Perceptron

Machine Learning

Arun Chauhan

The perceptron Algorithm

- Generalized linear model of the form

$$y(\mathbf{x}) = f\left(\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x})\right)$$

where the nonlinear activation function $f(\cdot)$ is given by a step function of the form

$$f(a) = \begin{cases} +1, & a \geqslant 0 \\ -1, & a < 0. \end{cases}$$

Error function?

- To determine w minimize error function?
- Error function would be the total number of misclassified patterns.
- Is this a simple algorithm?
- Will gradient decent will work here?
  - Error function is piece wise constant with respect to w
  - The gradient is zero almost everywhere

Alternative error function: Perceptron Criterion

- We want to find w such that

$$
\begin{aligned}
& \mathbf{x}_n \text{ in class } \mathcal{C}_1 \text{ will have } \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n) > 0, \\
\text{whereas} \quad & \mathbf{x}_n \text{ in class } \mathcal{C}_2 \text{ have } \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n) < 0.
\end{aligned}
$$

- Using $t \in \{-1, +1\}$

$$\text{all patterns to satisfy } \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n)t_n > 0.$$

Perceptron Criterion

- The perceptron criterion is therefore given by

$$E_{\mathrm{P}}(\mathbf{w}) = -\sum_{n \in \mathcal{M}} \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}_n t_n$$

where $\mathcal{M}$ denotes the set of all misclassified patterns.

- Error: Linear function of **w** where pattern is misclassified
  and **Zero** where it is correctly classified

Stochastic Gradient Decent

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E_{\mathrm{P}}(\mathbf{w}) = \mathbf{w}^{(\tau)} + \eta \boldsymbol{\phi}_n t_n$$

- Scaling w will not change the error

  we can set the learning rate parameter η equal to 1

- Therefore

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} + \boldsymbol{\phi}_n t_n$$

Perceptron learning algorithm

Simple interpretation

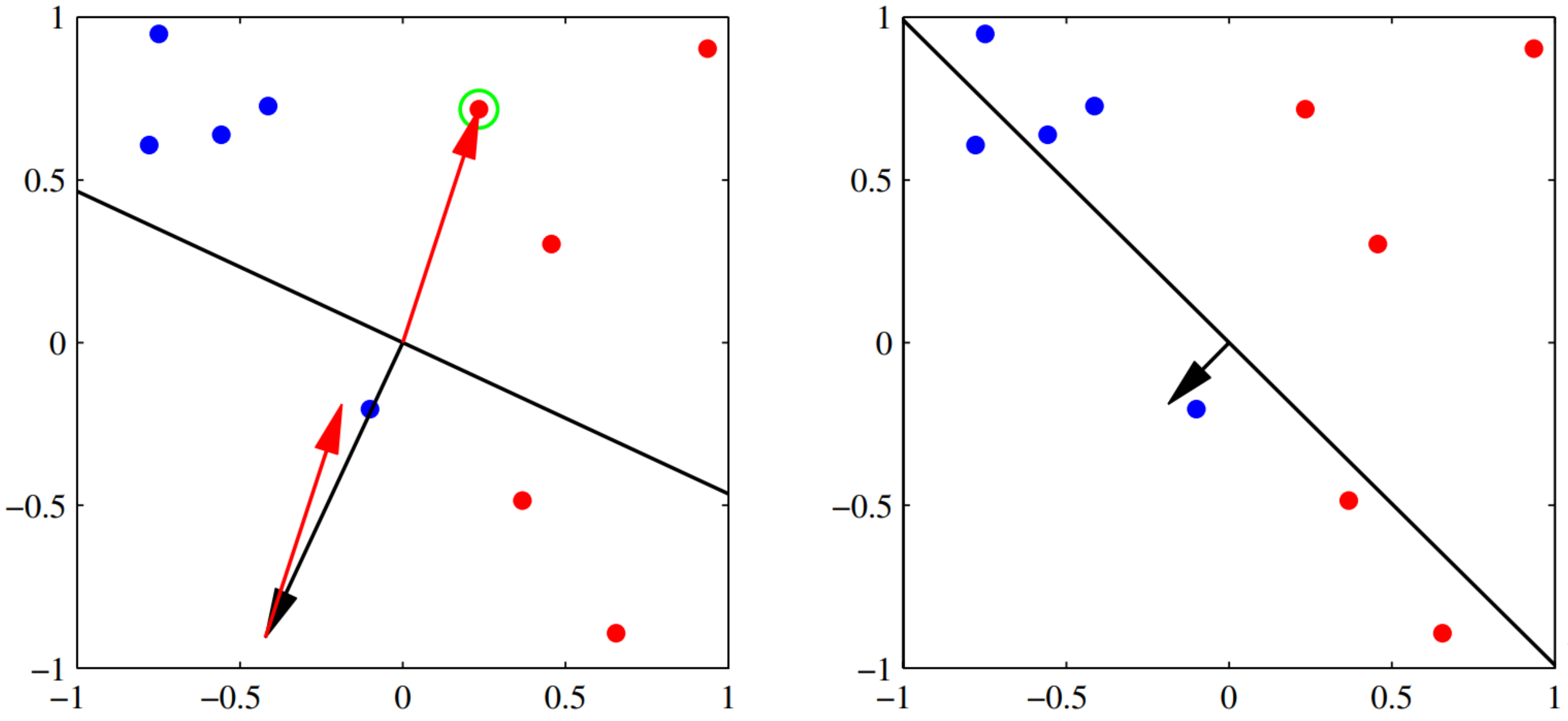
- If pattern is classified incorrectly using $y(\mathbf{x}) = f\left(\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x})\right)$

