Optimization

Mahdi Roozbahani
Georgia Tech

Outline

Motivation

Entropy

Conditional Entropy and Mutual Information

Cross-Entropy and KL-Divergence

Let's work on this subject in our Optimization lecture

Cross Entropy

Cross Entropy: The expected number of bits when a wrong distribution Q is assumed while the data actually follows a distribution P

Actual pd predicted pd

$$H(p, q) = -\sum_{x \in \mathcal{X}} p(x)\, \log q(x) = \underbrace{H(P)} + \underbrace{KL[P][Q]}$$

This is because:

$$H(p, q) = \mathrm{E}_p[l_i] = \mathrm{E}_p\left[\log \frac{1}{q(x_i)}\right]$$

$$H(p, q) = \sum_{x_i} p(x_i)\, \log \frac{1}{q(x_i)}$$

$$H(p, q) = -\sum_{x} p(x)\, \log q(x).$$

Labeling target values
Label encoding (ordinal) and One-hot encoding

Multi class classification

$$X = \begin{bmatrix} w & h & age & \cdots \\ & & & \\ & & & \end{bmatrix}_{n\times d}$$

$$Y_a = \begin{bmatrix} cat \\ fish \\ dog \\ fish \\ \vdots \end{bmatrix}_{n\times 1} = \begin{bmatrix} 0 \\ 1 \\ 2 \\ 1 \\ \vdots \end{bmatrix} \qquad \hat{Y}_p = \begin{bmatrix} 0 \\ 1 \\ 1.5 \\ \vdots \end{bmatrix}_{n\times 1}$$

$$Y_a = \begin{array}{c c} & \begin{array}{ccc} cat & & \end{array} \\ \begin{array}{c} \\ fish \\ dog \\ \\ \end{array} & \begin{bmatrix} [1 & 0 & 0] \\ [0 & 1 & 0] \\ [0 & 0 & 1] \\ [0 & 1 & 0] \\ & \vdots & \end{bmatrix} \end{array} \qquad \hat{Y}_p = \begin{bmatrix} [0.8 & 0.1 & 0.1] \\ [0.2 & 0.7 & 0.1] \\ & \vdots & \end{bmatrix}$$

