

---

# **PATTERN RECOGNITION AND MACHINE LEARNING**

## **CHAPTER 4: LINEAR MODELS FOR CLASSIFICATION**

---

# Learning Objectives

---

- 1、 What are linear classification models?
  - 2、 What are the three linear classification approaches?
  - 3、 What is the Fisher's discriminant method?
  - 4、 What is the Perceptron method?
  - 5、 What is the Gaussian mixture model method?
  - 6、 What is the logistic regression method?
  - 7、 How to compare the discriminative and generative methods?
  - 8、 What is the Bayesian Information Criterion?
-

# Outlines

---

- Three Approaches to Linear Classification
    - Approach I: Discriminant Functions
      - Least Square Classification
      - Fisher's Linear Discriminants
      - Perceptrons
    - Approach II: Probabilistic Generative Models
    - Approach III: Probabilistic Discriminative Models
      - Bayesian Information Criterion
-

# Linear Classification Models

---

## □ Classification is intrinsically non-linear

It puts non-identical things in the same class, so a difference in the input vector sometimes causes zero change in the answer

## □ Linear classification: linear adaptive part

- ✓ followed by a fixed non-linearity.
- ✓ preceded by a fixed non-linearity (e.g. nonlinear basis functions).

$$y(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0, \quad \textit{Decision} = f(y(\mathbf{x}))$$

  
adaptive linear function

  
fixed non-linear function

# Target Values for Classification

---

- ❑ **Two classes:** 1 and 0 (or sometimes -1)
    - ✓ Probabilistic class labels: the probability of the positive class as the target value.
  
  - ❑ **N classes:** N target values containing a single 1, and 0 for else
    - ✓ Probabilistic class labels: a vector of class probabilities as the target vector.
-

# Three Approaches to Classification

---

## □ Use discriminant functions directly (without probabilities):

- ✓ Convert the input vector into one or more real values so that a simple operation (like thresholding) can be applied to get the class.

$$y = f(w^T \mathbf{x})$$

## □ Infer the posterior probabilities with generative models.

- ✓ Use prior, and likelihood models to infer posterior models.

$$p(C_k|\mathbf{x}) = \frac{p(\mathbf{x}|C_k)p(C_k)}{p(\mathbf{x})}$$

## □ Directly construct posterior conditional class probabilities:

- ✓ Compute the posterior conditional probability of each class. Then make a decision that minimizes some loss function.

