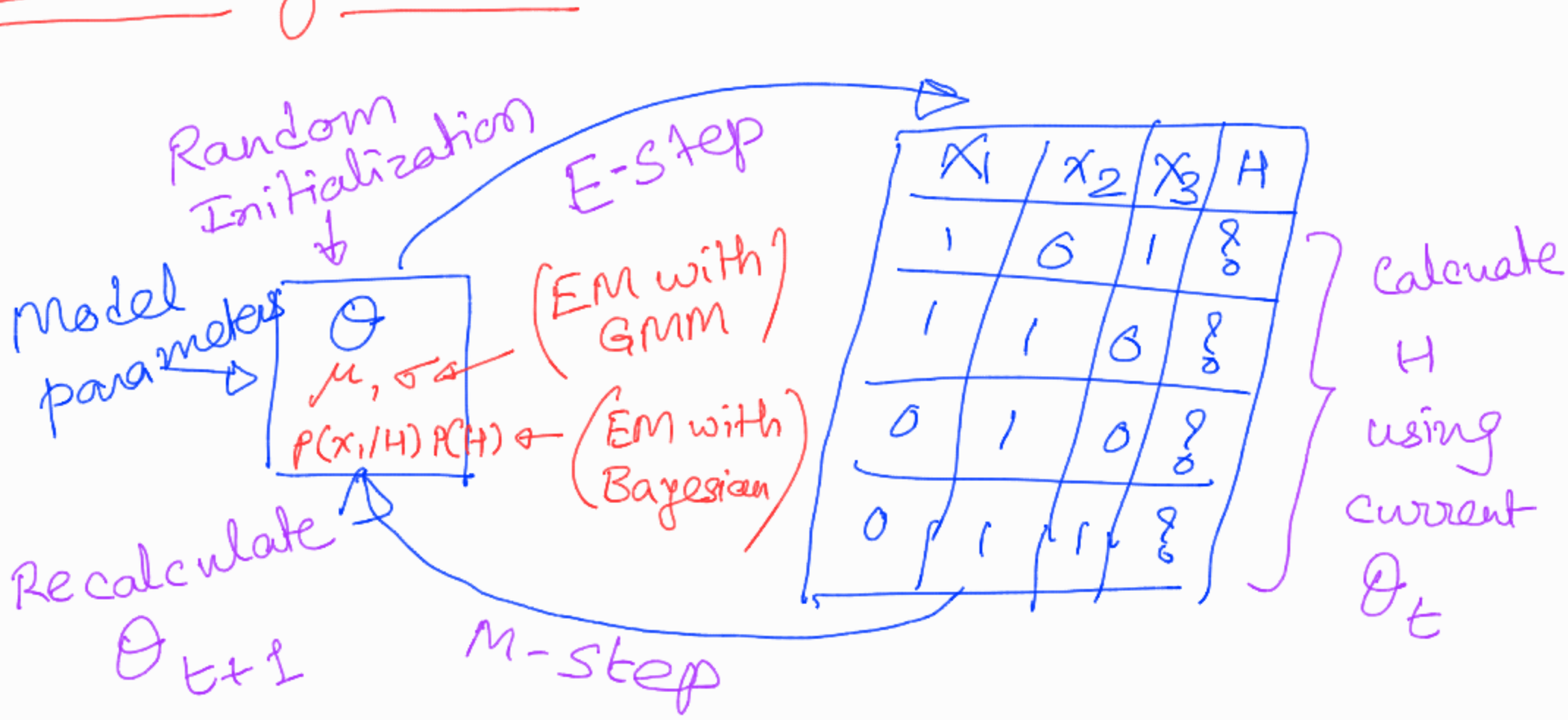
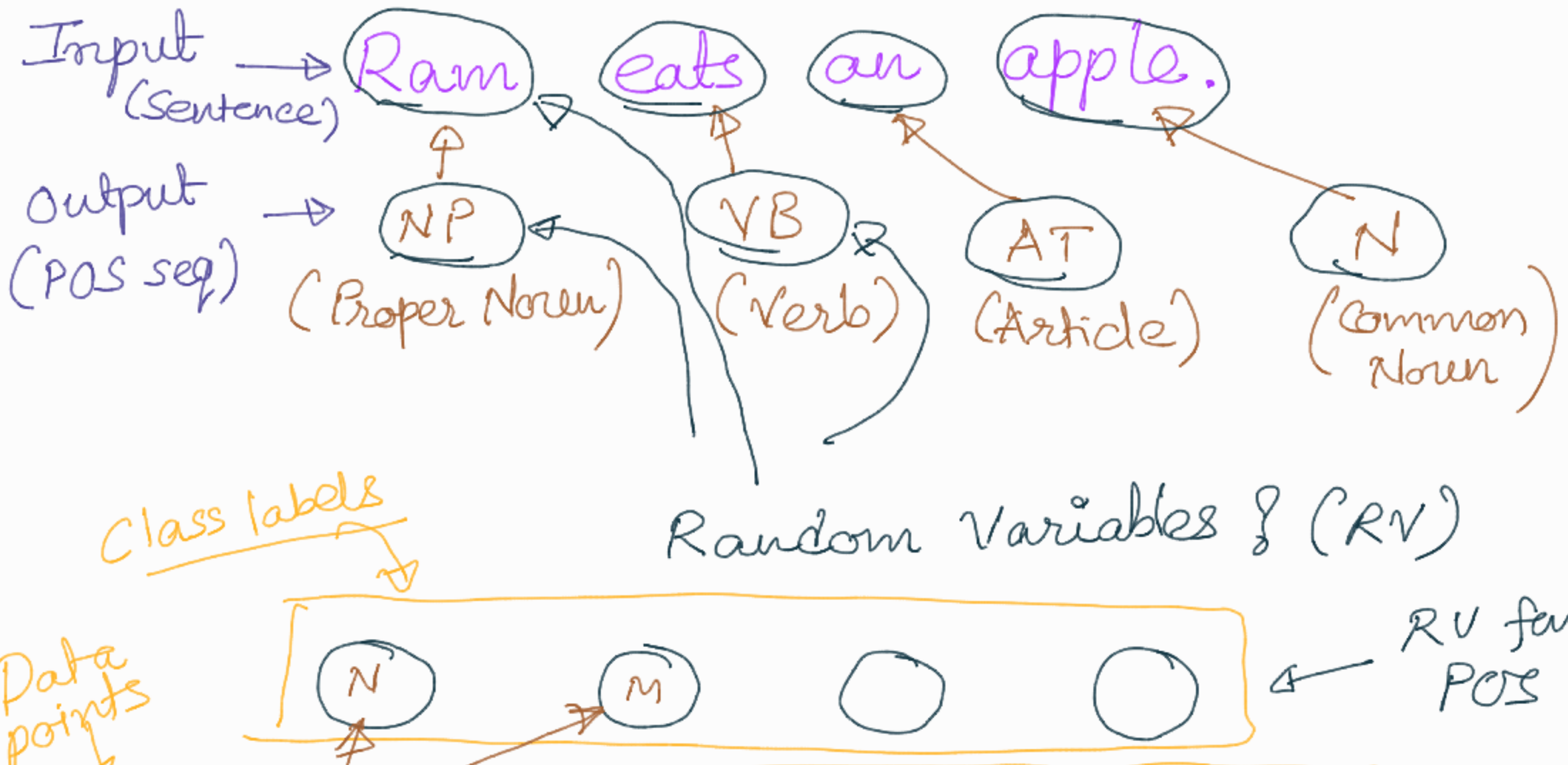
- Specialized Graphical Model
- Application of EM & Dynamic programming
- Part-of-Speech tagging
  - (Sequence Labeling task)