- If class $\mathcal{C}_1$????
- If class $\mathcal{C}_2$????

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} + \phi_n t_n$$

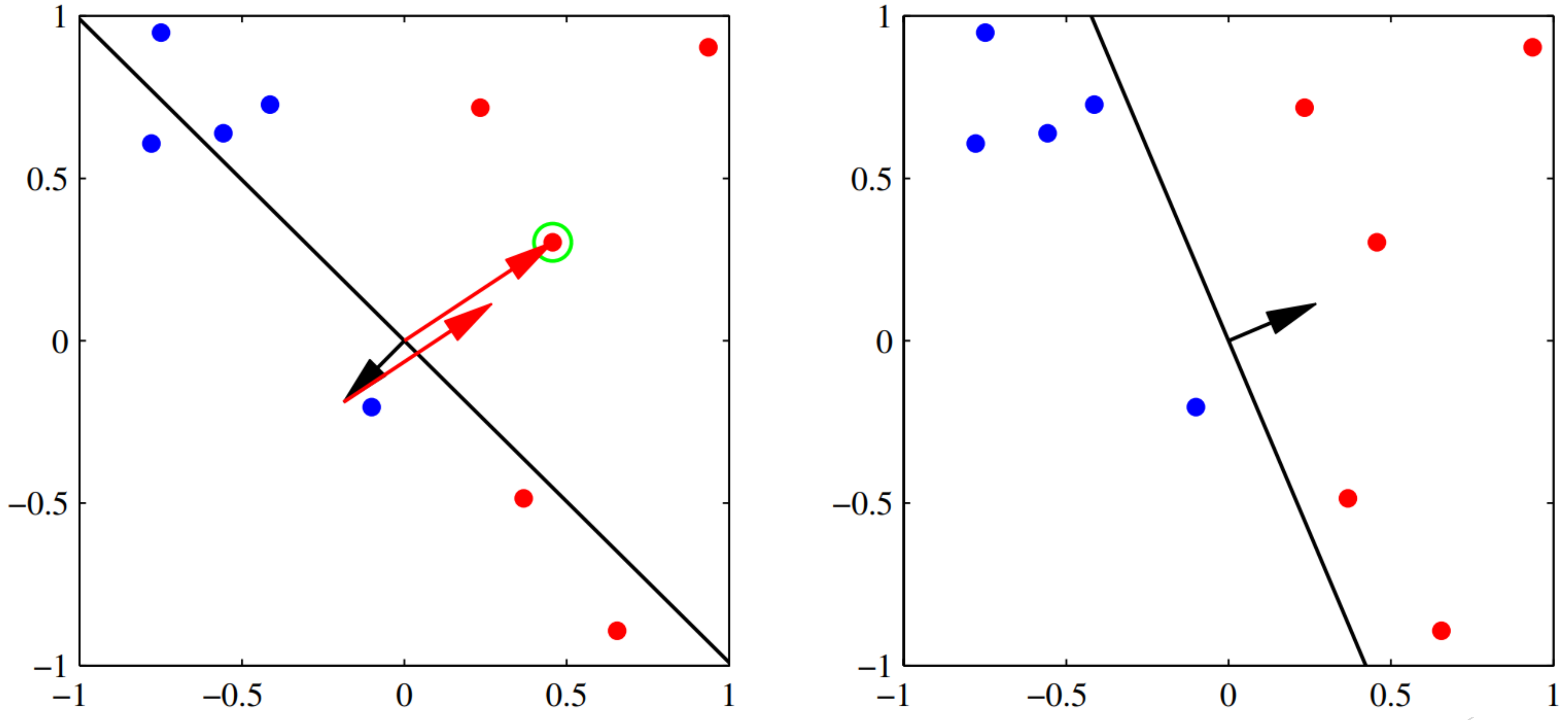
Perceptron learning algorithm

Simple interpretation



Perceptron learning algorithm

Simple interpretation



Perceptron Algo. iteration reduce error

$$-\mathbf{w}^{(\tau+1)\mathrm{T}}\phi_n t_n = -\mathbf{w}^{(\tau)\mathrm{T}}\phi_n t_n - (\phi_n t_n)^{\mathrm{T}}\phi_n t_n < -\mathbf{w}^{(\tau)\mathrm{T}}\phi_n t_n$$

where $\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} + \phi_n t_n$

and $\|\phi_n t_n\|^2 > 0$

Guaranteed to reduce the total error function at each stage ???

Perceptron learning algorithm

- If linearly separable; definitely converge (theorem)
- However # steps could be substantial for convergence
- For non linearly separable data never stop because never converge
- Cannot distinguish between non-sep and simply slow to converge.
- For linearly separable data:
  - Many Solutions
  - Solution depends upon the order of data present
  - And parameter initialization
- Limitation:
  - The perceptron does not provide probabilistic outputs
  - Nor does it generalize readily to $K > 2$ classes.

Probabilistic Generative Models

- Binary Classification

$$
\begin{aligned}
p(\mathcal{C}_1|\mathbf{x}) &= \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1) + p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)} \\
&= \frac{1}{1+\exp(-a)} = \sigma(a)
\end{aligned}
$$

