Single-Layer Perceptron & Bayes Classifier

Dr. Ratna Babu Chinnam

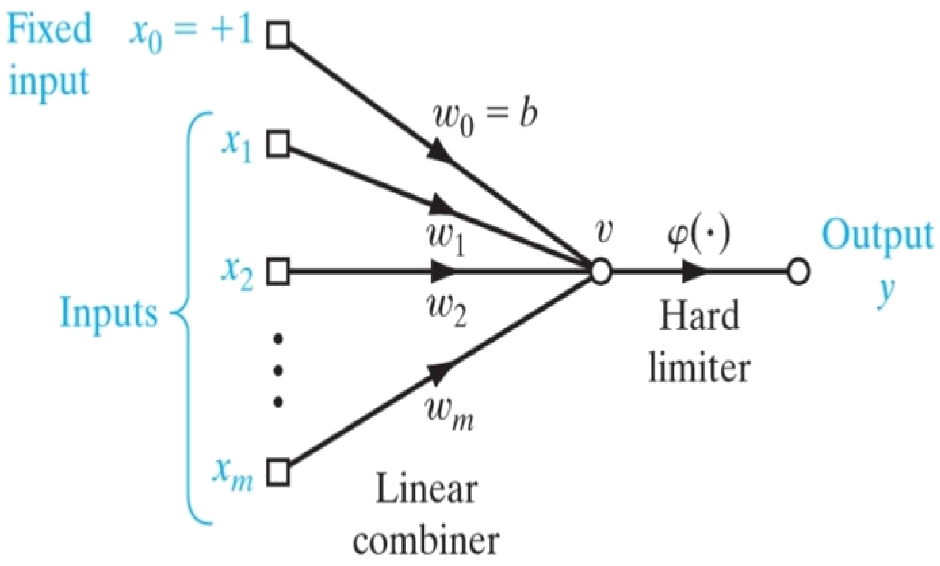
Industrial & Systems Engineering

Wayne State University

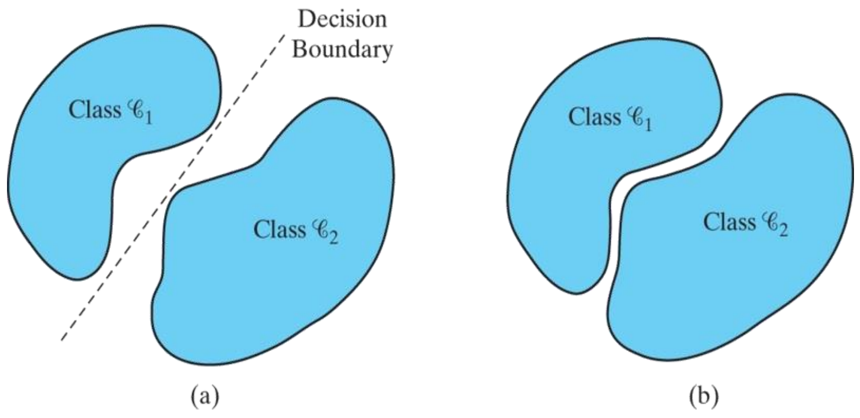
Primary Source: Simon Haykin (2009)

Introduction

- Resonblatt's [1958] ***perceptron*** is a neural network used for classification of ***linearly separable*** patterns
- Consists of a single layer of neurons
- **Perceptron Convergence Theorem**:
  - If patterns are linearly separable, converges and positions the decision surface in the form of a “hyperplane” between the two classes
  - Can be extended for multiple classes
    - One output neuron per class



