Artificial Intelligent – Week 2

Dr Salem Ameen
S.A.AMEEN1@SALFORD.AC.UK

Lecture 3: Adversarial Search
In Week 1&2:

- AI
- Search
- Logic

Lecture 3: Optimization & Uncertainty

- Optimization
- Uncertainty
  - Estimation

Lecture 3: AI Paradigm

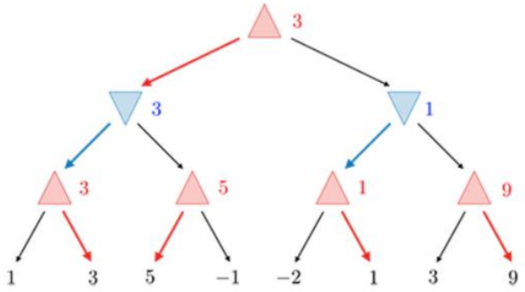
University of Salford MANCHESTER

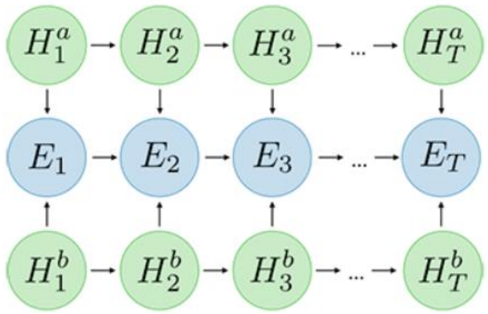
Modeling

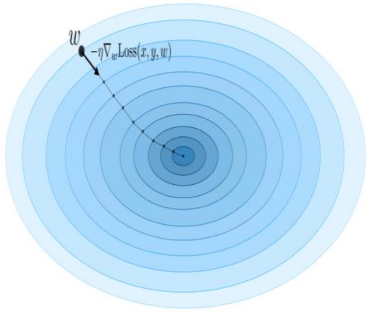
Inference

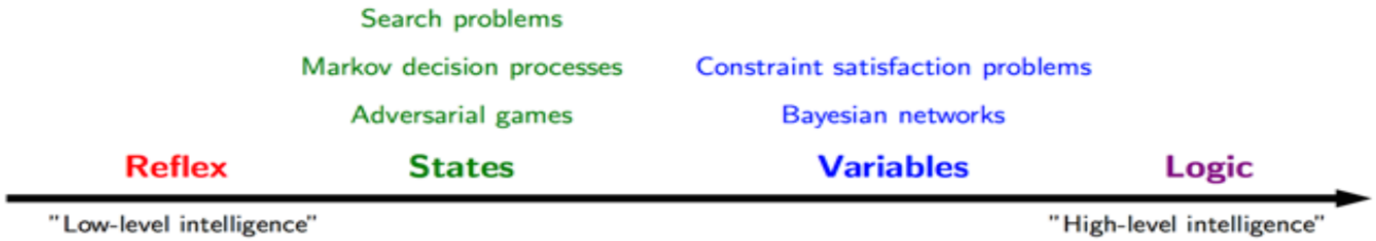
Learning

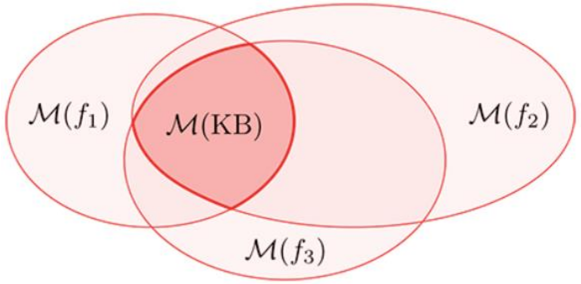
Lecture 3: Agent Types











Credit: Courtesy Percy Liang

Lecture 3: Optimization & Uncertainty

University of Salford MANCHESTER

Part 1 : Optimization

Lecture 3 – Part 1: Outline

- Optimization :
  - Minimize, Maximize
- Algorithms:
  - Hill Climbing
  - Simulated Annealing
  - Genetic algorithm
- Examples
  - Travelling salesman problem
  - 8 queen

Lecture 3: Optimization

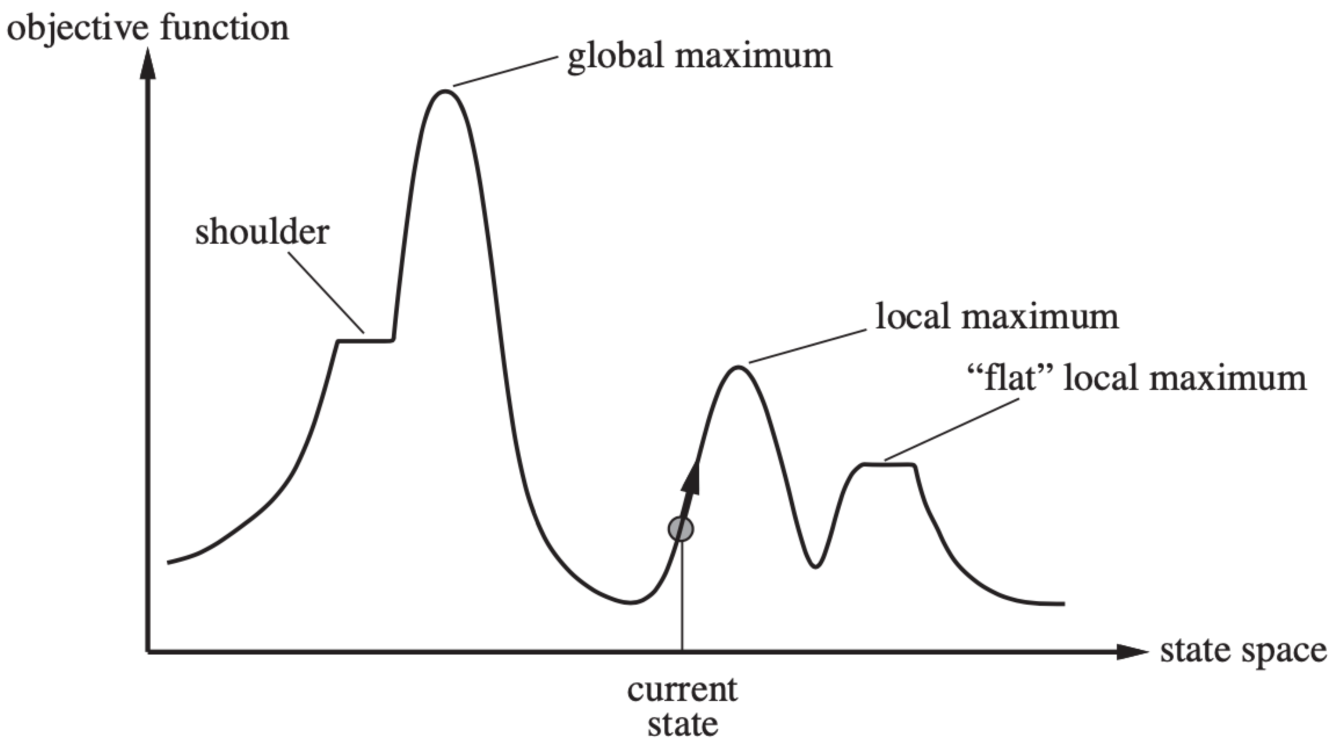
If you want to minimize $f(x)$ and your optimizer program seeks to maximize the objective function, then define $g(x) = -f(x)$ and find $\max(g(x)) = \max(-f(x)) = \min(f(x))$.

Similarly, if your optimization problem needs to maximize $f(x)$ and your optimizer program seeks to minimize the objective function, define $g(x) = -f(x)$ and find $\min(g(x)) = \min(-f(x)) = \max(f(x))$

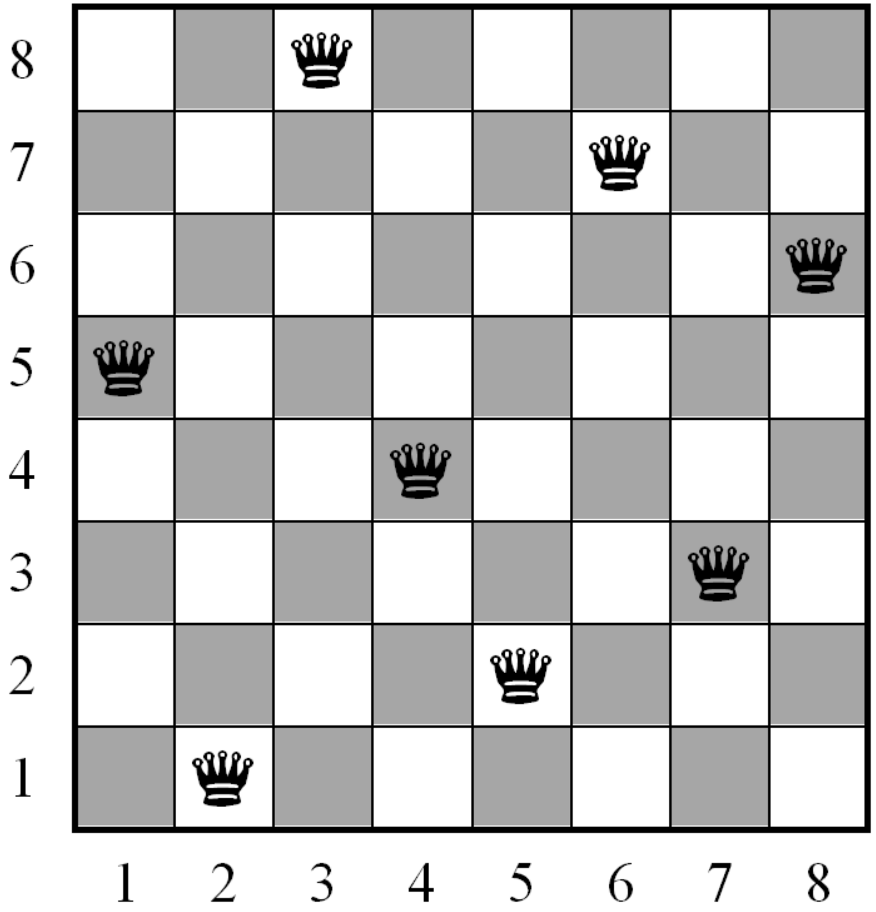
Lecture 3: Optimization

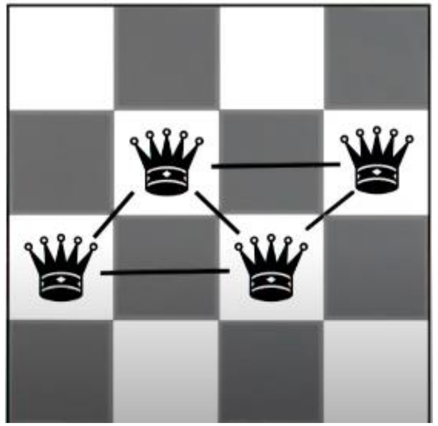
The potential limitation in the optimization algorithm, where it can get stuck in one of the local minima (plural) and fails to reach the global minimum (singular) in the problem's state space.