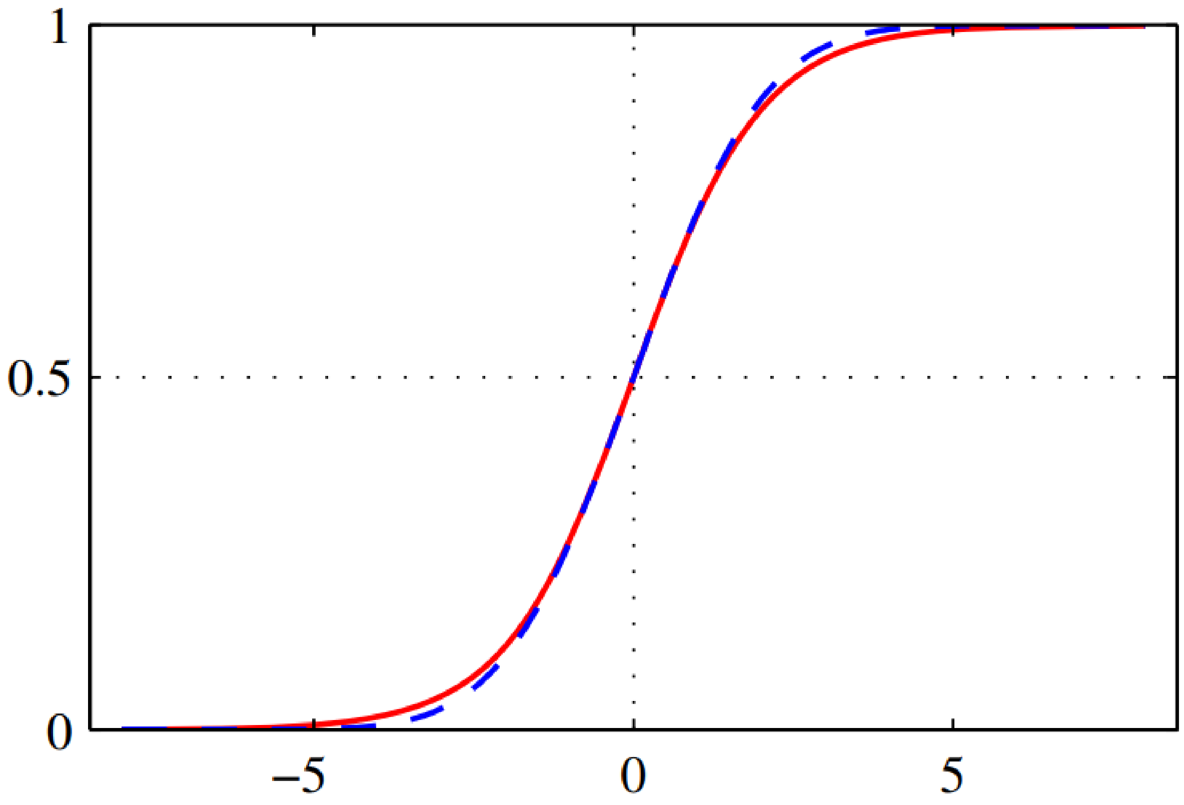
where we have defined

$$
a = \ln \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)}
$$

and $\sigma(a)$ is the *logistic sigmoid* function defined by

$$
\sigma(a) = \frac{1}{1+\exp(-a)}
$$

Logistic sigmoid function



Logistic sigmoid function

- Interesting properties

$$\sigma(-a) = 1 - \sigma(a)$$

- The inverse of the logistic sigmoid is given by

$$a = \ln\left(\frac{\sigma}{1-\sigma}\right)$$

and is known as the *logit* function.

Probabilistic Generative Models

For the case of $K > 2$ classes, we have