EM Algorithm



Hidden Markov Model (HMM)

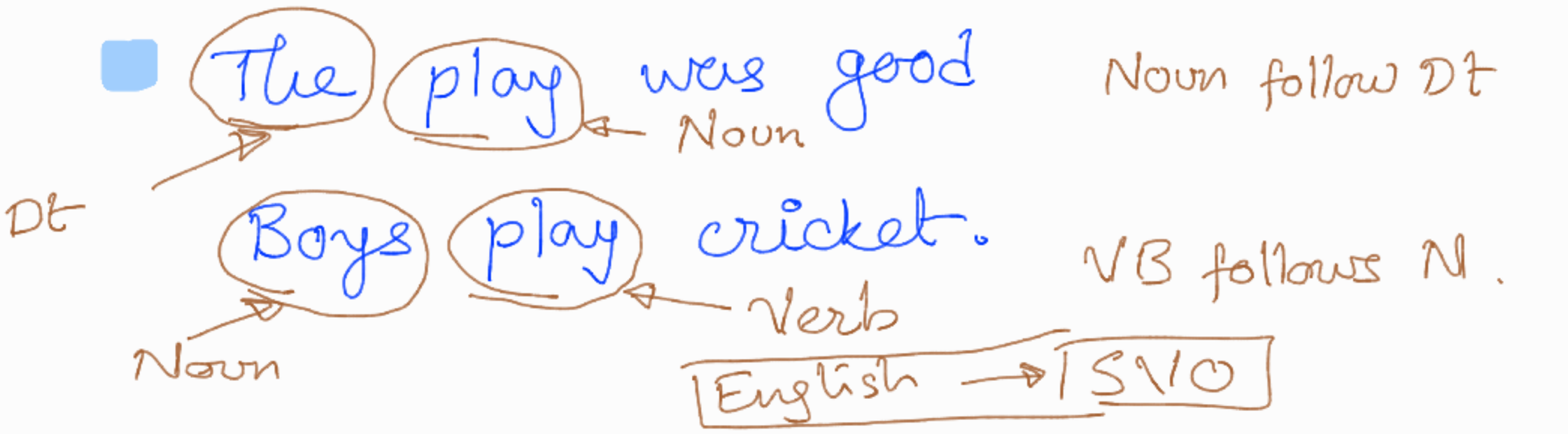
Part-of-Speech (POS) Tagging



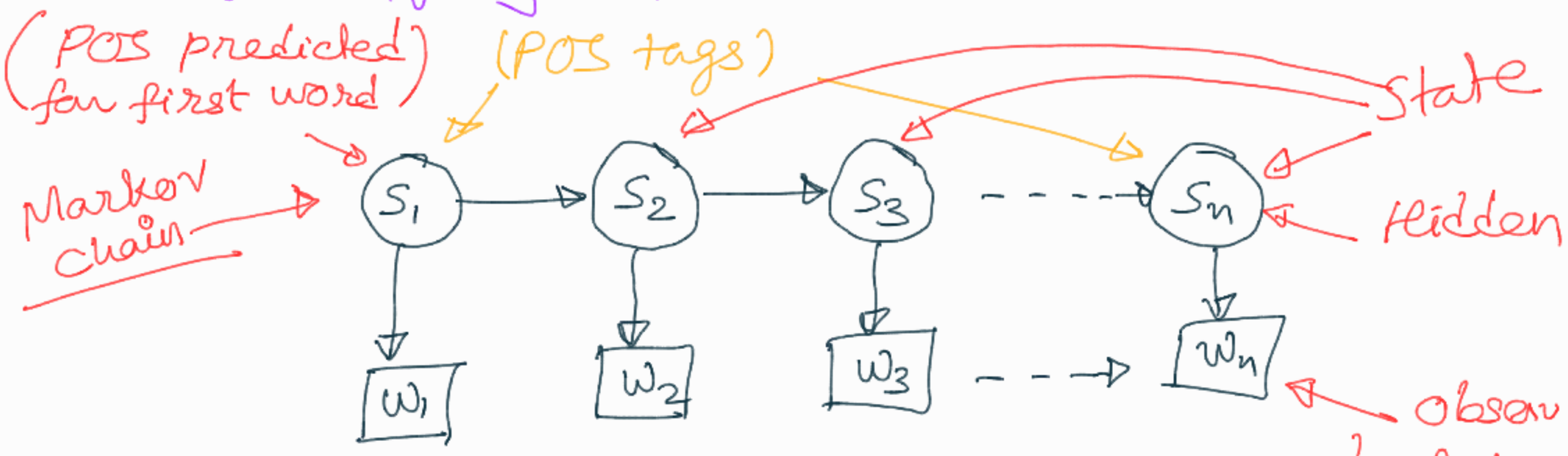
(1) Set of RV for POS is finite.
(AT, N, V, Adj, Adv, pre ...)
- Set of word is infinite.

(2) POS as class label, words as data points.
→ A word can be in more than one POS, will (N, M, auxverb)
can (N,

(3) Given a sentence can we disambiguate POS of a word?



④ Tag of previous word helps in identifying POS of the current word.



