Lecture-16

Independent Component Analysis:-

- Sources $S \in \mathbb{R}^n$

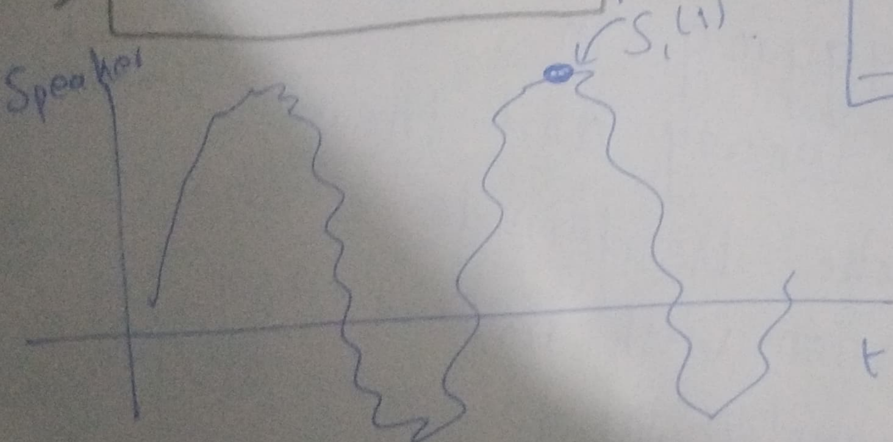
$S_j^{(i)}$ = Speaker j at time i.

$\boxed{x^{(i)} = A s^{(i)}}$ $x \in \mathbb{R}^n$

Finding individual speaker source

Goal find $W = A^{-1}$ so

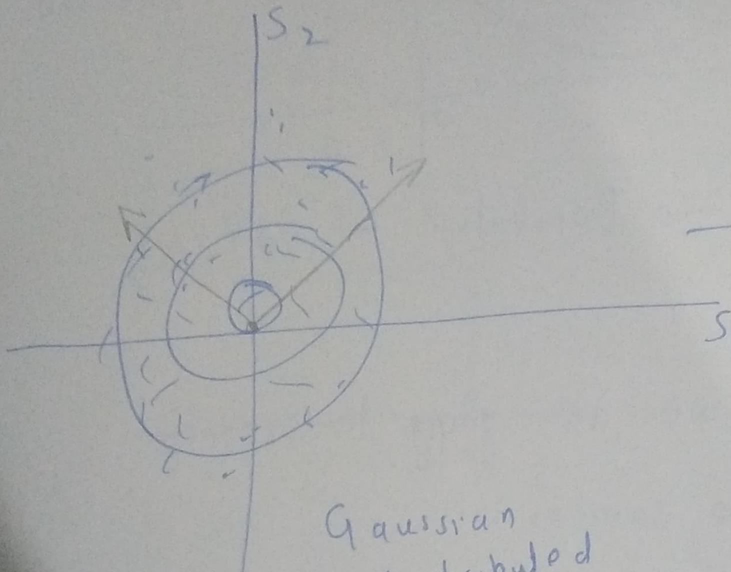
$\boxed{s^{(i)} = W x^{(i)}}$ $= W = \begin{bmatrix} - w_1^T - \\ \vdots \\ - w_n^T - \end{bmatrix}$



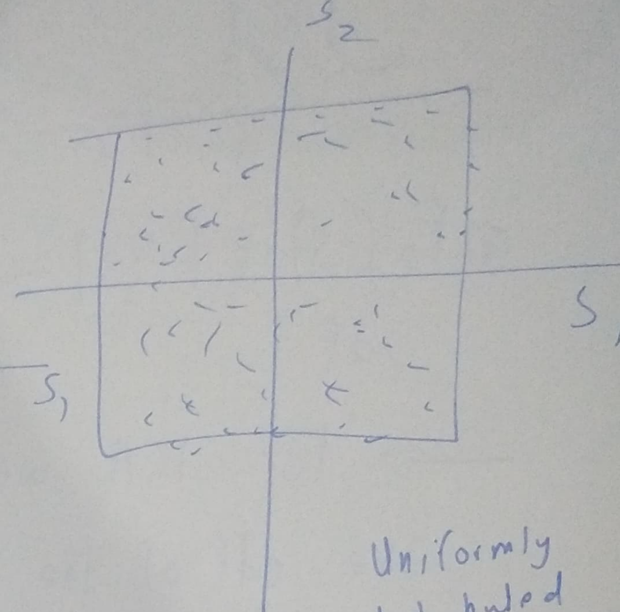
→ If S is ~~uniformly dis~~ gaussian distributed then ICA is not possible.