$$p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{\sum_j p(\mathbf{x}|\mathcal{C}_j)p(\mathcal{C}_j)} = \frac{\exp(a_k)}{\sum_j \exp(a_j)}$$

softmax function

$$a_k = \ln p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)$$

Continuous inputs : Class-conditional densities are Gaussian

Assume that all classes share the same covariance matrix

$$p(\mathbf{x}|\mathcal{C}_k) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right\}$$

- For two classes

$$a = \ln \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)}$$

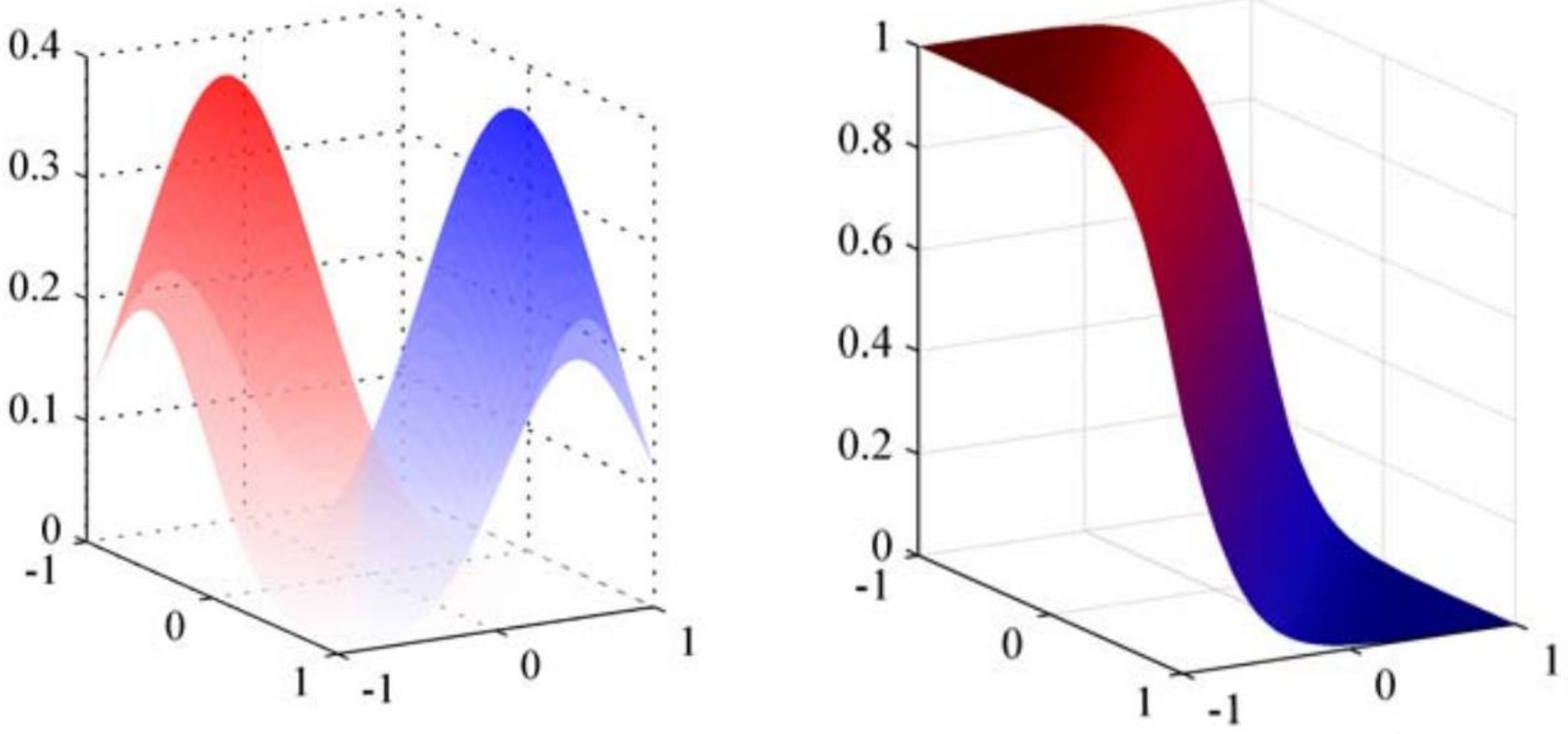
$$p(\mathcal{C}_1|\mathbf{x}) = \frac{1}{1+\exp(-a)} = \sigma(a)$$