$$\boxed{\text{Softmax}}$$

Min

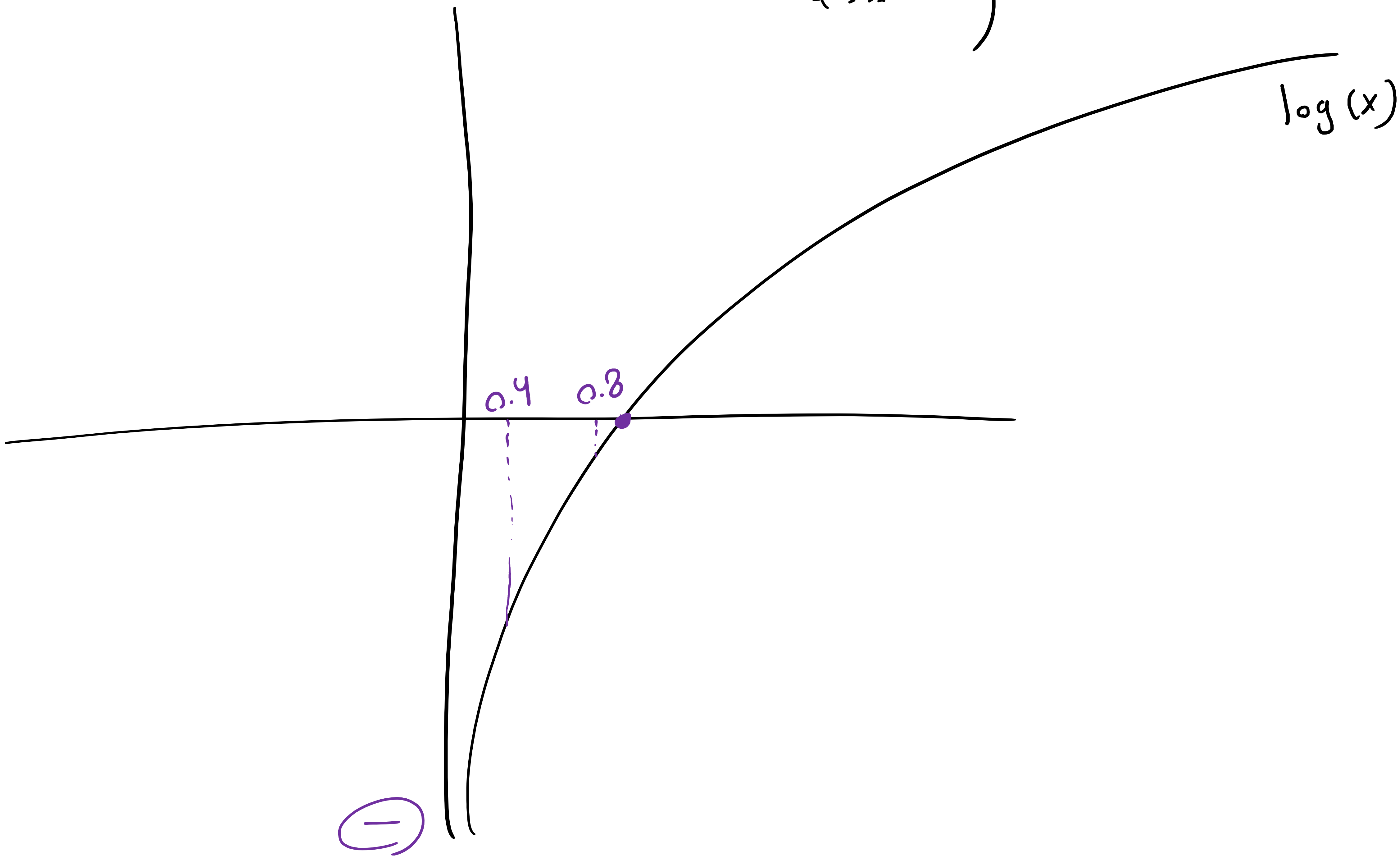
$$\| Y_a - Y_p \|_2^2 = \left[(1-0.8)^2 + (0-0.1)^2 + (0-0.1)^2\right] + \left[(0-0.2)^2 + (1-0.7)^2 + (0-0.1)^2\right] + \cdots$$

Max

$$\sum Y_a \cdot \hat{Y}_p = [1 \quad 0 \quad 0]\begin{bmatrix} 0.8 \\ 0.1 \\ 0.1 \end{bmatrix} + [0 \quad 1 \quad 0]\begin{bmatrix} 0.2 \\ 0.7 \\ 0.1 \end{bmatrix} + \cdots$$

Why Cross entropy and not simply use dot product?

$$H(p,q) = -\sum p(x)\log q(x) = -\left(\begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \log 0.8 \\ \log 0.1 \\ \log 0.1 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \log 0.2 \\ \log 0.7 \\ \log 0.1 \end{bmatrix} + \cdots \right)$$



Kullback-Leibler Divergence

Another useful information theoretic quantity measures the difference between two distributions.

$$\mathbf{KL}[P(S)\|Q(S)] = \sum_s P(s)\log\frac{P(s)}{Q(s)}$$

$$= \underbrace{\sum_s P(s)\log\frac{1}{Q(s)}}_{\text{cross entropy}} - \mathbf{H}[P] = H(P,Q) - H(P)$$

KL Divergence is a **KIND OF** distance measurement

Excess cost in bits paid by encoding according to Q instead of P.

$$-\mathbf{KL}[P\|Q] = \sum_s P(s)\log\frac{Q(s)}{P(s)}$$

log function is concave or convex?

