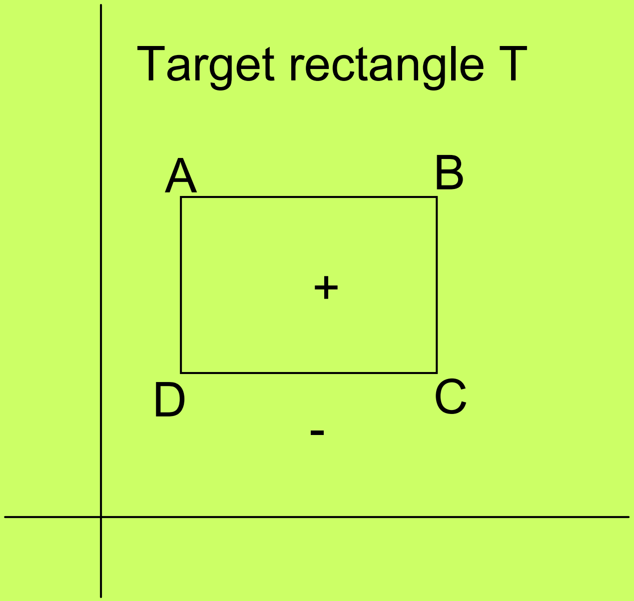
# CS623: Introduction to Computing with Neural Nets *(lecture-18)*

Pushpak Bhattacharyya
Computer Science and Engineering Department
IIT Bombay

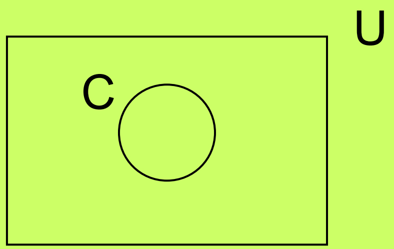
# Learning in Boltzmann m/c

- The meaning of learning probability distribution
- Example: rectangle learning
- Learning algo presented with + and – examples.
- + : points within ABCD
- - : points outside ABCD



# Learning Algorithm

- The algorithm has to output a hypotheses H, that is a good estimate of rectangle T.
- Probably Approximately Correct (PAC) learning used
- Let, U is universe of points
- Let, $c \subset U$, where c is called a concept.
- A collection C of c is called a concept class



# Learning Algorithm

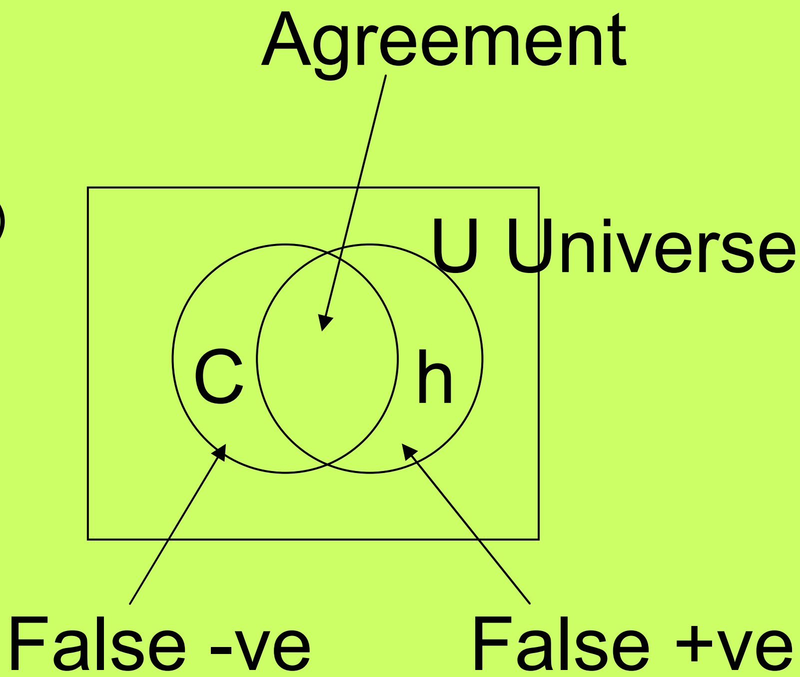
- There is a probability distribution Pr which is unknown, arbitrary but fixed, which produces examples over time

$<s_1, \pm>, <s_2, \pm>, <s_3, \pm>, \ldots, <s_n, \pm>$

- The probability distribution is generated by a “teacher” or “oracle”.
- We want the learning algo to learn ‘c’