$$p(\mathcal{C}_1|\mathbf{x}) = \sigma(\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0)$$

$$\mathbf{w} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$

$$w_0 = -\frac{1}{2}\boldsymbol{\mu}_1^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2 + \ln\frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)}$$

Continuous inputs :
Class-conditional densities are Gaussian



Probabilistic Generative Models

For the general case of K classes we have,

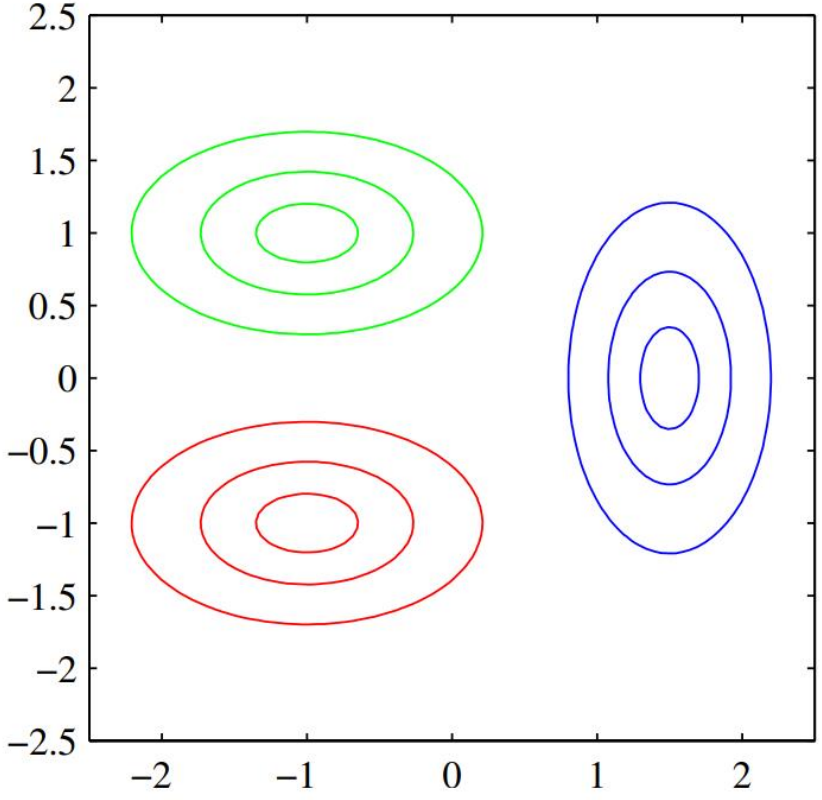
$$a_k(\mathbf{x}) = \mathbf{w}_k^{\mathrm{T}}\mathbf{x} + w_{k0}$$