It only depends on (immediate) previous state, S_n depends on S_{n-1} only

(Markov property)

$P(S_{i+1}/S_i)$ ← Transition Probability

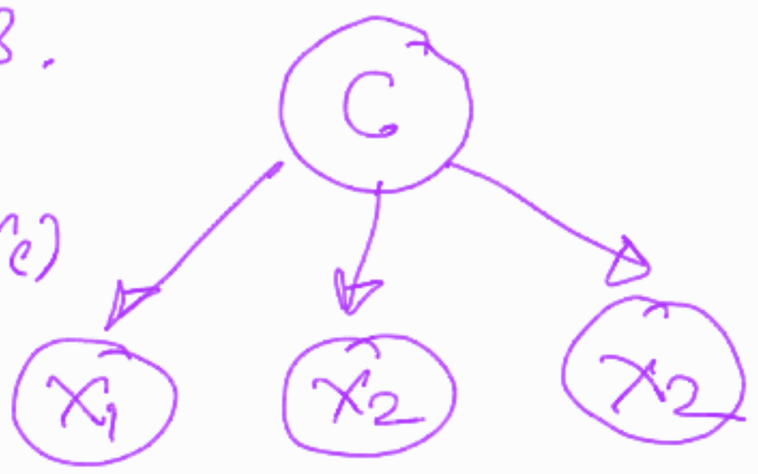
$P(w_i/S_i)$ ← Emission probability

$$P(S_i, w_1, w_2 \ldots w_{i-1}, S_1, S_2 \ldots S_{i-1})$$

$$\approx \left[\prod P(w_i/S_i) \cdot P(S_i/S_{i-1})\right] P(S_0)$$

(5) - Ws are observed variables.
S are hidden variables.

$$P(x_1 \ldots x_n, c) = P(x_1|c) \cdot P(x_2|c) \ldots P(x_n|c) \cdot P(c)$$

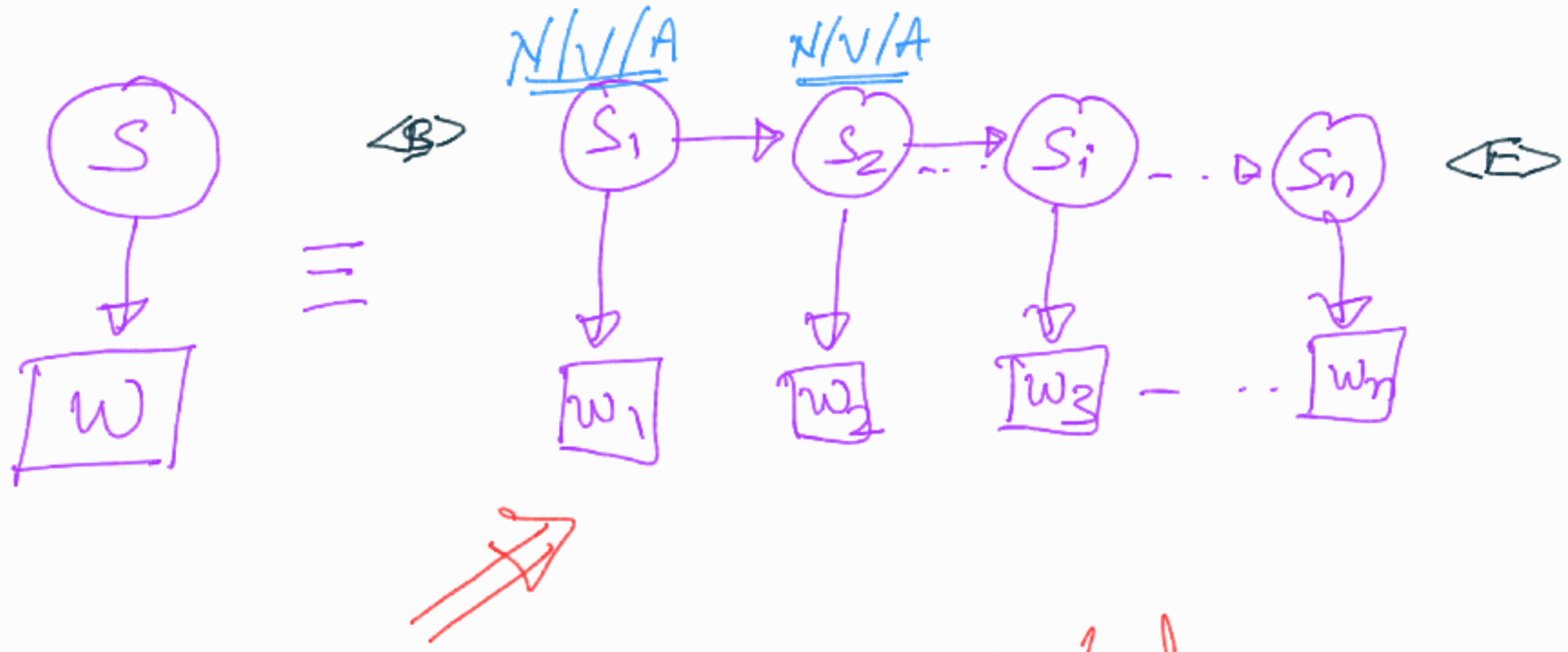


(6) S are states, (Different from traditional RVs)