$$\sum_s P(s)\log\frac{Q(s)}{P(s)} \le \log\sum_s P(s)\frac{Q(s)}{P(s)}$$

$$= \log\sum_s Q(s) = \log 1 = 0$$

By Jensen Inequality

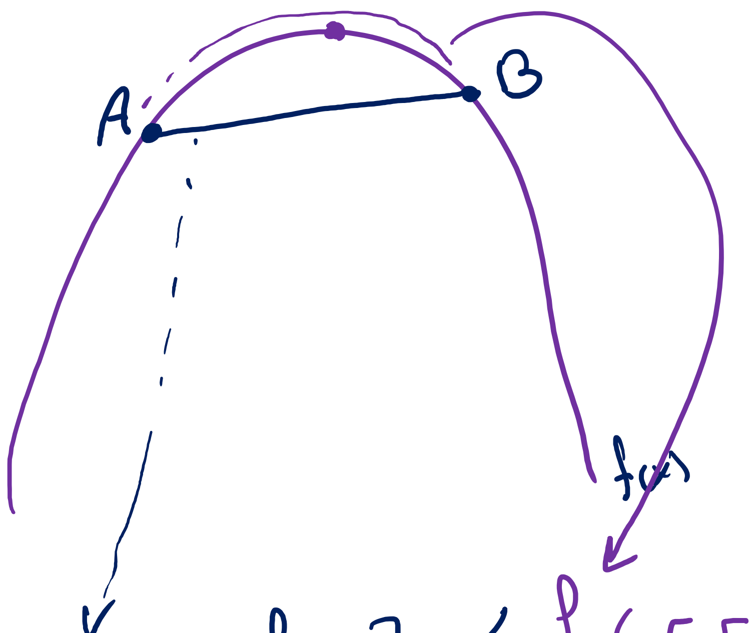
So $\mathbf{KL}[P\|Q] \ge 0$. Equality iff $P = Q$