Perceptron Signal-Flow Graph

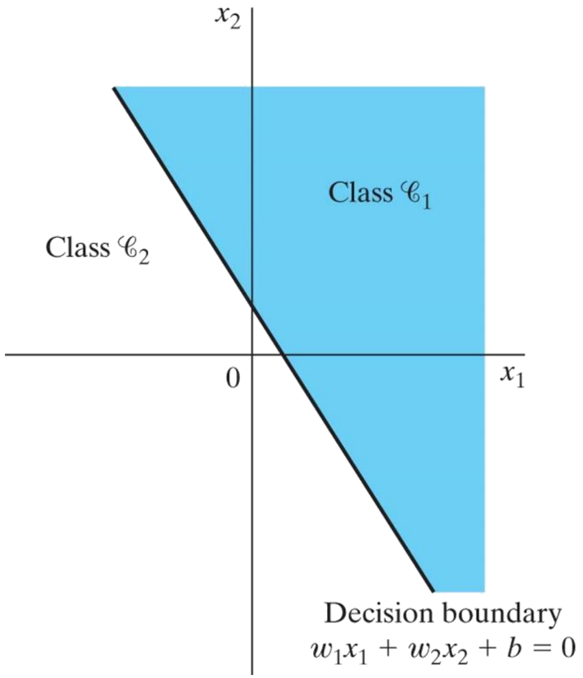


Linearly Separable & Unseparable Classes

Perceptron

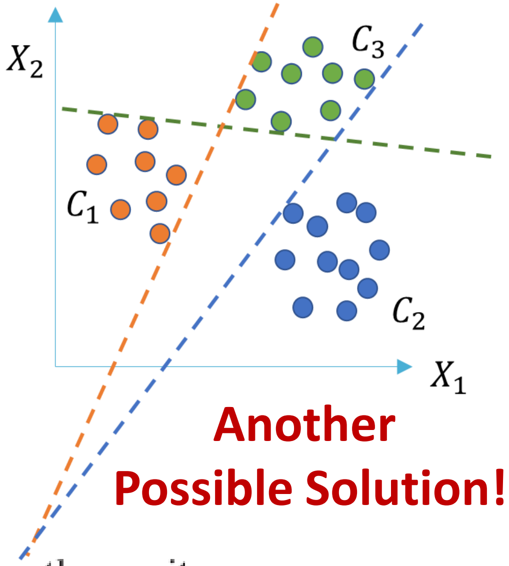
- Employs threshold transfer function (outputs ± 1)
- **Goal**: Correctly classify points $x_1, x_2, \ldots, x_m$ into one of two ***linearly separable*** classes C_1 or C_2
- **Decision**: Assign input to class based on output
- **Extensible to Multiple Classes**: Assign input to neuron with largest activation potential (v)
- Neuron bias allows boundary to shift from origin
- **Iterative Process**: Present points in random order
- Weights updated using ***error-correction rule***:

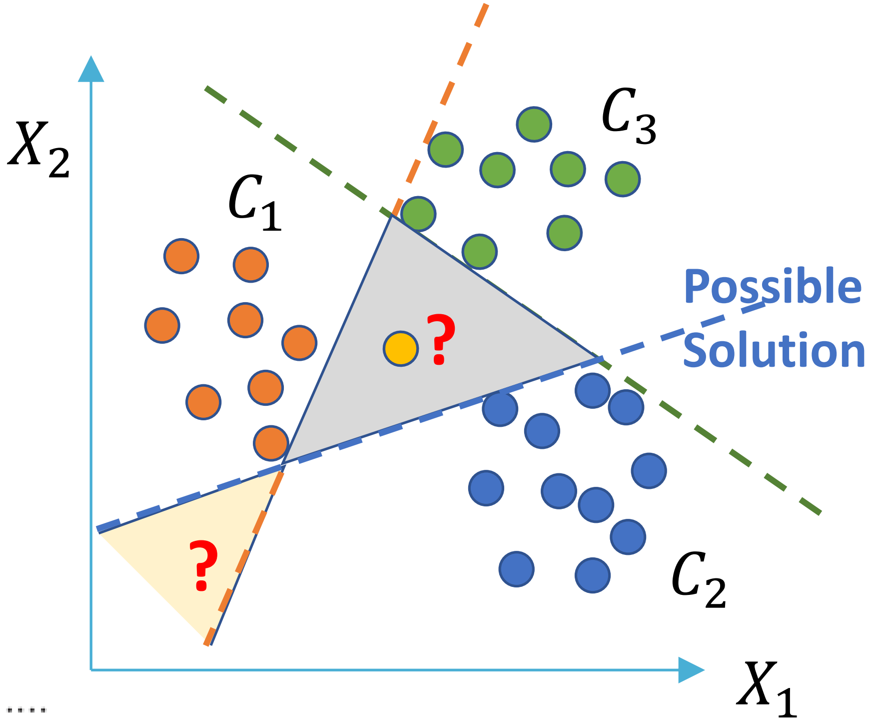
$$w(n+1) = w(n) + \eta[d(n) - y(n)]x(n)$$