↳ This is due to rotational ambiguity because gaussian distribution is rotationally symmetric

↳ If S_1 and S_2 (Example) are standard Gaussian then distribution is rotationally symmetric and you dont have information to recover the direction that correspond to orginal sources.



Gaussian distributed (Rotational Ambiguity)



Uniformly distributed (No rotational ambiguity)

* Develop ICA algorithm assuming independent source is non-gaussian:

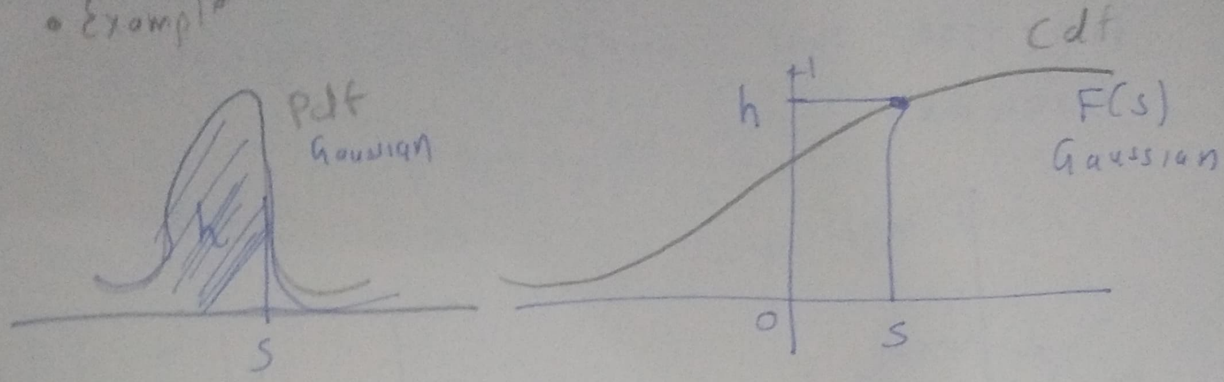
$P_s(S)$

CDF (Cummulative Distribution Function)

$$F(s) = P(\underbrace{S \leq s}_{\text{Constant}})$$

Random variable

• Example



$P_s(s) = F'(s)$ → Derivative of cdf is pdf

→ For ICA we are going to specify the cdf of the sources.

Choose $F(s)$

Density of s: $p(s)$

$$X = AS = W^{-1}s$$

$$S = WX$$

• Deriving maximum likelihood :-

$$\boxed{P_x(x) = P_s(Wx)\ ?}$$

↑ Density of X

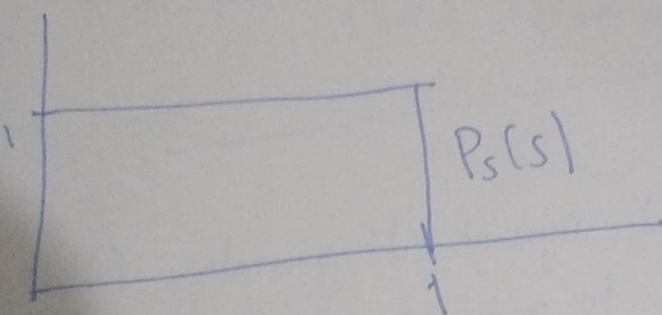
→ This is incorrected for continous probability density.