- not convex



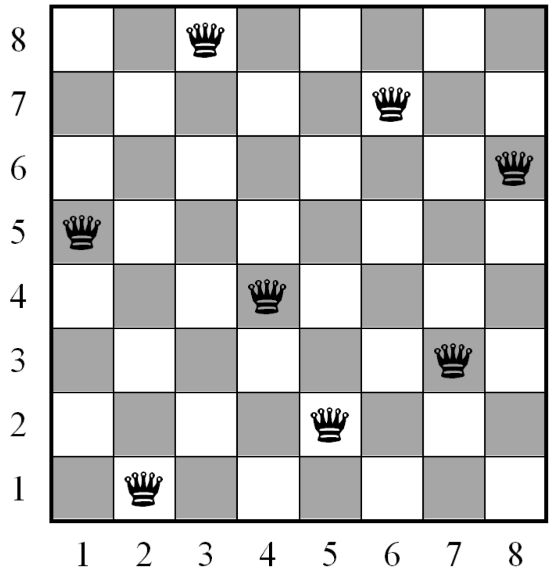
Lecture 3: Optimization
Example : 8 queen: Representation



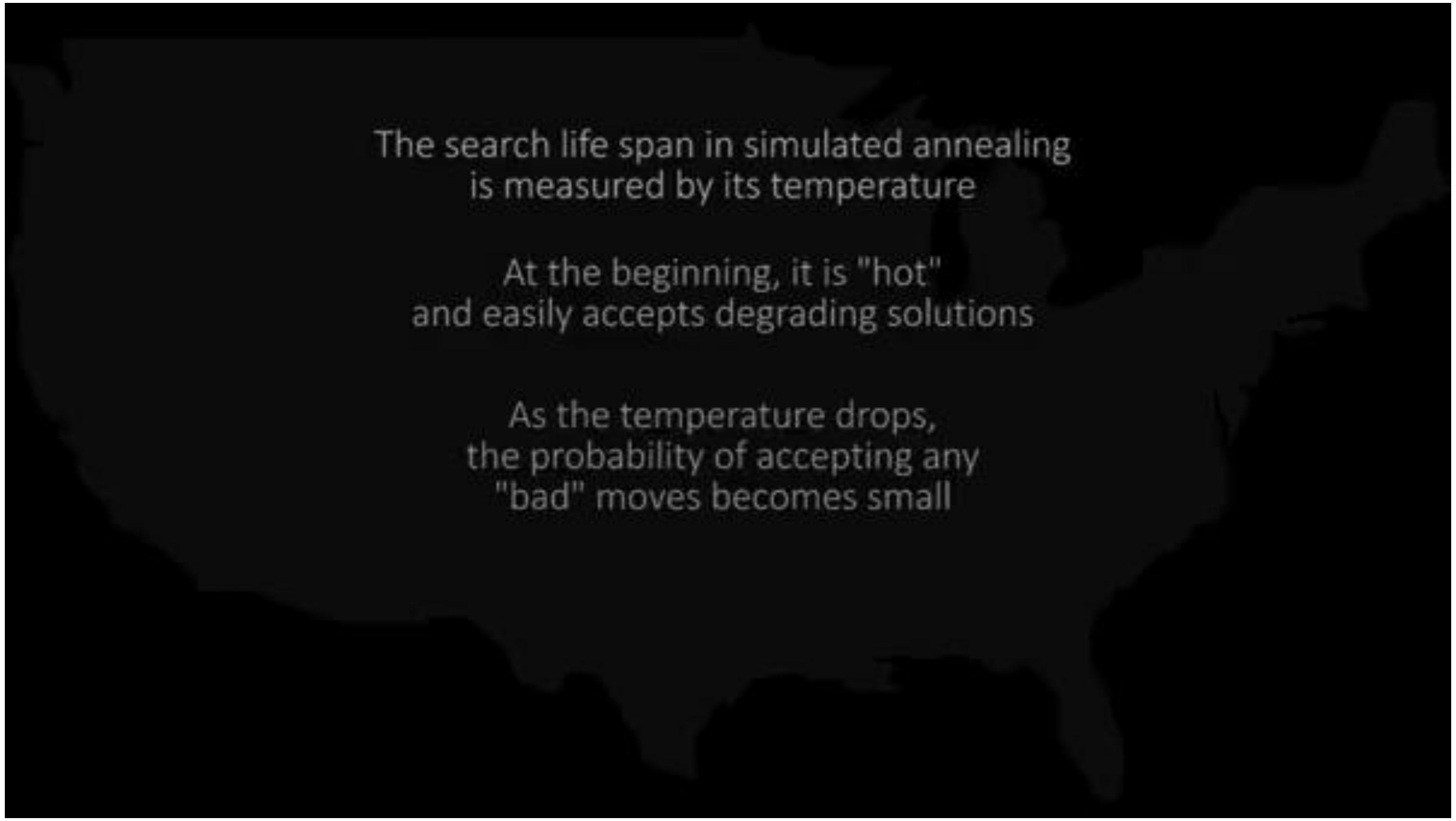


Lecture 3: Optimization
Example : 8 queen: Representation

In the n-Queens problem, we call local or global minima since the objective function to minimize the value. When the objective function is to maximize the value, we call the peaks as either **local** or **global** maxima as shown in the figure below.



Lecture 3: Optimization
Example : Travelling salesman problem

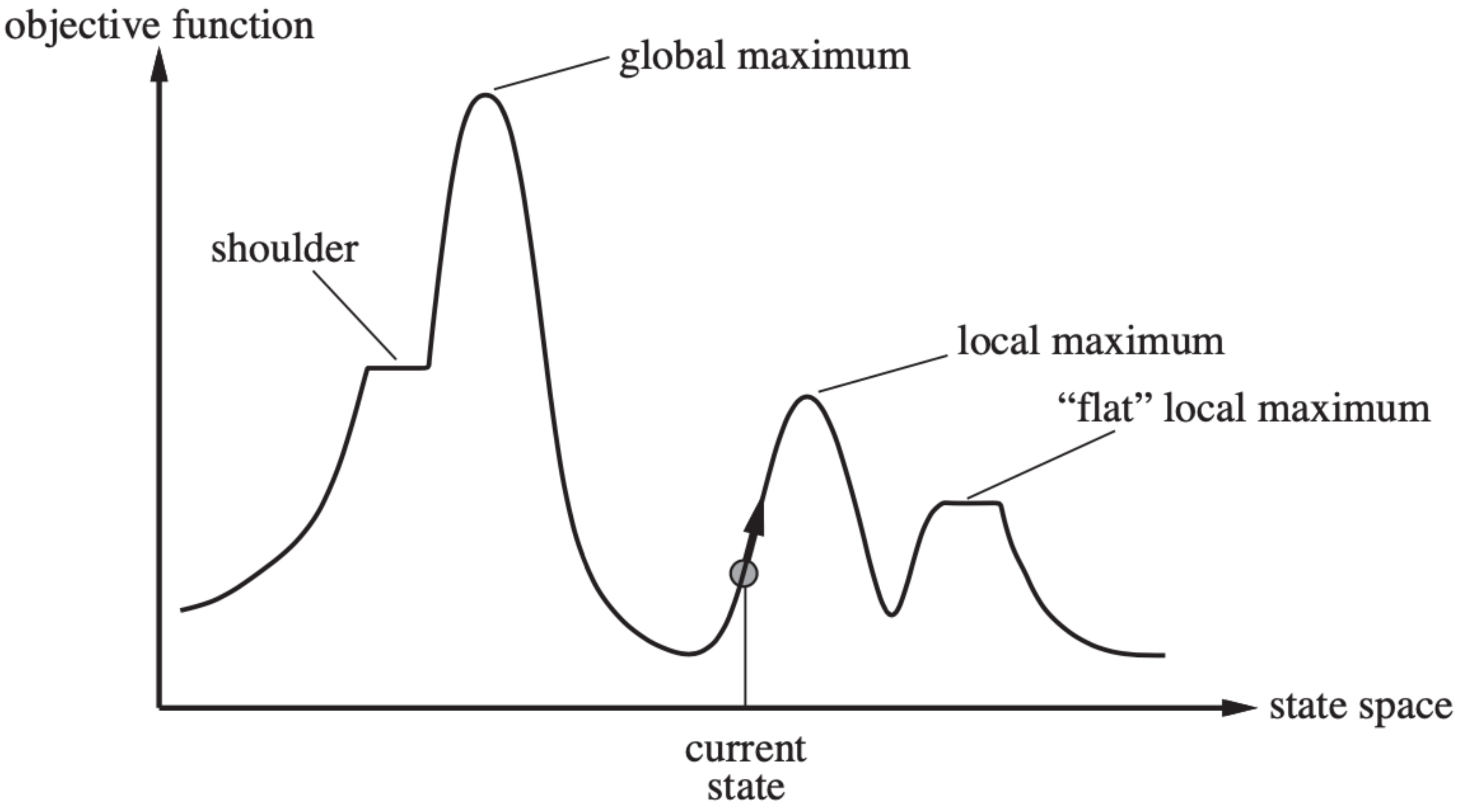


Optimization - Hill Climbing Algorithm

Hill Climbing is also known as ***greedy local search***.

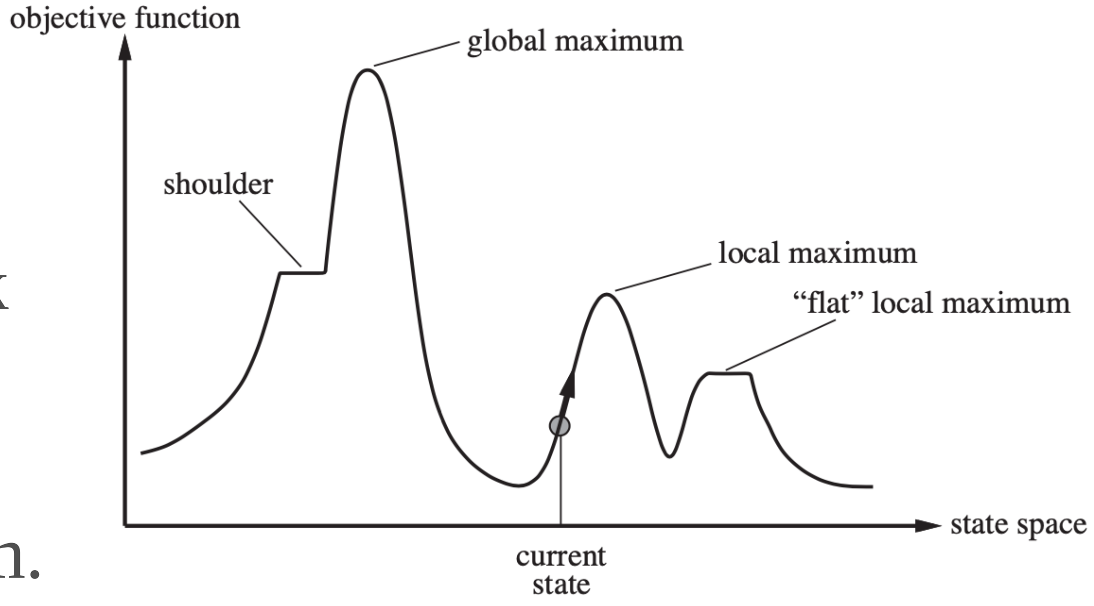
- This algorithm only looks to the immediate neighbours without knowing where to go next.
- The algorithm evaluates the values of immediate neighbours and continually moves to the direction of the increasing value, hence the name "hill climbing".
- The algorithm will terminate at a *peak* where there are no higher values among the neighbours.

Optimization - Hill Climbing Algorithm



Optimization - Hill Climbing Algorithm

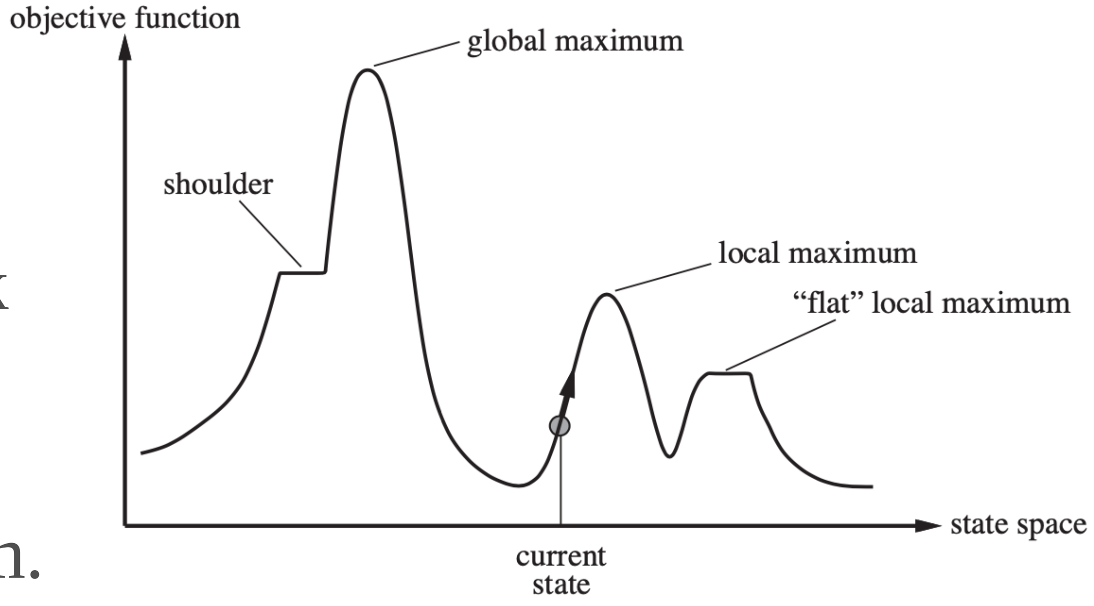
- Hill Climbing algorithm is said to be incomplete because it can get stuck in a local maximum and does not guarantee finding the global maximum.



Solution ???