When $P = Q,\ KL[P||Q] = 0$

Concave

$f(x) = -x^2$

$f''(x) = -2$

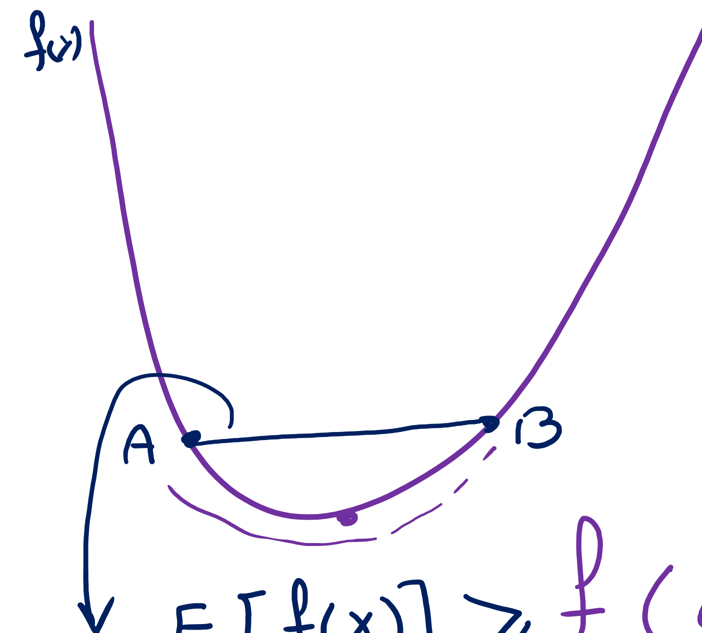


$E[f(x)] \leq f(E[x])$

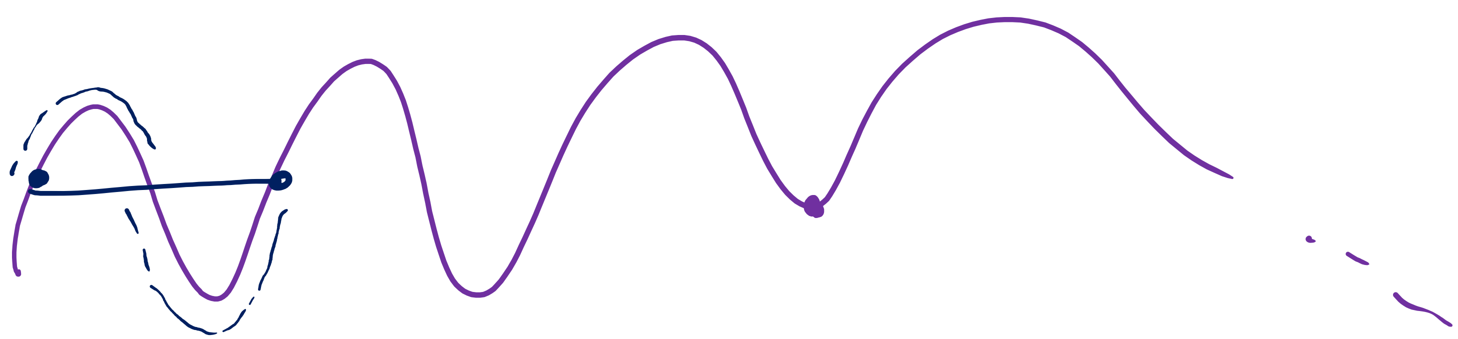
Convex

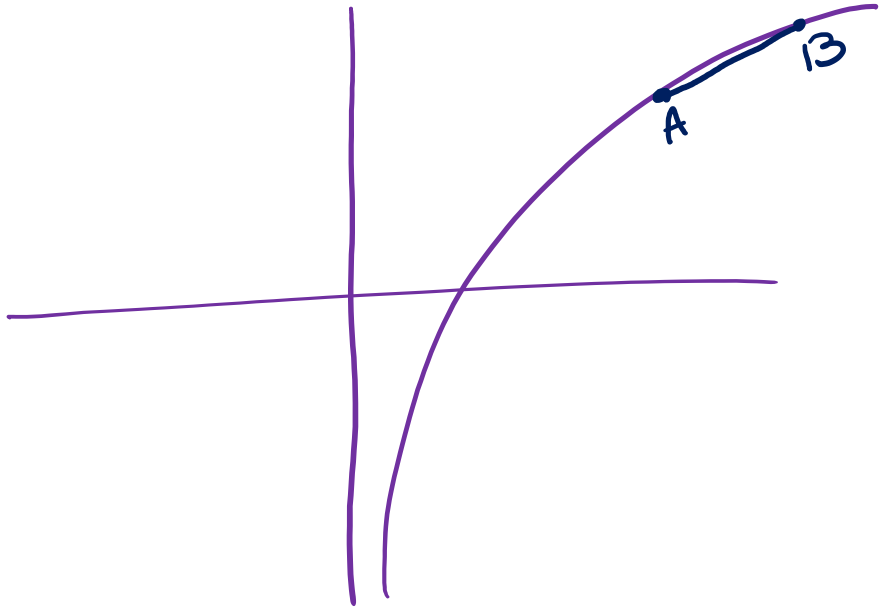
$f(x) = x^2$

$f''(x) = 2$



$E[f(x)] \geq f(E[x])$





$\log x = f(x)$

$E[f(x)] \leq f(E[x])$

$E[\log(x)] \leq \log(E[x])$

$$\frac{Q(s)}{P(s)} = g(s)$$

$$-KL[P][Q] = \sum P(s) \log \frac{Q(s)}{P(s)} = \sum \underbrace{P(s)} \underbrace{\log g(s)}$$

$$-KL[P][Q] = E[\log g(s)] \leq \log E[g(s)]$$

〃 $\leq \log \sum P(s)\, g(s)$

〃 $\leq \log \sum \cancel{P(s)} \frac{Q(s)}{\cancel{P(s)}}$

〃 $\leq \log \left(\underbrace{\sum Q(s)}\right)$

$$-KL[P][Q] \leq \log 1$$

$$KL[P][Q] \geq 0$$

Optimization

→ Probabilistic models → $N(X \mid \overset{a,b}{\mu, \Sigma})$ → MLE → L → $\frac{\partial L}{\partial \cdot} = 0$