$$p(class = C_k|\mathbf{x})$$

---

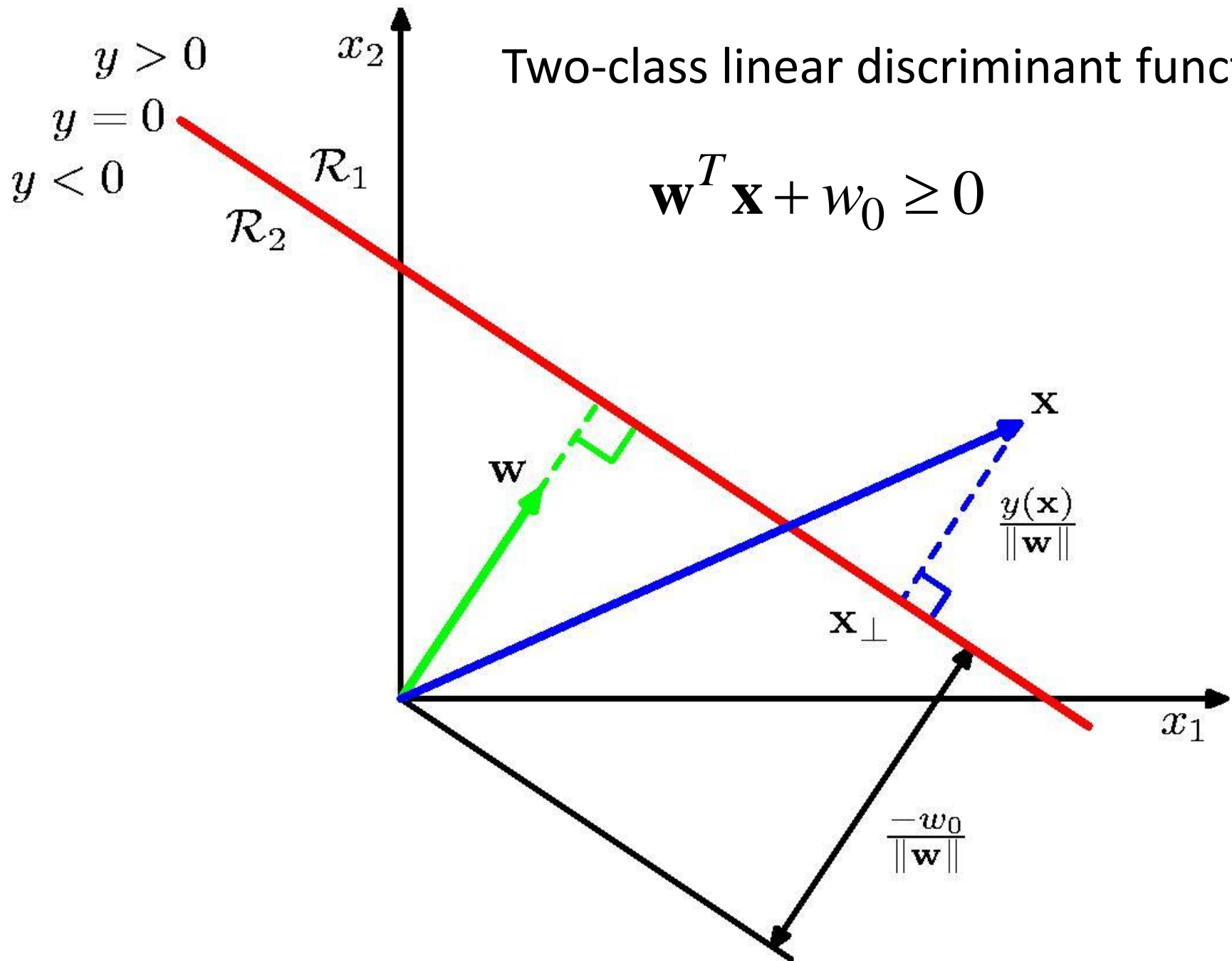
# Outlines

---

- Three Approaches to Linear Classification
  - Approach I: Discriminant Functions
  - Least Square Classification
  - Fisher's Discriminants
  - Perceptrons
  - Approach II: Probabilistic Generative Models
  - Approach III: Probabilistic Discriminative Models
  - Bayesian Information Criterion
-