# Learning Algorithm

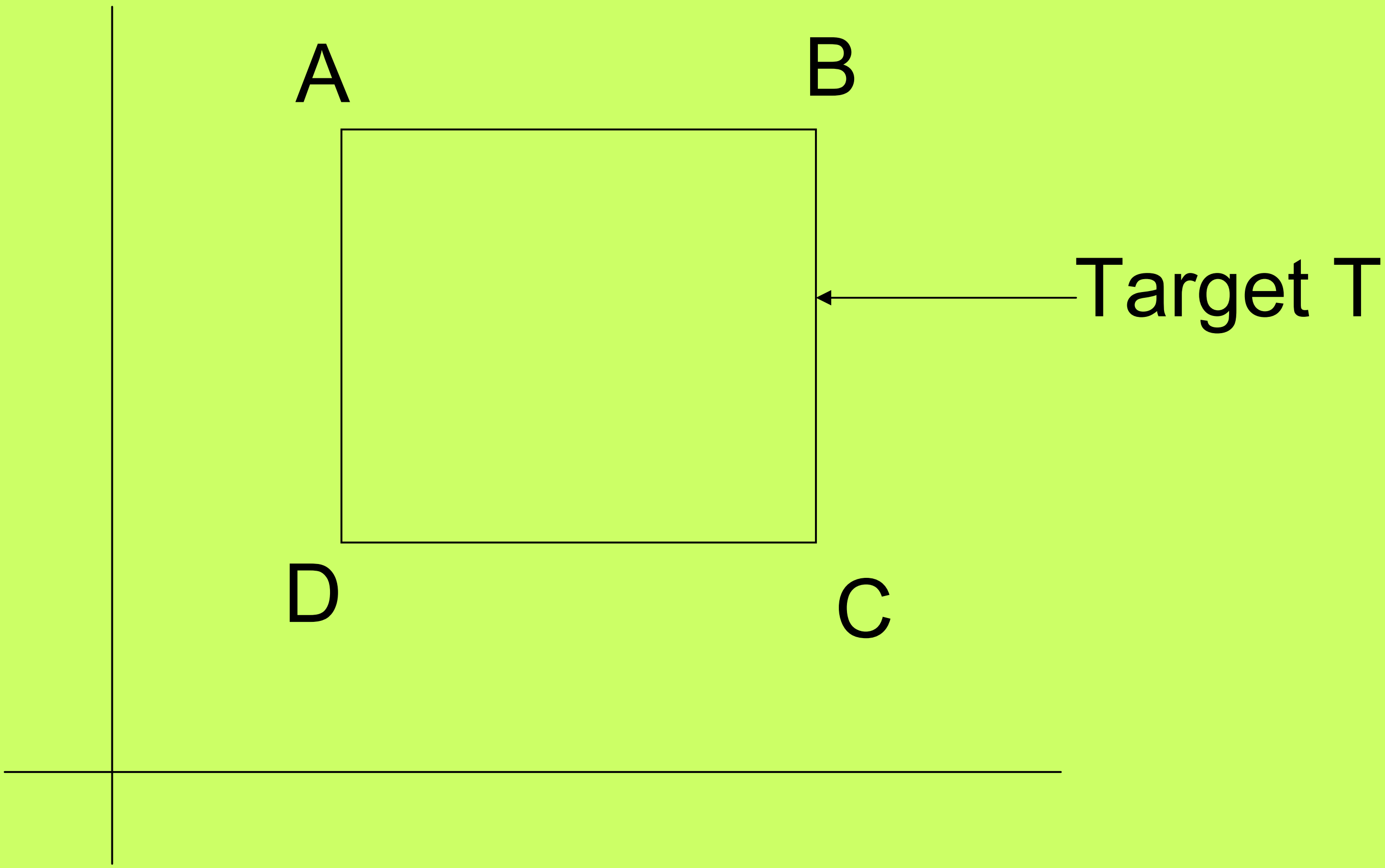
- We say that 'c' is PAC learnt by a hypothesis 'h'
- Learning algo takes place in 2 phases:
  - Training phase (loading)
  - Testing phase (generalization)
- $c \oplus h$ is the error region
- If the examples coming from $c \oplus h$ are of low probability then generalization is good.
- Learning means learning the following things
  - Approximating the Distribution
  - Assigning Name (community accepted)




$C \oplus h =$
Error region

# Key insights from 40 years of Machine Learning

- Learning in vacuum is impossible
  - Learner must already have a lot of knowledge
- The learner must know "what to" learn, called as Inductive Bias
- Example: Learning the rectangle ABCD. The target is to learn the boundary defined by the points A, B, C and D.


A
B
Target T
D
C

# PAC

- We want, $P(C \oplus h) <= \in$
- Definition of Probably Approximately Correct learning is:

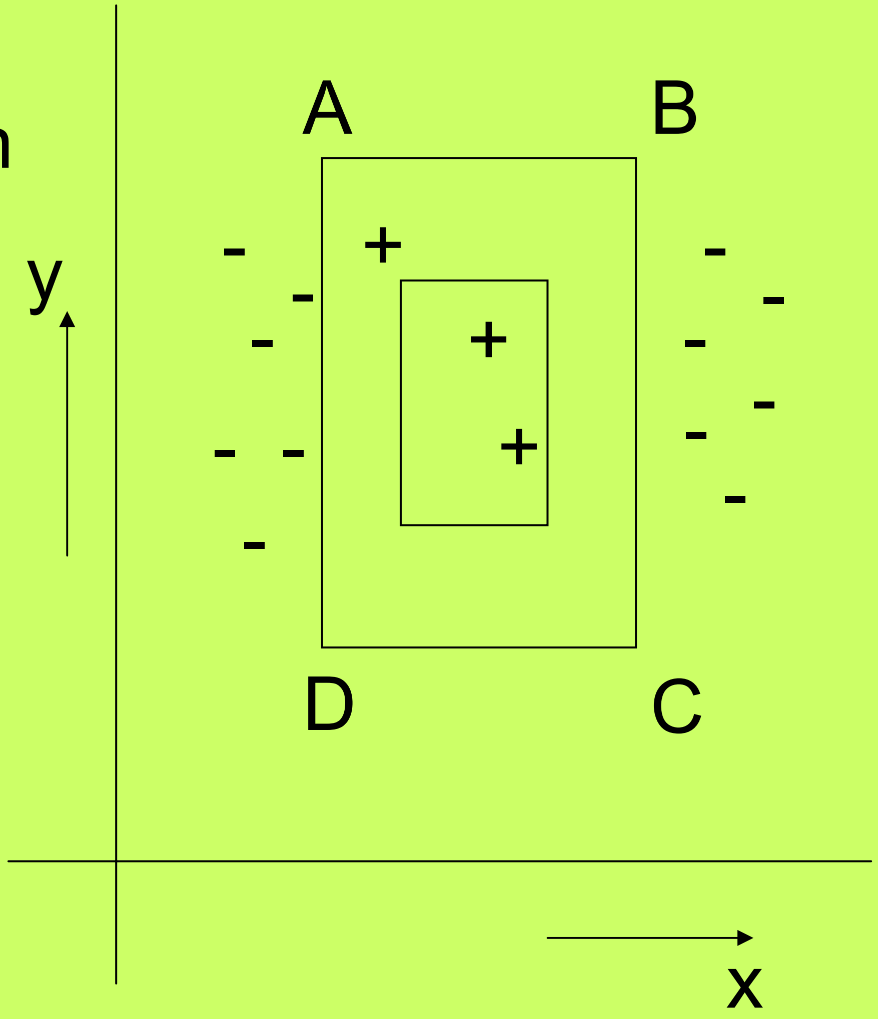
  $P[\ Pr(C \oplus h) <= \in\ ] >= 1 - \delta$

where $\in$ = accuracy factor

$\delta$ = confidence factor

# PAC - Example

- Probability distribution is producing +ve and –ve examples
- Keep track of ($x_{min}$, $y_{min}$) and ($x_{max}$, $y_{max}$) and build rectangle out of it.



# PAC - Algorithm

- In case of rectangle one can prove that the learning algorithm, *viz.*,

1. Ignore –ve examples.
2. Producing $<x_{min}, y_{min}>$ and $<x_{max}, y_{max}>$ of the points, PAC learns the target rectangle if the number of examples is $>= \in/4 \ln \delta/4$

# Illustration of the basic idea of Boltzmann Machine

- To learn the identity function
- The setting is probabilistic, x = 1 or

  x = -1, with uniform probability.
- P(x=1) = 0.5, P(x=-1) = 0.5
- For, x=1, y=1 with P=0.9
- For, x=-1, y=-1 with P=0.9


| x | y |
|---|---|
| 1 | 1 |
| -1 | -1 |

# Illustration of the basic idea of Boltzmann Machine (contd.)

- Let $\alpha$ = output neuron states

  $\beta$ = input neuron states

  $P_{\alpha|\beta}$ = observed probability distribution

  $Q_{\alpha|\beta}$ = desired probability distribution

  $Q_{\beta}$ = probability distribution on input states $\beta$

# Illustration of the basic idea of Boltzmann Machine (contd.)

- The divergence D is given as:

$D = \sum_\alpha \sum_\beta Q_{\alpha|\beta} Q_\beta \ln Q_{\alpha|\beta} / P_{\alpha|\beta}$

called KL divergence formula

$D = \sum_\alpha \sum_\beta Q_{\alpha|\beta} Q_\beta \ln Q_{\alpha|\beta} / P_{\alpha|\beta}$

$>= \sum_\alpha \sum_\beta Q_{\alpha|\beta} Q_\beta ( 1 - P_{\alpha|\beta} / Q_{\alpha|\beta})$

$>= \sum_\alpha \sum_\beta Q_{\alpha|\beta} Q_\beta - \sum_\alpha \sum_\beta P_{\alpha|\beta} Q_\beta$

$>= \sum_\alpha \sum_\beta Q_{\alpha\beta} - \sum_\alpha \sum_\beta P_{\alpha\beta}$

{$Q_{\alpha\beta}$ and $P_{\alpha\beta}$ are joint distributions}

$>= 1 – 1 = 0$