SINGLE-LAYER PERCEPTRON

Source: Haykin, 2009

- Resonblatt's [1958] perceptron is a neural network used for classification of *linearly separable* patterns.
- Network consists of a single layer of neurons with adjustable synaptic weights and bias.
- *Perceptron convergence theorem:*
  If patterns are drawn from two linearly separable classes, the perceptron learning procedure converges and positions the decision surface in the form of a hyperplane between the two classes. The theorem can be extended for multiple classes.

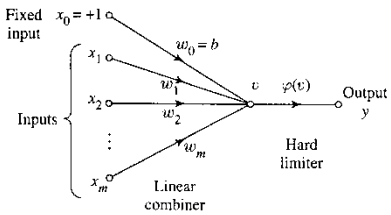
PERCEPTRON

- Employs a nonlinear neuron using the hard limiter or threshold function (with outputs normally set at ± 1).
- The goal of perceptron is to correctly classify the set of externally applied stimuli $x_1, x_2, \ldots, x_m$ into one of two *linearly separable* classes C_1 or C_2. The decision rule is to assign the input to class C_1 or C_2 depending on the sign of the neuron output.
- The framework can be extended to multiple classes (for example, a neuron can be assigned for each of the classes with the winning neuron taking it all).
- The neuron bias allows the boundary to shift away from the origin when necessary.
- The perceptron synaptic weights are normally adapted on an iteration-by-iteration basis using an error-correction rule known as the perceptron convergence algorithm.
- *Performance Measure:*
  A performance function proposed by Shynk [1990] that befits the operation of a perceptron is
  $J = -E[e(n)v(n)]$ ← MSE is inappropriate for being not-differentiable
  where $e(n)$ is the error signal and $v(n)$ is the hard limiter input.
  The instantaneous estimate of the function is
  $\hat{J}(n) = -[d(n) - y(n)]v(n)$.
  The instantaneous gradient vector of $\hat{J}(n)$ with respect to $\mathbf{w}(n)$ is

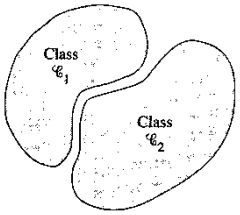
$$\nabla_{\mathbf{w}}\hat{J}(n) = -[d(n) - y(n)]\frac{\partial v(n)}{\partial \mathbf{w}(n)}$$
$$= -[d(n) - y(n)]x(n).$$

  Hence, in accordance with error-correction rule, the weight update rule is

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



Perceptron Signal-Flow Graph.



Linearly Unseparable Classes.

Table: Summary of Perceptron Convergence 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
$= [b(n), w_1(n), w_2(n), \ldots, w_m(n)]^T$
$b(n)$ = 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:

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

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

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

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

where

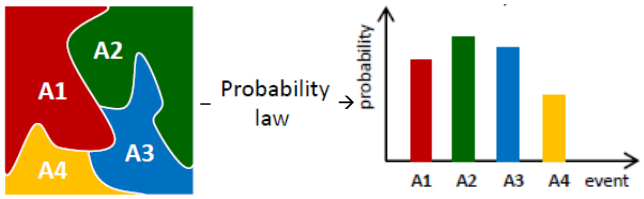
$$d(n) = \begin{cases} +1 & \text{if } \mathbf{x}(n) \text{ belongs to class } \mathscr{C}_1 \\ -1 & \text{if } \mathbf{x}(n) \text{ belongs to class } \mathscr{C}_2 \end{cases}$$

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

RELATION BETWEEN THE PERCEPTRON AND BAYES CLASSIFIER

- Perceptron bears a relationship to a classical pattern classifier known as the Bayes classifier
- When the environment is Gaussian, the Bayes classifier also reduces to a linear classifier

REVIEW OF PROBABILITY THEORY



Sample Space **Probability Distribution**

Definitions (informal)

- Probabilities are numbers assigned to events that indicate "*how likely*" it is that the event will occur when a random experiment is performed
- A probability law for an experiment is a rule that assigns probabilities to the events in the experiment
- Sample space S of a random experiment is the set of all possible outcomes