Two-class linear discriminant function:

$$\mathbf{w}^T \mathbf{x} + w_0 \geq 0$$





# Discriminant Functions for N classes

---

- ❑ **To use N two-way discriminant functions**

Each function discriminates one class from the rest.

- ❑ **To use  $N(N-1)/2$  two-way discriminant functions**

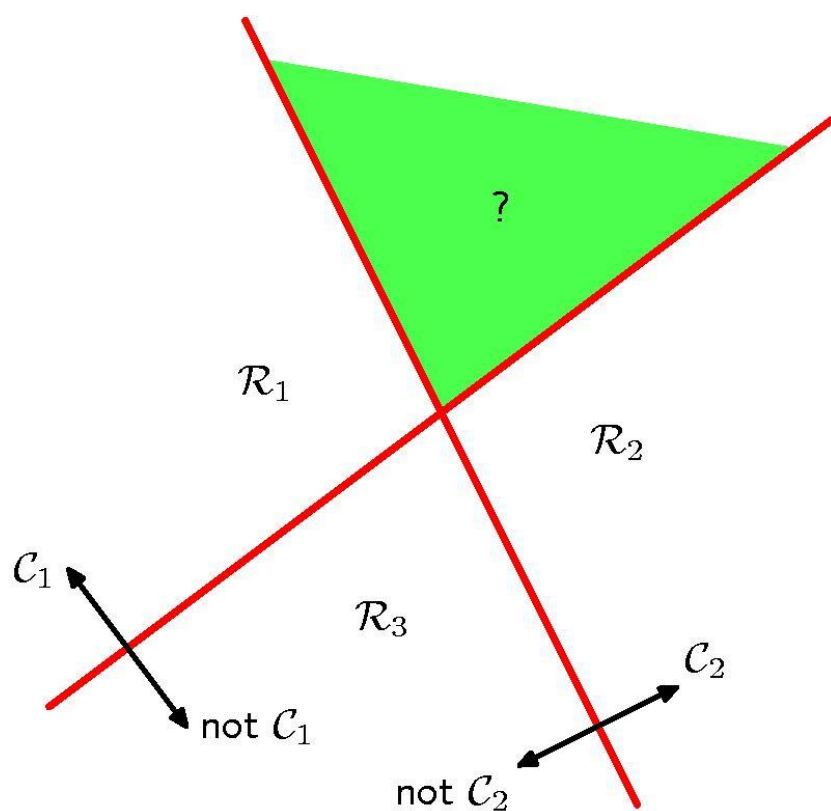
Each function discriminates between two particular classes.