Decision Boundary Hyperplane (straight line for 2-D two-class problem)

Perceptron Algorithm





Variables and Parameters:

$\mathbf{x}(n)$ = $(m+1)$-by-1 input vector
= $[+1, x_1(n), x_2(n), \ldots, x_m(n)]^T$
$\mathbf{w}(n)$ = $(m+1)$-by-1 weight vector
= $[b, w_1(n), w_2(n), \ldots, w_m(n)]^T$
b = bias
$y(n)$ = actual response (quantized)
$d(n)$ = desired response
η = learning-rate parameter, a positive constant less than unity

1. *Initialization.* Set $\mathbf{w}(0) = \mathbf{0}$. Then perform the following computations for time-step $n = 1, 2, \ldots$.
2. *Activation.* At time-step n, activate the perceptron by applying continuous-valued input vector $\mathbf{x}(n)$ and desired response $d(n)$.
3. *Computation of Actual Response.* Compute the actual response of the perceptron as

$$y(n) = \operatorname{sgn}[\mathbf{w}^T(n)\mathbf{x}(n)]$$

where $\operatorname{sgn}(\cdot)$ is the signum function.

4. *Adaptation of Weight Vector.* Update the weight vector of the perceptron to obtain

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \eta[d(n) - y(n)]\mathbf{x}(n)$$

where

$$d(n) = \begin{cases} +1 & \text{if } \mathbf{x}(n) \text{ belongs to class } \mathscr{C}_1 \\ -1 & \text{if } \mathbf{x}(n) \text{ belongs to class } \mathscr{C}_2 \end{cases}$$

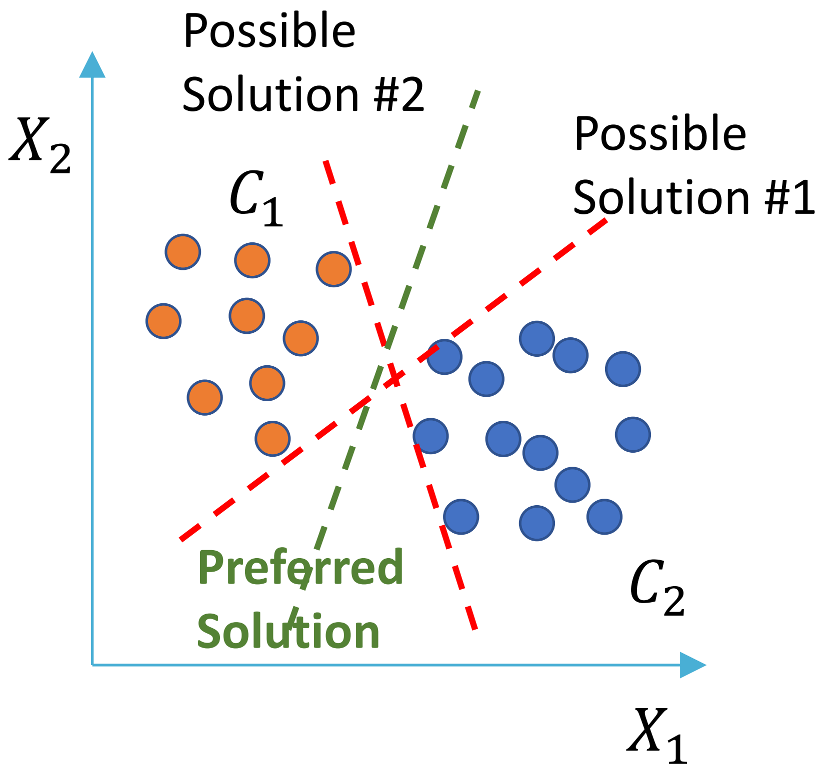
5. *Continuation.* Increment time step n by one and go back to step 2.