Axioms of Probability

Axiom I: $P[A_i] \geq 0$
Axiom II: $P[S] = 1$
Axiom III: $A_i \cap A_j = \emptyset \Rightarrow P[A_i \cup A_j] = P[A_i] + P[A_j]$

More Properties of Probability

$P[A^C] = 1 - P[A]$

$P[A] \leq 1$

$P[\emptyset] = 0$

$given\ \{A_1 \dots A_N\}, \{A_i \cap A_j = \emptyset, \forall ij\} \Rightarrow P\left[\cup_{k=1}^{N} A_k\right] = \sum_{k=1}^{N} P[A_k]$

$P[A_1 \cup A_2] = P[A_1] + P[A_2] - P[A_1 \cap A_2]$

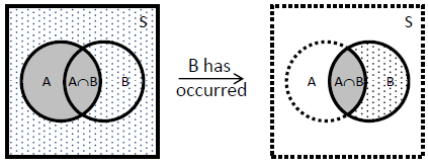
$P\left[\cup_{k=1}^{N} A_k\right] = \sum_{k=1}^{N} P[A_k] - \sum_{j<k}^{N} P\left[A_j \cap A_k\right] + \cdots + (-1)^{N+1} P[A_1 \cap A_2 \dots \cap A_N]$

Conditional Probability

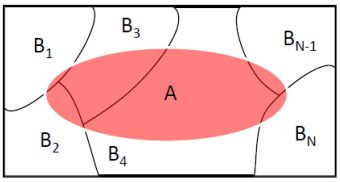
- If A and B are two events, the probability of event A when we already know that event B has occurred is

$$P[A|B] = \frac{P[A \cap B]}{P[B]}\ if\ P[B] > 0$$

- Interpretation
  - New evidence "*B has occurred*" has following effects
  - Original sample space S (the square) becomes B (the rightmost circle)
  - Event A becomes A∩B
  - P[B] simply re-normalizes the probability of events that occur jointly with B



Theorem of Total Probability



- Let $B_1, B_2 \dots B_N$ be a partition of S (mutually exclusive that add to S)
- Any event A can be represented as

$A = A \cap S = A \cap (B_1 \cup B_2 \dots B_N) = (A \cap B_1) \cup (A \cap B_2) \dots (A \cap B_N)$

- Since $B_1, B_2 \dots B_N$ are mutually exclusive, then

$$P[A] = P[A \cap B_1] + P[A \cap B_2] + \cdots + P[A \cap B_N]$$

- and, therefore

$P[A] = P[A|B_1]P[B_1] + \cdots P[A|B_N]P[B_N] = \sum_{k=1}^{N} P[A|B_k]P[B_k]$

Bayes Theorem

- Assume $\{B_1, B_2 \dots B_N\}$ is a partition of S
- Suppose that event A occurs
- What is the probability of event B_j?
- Using the definition of conditional probability and the Theorem of total probability we obtain

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

- This is known as Bayes Theorem or Bayes Rule, and is (one of) the most useful relations in probability and statistics

Bayes Theorem & Statistical Pattern Recognition

- When used for pattern classification, Bayes Theorem is generally expressed as

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

where ω_j is the j-th class and x is the feature/observation vector

- Decision rule: Should we just choose the class ω_j with highest $P\left[\omega_j|x\right]$?
  - Why not? – This class is more "likely" given observation x
- Each term in the Bayes Theorem has a special name
  - $P[\omega_j]$ prior probability (of class ωj)
  - $P\left[\omega_j|x\right]$ posterior probability (of class ωj given the observation x)
  - $p[x|\omega_j]$ likelihood (probability of observation x given class ωj)
  - $p[x]$ normalization constant (does not affect the decision)