- ❑ **Both methods have problems**

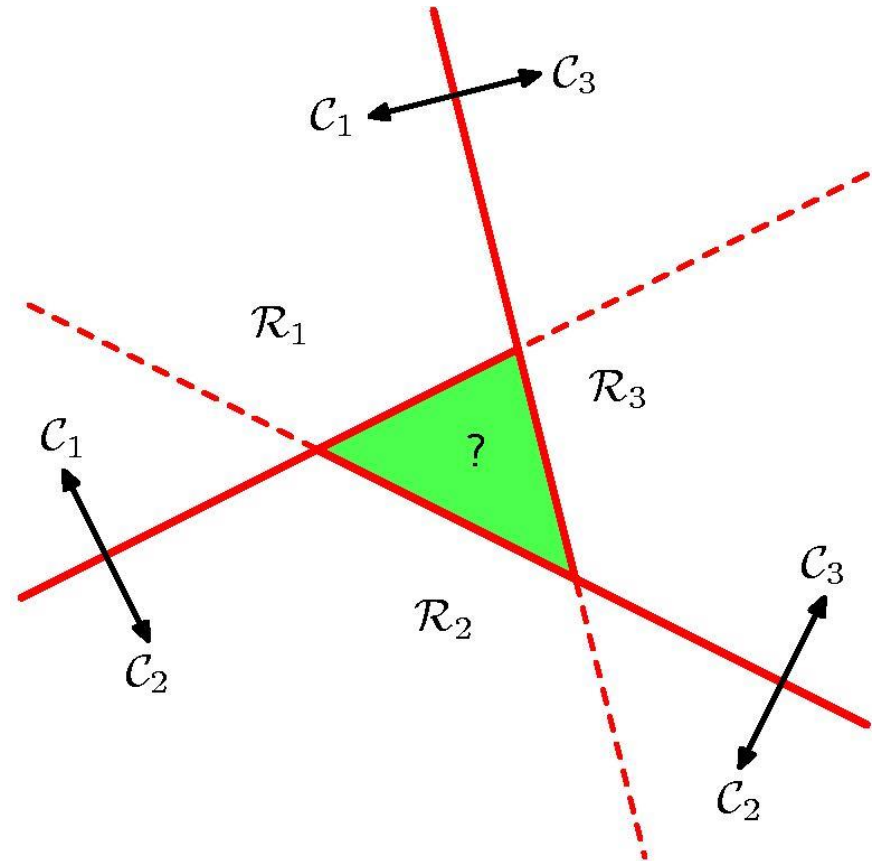
---

# Multi-class Using Two-class Discriminants

---



More than one good answer



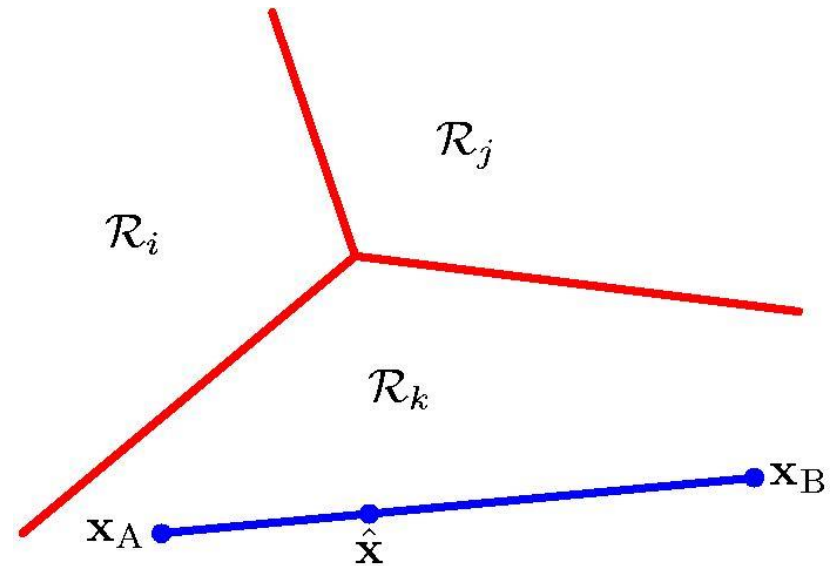
Two-way preferences need not be transitive!

# A Simple Solution

---

Use  $K$  discriminant functions,  
and pick the max.  $y_i, y_j, y_k \dots$

This is guaranteed to give  
consistent and convex decision  
regions if  $y$  is linear.



$$y_k(\mathbf{x}_A) > y_j(\mathbf{x}_A) \text{ and } y_k(\mathbf{x}_B) > y_j(\mathbf{x}_B)$$

*implies (for positive  $\alpha$ ) that*

$$y_k(\alpha \mathbf{x}_A + (1-\alpha) \mathbf{x}_B) > y_j(\alpha \mathbf{x}_A + (1-\alpha) \mathbf{x}_B)$$

---

# Outlines

---

- Three Approaches to Linear Classification
  - Approach I: Discriminant Functions
  - Least Square Classification
  - Fisher Discriminant Function
  - Perceptrons
  - Approach II: Probabilistic Generative Models
  - Approach III: Probabilistic Discriminative Models
  - Bayesian Information Criterion
-

# Least Squares for Classification

---

- ❑ This is not the right thing to do and it doesn't work as well as better methods, but it is easy:
    - ✓ It reduces classification to least squares regression.
    - ✓ We already know how to do regression. We can just solve for the optimal weights with some matrix algebra .
  
  - ❑ We use targets that are equal to the conditional probability of the class given the input.
    - ✓ When there are more than two classes, we treat each class as a separate problem (we cannot get away with this if we use the “max” decision function).
-