Optimization - Hill Climbing Algorithm

- Hill Climbing algorithm is said to be incomplete because it can get stuck in a local maximum and does not guarantee finding the global maximum.
- Solution : Random Restart



Optimization - Simulated Annealing algorithm

University of Salford MANCHESTER

- **SA:** is inspired by the *annealing* process in metallurgy.
- An annealing process reshapes a hard metal or glass by exposing it to a high temperature and gradually cool it down until it maintains the new shape.
- Unlike the Hill Climbing algorithm, which can get stuck in the local maxima, Simulated Annealing is guaranteed to find the global maximum.

Optimization - Simulated Annealing algorithm

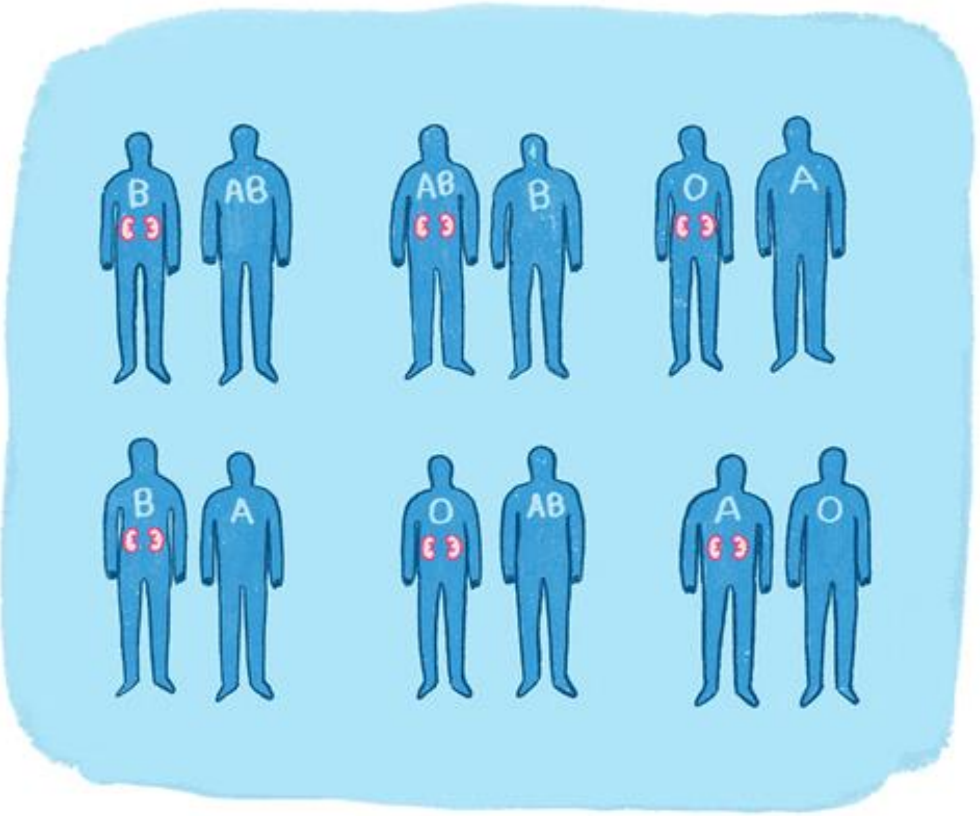
```
For t=1 to ∞ do
    T ← SCHEDULE(t)
    if T=0: return current
    next ← GET_RANDOM_SUCCESSOR(current)
    if Δe>0: current ← next
    else: current ← next with probability e^(ΔE/T)
```

schedule(t) function, the temperature will be decreased at each step. When temperature is zero, it will return the current state.

