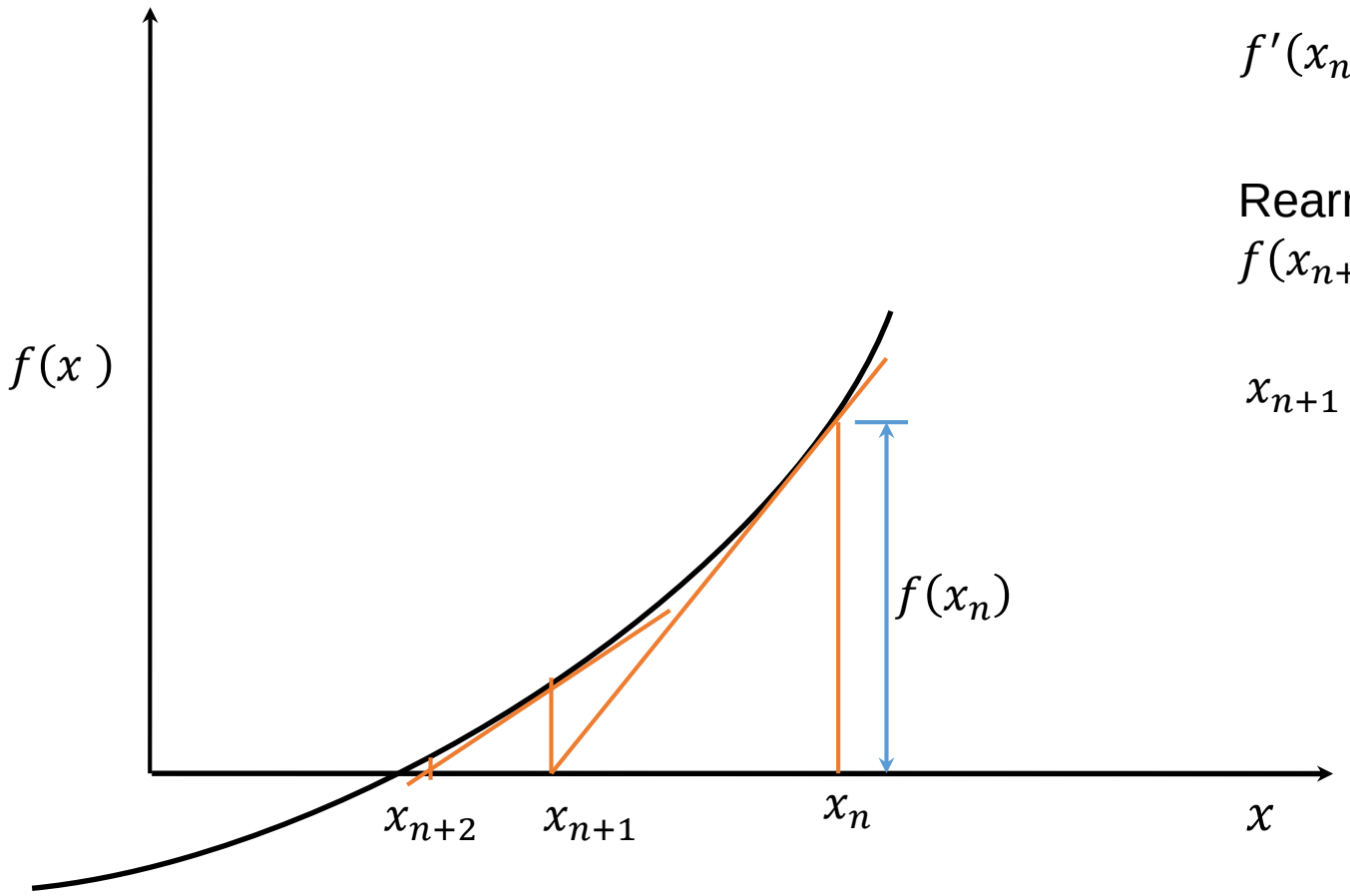
Apply sufficient condition $f''(x) = 6x - 4$

$f''(2.61) = 11.66$ *positive and n is odd* $f''(-1.28) = -11.68$ *negative and n is odd*