$S \longrightarrow \{ N, VB, ADJ \ldots \}$

$W \longrightarrow \{ a, cat, dog \ldots \}$

(7) n (number of states) = No. of words in the sentence



Hidden Markov Model

$\pi(S_0) = P(S_0)$ ⟵ Initial state

Transition Pro: $P(S_i / S_{i-1})$

Emission Pro: $P(W_i / S_i)$

Supervised Learning:

Text corpus with every tagged.

Ex:

1. The_DT boys_N Play_VB Cricket_NP
2. Ram_NP likes_VB the_DT play_N

Transition Prob.

$P(S_{i+1} / S_i)$

S_i \ S_{i+1}	Dt	N	VB	NP
Dt	0	2	0	0
N	0	0	1	0
VB	1	0	0	1
NP	0	0	1	0

Emission Prob

$P(W_i / S_i)$

	The	boys	Play	Cricket	RAM	like
Dt	2	0	0	0	0	0
N	0	1	1	0	0	0
VB	0	0	1	0	0	1
NP	0	0	0	1	1	0

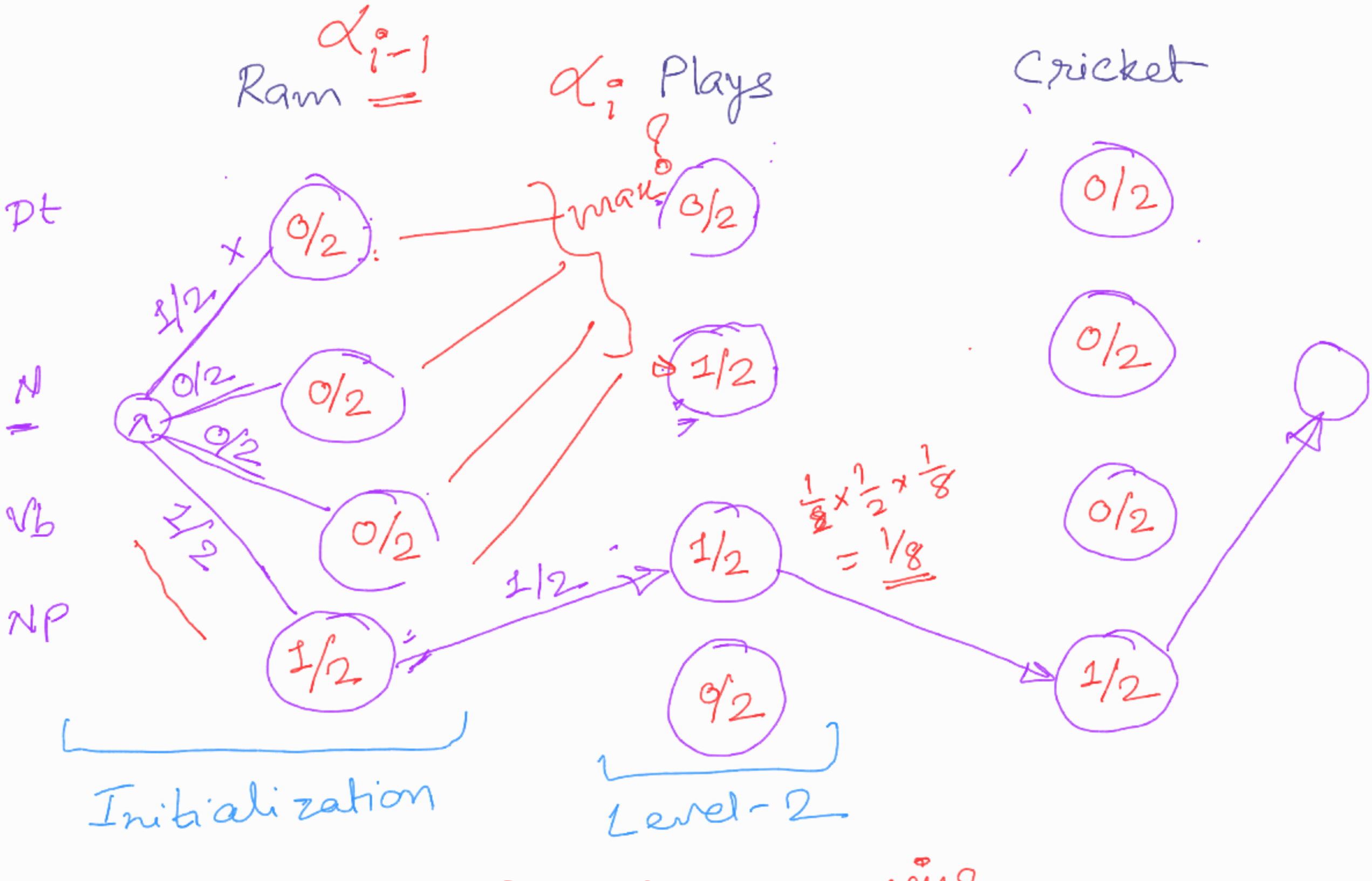
$P(S_0)$

S	Dt	N	Vb	NP
^	1	0	0	1

Brown Corpus

Inference :

Input : Ram _ ? Plays _ ? Cricket _ ?



Dynamic programming :

Viterbi Algorithm (Decoding)

Outline

→ Finish HMM and discussion on probabilistic graphical model

→ Begin with discriminative classifier

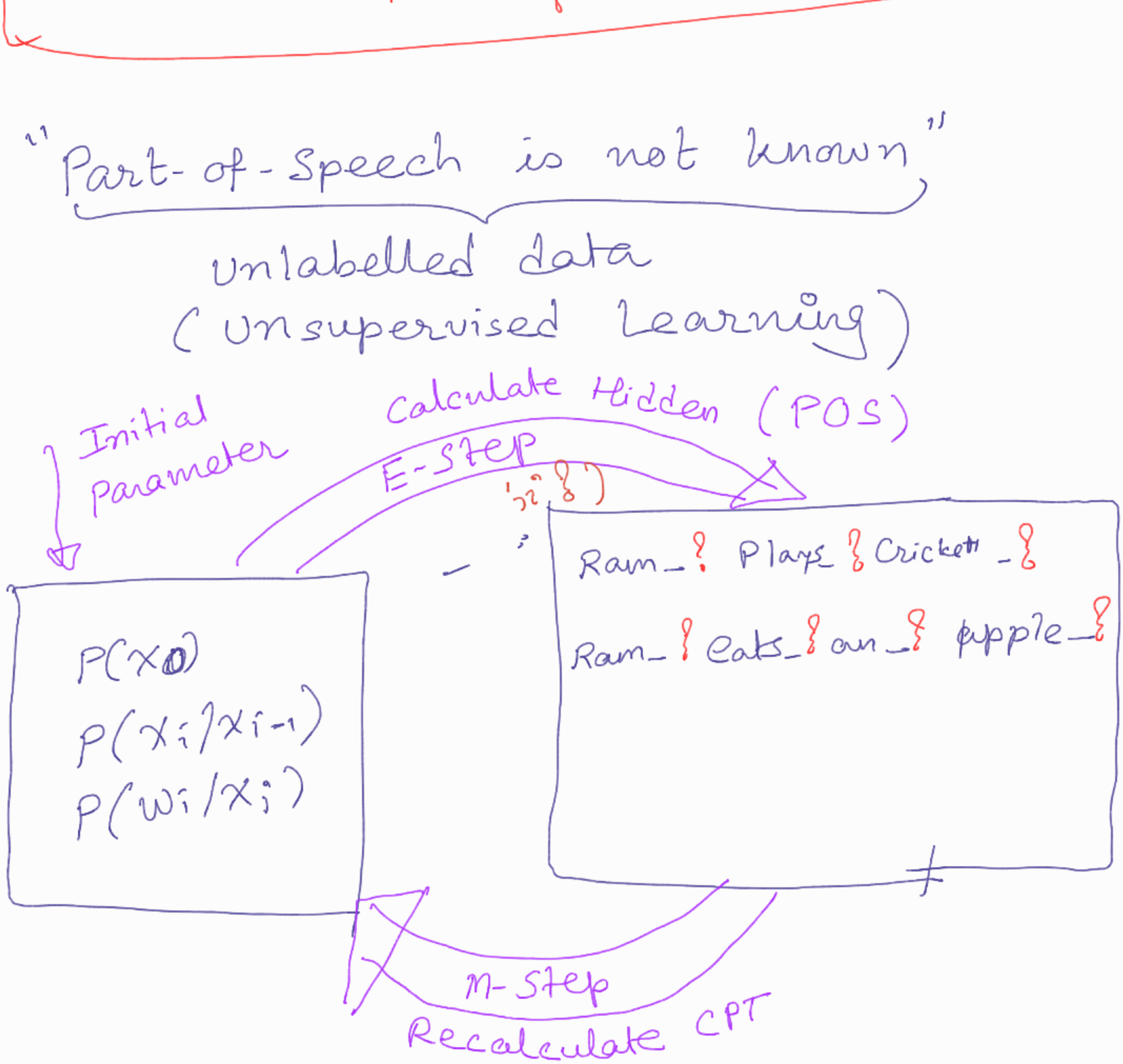
→ Neural Network basics

→ Perceptron

"Part-of-Speech is not known"