# Least Squares Regression

---

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

$$y(\mathbf{x}) = \widetilde{\mathbf{W}}^T \widetilde{\mathbf{x}}$$

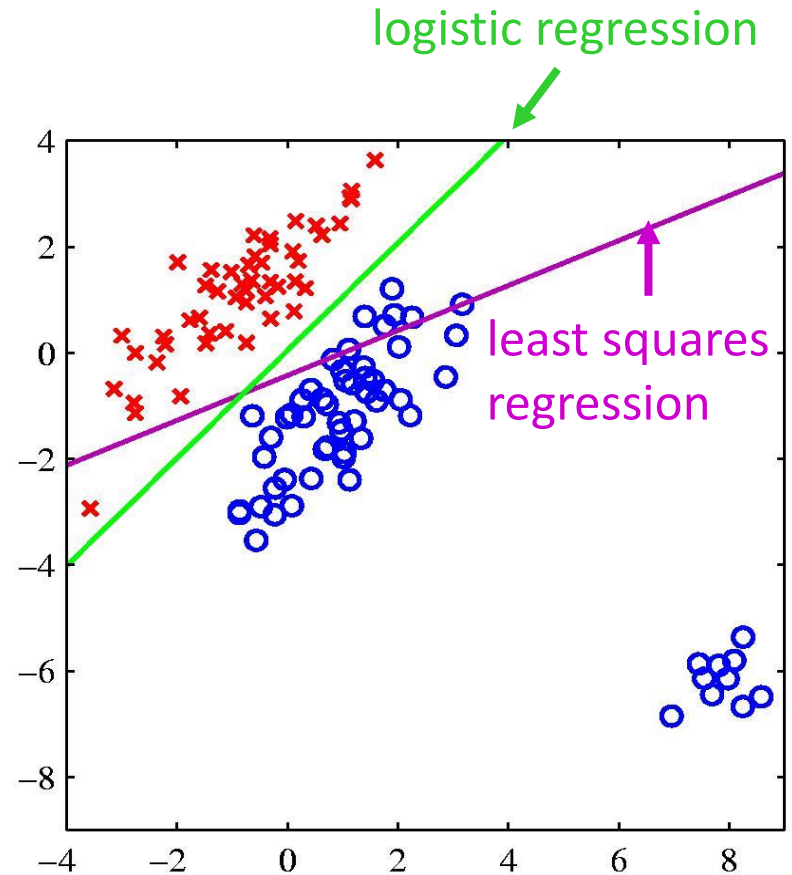
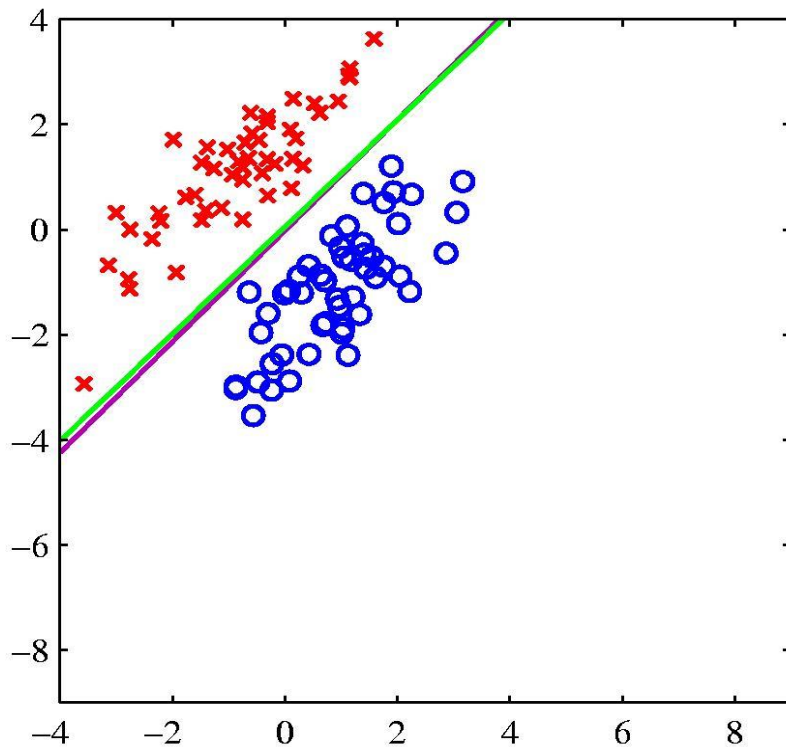
$$E_D(\widetilde{\mathbf{W}}) = \frac{1}{2} \text{Tr} \left\{ (\widetilde{\mathbf{X}} \widetilde{\mathbf{W}} - \mathbf{T})^T (\widetilde{\mathbf{X}} \widetilde{\mathbf{W}} - \mathbf{T}) \right\}$$

$$\widetilde{\mathbf{W}} = (\widetilde{\mathbf{X}}^T \widetilde{\mathbf{X}})^{-1} \widetilde{\mathbf{X}}^T \mathbf{T} = \widetilde{\mathbf{X}}^\dagger \mathbf{T}$$