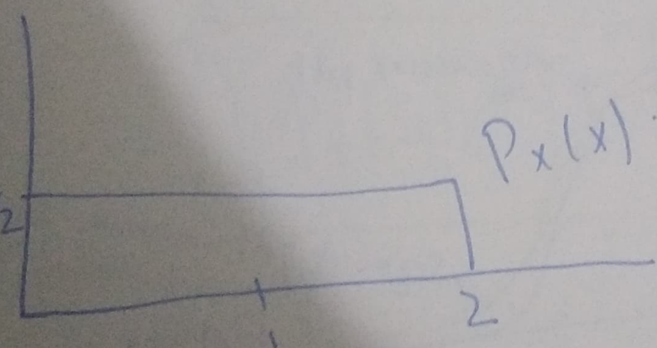
→ So now what: Lets take an example:

→ $P_s(s) = \mathbb{1}\{0 \leq s \leq 1\}$ $(S \sim \text{Uniform}(0,1))$.

$X = 2S$ $(A = 2,\ W = \frac{1}{2})$

$X \sim \text{Uniform}(0,2)$.





$P_x(x) = \frac{1}{2}\mathbb{1}\{0 \leq x \leq 2\}$.

$|W|$

→ Returning back (the Correct formula) is:-

$$P_x(x) = P_s(Wx)\,|W|$$

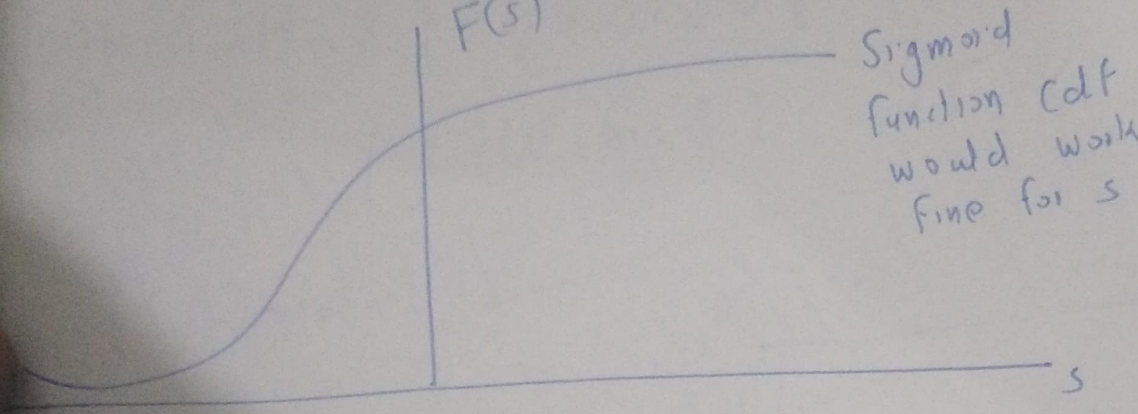
Correct Formula

determinant of matrix W

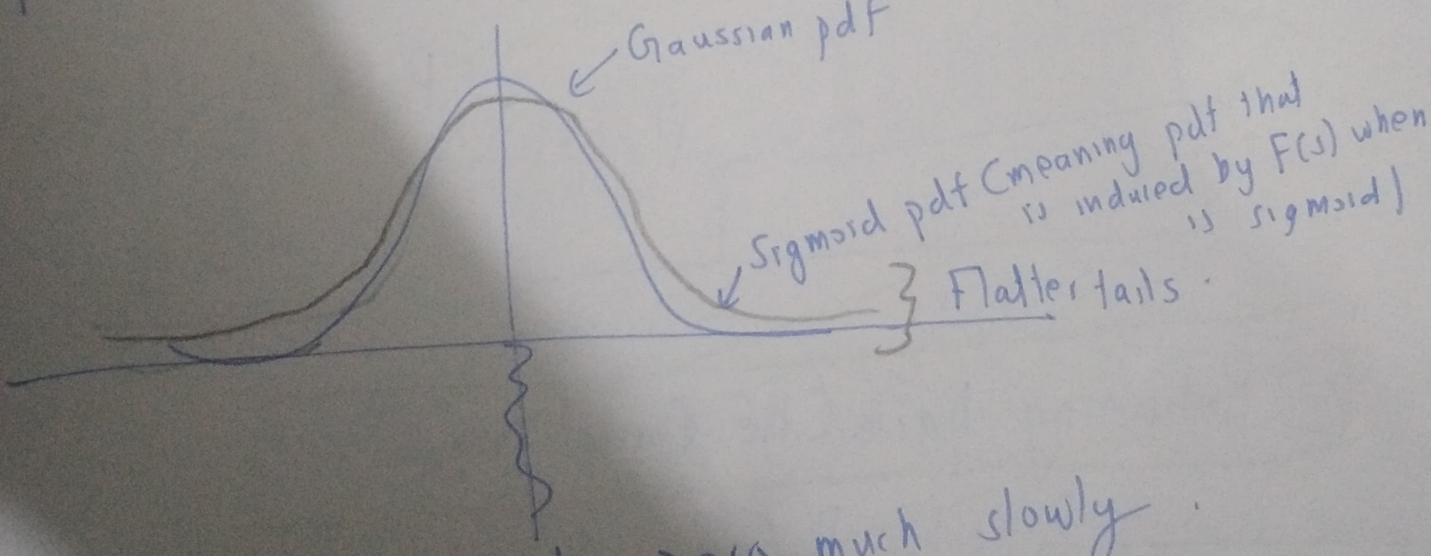
(181)