Optimization - Simulated Annealing algo. Example : Travelling salesman problem

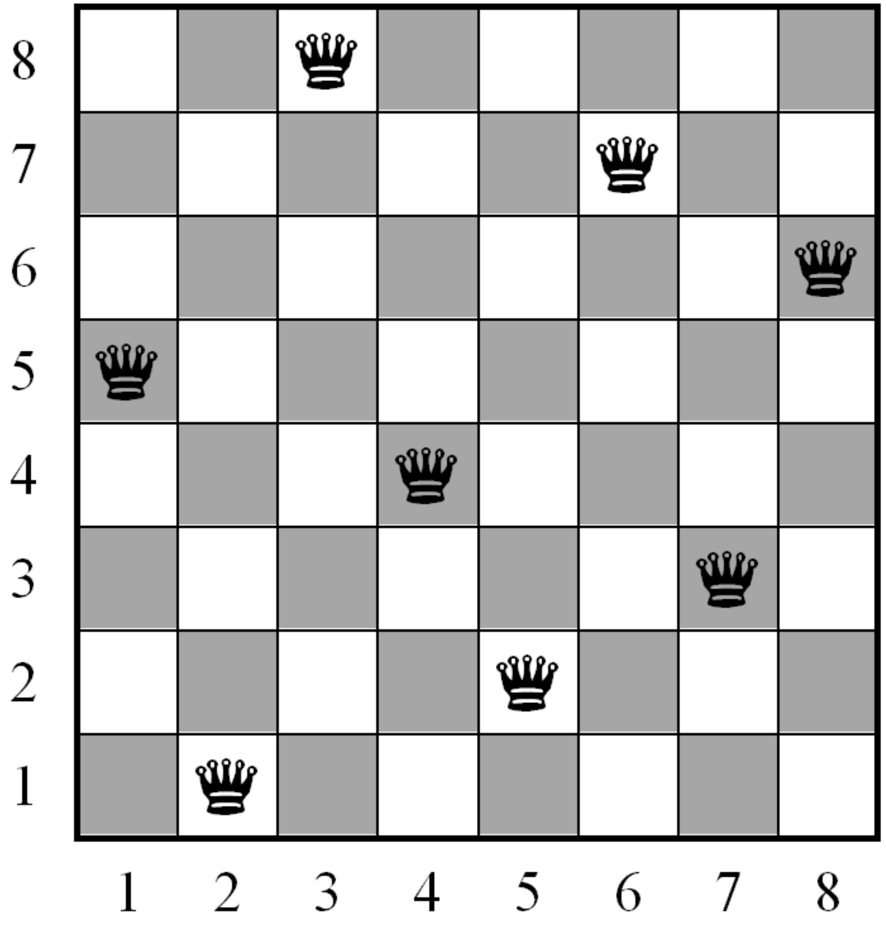
Code

Lecture 3: Optimization - Genetic algorithm

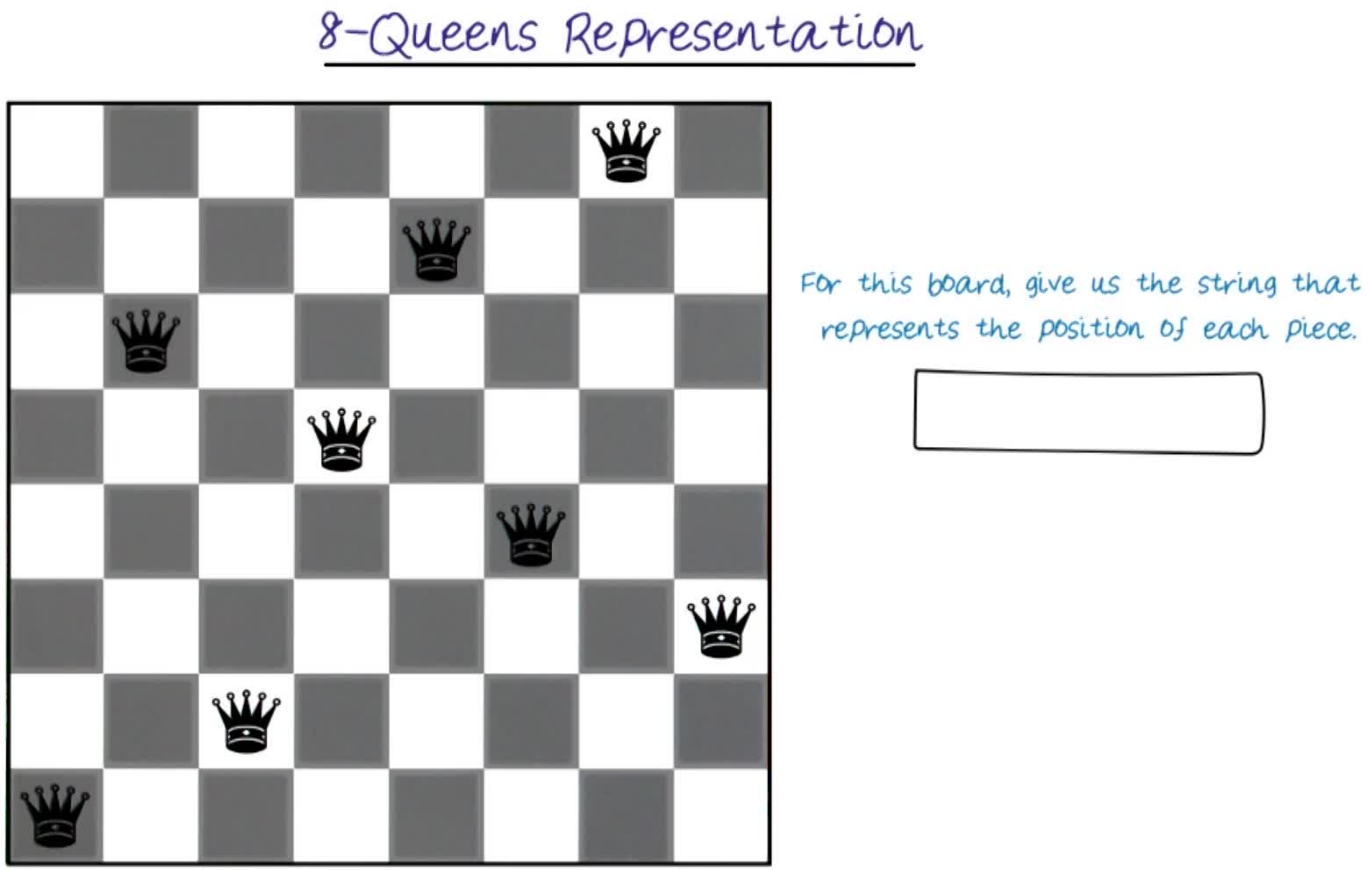


Blood types: A, B, AB and O

Optimization - Genetic algorithm Example : 8 queen: Representation

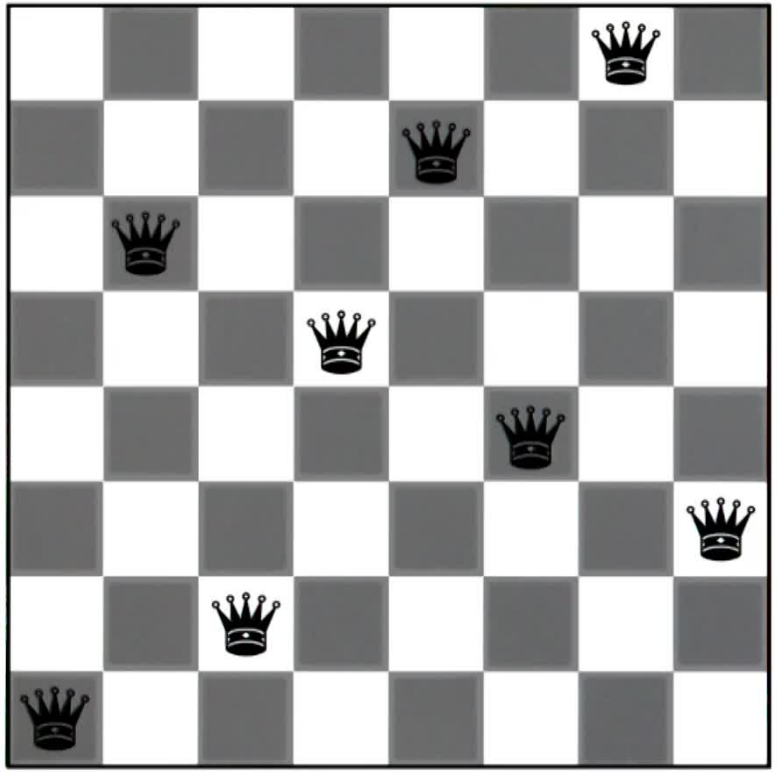


Optimization - Genetic algorithm Example : 8 queen: Representation



Optimization - Genetic algorithm Example : 8 queen: Representation

8-Queens Representation

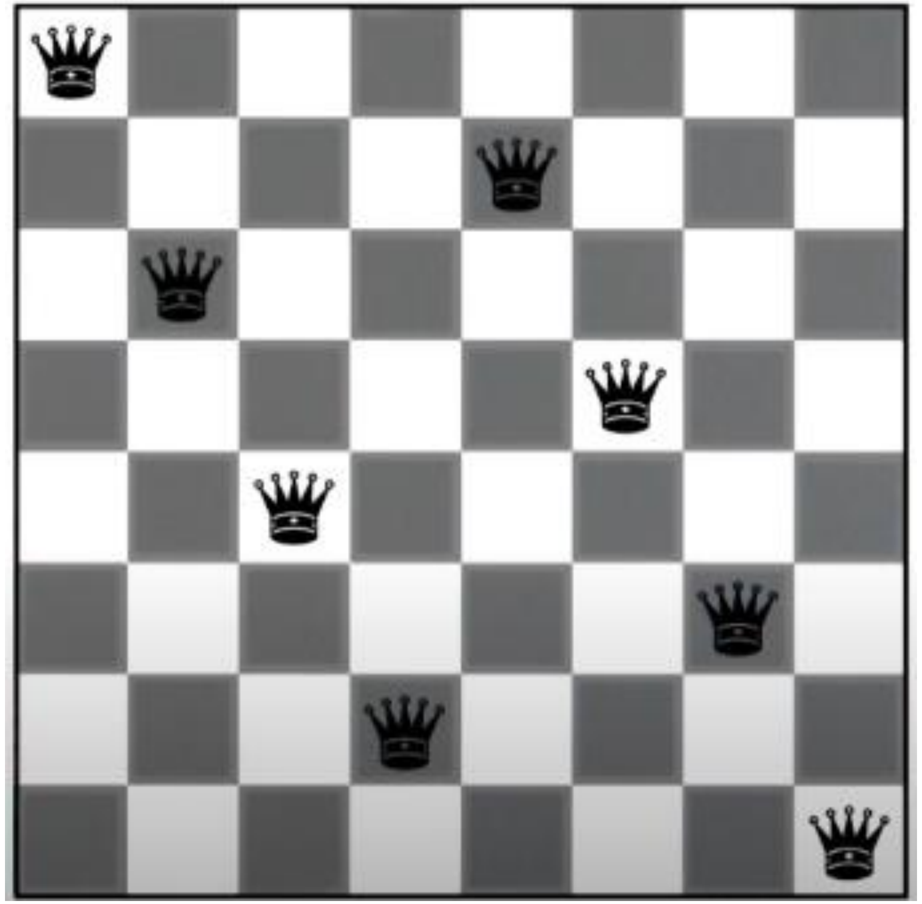


For this board, give us the string that represents the position of each piece.

16257483

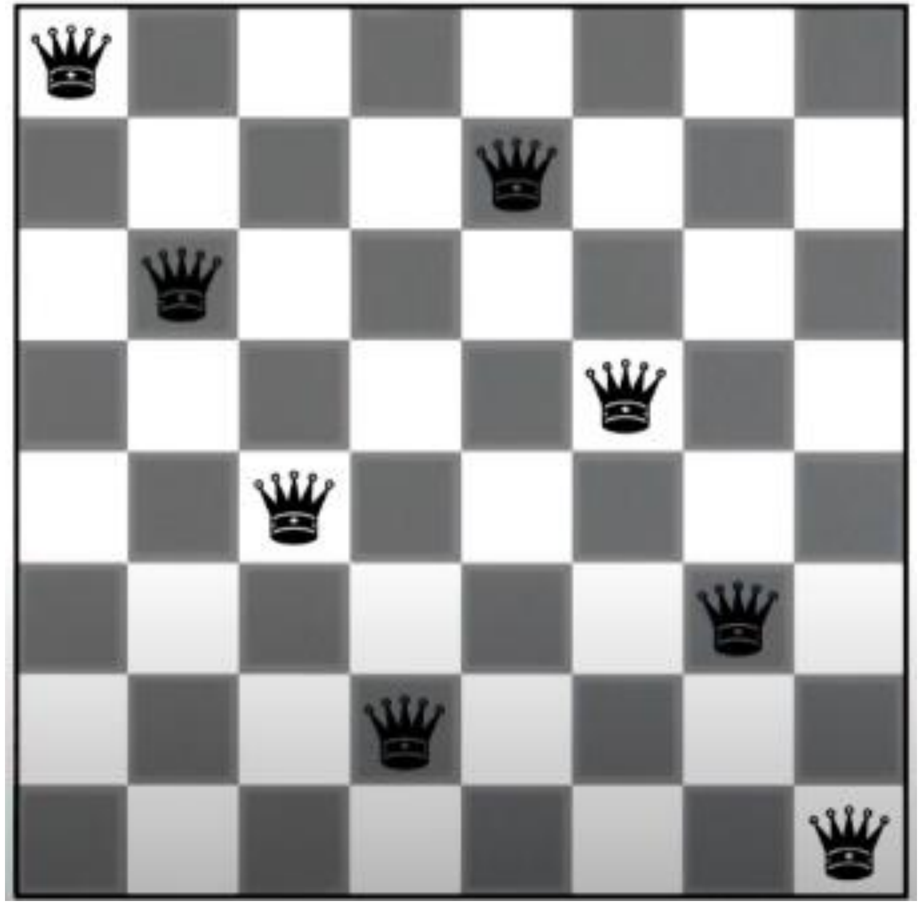
Optimization - Genetic algorithm Example : 8 queen

Fitness function ??