Bayes Pattern Recognition - EXAMPLE

- Consider a clinical problem where we need to decide if a patient has a particular medical condition on the basis of an imperfect test
  - Someone with the condition may go undetected (false-negative)
  - Someone free of the condition may yield a positive result (false-positive)
- Nomenclature
  - The true-negative rate P(NEG|¬COND) of a test is called its SPECIFICITY
  - The true-positive rate P(POS|COND) of a test is called its SENSITIVITY
- Problem
  - Assume a population of 10,000 with a 1% prevalence for the condition
  - Assume that we design a test with 98% specificity and 90% sensitivity
  - Assume you take the test, and the result comes out POSITIVE
  - What is the probability that you have the condition?
- Solution
  - Fill in the joint frequency table

	TEST IS POSITIVE	TEST IS NEGATIVE	ROW TOTAL
HAS CONDITION	*True-positive P(POS|COND)*	*False-negative P(NEG|COND)*	
FREE OF CONDITION	*False-positive P(POS|¬COND)*	*True-negative P(NEG|¬COND)*	
COLUMN TOTAL			

  - Apply Bayes rule

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

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

BAYES CLASSIFIER

- In the Bayes classifier, or Bayes hypothesis testing procedure, we minimize the average risk, denoted by $\mathcal{R}$ (***Bayes Risk***)
- For a two-class problem, represented by classes C_1 and C_2, $\mathcal{R}$ is defined as:

$$\mathcal{R} = c_{11}p_1 \int_{\mathcal{H}_1} p_{\mathbf{x}}(\mathbf{x}|C_1)d\mathbf{x} + c_{22}p_2 \int_{\mathcal{H}_2} p_{\mathbf{x}}(\mathbf{x}|C_2)d\mathbf{x}$$

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

$p_i =$ *prior probability* that the observation vector $\mathbf{x}$ is drawn from subspace $\mathcal{H}_i$, with $i = 1,2$, and $p_1 + p_2 = 1$

$c_{ij} =$ cost of deciding in favor of class C_i represented by subspace $\mathcal{H}_i$ when class C_j is true

$p_{\mathbf{x}}(\mathbf{x}|C_i) =$ conditional probability density function of the random vector $\mathbf{x}$, given that the observation vector $\mathbf{x}$ is drawn from subspace $\mathcal{H}_i$

- The first two terms on the right-hand side represent correct decisions/classifications, whereas the last two terms represent incorrect decisions/misclassifications.
- Each decision is weighted by the product of two factors: the cost involved in making the decision and the relative frequency with which it occurs (i.e., prior probability).
- Intention is to determine a strategy for the minimum average risk.
- Because we require that a decision be made, each observation vector x must be assigned in the over-all observation space to either $\mathcal{H}_1$ or $\mathcal{H}_2$. Thus, $\mathcal{H} = \mathcal{H}_1 + \mathcal{H}_2$.
- Accordingly, we may rewrite $\mathcal{R}$ in the equivalent form:

$$\mathcal{R} = c_{11}p_1\int_{\mathcal{H}_1} p_{\mathbf{x}}(\mathbf{x}|C_1)d\mathbf{x} + c_{22}p_2\int_{\mathcal{H}-\mathcal{H}_1} p_{\mathbf{x}}(\mathbf{x}|C_2)d\mathbf{x}$$
$$+c_{21}p_1\int_{\mathcal{H}-\mathcal{H}_1} p_{\mathbf{x}}(\mathbf{x}|C_1)d\mathbf{x} + c_{12}p_2\int_{\mathcal{H}_1} p_{\mathbf{x}}(\mathbf{x}|C_2)d\mathbf{x}$$

where $c_{11} < c_{21}$ and $c_{22} < c_{12}$.
Given that

$$\int_{\mathcal{H}} p_{\mathbf{x}}(\mathbf{x}|C_1)d\mathbf{x} = \int_{\mathcal{H}} p_{\mathbf{x}}(\mathbf{x}|C_2)d\mathbf{x} = 1$$

we may rewrite $\mathcal{R}$ as follows:

$$\mathcal{R} = c_{21}p_1 + c_{22}p_2$$
$$+\int_{\mathcal{H}_1}[p_2(c_{12}-c_{22})p_{\mathbf{x}}(\mathbf{x}|C_2) - p_1(c_{21}-c_{11})p_{\mathbf{x}}(\mathbf{x}|C_1)]d\mathbf{x}$$