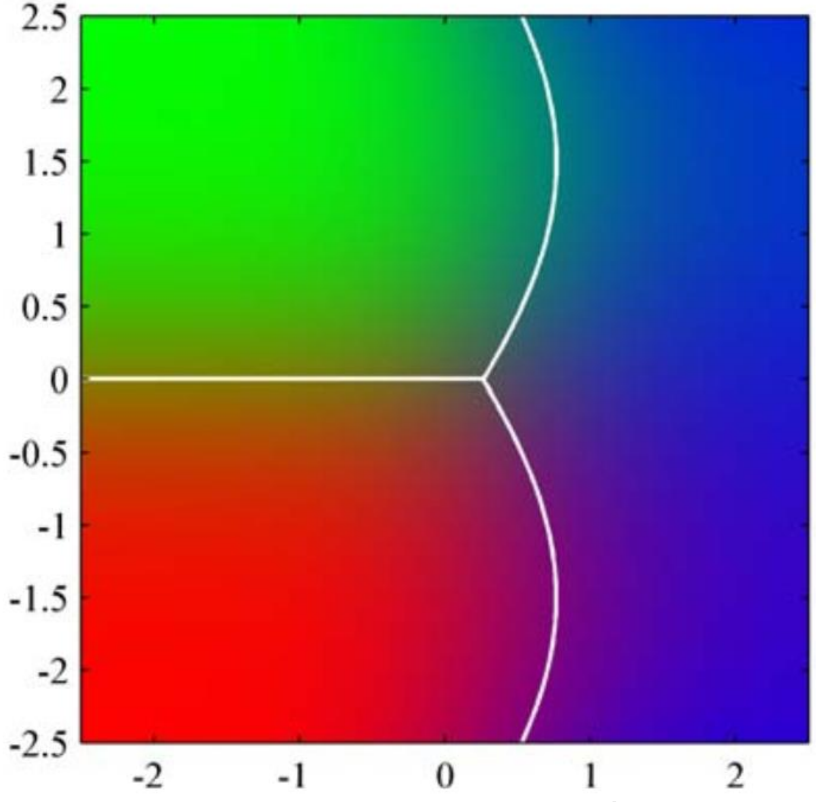
where we have defined

$$\begin{aligned} \mathbf{w}_k &= \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k \\ w_{k0} &= -\frac{1}{2}\boldsymbol{\mu}_k^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k + \ln p(\mathcal{C}_k) \end{aligned}$$

Continuous inputs : Class-conditional densities are Gaussian

Assume that all classes have different covariance matrix
Quadratic Discriminant





Maximum likelihood solution:

Class-conditional densities; Gaussian, Shared Covariance

- For K=2

$$p(\mathbf{x}_n, \mathcal{C}_1) = p(\mathcal{C}_1)p(\mathbf{x}_n|\mathcal{C}_1) = \pi\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_1, \boldsymbol{\Sigma})$$

$$p(\mathbf{x}_n, \mathcal{C}_2) = p(\mathcal{C}_2)p(\mathbf{x}_n|\mathcal{C}_2) = (1-\pi)\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_2, \boldsymbol{\Sigma})$$

Thus the likelihood function is given by

$$p(\mathbf{t}|\pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}) = \prod_{n=1}^{N} [\pi\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_1, \boldsymbol{\Sigma})]^{t_n} [(1-\pi)\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_2, \boldsymbol{\Sigma})]^{1-t_n}$$

Maximum likelihood solution:

Class-conditional densities; Gaussian, Shared Covariance

- Consider first the maximization with respect to π
- The terms in the log likelihood function that depend on π are

$$\sum_{n=1}^{N} \{t_n \ln \pi + (1 - t_n) \ln(1 - \pi)\}$$

Setting the derivative with respect to π equal to zero and rearranging, we obtain

$$\pi = \frac{1}{N} \sum_{n=1}^{N} t_n = \frac{N_1}{N} = \frac{N_1}{N_1 + N_2}$$

END