Optimization - Genetic algorithm Example : 8 queen

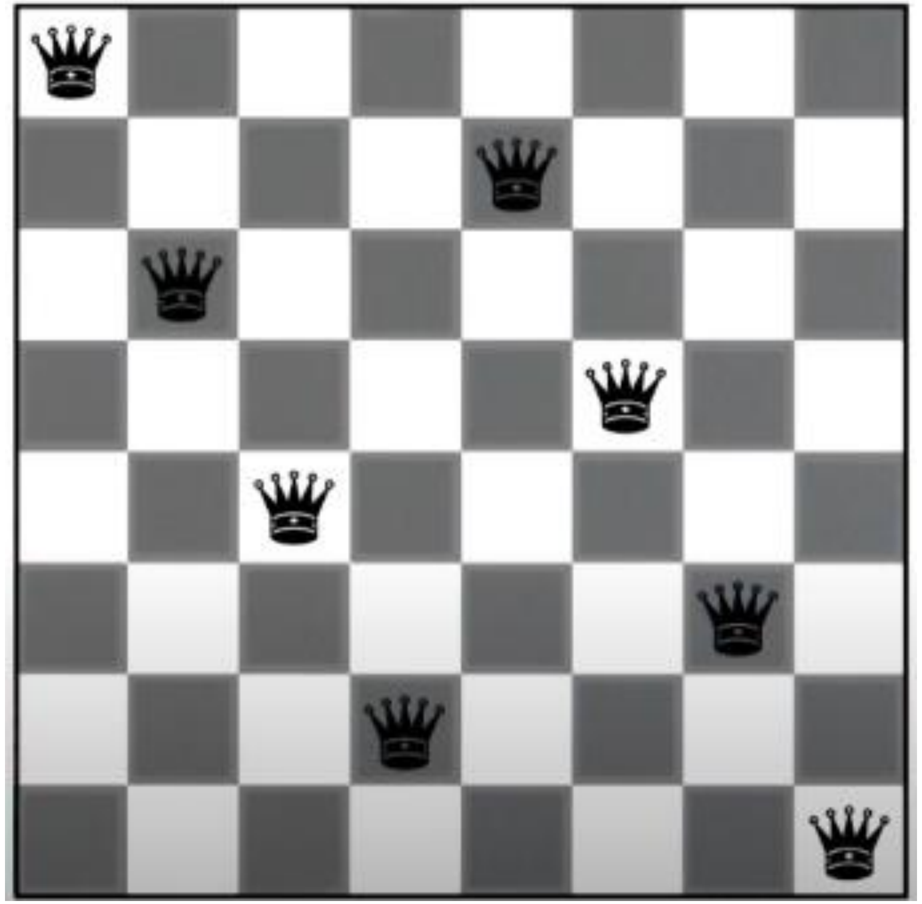
Fitness function = non-attacking



Optimization - Genetic algorithm Example : 8 queen

Fitness function = non-attacking

non-attacking = Total-attacking – attacking

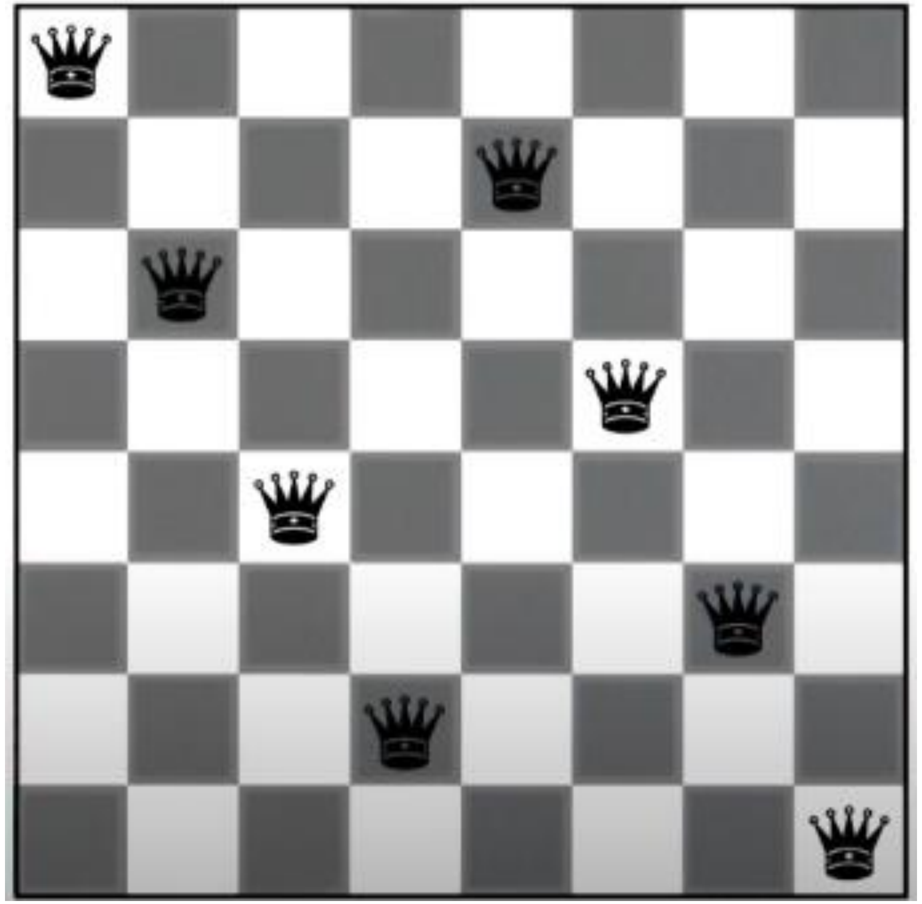


Optimization - Genetic algorithm Example : 8 queen

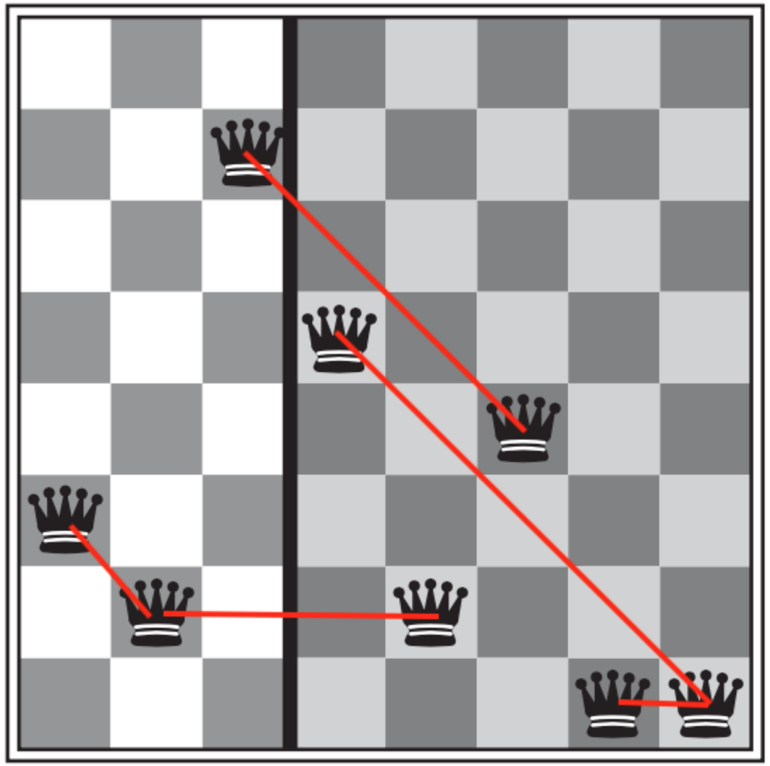
Fitness function = non-attacking

non-attacking = Total-attacking – attacking

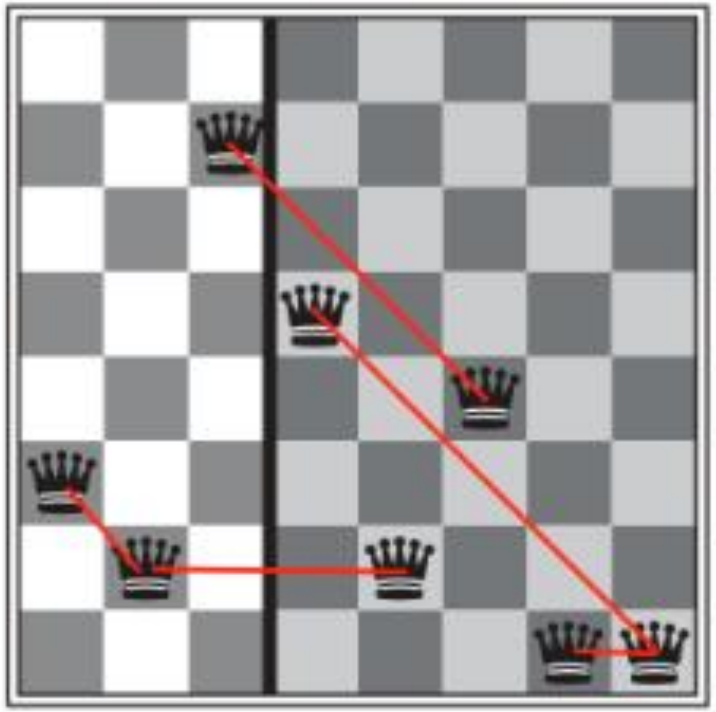
Total-attacking The maximum fitness value for this problem is 28 (8 chooses 2)



Optimization - Genetic algorithm Example : 8 queen



Optimization - Genetic algorithm Example : 8 queen



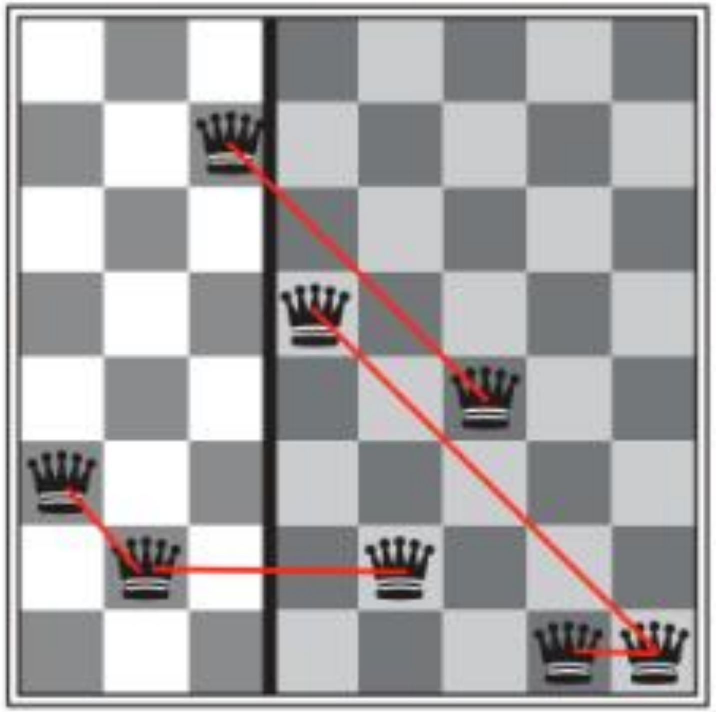
Board state 32752411 with 5 pairs of attacking queens

Optimization - Genetic algorithm
Example : 8 queen

Fitness function = non-attacking

= Total-attacking – attacking

= 28 – 5 = 23



Board state 32752411 with 5 pairs of attacking queens

Optimization - Genetic algorithm Example : 8 queen

24748552

32752411

24415124

32543213

Optimization - Genetic algorithm Example : 8 queen

Fitness function

24748552	??
32752411	??
24415124	??
32543213	??

Optimization - Genetic algorithm Example : 8 queen

Fitness function

24748552	24
32752411	23
24415124	20
32543213	11

Optimization - Genetic algorithm Example : 8 queen

Fitness function

24748552	**24**	**??**
32752411	**23**	**??**
24415124	**20**	**??**
32543213	**11**	**??**

Optimization - Genetic algorithm Example : 8 queen

Fitness function

24748552	24	31%
32752411	23	29%
24415124	20	26%
32543213	11	14%

Optimization - Genetic algorithm Example : 8 queen

Fitness function

24748552	24	31%
32752411	23	29%
24415124	20	26%
32543213	11	14%

Optimization - Genetic algorithm
Example : 8 queen

Fitness function

55

24748552	24	31%
32752411	23	29%
24415124	20	26%
32543213	11	14%

Optimization - Genetic algorithm Example : 8 queen

Fitness function

55

24748552	24	31%	32752411
32752411	23	29%	
24415124	20	26%	
32543213	11	14%	

Optimization - Genetic algorithm Example : 8 queen

Fitness function

11

24748552	24	31%	32752411
32752411	23	29%	
24415124	20	26%	
32543213	11	14%	

Optimization - Genetic algorithm Example : 8 queen

Fitness function

11

24748552	24	31%	32752411
32752411	23	29%	24748552
24415124	20	26%	
32543213	11	14%	

Optimization - Genetic algorithm
Example : 8 queen

Fitness function

33

24748552	24	31%	32752411
32752411	23	29%	24748552
24415124	20	26%	
32543213	11	14%	

Optimization - Genetic algorithm Example : 8 queen

Fitness function

80

24748552	24	31%	32752411
32752411	23	29%	24748552
24415124	20	26%	32752411
32543213	11	14%	

Optimization - Genetic algorithm Example : 8 queen

Fitness function

80

24748552	24	31%	32752411
32752411	23	29%	24748552
24415124	20	26%	32752411
32543213	11	14%	24415124

Optimization - Genetic algorithm Example : 8 queen

Fitness function

24748552	24	31%	32752411
32752411	23	29%	24748552
24415124	20	26%	32752411
32543213	11	14%	24415124

Optimization - Genetic algorithm Example : 8 queen

Fitness function

24748552	24	31%
32752411	23	29%
24415124	20	26%
32543213	11	14%

Crossover

32752411	32748552
24748552	24752411
32752411	
24415124	

Optimization - Genetic algorithm Example : 8 queen

University of Salford MANCHESTER

Fitness function

Crossover

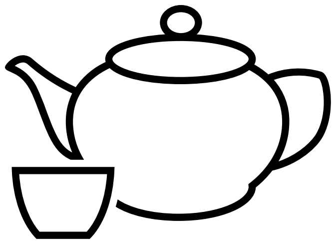
24748552	24	31%	32752411	32748552
32752411	23	29%	24748552	24752411
24415124	20	26%	32752411	32752411
32543213	11	14%	24415124	24415124

Optimization - Genetic algorithm Example : 8 queen

Fitness function			Crossover		Mutation = Randomness
24748552	24	31%	32752411	32748552	32748512
32752411	23	29%	24748552	24752411	24752411
24415124	20	26%	32752411	32752411	32252411
32543213	11	14%	24415124	24415124	24415127

Lecture 3 – Part 2

Part 2 : Uncertainty



Lecture 3 – Part 2: Outline

- Probabilities:
  - Dependence, Independence, Conditional Independence
- Parameter estimation:
  - Maximum Likelihood Estimation (MLE)
  - Maximum Aposteriori (MAP)
- Anomaly detection

Lecture 3 - Outline

- Probabilities:
  - Dependence, Independence, Conditional Independence
- Parameter estimation:
  - Maximum Likelihood Estimation (MLE)
  - Maximum Aposteriori (MAP)
- Anomaly detection

Lecture 3: Probability

Probability

- **Classical**, Naïve, Equally likely,
- ***Relative***, *frequency, data, by experimentation*
- **Subjective**

Lecture 3: Conditional Probability

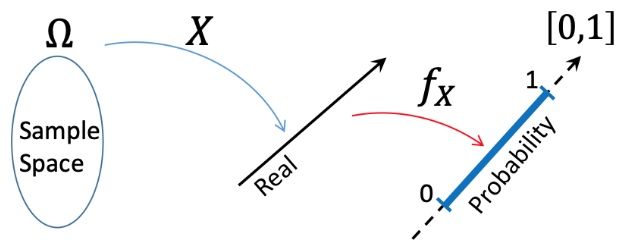
The soul of statistics

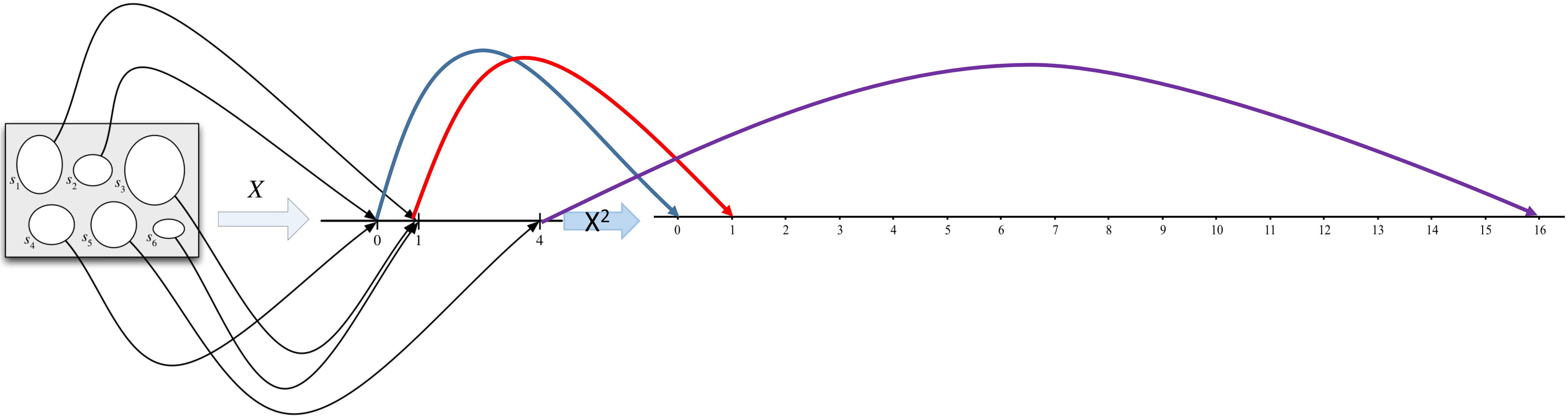
Three important tools:

- Multiplication rule
- Independent
- Total probability theorem
- Bayes' rule ($\rightarrow$ inference)

Lecture 3: Random Variables

DEFINITION (RANDOM VARIABLE).