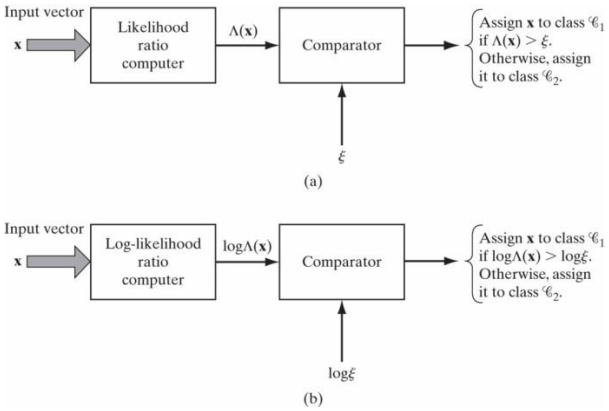
- First two terms on the right- hand side of $\mathcal{R}$ represent a fixed cost.
- The following strategy can be deduced from $\mathcal{R}$ for optimum classification:
  1. All $\mathbf{x}$ for which the integrand (i.e., the expression inside the square brackets) is negative should be assigned to subspace $\mathcal{H}_1$ (i. e., class C_1).
  2. All $\mathbf{x}$ for which the integrand is positive should be excluded from subspace $\mathcal{H}_1$ (i.e., assigned to class C_2).
  3. Values of $\mathbf{x}$ for which the integrand is zero have no effect on $\mathcal{R}$ and may be assigned arbitrarily. Let us assume that these points are assigned to subspace $\mathcal{H}_2$ (i.e., class C_2).
- We may then formulate the Bayes classifier as follows:
  If condition $p_1(c_{21}-c_{11})p_{\mathbf{x}}(\mathbf{x}|C_1) > p_2(c_{12}-c_{22})p_{\mathbf{x}}(\mathbf{x}|C_2)$ *holds, assign observation vector* $\mathbf{x}$ *to subspace* $\mathcal{H}_1$ *(i.e., class* C_1*). Otherwise, assign* $\mathbf{x}$ *to* $\mathcal{H}_2$ *(i.e., class* C_2*).*
- To simplify, let us define the following:

$$\Lambda(\mathbf{x}) = \frac{p_{\mathbf{x}}(\mathbf{x}|C_1)}{p_{\mathbf{x}}(\mathbf{x}|C_2)} \quad \leftarrow \text{Likelihood ratio}$$
$$\xi = \frac{p_2(c_{12}-c_{22})}{p_1(c_{21}-c_{11})} \quad \leftarrow \text{Threshold}$$

- We may now reformulate the Bayes classifier:
  If, for an observation vector $\mathbf{x}$*, the likelihood ratio* $\Lambda(\mathbf{x})$ *is greater than the threshold* ξ*, assign* $\mathbf{x}$ *to class* C_1*. Otherwise, assign it to class* C_2*.*

Notes:

- The data processing involved is confined entirely to the computation of $\Lambda(\mathbf{x})$.
- ξ is affected by values of prior probabilities and costs involved in the decision-making process.
- From a computational point of view, working with the logarithm of the likelihood ratio is preferred. This is valid for two reasons: 1) Logarithm is a monotonic function. 2) Likelihood ratio and threshold are both positive.



Two equivalent implementations of the Bayes classifier: (a) Likelihood ratio test, (b) Log-likelihood ratio test.