- Also choose $P_s(s)$:

$$F(s) = P(S \leq s) = \frac{1}{1+e^{-s}}$$ (This will work fine).



→ Turns out that sigmoid cdf generates follow pdf as compare to gaussian:



→ Sigmoid goes to zero much slowly. and this captures many pheonomenon of real life much vividly.

→ Find Step

$$P_s(s) = \prod_{i=1}^{n} P_s(s_i)$$

(n speakers are independent) that's why we are taking the product).

Key assumption of ICA!!!

$$P(x) = P_s(Wx)\,|W|$$

$$\boxed{= \prod_{i=1}^{n} P_s(W_i^T x)\,|W|}$$

$$W = \begin{pmatrix} - & W_1^T & - \\ & \vdots & \\ - & W_n^T & - \end{pmatrix}$$

Remeber
$$\boxed{S_j = W_j^T X}$$

→ We are able to write down the density of X in terms of W

→ Using MLE

$$\boxed{l(W) = \sum_{i=1}^{n} \log\left(\prod_j P_s(W_j^T x^{(i)})\,|W|\right)}$$ → log likelihood

→ Sochastic gradient ascent

$$\nabla_W l(W) = \begin{bmatrix} 1-2g(W_1^T x) \\ \vdots \\ 1-2g(W_n^T x) \end{bmatrix} x^{(i)T} + (W^T)^{-1}$$

where g is sigmoid function.

* Running SGD for W for number of iteration would find us good parameter.
  - ↳ Then you can recover S

$$\boxed{\text{by } S = WX}$$

→ Reinforcement Learning:-

- R → Reward
  S → Action State

- Credit Assignment makes reinforcement learning really hard.
- ↳ How to distribute reward at each state
- ↳ How to figure out what agent did well and what agent did poorly

→ Markov Decision Process (MDP):-

MDP is a five tuple:

$(S, A, \{P_{sa}\}, \gamma, R)$

S - Set of states.

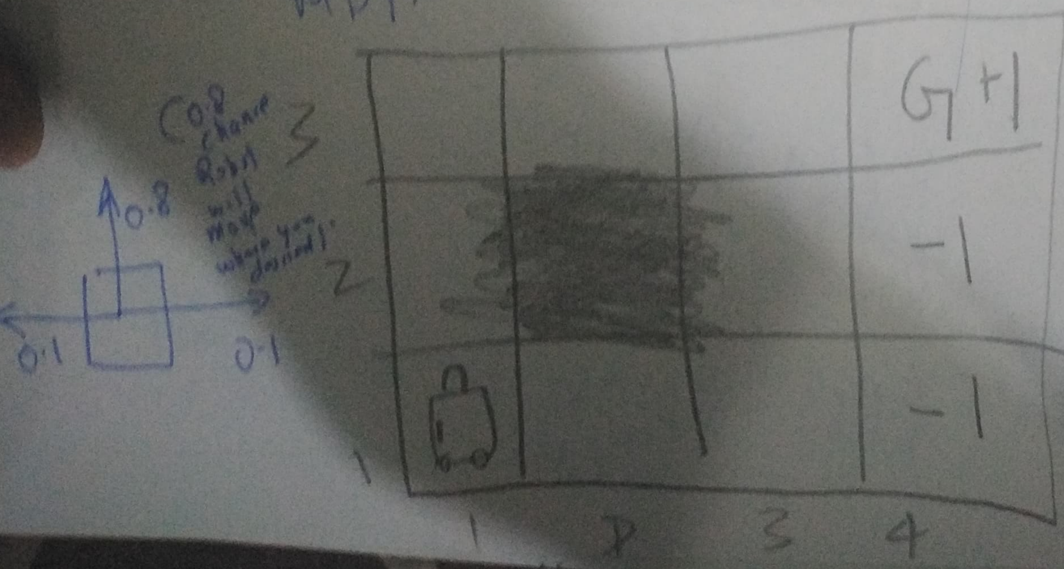
A - Actions.