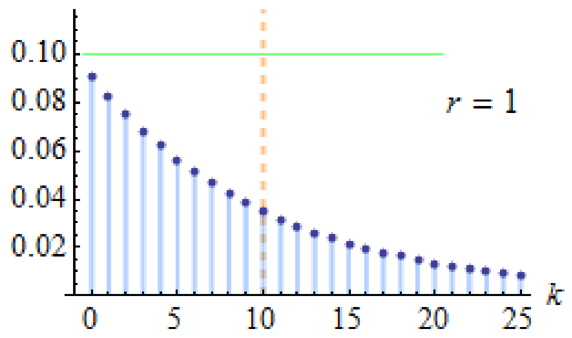


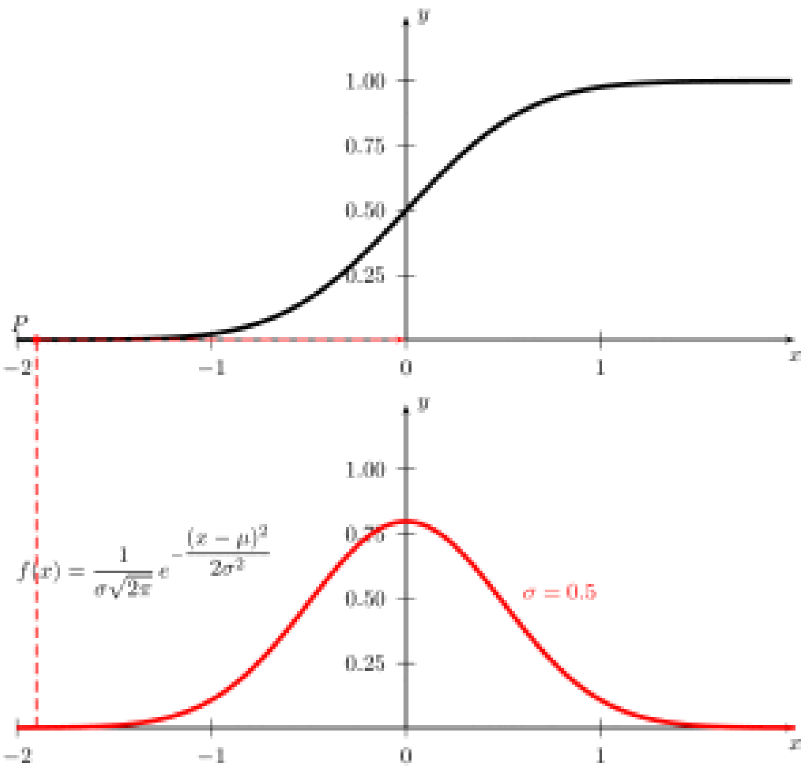
Lecture 3: Random Variables

- Random Variables
  - Discrete Random Variables
  - Continuous Random Variables
  - Combination
    - $[0,2] \cup \{ e , 2.7\}$

Lecture 3: Random Variables

- **CUMULATIVE DISTRIBUTION FUNCTION (CDF)**
- **PROBABILITY MASS FUNCTION (PMF)**
- **PROBABILITY DENSITY FUNCTION (PDF)**





Lecture 3: Random Variables

	Discrete	Continuous
Prob. Fun.	pmf - p	pdf - f
≥ 0	$p(x) \geq 0$	$f(x) \geq 0$
∑ = 1	$\sum p(x) = 1$	$\int f(x)dx = 1$
P(A)	$\sum_{x \in A} p(x)$	$\int_{x \in A} f(x)dx$
F(X)	$\sum_{u \leq x} p(u)$	$\int_{-\infty}^{x} f(u)du$
μ = E(X)	$\sum xp(x)$	$\int xf(x)dx$
V(X)	$\sum (x-\mu)^2 p(x)$	$\int (x-\mu)^2 f(x)dx$

Lecture 3: Random Variables

- Bernoulli
- Binomial
- Poisson
- Geometric
- Negative Binomial (Pascal)
- Hypergeometric
- Discrete Uniform
- More

- Uniform
- Normal
- Exponential
- The Cauchy
- More

Lecture 3 - Parameter Estimation

- Probabilities:
  - Dependence, Independence, Conditional Independence
- Parameter estimation:
  - Maximum Likelihood Estimation (MLE)
  - Maximum Aposteriori (MAP)
- Anomaly detection

Lecture 3: Maximum Likelihood Estimation (MLE for Binomial)

Data, D =

$$D = \{X_i\}_{i=1}^{n}, \;\; X_i \in \{\mathrm{H}, \mathrm{T}\}$$

$$P(Heads) = \theta, \;\; P(Tails) = 1\text{-}\theta$$

MLE: Choose θ that maximizes the probability of observed data

Lecture 3: MLE for Binomial

- MLE: Choose θ that maximizes the probability of observed data

$$\hat{\theta}_{MLE} = \underset{\theta}{\operatorname{argmax}}\ P(D;\theta)$$

Lecture 3: MLE for Binomial

- MLE: Choose θ that maximizes the probability of observed data

$$p(X = k) = \binom{n}{k} p^k q^{n-k} \text{ for } x = 0, 1, 2, 3 \ldots, n \text{ and } q = 1 - p.$$

$$\begin{aligned}\hat{\theta}_{MLE} &= \underset{\theta}{\operatorname{argmax}}\ P(D;\theta) \\ &= \underset{\theta}{\operatorname{argmax}} \binom{n_H + n_T}{n_H} \theta^{n_H} (1-\theta)^{n_T}\end{aligned}$$

Lecture 3: MLE for Binomial

- MLE: Choose θ that maximizes the probability of observed data

$$
\begin{aligned}
\hat{\theta}_{MLE} &= \underset{\theta}{\operatorname{argmax}}\ P(D;\theta) \\
&= \underset{\theta}{\operatorname{argmax}} \binom{n_H + n_T}{n_H} \theta^{n_H} (1-\theta)^{n_T} \\
&= \underset{\theta}{\operatorname{argmax}}\ \log \binom{n_H + n_T}{n_H} + n_H \cdot \log(\theta) + n_T \cdot \log(1-\theta)
\end{aligned}
$$

Lecture 3: MLE for Binomial

- MLE: Choose θ that maximizes the probability of observed data

$$
\begin{aligned}
\hat{\theta}_{MLE} &= \underset{\theta}{\operatorname{argmax}}\ P(D;\theta) \\
&= \underset{\theta}{\operatorname{argmax}} \binom{n_H + n_T}{n_H} \theta^{n_H} (1-\theta)^{n_T} \\
&= \underset{\theta}{\operatorname{argmax}}\ \log \binom{n_H + n_T}{n_H} + n_H \cdot \log(\theta) + n_T \cdot \log(1-\theta) \\
&= \underset{\theta}{\operatorname{argmax}}\ n_H \cdot \log(\theta) + n_T \cdot \log(1-\theta)
\end{aligned}
$$

Lecture 3: MLE for Binomial

- MLE: Choose θ that maximizes the probability of observed data

$$\begin{aligned}\hat{\theta}_{MLE} &= \underset{\theta}{\operatorname{argmax}}\ P(D;\theta) \\ &= \underset{\theta}{\operatorname{argmax}} \binom{n_H + n_T}{n_H} \theta^{n_H} (1-\theta)^{n_T} \\ &= \underset{\theta}{\operatorname{argmax}}\ \log \binom{n_H + n_T}{n_H} + n_H \cdot \log(\theta) + n_T \cdot \log(1-\theta) \\ &= \underset{\theta}{\operatorname{argmax}}\ n_H \cdot \log(\theta) + n_T \cdot \log(1-\theta)\end{aligned}$$

We can then solve for θ by taking the derivative and equating it with zero. This results in

$$\frac{n_H}{\theta} = \frac{n_T}{1-\theta} \Longrightarrow n_H - n_H\theta = n_T\theta \Longrightarrow \theta = \frac{n_H}{n_H + n_T}$$

Lecture 3: MLE for Binomial

- MLE: Choose θ that maximizes the probability of observed data
- Simple scenario: coin toss with prior knowledge

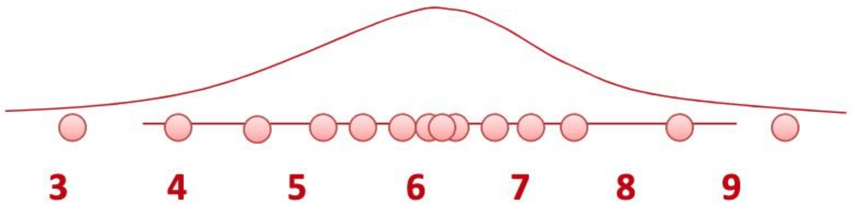
Assume you have a hunch that θ is close to 0.5. But your sample size is small, so you don't trust your estimate.

Simple fix: Add m imaginery throws that would result in θ' (e.g. $\theta = 0.5$). Add m Heads and m Tails to your data.

$$\hat{\theta} = \frac{n_H + m}{n_H + n_T + 2m}$$

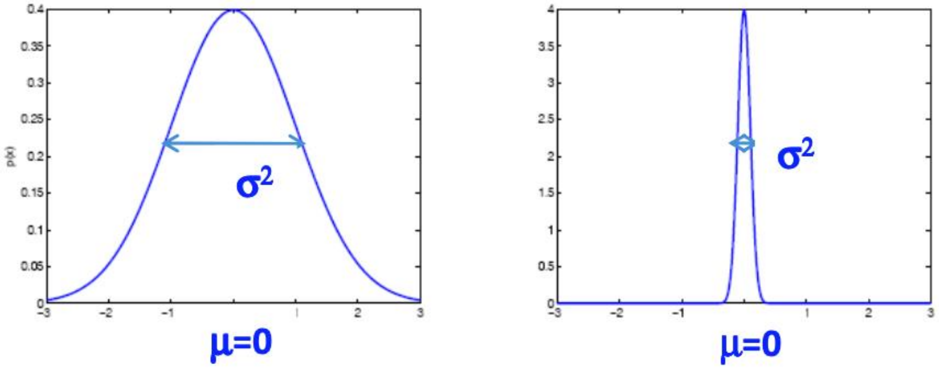
For large n, this is an insignificant change. For small n, it incorporates your "prior belief" about what θ should be. Can we derive this formally?

Lecture 3: MLE for Gaussian



Let us try Gaussians...

$$p(x \mid \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) = \mathcal{N}_x(\mu, \sigma)$$



Lecture 3: MLE for Gaussian

$$\hat{\theta}_{MLE} = \arg\max_{\theta} P(D \mid \theta)$$

Likelihood: $p(\{x_i\}|\mu, \sigma)$

Observed data Unknown parameters

Lecture 3: MLE for Gaussian

$$\hat{\theta}_{MLE} = \arg\max_{\theta} P(D \mid \theta)$$

Likelihood: $p(\{x_i\}|\mu,\sigma)$

Observed data Unknown parameters

$$\hat{\mu},\hat{\sigma} = \arg\max_{\mu,\sigma} p(\{x_i\}|\mu,\sigma)$$

Lecture 3: MLE for Gaussian

$$\hat{\theta}_{MLE} = \arg\max_{\theta} P(D \mid \theta)$$

$$\hat{\mu}, \hat{\sigma} = \arg\max_{\mu,\sigma} p(\{x_i\}|\mu, \sigma)$$

$$\hat{\mu}, \hat{\sigma} = \arg\max_{\mu,\sigma} \prod_{i=1}^{N} p(x_i|\mu, \sigma)$$

$$\hat{\mu}, \hat{\sigma} = \arg\max_{\mu,\sigma} \sum_{i=1}^{N} \underline{\ln p(x_i|\mu, \sigma)}$$

$$\ln \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(x_i-\mu)^2}{2\sigma^2}\right\}$$

$$= \left\{-\frac{(x_i-\mu)^2}{2\sigma^2} - \ln\sigma - \ln\sqrt{2\pi}\right\}$$

Lecture 3: MLE for Gaussian

$$\hat{\theta}_{MLE} = \arg\max_{\theta} P(D \mid \theta)$$

$$\widehat{\mu}_{MLE} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\widehat{\sigma}^2_{MLE} = \frac{1}{n}\sum_{i=1}^{n} (x_i - \widehat{\mu})^2$$

Note: MLE for the variance of a Gaussian is biased [Expected result of estimation is not the true parameter!]

Unbiased variance estimator: $\widehat{\sigma}^2_{unbiased} = \dfrac{1}{n-1}\sum_{i=1}^{n} (x_i - \widehat{\mu})^2$

Lecture 3: The Bayesian Way

- Model θ as a **random variable**, drawn from a distribution P(θ)

$$P(\theta \mid D) = \frac{P(D|\theta)P(\theta)}{P(D)}$$

- P(θ) is the **prior** distribution over the parameter(s) θ, before we see any data.
- P(D|θ) is the **likelihood** of the data given the parameter(s) θ.
- P(θ|D) is the **posterior** distribution over the parameter(s) θ after we have observed the data.