NAÏVE-BAYES CLASSIFIER

- Bayes classifier requires knowledge of the following:
  $p_{\mathbf{X}}(\mathbf{X}|C_i) = p_{\mathbf{X}}(X_1, X_2, \dots, X_m|C_i)$
- Difficulty: Learning this joint distribution as the size of the input vector grows
- Applying the "chain rule" repeatedly, we get the following:
  $p_{\mathbf{X}}(X_1, X_2, \dots, X_m|C_i)$
  $= p_{\mathbf{X}}(X_1|C_i)p_{\mathbf{X}}(X_2, \dots, X_m|C_i, X_1)$
  $= p_{\mathbf{X}}(X_1|C_i)p_{\mathbf{X}}(X_2|C_i, X_1)p_{\mathbf{X}}(X_3, \dots, X_m|C_i, X_1, X_2)$
  $= p_{\mathbf{X}}(X_1|C_i)p_{\mathbf{X}}(X_2|C_i, X_1)\dots p_{\mathbf{X}}(X_m|C_i, X_1, X_2, X_3, \dots, X_{m-1})$
- Naïve Bayes Assumption: All input attributes are conditionally independent!
  $p_{\mathbf{X}}(X_1, X_2, \dots, X_m|C_i)$
  $= p_X(X_1|C_i)p_X(X_2|C_i, X_1)\dots p_X(X_m|C_i, X_1, X_2, X_3, \dots, X_{m-1})$
  $= p_X(X_1|C_i)p_X(X_2|C_i)p_X(X_3|C_i)\dots p_X(X_m|C_i)$
  $= \prod_{i=1}^{m} p_{\mathbf{X}}(X_i|C_i)$

BAYES CLASSIFIER FOR A GAUSSIAN DISTRIBUTION

- Consider the special case of a two-class problem, for which underlying distribution is Gaussian.
- The random vector $\mathbf{X}$ has a mean value that depends on whether it belongs to class C_1 or class C_2, but the covariance matrix of $\mathbf{X}$ is the same for both classes.

$$\text{Class } C_1: \quad \mathrm{E}[\mathbf{X}] = \boldsymbol{\mu}_1$$
$$\mathrm{E}[(\mathbf{X}-\boldsymbol{\mu}_1)(\mathbf{X}-\boldsymbol{\mu}_1)^{\boldsymbol{T}}] = \boldsymbol{C}$$
$$\text{Class } C_2: \quad \mathrm{E}[\mathbf{X}] = \boldsymbol{\mu}_2$$
$$\mathrm{E}[(\mathbf{X}-\boldsymbol{\mu}_2)(\mathbf{X}-\boldsymbol{\mu}_2)^{\boldsymbol{T}}] = \boldsymbol{C}$$

- Covariance matrix $\boldsymbol{C}$ is non-diagonal, which means that the samples drawn from classes and are correlated.
- It is assumed that $\boldsymbol{C}$ is nonsingular, so that its inverse matrix $\boldsymbol{C}^{-1}$ exists.
- Thus, we may express the conditional probability density function of $\mathbf{X}$ as the multivariate Gaussian distribution, where m is the dimensionality of the observation vector $\mathbf{x}$.

$$p_{\mathbf{x}}(\mathbf{x}|C_i) = \frac{1}{(2\pi)^{m/2}(\det(\boldsymbol{C}))^{\frac{1}{2}}} \cdot \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^{\boldsymbol{T}}\boldsymbol{C}^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)\right), \quad i = 1,2$$

- We further assume the following:
  1. The two classes are equiprobable: $p_1 = p_2 = 1/2$
  2. Misclassifications carry the same cost and no cost is incurred on correct classifications: $c_{21} = c_{12}$ and $c_{11} = c_{22} = 0$.
- We can now design the Bayes classifier for the two-class problem as follows:

$$\log\Lambda(\mathbf{x}) = -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)^{\boldsymbol{T}}\boldsymbol{C}^{-1}(\mathbf{x}-\boldsymbol{\mu}_1) + \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_2)^{\boldsymbol{T}}\boldsymbol{C}^{-1}(\mathbf{x}-\boldsymbol{\mu}_2)$$
$$= (\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)^{\boldsymbol{T}}\boldsymbol{C}^{-1}\mathbf{x} + \frac{1}{2}(\boldsymbol{\mu}_2^T\boldsymbol{C}^{-1}\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1^T\boldsymbol{C}^{-1}\boldsymbol{\mu}_1)$$
$$\log\xi = 0$$