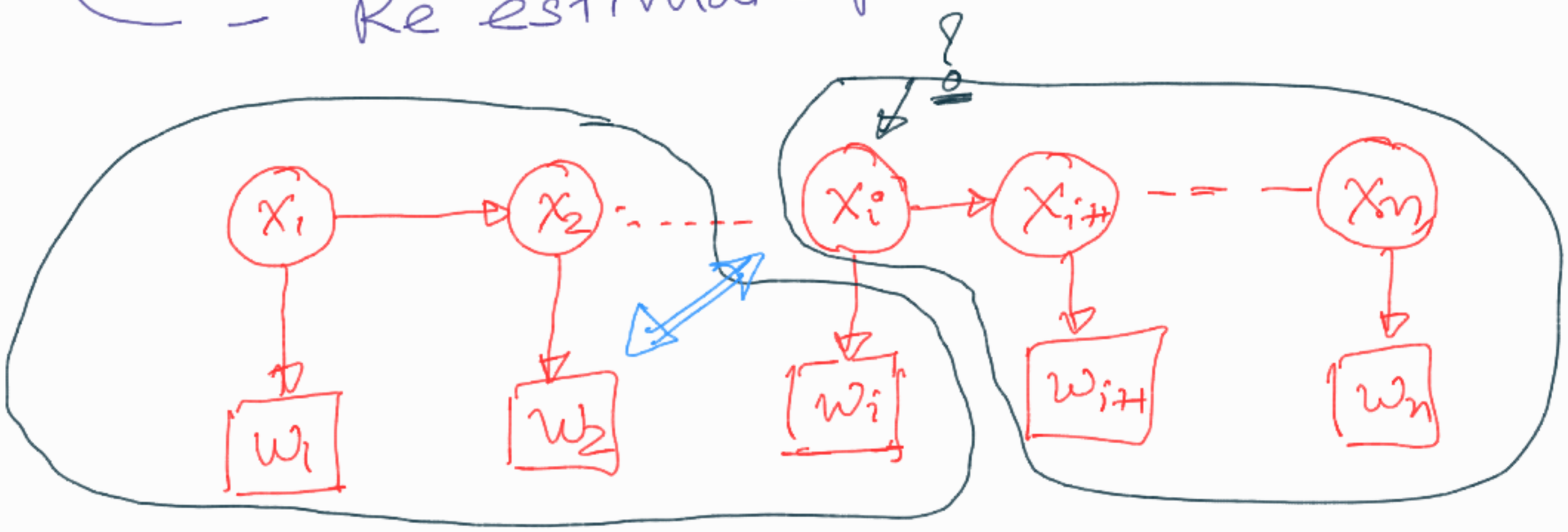
Unlabelled data

(Unsupervised Learning)



Baum-Welch Algorithm

- Fine tuning parameters
- Special case of EM
- start with initial parameter
- Based on parameter calculate hidden variable
- Re estimate parameter



When deciding x_i

→ All observed words $w_1 \ldots w_n$

→ All previous labels $x_1 \ldots x_{i-1}$

$$P(x_i / x_1 \ldots x_{i-1}, w_1 \ldots w_n)$$

$\alpha = P(w_1 \ldots w_i, x_i)$ ⇐ observed before deciding x_i

$\beta = P(w_{i+1} \ldots w_n / x_i)$ ⇐ All the forth coming observations

α_i - depends on $w_i, w_{i-1} \cdots w_1$

$\propto P(x_i/x_{i-1}) \cdot P(w_i/x_i)$

$\times$ (whatever values upto x_{i-1})

Forward calculation

$$\alpha(i) = \sum \alpha(i-1) \cdot P(x_i/x_{i-1}) \cdot P(w_i/x_i)$$

Forward

sum over all possible labels at $\alpha(i-1)$

Backward Computation

$$\beta(i) = \sum \beta(y_{i+1}) \cdot P(x_{i+1}/x_i) \cdot P(w_{i+1}/x_{i+1})$$

Backward

Complete distri $\propto \alpha \times \beta$

Forward - Backward Algo.