P_{sa} — State transistion Probabilities ($\sum_{s'} P_{sa}(s') = 1$).

γ → Discount Factor.

R - Reward.

⊛ Running example of a simplified MDP:



G - Goal

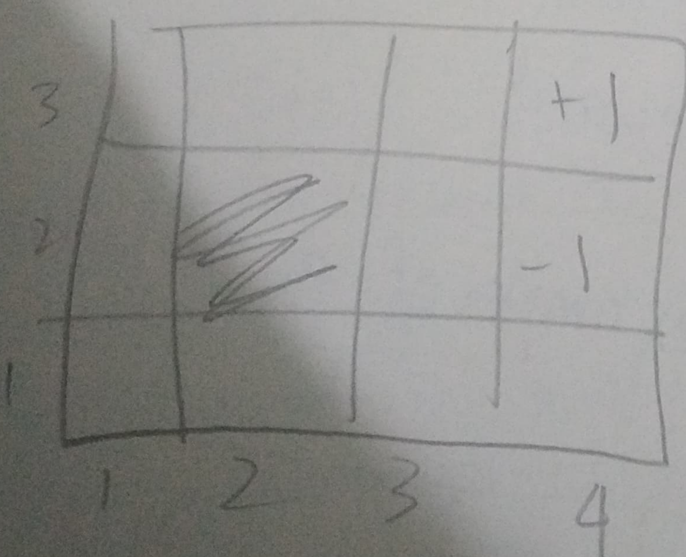
→ This MDP has:-

① 11 states

② Actions: {N, S, E, W}.

③ P_{sa}
$$\begin{cases} P_{(3,1)N}((3,2)) = 0.8 \\ P_{(3,1)N}((4,1)) = 0.1 \\ P_{(3,1)N}((2,1)) = 0.1 \\ P_{(3,1)N}((3,3)) = 0. \end{cases}$$

④ Let Specify the Reward for this Simplifying MDP:-



$R((4,3)) = +1$

$R((4,2)) = -1.$

$R(s) = -0.02$ for all other states

186

(a)

$S_0 \rightarrow$ Robot Intial State

Choose some action a_0

Get to $\boxed{S_1 \sim P_{S_0 a_0}}$

Choose action a_1

Get to $\boxed{S_2 - P_{S_1 a_1}}$

⋮

Total Reward:

$$\boxed{R(S_0) + \gamma R(S_1) + \gamma^2 R(S_2) + \cdots}$$

$\gamma = 0.99$ (Gamma is slightly less than One).

→ Further the reward is in future the smaller the gamma is power of time that reward is multiplied by therefore discounting the future rewards. (Link to Time Value of Money).

→ Goal: Choose actions over time to maximize $E[R(S_0) + \gamma R(S_1) + \gamma^2 R(S_2) + \cdots]$

→ Most Reinforcement learning algorithm will come up with a policy

$\pi : S \longmapsto A$ (State Map to Action)

(~~Also~~ AKA controller)

→ Optimal policy for our MDP.

3	→	→	→	+1
2	↑		↑	−1
1	←	←	$\boxed{←}$	←
	1	2	3	4

$\boxed{\pi((3,1)) = W}$ → When in state (3,1) take the action w.

→ When in s take action $\pi(s)$.

→ Optimal policy maximizes

$E[R(S_0) + R(S_1) + R(S_2) + \cdots]$