→ Non-probabilistic models

No-constraint

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

$\frac{\partial f(x,y)}{\partial x} = 0$

$\frac{\partial f(x,y)}{\partial y} = 0$

Equality constraint

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

s.t. $x - y = 8$

→ lagrange function

$\frac{\partial L}{\partial \cdot} = 0$

Inequality constraint

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

s.t. $x - y \leq 8$

→ $\frac{\partial L}{\partial \cdot} = 0$

& Satisfy KKT conditions

$$f(M,S) = \underbrace{6M^2}_{} + \underbrace{3S^2}_{}$$

$$f'' = \begin{bmatrix} \frac{f^2}{\partial^2 M} & \frac{f^2}{\partial M \partial S} \\ \frac{f^2}{\partial S \partial M} & \frac{f^2}{\partial^2 S} \end{bmatrix}$$

M: # hours study ML

S: # " sleep

$$\frac{\partial f(M,S)}{\partial M} = 12M = 0 \Rightarrow M = 0$$

$$\frac{\partial f(M,S)}{\partial S} = 6S = 0 \Rightarrow S = 0$$

objective function

$$f(m,s) = 6m^2 + 3s^2$$

s.t. $m+s=24$

s.t. $m-s=10$

$$\Rightarrow \quad g(m,s) = m+s-24=0$$

$$h(m,s) = m-s-10$$

constraint function

λ, α, β

lagrange mulitiplier

$$L(m,s,\lambda_1,\lambda_2) = f(m,s) - \lambda_1 g(m,s) - \lambda_2 h(m,s)$$

$$f(m,s) = 6m^2+3s^2$$

s.t. $m+s=24$

$$L(m,s,\lambda) = 6m^2+3s^2 - \lambda(m+s-24)$$

$$\frac{\partial L}{\partial m}=0 \Rightarrow 12m-\lambda=0 \Rightarrow m=\frac{\lambda}{12}=\frac{96}{12}=8$$

$m+s=24$

$8+16=24$

$$\frac{\partial L}{\partial s}=0 \Rightarrow 6s-\lambda=0 \Rightarrow s=\frac{\lambda}{6}=\frac{96}{6}=16$$

$$\frac{\partial L}{\partial \lambda}=0 \Rightarrow -(m+s-24)=0 \Rightarrow m+s=24 \Rightarrow \frac{\lambda}{12}+\frac{\lambda}{6}=24 \Rightarrow \lambda=96$$

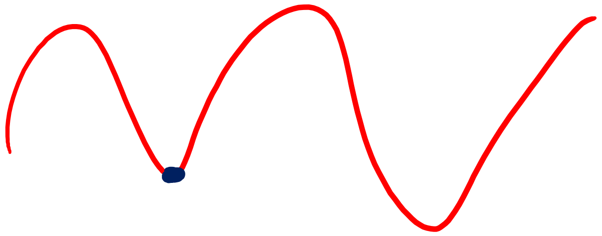
$$\nabla f(m,s) \sim \nabla g(m,s)$$
$$\Downarrow$$
$$\nabla f(m,s) = \lambda \nabla g(m,s)$$