$$\boxed{\beta_n = 1}$$

HMM

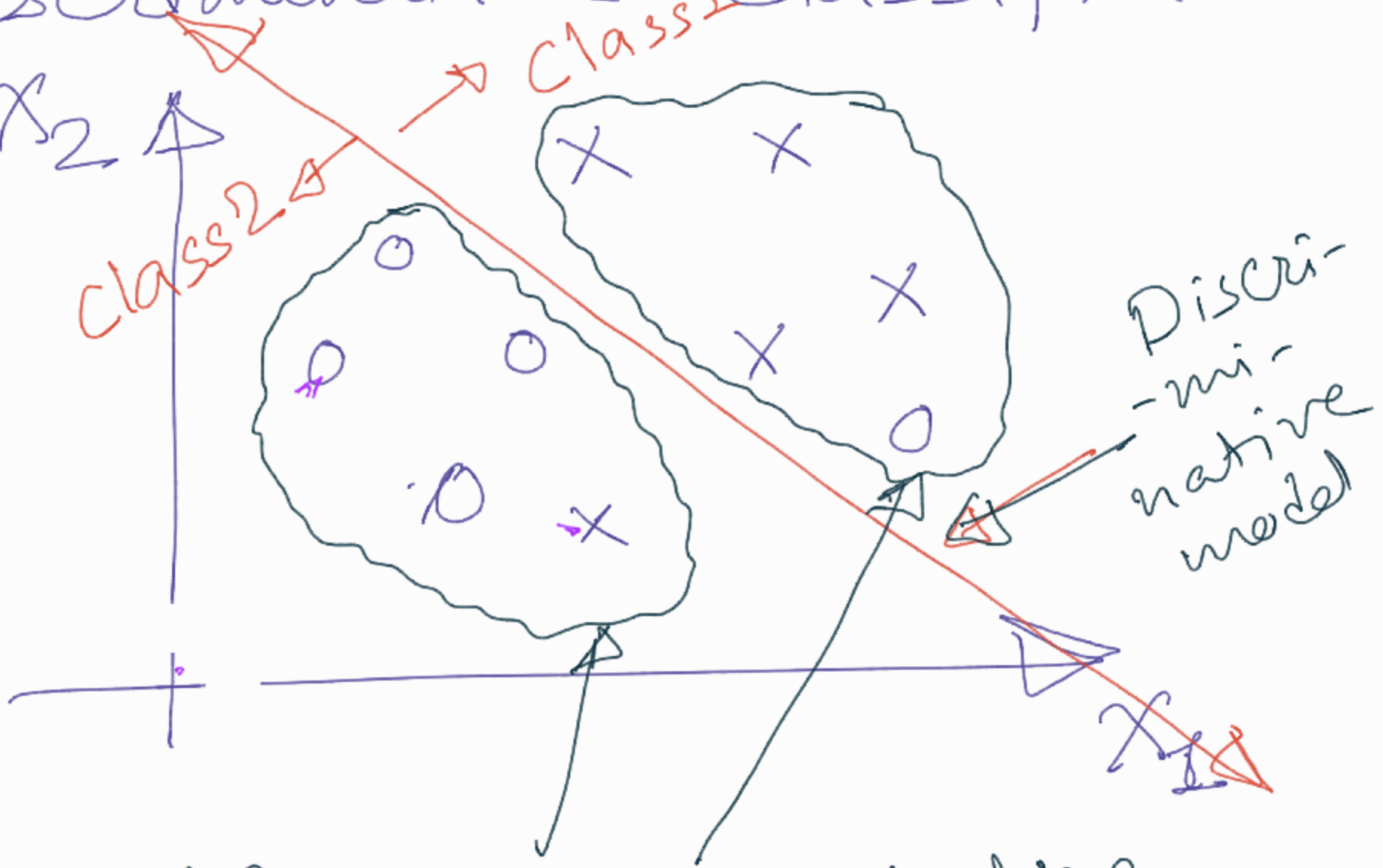
↳ Application to POS created interest in application of ML to NLP,

↳ Speech processing / Recognition

↳ Sequence labeling

- "Every feature is a vector in N-dimensional space."

- Discriminative classifiers



Discri-
-mi-
native
model

Generative → Learn distribution of data points

Neural Network

↳ Perceptron