Lecture 3: Maximum a Posteriori Probability Estimation (MAP)

- MAP Principle: Find $\hat{\theta}$ that maximizes the posterior distribution $P(\theta|D)$:

$$\hat{\theta}_{MAP} = \underset{\theta}{\operatorname{argmax}}\ P(\theta \mid D)$$

Lecture 3: Maximum a Posteriori Probability Estimation (MAP)

University of Salford MANCHESTER

- <u>MAP Principle:</u> Find $\hat{\theta}$ that maximizes the posterior distribution $P(\theta|D)$:

$$P(\theta \mid D) = \frac{P(D|\theta)P(\theta)}{P(D)}$$

$$\begin{aligned}\hat{\theta}_{MAP} &= \underset{\theta}{\operatorname{argmax}}\ P(\theta \mid D) \\ &= \underset{\theta}{\operatorname{argmax}}\ \log P(D \mid \theta) + \log P(\theta)\end{aligned}$$

Lecture 3: Maximum a Posteriori Probability Estimation (MAP)

- MAP Principle: Find $\hat{\theta}$ that maximizes the posterior distribution $P(\theta|D)$:

$$\begin{aligned}\hat{\theta}_{MAP} &= \underset{\theta}{\operatorname{argmax}}\ P(\theta \mid D) \\ &= \underset{\theta}{\operatorname{argmax}} \log P(D \mid \theta) + \log P(\theta)\end{aligned}$$

For out coin flipping scenario, we get:

$$\begin{aligned}\hat{\theta}_{MAP} &= \underset{\theta}{\operatorname{argmax}}\ P(\theta|Data) \\ &= \underset{\theta}{\operatorname{argmax}}\ \frac{P(Data|\theta)P(\theta)}{P(Data)} \\ &= \underset{\theta}{\operatorname{argmax}}\ \log(P(Data|\theta)) + \log(P(\theta))\end{aligned}$$

$$P(\theta) = \frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{B(\alpha, \beta)}$$

Lecture 3: Maximum a Posteriori Probability Estimation (MAP)

- <u>MAP Principle:</u> Find $\hat{\theta}$ that maximizes the posterior distribution $P(\theta|D)$:

$$
\begin{aligned}
\hat{\theta}_{MAP} &= \underset{\theta}{\operatorname{argmax}}\ P(\theta \mid D) \\
&= \underset{\theta}{\operatorname{argmax}}\ \log P(D \mid \theta) + \log P(\theta)
\end{aligned}
$$

For out coin flipping scenario, we get:

$$
\begin{aligned}
\hat{\theta}_{MAP} &= \underset{\theta}{\operatorname{argmax}}\ P(\theta|Data) \\
&= \underset{\theta}{\operatorname{argmax}}\ \frac{P(Data|\theta)P(\theta)}{P(Data)} && \text{(By Bayes rule)} \\
&= \underset{\theta}{\operatorname{argmax}}\ \log(P(Data|\theta)) + \log(P(\theta)) \\
&= \underset{\theta}{\operatorname{argmax}}\ n_H \cdot \log(\theta) + n_T \cdot \log(1-\theta) + (\alpha - 1) \cdot \log(\theta) + (\beta - 1) \cdot \log(1-\theta)
\end{aligned}
$$

Lecture 3: Maximum a Posteriori Probability Estimation (MAP)

- MAP Principle: Find $\hat{\theta}$ that maximizes the posterior distribution $P(\theta|D)$:

$$\begin{aligned}\hat{\theta}_{MAP} &= \underset{\theta}{\operatorname{argmax}}\ P(\theta \mid D) \\ &= \underset{\theta}{\operatorname{argmax}}\ \log P(D \mid \theta) + \log P(\theta)\end{aligned}$$

For out coin flipping scenario, we get:

$$\begin{aligned}\hat{\theta}_{MAP} &= \underset{\theta}{\operatorname{argmax}}\ P(\theta|Data) \\ &= \underset{\theta}{\operatorname{argmax}}\ \frac{P(Data|\theta)P(\theta)}{P(Data)} && \text{(By Bayes rule)} \\ &= \underset{\theta}{\operatorname{argmax}}\ \log(P(Data|\theta)) + \log(P(\theta)) \\ &= \underset{\theta}{\operatorname{argmax}}\ n_H \cdot \log(\theta) + n_T \cdot \log(1-\theta) + (\alpha - 1) \cdot \log(\theta) + (\beta - 1) \cdot \log(1-\theta) \\ &= \underset{\theta}{\operatorname{argmax}}\ (n_H + \alpha - 1) \cdot \log(\theta) + (n_T + \beta - 1) \cdot \log(1-\theta)\end{aligned}$$

Lecture 3: Maximum a Posteriori Probability Estimation (MAP)

- <u>MAP Principle:</u> Find $\hat{\theta}$ that maximizes the posterior distribution P(θ|D):

$$\begin{aligned}\hat{\theta}_{MAP} &= \underset{\theta}{\operatorname{argmax}}\ P(\theta \mid D) \\ &= \underset{\theta}{\operatorname{argmax}} \log P(D \mid \theta) + \log P(\theta)\end{aligned}$$

For out coin flipping scenario, we get:

$$\begin{aligned}\hat{\theta}_{MAP} &= \underset{\theta}{\operatorname{argmax}}\ P(\theta|Data) \\ &= \underset{\theta}{\operatorname{argmax}}\ \frac{P(Data|\theta)P(\theta)}{P(Data)} && \text{(By Bayes rule)} \\ &= \underset{\theta}{\operatorname{argmax}}\ \log(P(Data|\theta)) + \log(P(\theta)) \\ &= \underset{\theta}{\operatorname{argmax}}\ n_H \cdot \log(\theta) + n_T \cdot \log(1-\theta) + (\alpha - 1) \cdot \log(\theta) + (\beta - 1) \cdot \log(1-\theta) \\ &= \underset{\theta}{\operatorname{argmax}}\ (n_H + \alpha - 1) \cdot \log(\theta) + (n_T + \beta - 1) \cdot \log(1-\theta) \\ &\implies \hat{\theta}_{MAP} = \frac{n_H + \alpha - 1}{n_H + n_T + \beta + \alpha - 2}\end{aligned}$$

Lecture 3: MAP vs MLE

$$\hat{\theta}_{MAP} = \frac{n_H + \alpha - 1}{n_H + n_T + \beta + \alpha - 2}$$

$$\hat{\theta} = \frac{n_H + m}{n_H + n_T + 2m}$$

Lecture 3: MAP vs MLE
Bayesians vs Frequentists

$$\hat{\theta}_{MAP} = \frac{n_H + \alpha - 1}{n_H + n_T + \beta + \alpha - 2}$$

$$\hat{\theta} = \frac{n_H + m}{n_H + n_T + 2m}$$

Lecture 3: Multivariate Gaussian Distribution (Estimation)

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x}-\boldsymbol{\mu})\right\}$$

Σ Covariance *matrix*

* Diagonal terms: variance
* Off-diagonal terms: correlation

D Number of Dimensions

$\mathbf{x}$ Variable

$\boldsymbol{\mu}$ Mean *vector*

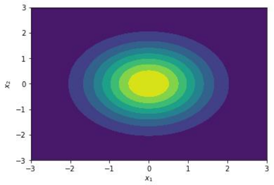
Σ Covariance *matrix*

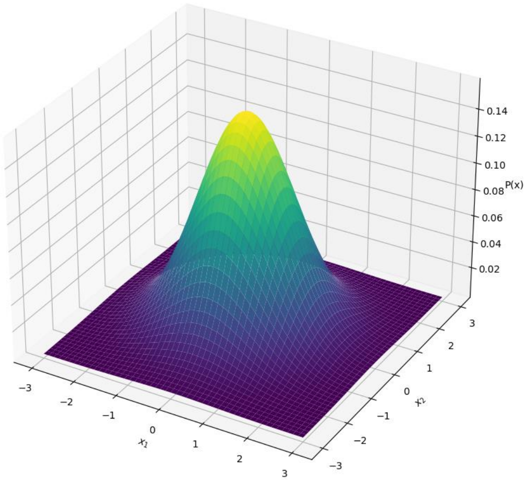
(Dimension = 2)

$$\boxed{\Sigma = \begin{bmatrix} \sigma_{x_1}^2 & \sigma_{x_1 x_2}^2 \\ \sigma_{x_2 x_1}^2 & \sigma_{x_2}^2 \end{bmatrix}} \quad (\sigma_{x_1 x_2}^2 = \sigma_{x_2 x_1}^2)$$

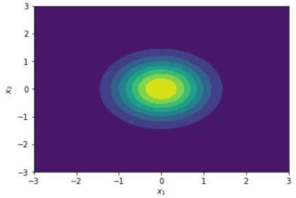
Lecture 3: Multivariate Gaussian Distribution (Estimation)

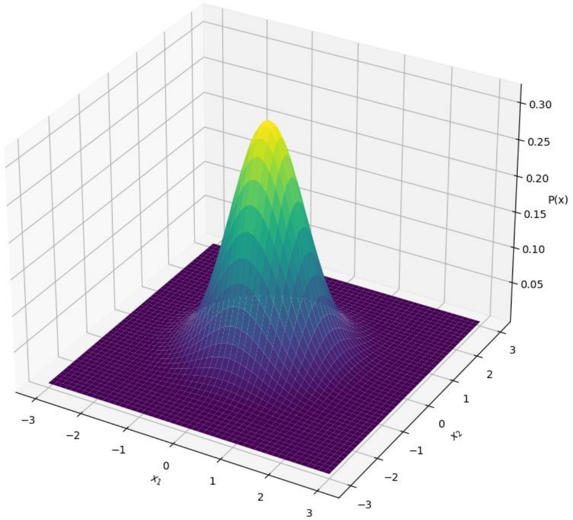
mean = [0, 0]
cov = [[1, 0], [0, 1]]



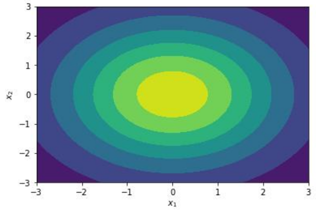


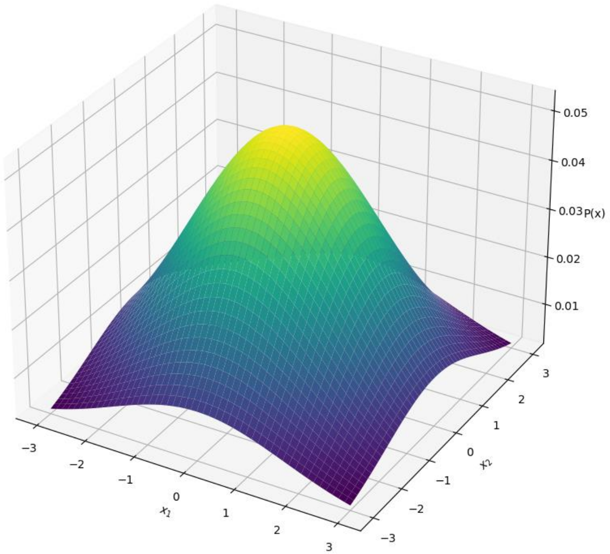
mean = [0, 0]
cov = [[0.5, 0], [0, 0.5]]





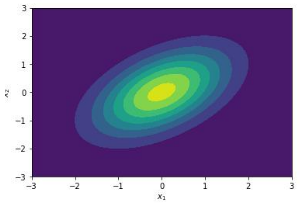
mean = [0, 0]
cov = [[3, 0], [0, 3]]

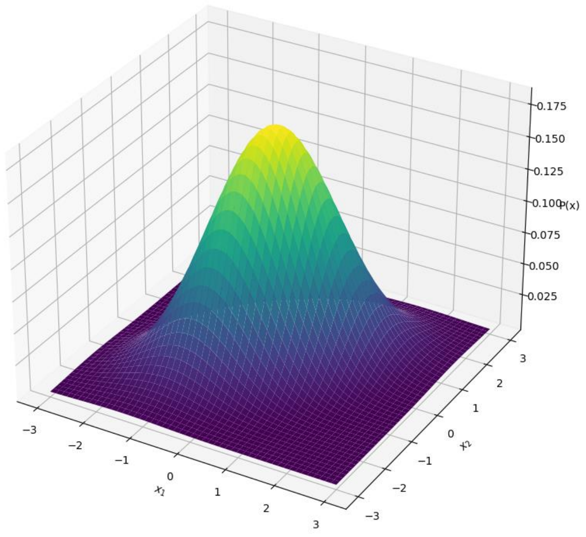




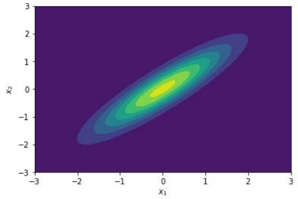
Lecture 3: Multivariate Gaussian Distribution (Estimation)

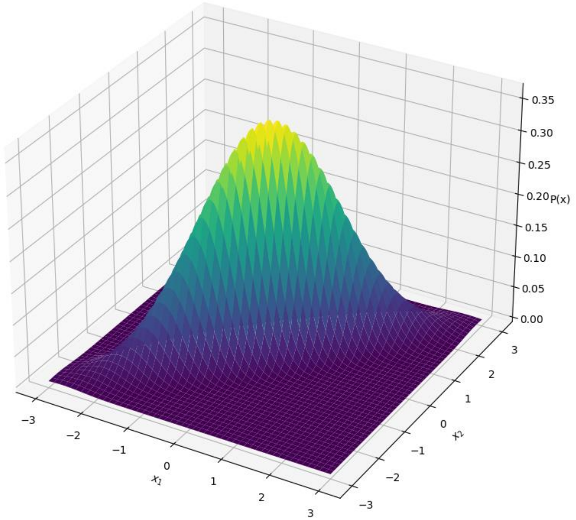
mean = [0, 0]
cov = [[1, 0.5], [0.5, 1]]