$x^* = 2.61$ is a minimum point

$x^* = -1.28$ is a maximum point

Newton-Raphson method



$$f'(x_n) = \frac{f(x_n) - f(x_{n+1})}{x_n - x_{n+1}}$$

Rearranging and putting $f(x_{n+1})$=0

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Continue iteration

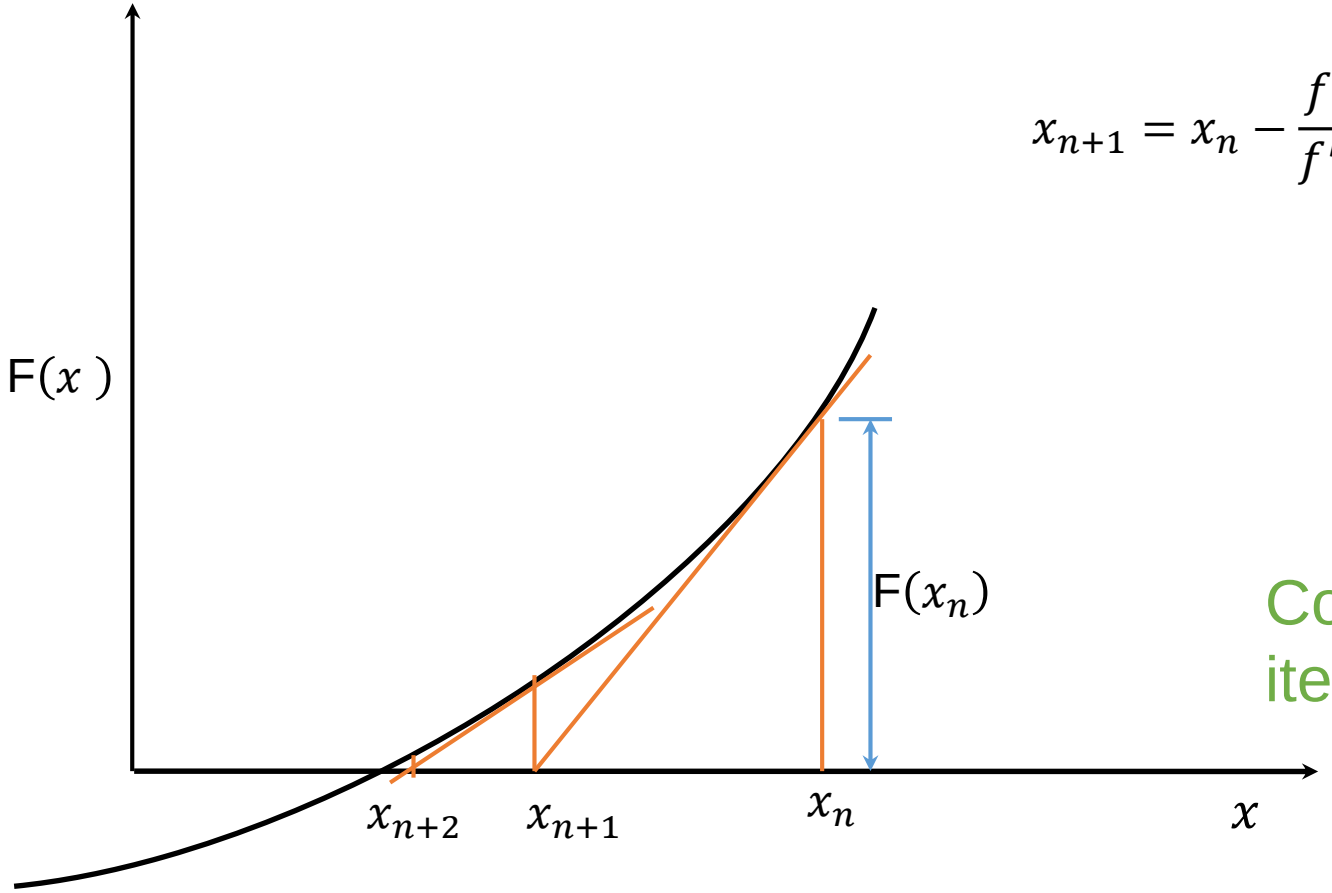