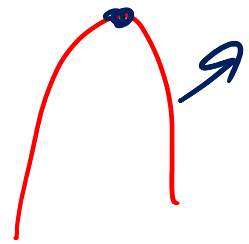
$$L(m,s,\lambda) = f(m,s) - \lambda\, g(m,s)$$

$$\nabla_{m,s,\lambda} L(m,s,\lambda) = 0$$

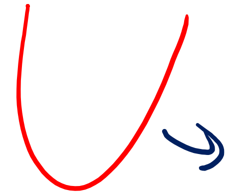
$$\nabla\big(f(m,s) - \lambda\, g(m,s)\big) = 0$$

$$\nabla f(m,s) - \cancel{\lambda}\, \nabla g(m,s) = 0$$

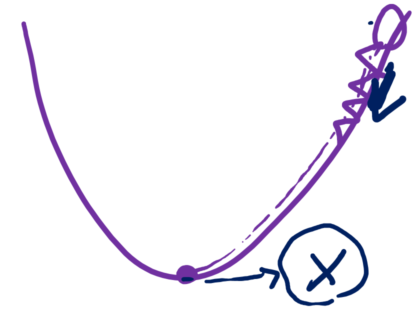




GA



GD



$f(x) = x^2$

$\frac{\partial f(x)}{\partial x} = 2x$

learning step $= 0.002$

$$x^{\{t+1\}} \leftarrow x^{\{t\}} - \alpha \frac{\partial f(x)}{\partial x} \rightarrow GD$$

$$x^{\{t+1\}} \leftarrow x^{\{t\}} - \alpha\, 2x^{\{t\}}$$

$$L(\mu, s, \delta) = f(\mu, s) - \delta\, g(\mu, s) \qquad \mu, s, \delta$$

$$\mu^{\{t+1\}} \leftarrow \mu^{\{t\}} - \alpha \frac{\partial L}{\partial \mu}$$

$$s^{\{t+1\}} \leftarrow s^{\{t\}} - \alpha \frac{\partial L}{\partial s}$$

$$\delta^{\{t+1\}} \leftarrow \delta^{\{t\}} - \alpha \frac{\partial L}{\partial \delta}$$

Min $f(M,S) = 6M^2 + 3S^2$

s.t. $M + S \leq 24 \Rightarrow \underbrace{M + S - 24 \leq 0}_{}$

$g(M,S) \leq 0$

KKT conditions

1. Stationary condition $\Rightarrow \underset{\text{Min}}{L} = \underbrace{6M^2 + 3S^2}_{\text{Min}} + \delta (M + S - 24)$
2. Primal feasibility $g(M,S) \leq 0$
3. Dual feasibility $\delta \geq 0$
4. Complementary slackness $g(M,S)\,\delta = 0$

$g(M,S)\,\delta = 0 \rightarrow g(M,S) = 0 \rightarrow$ Active $\rightarrow \delta \geq 0$

$g(M,S)\,\delta = 0 \rightarrow g(M,S) \neq 0 \rightarrow$ Inactive solution $\rightarrow \delta = 0$

Using Gradient Descent **as an alternative** to solve the equality constraint optimization example

This won't converge; do you know why?

```
import numpy as np

def minimize_gd():
    LEARNING_RATE = 0.01
    TOLERANCE = 1e-6
    # Initialize M, S, and lambda (make sure M + S = 24)
    M = 12.0
    S = 12.0
    lm = 0.0  # Lagrange Multiplier

    while True:
        # Calculate the gradients of the Lagrangian w.r.t. M, S, and lambda
        gradientM = 12 * M - lm
        gradientS = 6 * S - lm
        gradientLambda = - (M + S - 24)

        # Update M, S, and lambda
        newM = M - LEARNING_RATE * gradientM
        newS = S - LEARNING_RATE * gradientS
        newLambda = lm - LEARNING_RATE * gradientLambda

        # If the changes in M, S, and lambda are smaller than the tolerance, we break the loop
        if np.abs(newM - M) < TOLERANCE and np.abs(newS - S) < TOLERANCE and np.abs(newLambda - lm) < TOLERANCE:
            break

        M = newM
        S = newS
        lm = newLambda

    print("Minimum occurs at M = ", M , ", S = " , S , ", with lambda = ", lm)
    print("Minimum value of z = " , (6*M * M + 3*S * S))

minimize_gd()
```

Example 1:

https://www.geogebra.org/3d/srzmv8uh

Example 2:

https://www.geogebra.org/3d/syhkqpk7