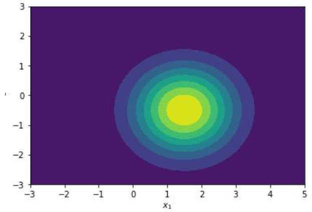


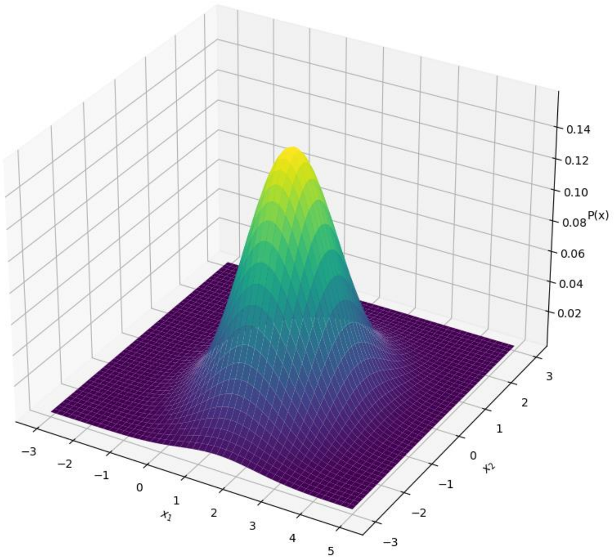
mean = [0, 0]
cov = [[0.9, 0], [0.9, 0.5]]



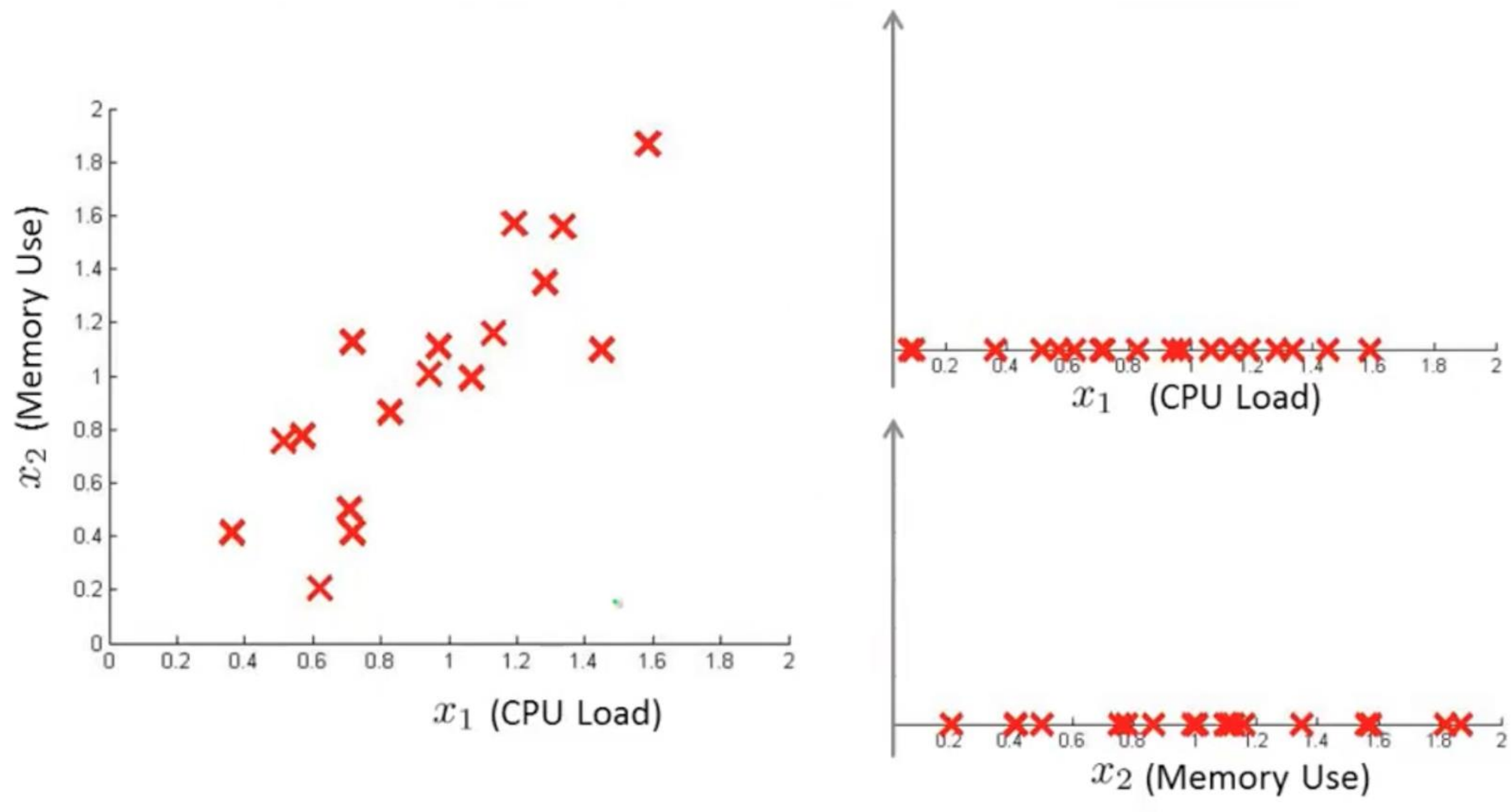


mean = [1.5, 0.5]
cov = [[1, 0], [0, 1]]



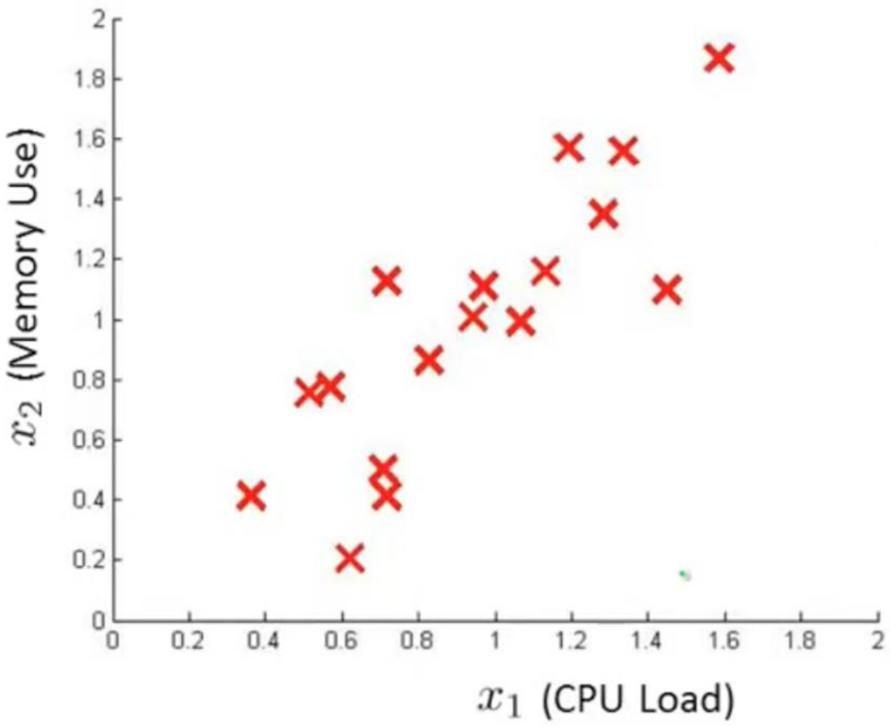


Lecture 3: Anomaly detection (Estimation)



Ng slide

Lecture 3: Anomaly detection (Estimation)



$$\hat{\mu}_{MLE} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\hat{\sigma}^2_{MLE} = \frac{1}{n}\sum_{i=1}^{n} (x_i - \hat{\mu})^2$$

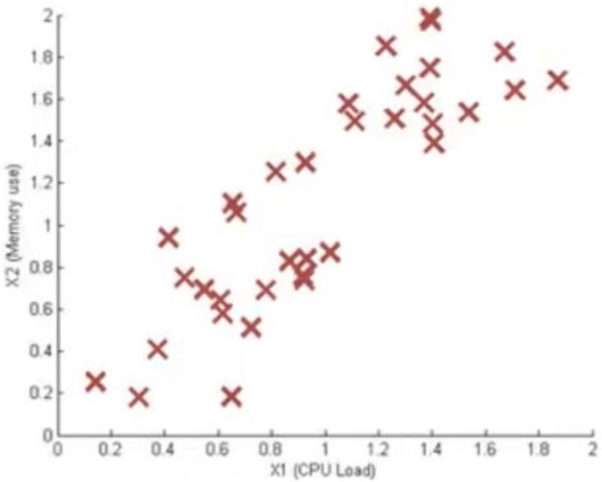
Lecture 3: Anomaly detection (Estimation)

Anomaly Detection using the Multivariate Gaussian Distribution

1. Fit model $p(x)$ by setting

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}$$

$$\Sigma = \frac{1}{m}\sum_{i=1}^{m}(x^{(i)} - \mu)(x^{(i)} - \mu)^T$$



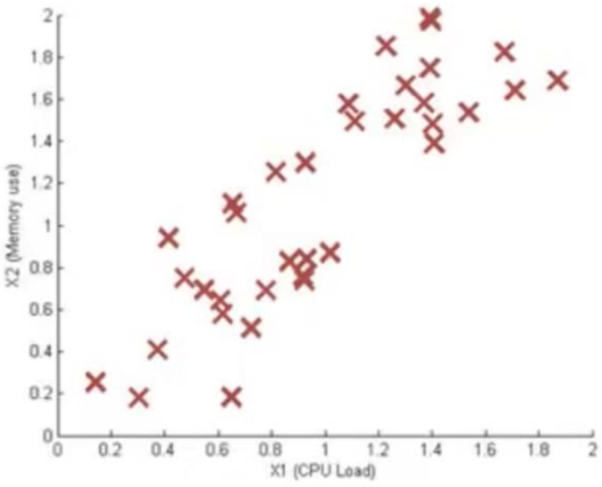
Lecture 3: Anomaly detection (Estimation)

Anomaly Detection using the Multivariate Gaussian Distribution

1. Fit model $p(x)$ by setting

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}$$

$$\Sigma = \frac{1}{m}\sum_{i=1}^{m}(x^{(i)} - \mu)(x^{(i)} - \mu)^T$$



X1 (CPU Load)

2. Given a new example x, compute

$$p(x) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}}\exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$

Flag an anomaly if $p(x) < \varepsilon$

Lecture 3: Anomaly detection (Estimation) Applications

- Fraud Detection
- Manufacturing (e.g. aircraft engines)
- Monitoring Machines in a data centre

Lecture 3: MAP vs MLE
Machine Learning and estimation

University of Salford MANCHESTER

Generative Model vs Discriminative Model

In supervised Machine learning you are provided with training data D. You use this data to train a model, represented by its parameters θ. With this model you want to make predictions on a test point x_t.

- **MLE** Prediction: $P(y|x_t;\theta)$ Learning: $\theta = \text{argmax}_\theta\, P(D;\theta)$. Here θ is purely a model parameter.
- **MAP** Prediction: $P(y|x_t,\theta)$ Learning: $\theta = \text{argmax}_\theta\, P(\theta|D) \propto P(D \mid \theta)P(\theta)$. Here θ is a random variable.

MLE we maximize $\log[P(D;\theta)]$ + Regularization

MAP we maximize $\log[P(D|\theta)] + \log[P(\theta)]$

Lecture 3 : Summary:

- ❑ Optimization :
  - Minimize, Maximize
- ❑ Algorithms:
  - Hill Climbing
  - Simulated Annealing
  - Genetic algorithm
- ❑ Examples
  - Travelling salesman problem
  - 8 queen
- ❑ Probabilities:
  - Dependence, Independence, Conditional Independence
- ❑ Parameter estimation:
  - Maximum Likelihood Estimation (MLE)
  - Maximum Aposteriori (MAP)
- ❑ Anomaly detection

Lecture 3: Next

- Machine Learning

Any Question