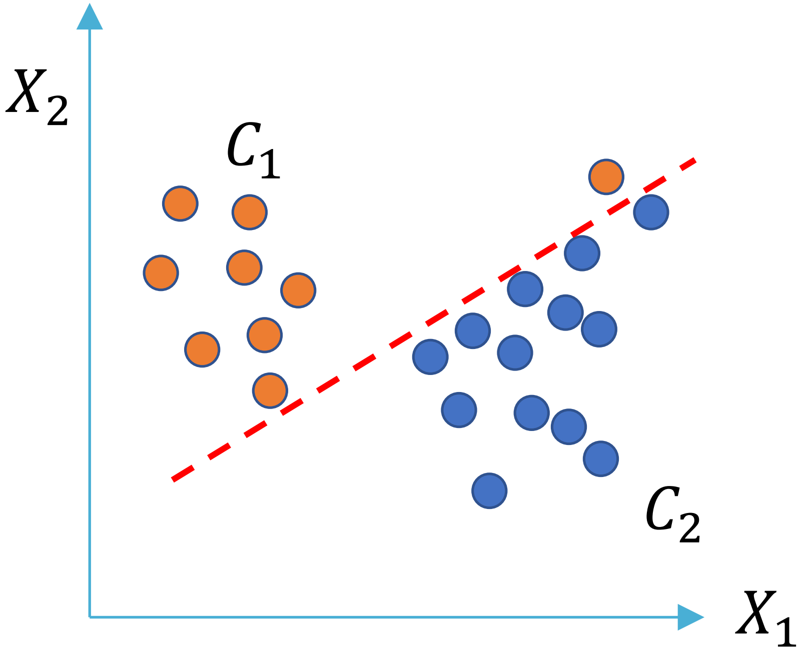
Termination Criteria?

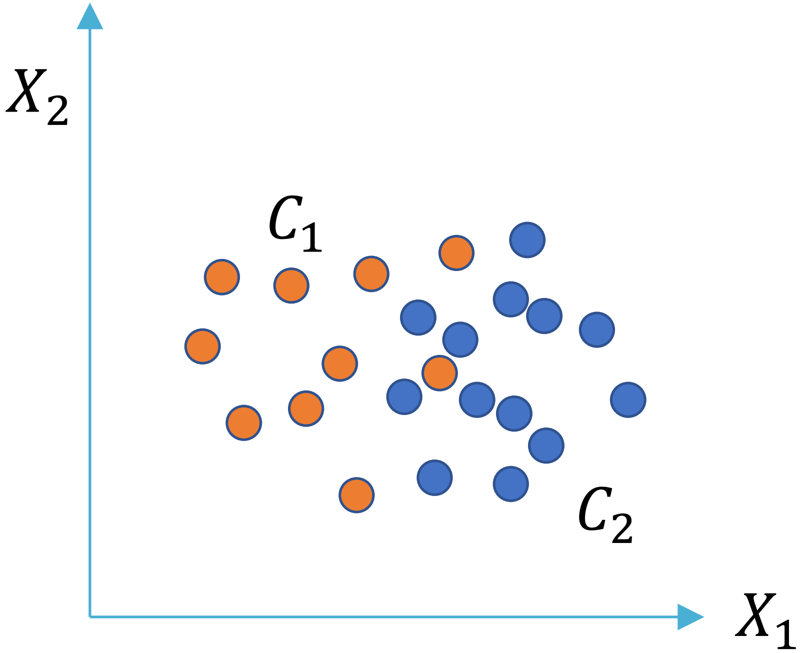
Learning Rate?

Perceptron: Limitations

- Optimality of Decision Surface!
  - Stops learning once data points are correctly classified
- Robustness to Noise
- What about Non-Separable Classes?