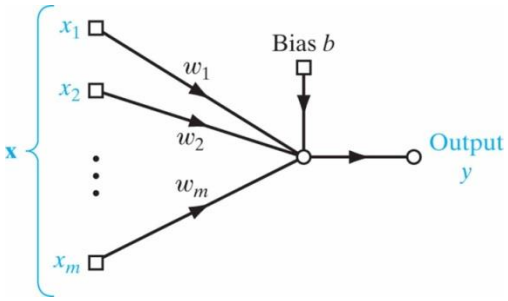
- These equations state that the Bayes classifier for the problem at hand is a *linear classifier*, as described by the relation

$$\mathrm{y} = \mathbf{w}^{\boldsymbol{T}}\mathbf{x} + b$$

where

$$\mathrm{y} = \log\Lambda(\mathbf{x})$$
$$\mathbf{w} = \boldsymbol{C}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$
$$b = \frac{1}{2}(\boldsymbol{\mu}_2^T\boldsymbol{C}^{-1}\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1^T\boldsymbol{C}^{-1}\boldsymbol{\mu}_1)$$

- More specifically, the classifier consists of a linear combiner with weight vector $\mathbf{w}$ and bias b



Signal-flow graph of Gaussian classifier.

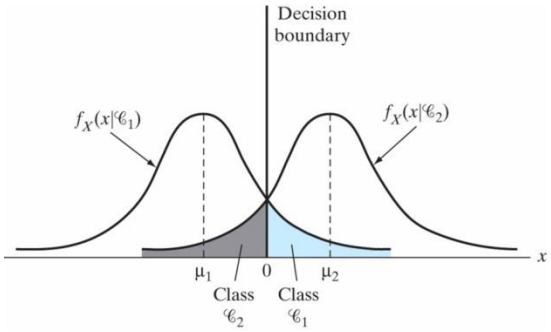
- We can now describe the log-likelihood ratio test for our two-class problem as follows:

If the output y of the linear combiner (including the bias b) is positive, assign the observation vector $\mathbf{x}$ to class C_1. Otherwise, assign it to class C_2.

- The operation of the Bayes classifier for the Gaussian environment described herein is analogous to that of the perceptron in that they are both linear classifiers.

Important Differences between the Perceptron and Bayes Classifier with Gaussian Distributions (Lippmann, 1987):

- The perceptron operates on the premise that the patterns to be classified are linearly separable. The Bayes classifier makes no such assumption; Under the Gaussian case, they certainly overlap (not separable). The extent of the overlap is determined by the mean vectors and the covariance matrix.



Two overlapping, one-dimensional Gaussian distributions.

- When the inputs are nonseparable and their distributions overlap, the perceptron convergence algorithm develops a problem because decision boundaries between the different classes may oscillate continuously.
- The Bayes classifier minimizes the probability of classification error. This minimization is independent of the overlap between the underlying Gaussian distributions of the two classes. For example, in the special case illustrated in the figure above, the Bayes classifier always positions the decision boundary at the point where the Gaussian distributions for the two classes cross each other.
- The perceptron convergence algorithm is nonparametric (unlike the Bayes classifier) in the sense that it makes no assumptions concerning the form of the underlying distributions.
- The perceptron convergence algorithm is both adaptive and simple to implement; its storage requirement is confined to the set of synaptic weights and bias. On the other hand, the design of the Bayes classifier is fixed; it can be made adaptive, but at the expense of increased storage requirements and more complex computations.

Bayes Pattern Recognition – EXAMPLE (Continuation)

- Bayes Rule Example: Joint frequency table

	TEST IS POSITIVE	TEST IS NEGATIVE	ROW TOTAL
HAS CONDITION	*True-positive* *P(POS|COND)* 100×0.90	*False-negative* *P(NEG|COND)* 100×(1-0.90)	100
FREE OF CONDITION	*False-positive* *P(POS|¬COND)* 9,900×(1-0.98)	*True-negative* *P(NEG|¬COND)* 9,900×0.98	9,900
COLUMN TOTAL	288	9,712	10,000

- Applying Bayes rule

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

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

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

$$= 0.3125$$