Newton-Raphson method

Incase optimization problem, $f'(x) = 0$

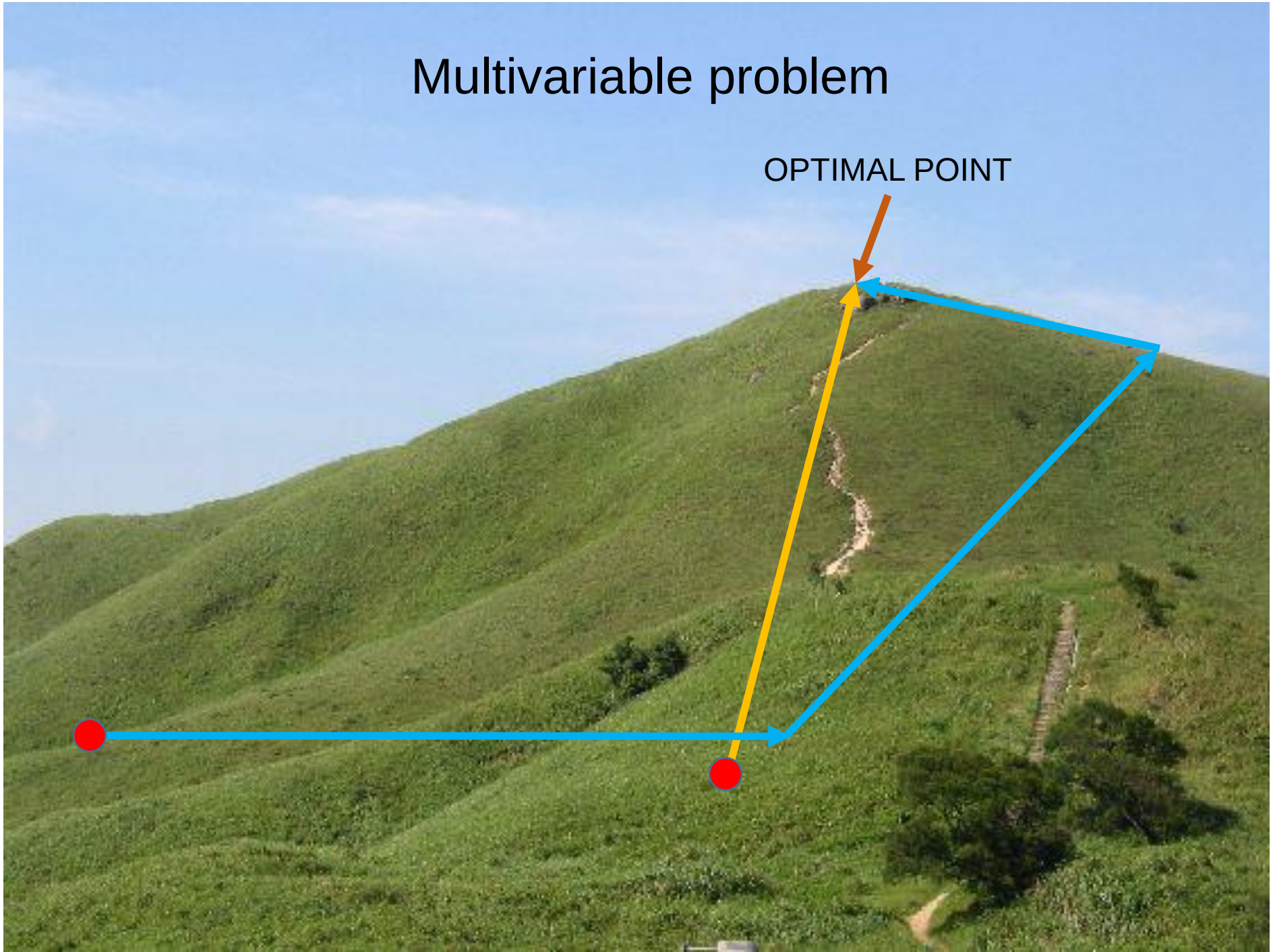
Considering $F(x) = f'(x)$

$$x_{n+1} = x_n - \frac{F(x_n)}{F'(x_n)}$$

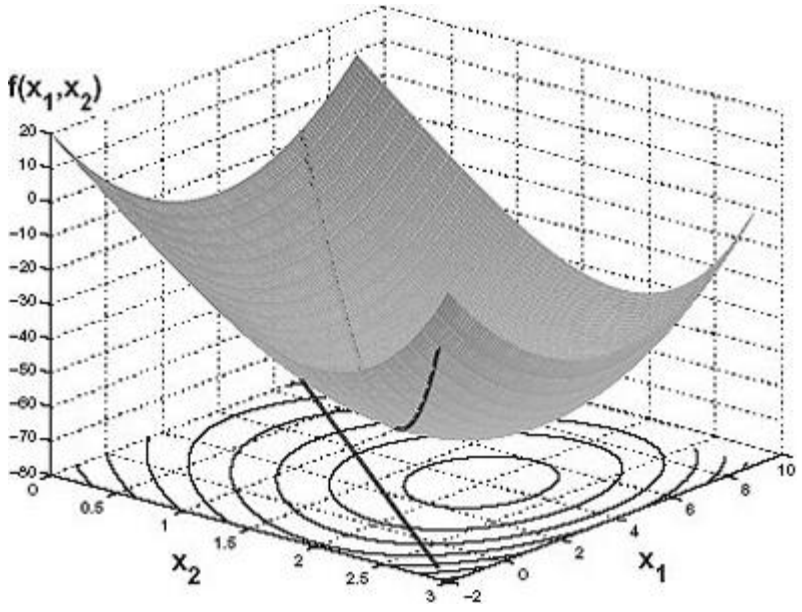
$$x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)}$$