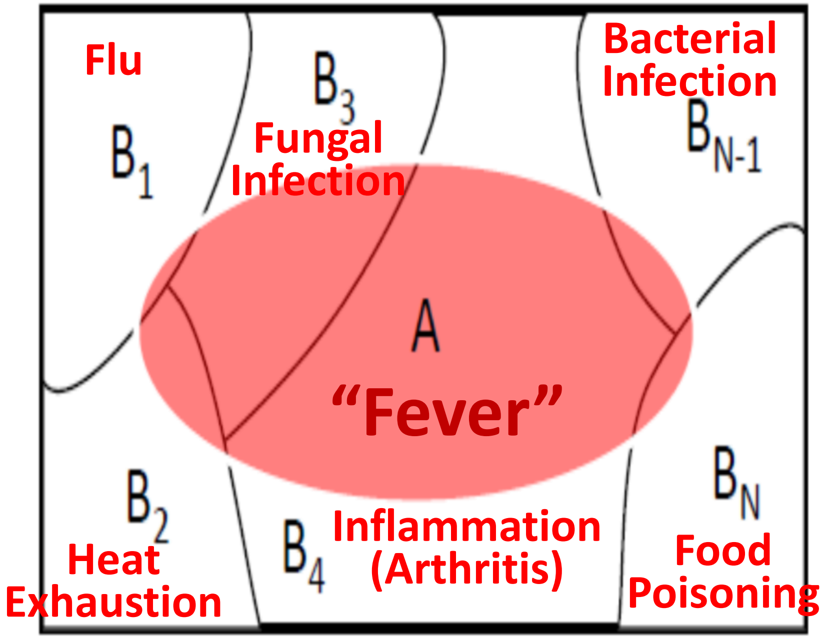


Bayes Theorem

- Assume $\{B_1, B_2 \ldots B_N\}$ is a partition of S
- Suppose that event A occurs
- What is the probability of event B_j?
- **Bayes Theorem/Rule**: From definition of conditional probability and Theorem of total probability:

$$P[B_j|A] = \frac{P[A \cap B_j]}{P[A]}$$

$$= \frac{P[A|B_j]P[B_j]}{\sum_{k=1}^{N} P[A|B_k]P[B_k]}$$



Bayes Classifier

- Has some resemblance to the Perceptron
  - Under Gaussian distributions, Bayes classifier reduces to a linear classifier
- When used for pattern classification:

$$P[\omega_j|x] = \frac{P[x|\omega_j]P[\omega_j]}{\sum_{k=1}^{N} P[x|\omega_k]P[\omega_k]} = \frac{p[x|\omega_j]P[\omega_j]}{p[x]}$$

  where ω_j is the j-th class and x is the feature/observation vector
- **Decision Rule**: Choose class ω_j with highest $P[\omega_j|x]$?
  - Class is more "likely" given observation x
- **Terminology**:

  $P[\omega_j]$: prior probability (of class ω_j)

  $P[\omega_j|x]$: posterior probability (of class ωj given the observation x)

  $p[x|\omega_j]$: likelihood (probability of observation x given class ω_j)

  $p[x]$: normalization constant (does not affect decision)

What if misclassification costs are not the same?

Bayes Classifier: General Likelihood Ratio Test

- **Bayes Risk**: Minimize average "risk"
- For a two-class problem (classes C_1 and C_2)

$$\mathcal{R} = c_{21}p_1 \int_{\mathcal{X}_2} p_{\mathbf{x}}(\mathbf{x}|C_1)d\mathbf{x} + c_{12}p_2 \int_{\mathcal{X}_1} p_{\mathbf{x}}(\mathbf{x}|C_2)d\mathbf{x}$$

p_i: prior prob that $\mathbf{x}$ is from subspace $\mathcal{X}_i$

$$p_1 + p_2 = 1$$

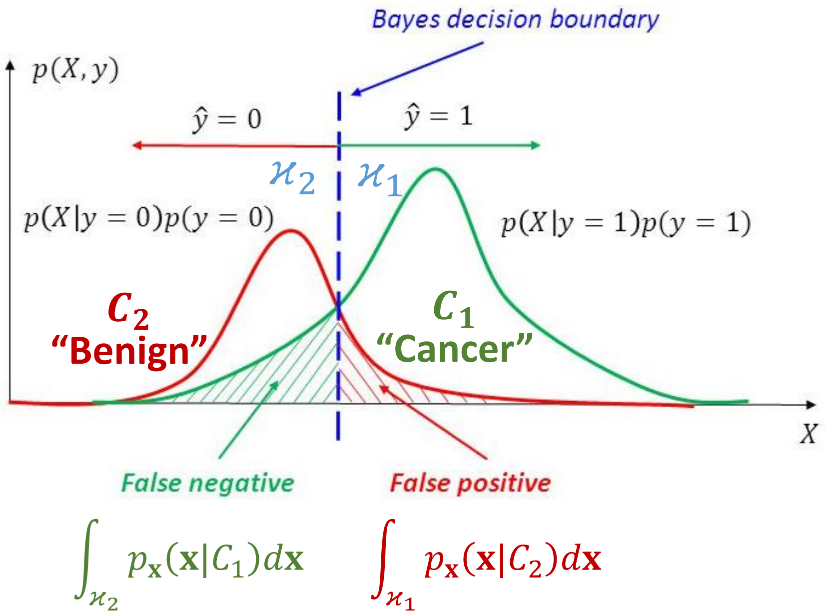
c_{ij}: cost of deciding in favor of C_i when C_j is true