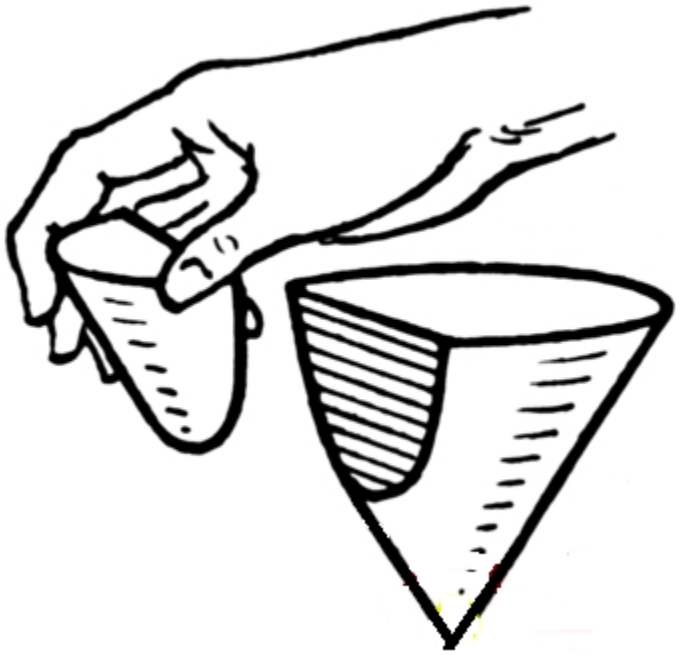


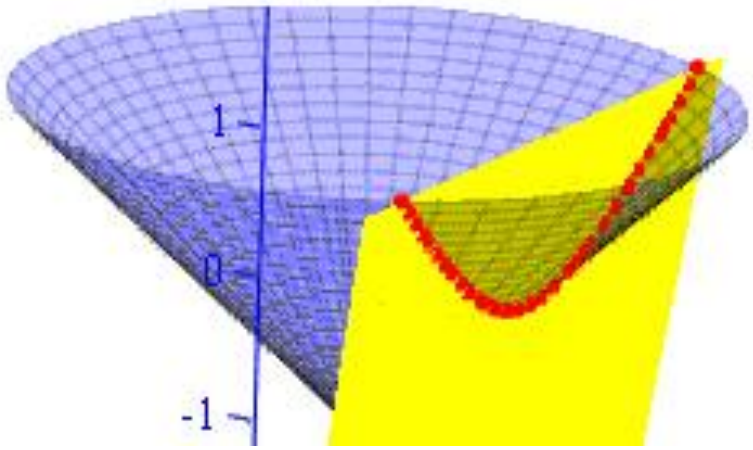
Continue iteration

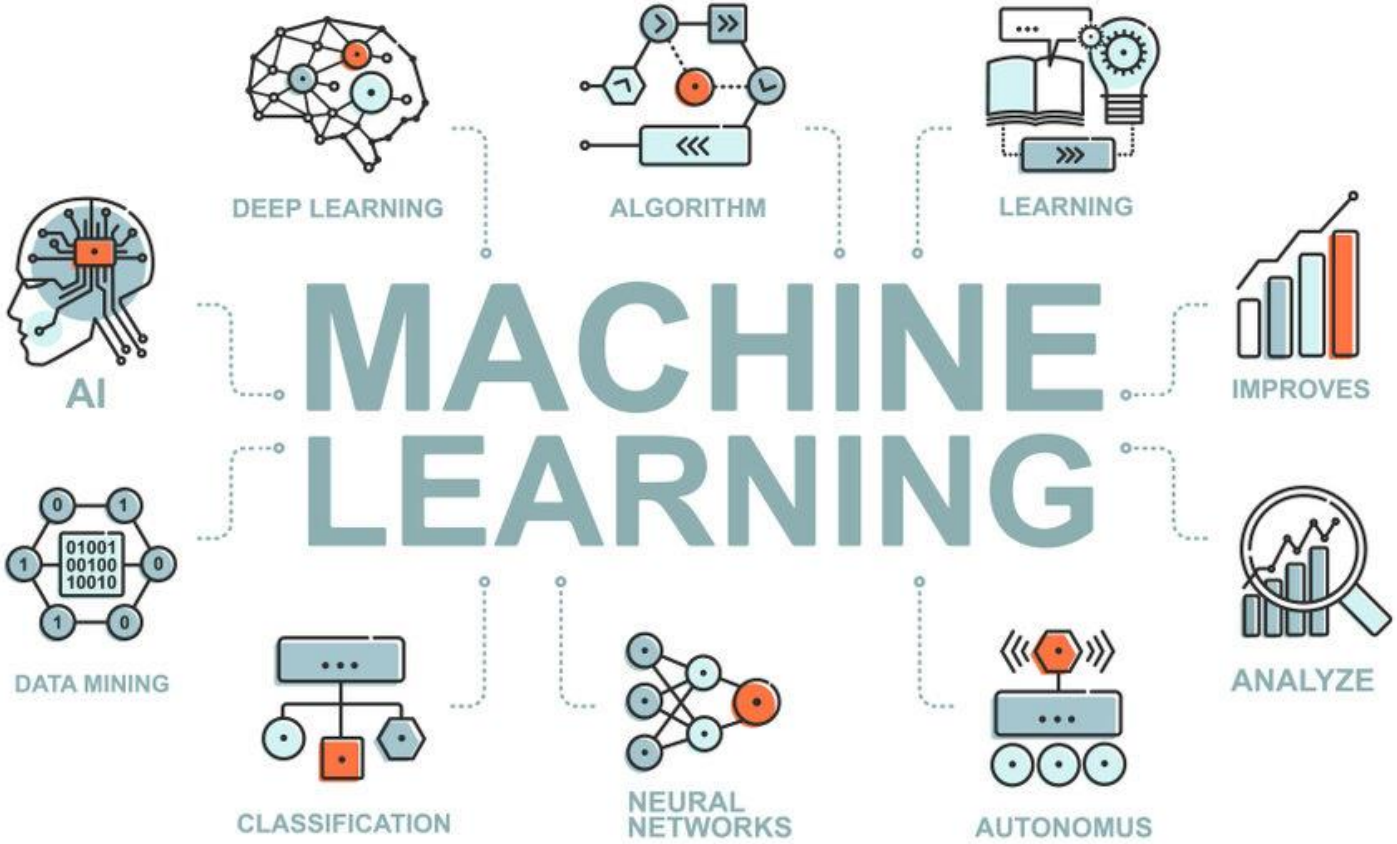

Multivariable problem
OPTIMAL POINT

Multivariable problem








MACHINE LEARNING
DEEP LEARNING
ALGORITHM
LEARNING
AI
IMPROVES
01001
00100
10010
DATA MINING
ANALYZE
CLASSIFICATION
NEURAL NETWORKS
AUTONOMUS