- Define: $\Lambda(\mathbf{x}) = \dfrac{p_{\mathbf{x}}(\mathbf{x}|C_1)}{p_{\mathbf{x}}(\mathbf{x}|C_2)}$ $\leftarrow$ Likelihood ratio

$$\xi = \frac{p_2(c_{12} - c_{22})}{p_1(c_{21} - c_{11})}$$ $\leftarrow$ Threshold

- **Bayes Classifier**: *If likelihood ratio $\Lambda(\mathbf{x})$ i> ξ, assign $\mathbf{x}$ to class C_1. Otherwise, assign it to class C_2.*

Where to establish decision boundary?



Bayes Classifier: Example

- **Clinical Problem**: Decide if a patient has a particular medical condition on the basis of an imperfect test!
- **Nomenclature**:
  - True-negative rate $P(-|\neg COND)$ of a test is called its SPECIFICITY
  - True-positive rate $P(+|COND)$ of a test is called its SENSITIVITY
- **Problem**:
  - Population of 10,000 with a 1% prevalence for condition
  - Test has 98% specificity and 90% sensitivity
  - Test result comes out POSITIVE
  - What is the probability that patient has condition?

Bayes Classifier: Example Solution

- **Applying Bayes Rule:**

$$P[COND|+] = \frac{p[+|COND]\ P[COND]}{p[+]}$$

$$= \frac{p[+|COND]\ P[COND]}{p[+|COND]P[COND]+p[+|\neg COND]\ P[\neg COND]}$$

$$= \frac{0.90 \times 0.01}{0.90 \times 0.01+(1-0.98) \times 0.99}$$

$$= 0.3125$$

Naïve-Bayes Classifier

- **Pre-requisite for Bayes Classifier**: $p_{\mathbf{X}}(\mathbf{X}|C_i) = p_{\mathbf{X}}(X_1, X_2, \ldots, X_m|C_i)$
- **Difficulty**: Learning this joint distribution as the size of the input vector grows
- Applying the "**chain rule**" repeatedly:

$$p_{\mathbf{X}}(X_1, X_2, \ldots, X_m|C_i) = p_{\mathbf{X}}(X_1|C_i)p_{\mathbf{X}}(X_2, \ldots, X_m|C_i, X_1)$$
$$= p_{\mathbf{X}}(X_1|C_i)p_{\mathbf{X}}(X_2|C_i, X_1)p_{\mathbf{X}}(X_3, \ldots, X_m|C_i, X_1, X_2)$$
$$\cdots$$
$$= p_{\mathbf{X}}(X_1|C_i)p_{\mathbf{X}}(X_2|C_i, X_1)p_{\mathbf{X}}(X_3|C_i, X_1, X_2) \ldots p_{\mathbf{X}}(X_m|C_i, X_1, X_2, X_3, \ldots, X_{m-1})$$

Symptoms for Bronchitis:
Cough; Mucus; Fatigue;
Shortness of breath;
Slight fever and chills …

- **Naïve Bayes Assumption**: All input attributes are conditionally independent!

$$p_{\mathbf{X}}(X_1, X_2, \ldots, X_m|C_i)$$
$$= p_X(X_1|C_i)p_X(X_2|C_i, X_1)p_X(X_3|C_i, X_1, X_2) \ldots p_X(X_m|C_i, X_1, X_2, X_3, \ldots, X_{m-1})$$
$$= p_X(X_1|C_i)p_X(X_2|C_i)p_X(X_3|C_i) \ldots p_X(X_m|C_i)$$
$$= \prod_{i=1}^{m} p_{\mathbf{X}}(X_i|C_i)$$

Assuming binary variables, compare # of parameters in the two distributions!

More importantly, effective in practice!