---

# Problems with Least Square Classification

---

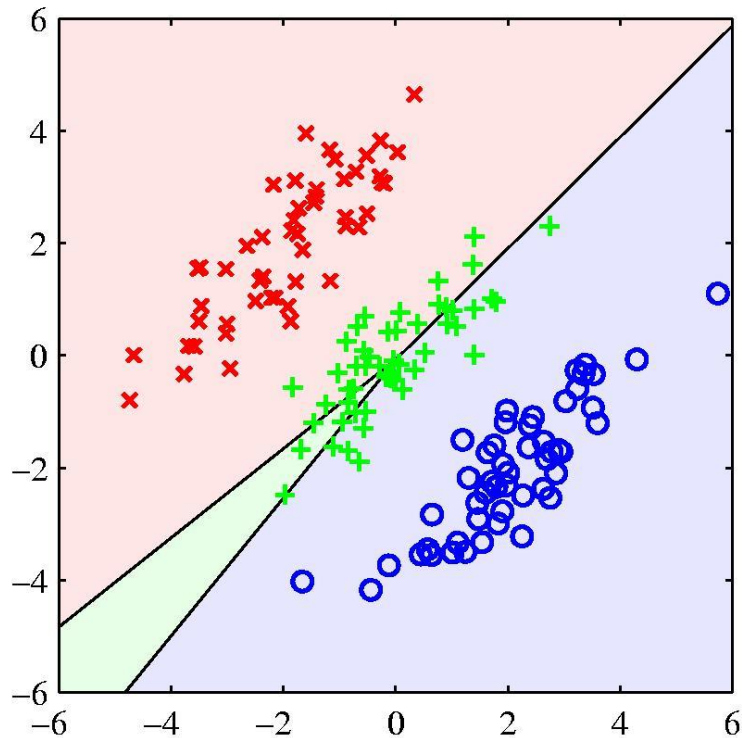


If the right answer is 1 and the model says 1.5, it loses, so it changes the boundary to avoid being “too correct”

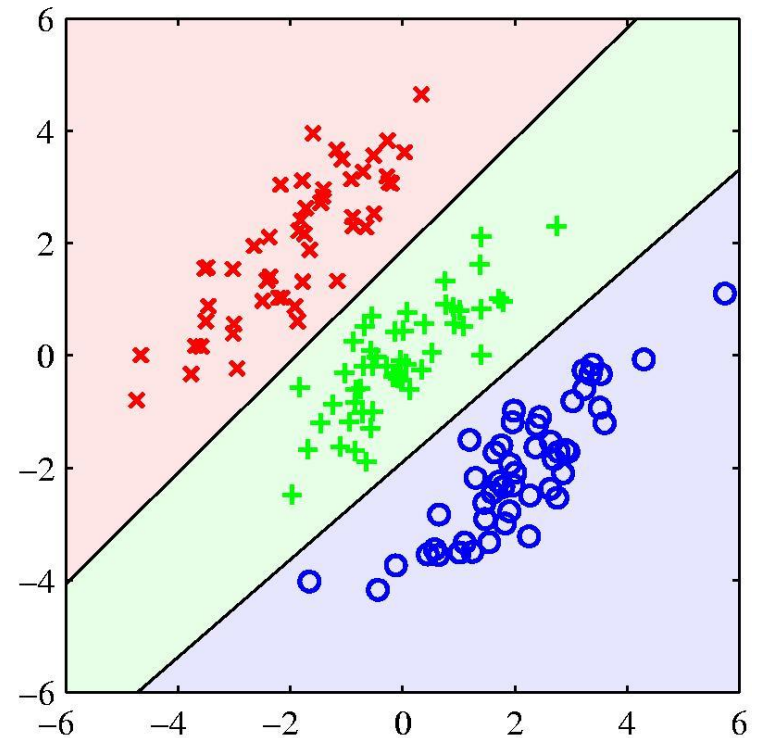
---

# Problems with Least Squares Classification

---



least squares regression



logistic regression



# Outlines

---

- Three Approaches to Linear Classification
    - Approach I: Discriminant Functions
      - Least Square Classification
      - Fisher's Linear Discriminants
      - Perceptrons
    - Approach II: Probabilistic Generative Models
    - Approach III: Probabilistic Discriminative Models
    - Bayesian Information Criterion
-

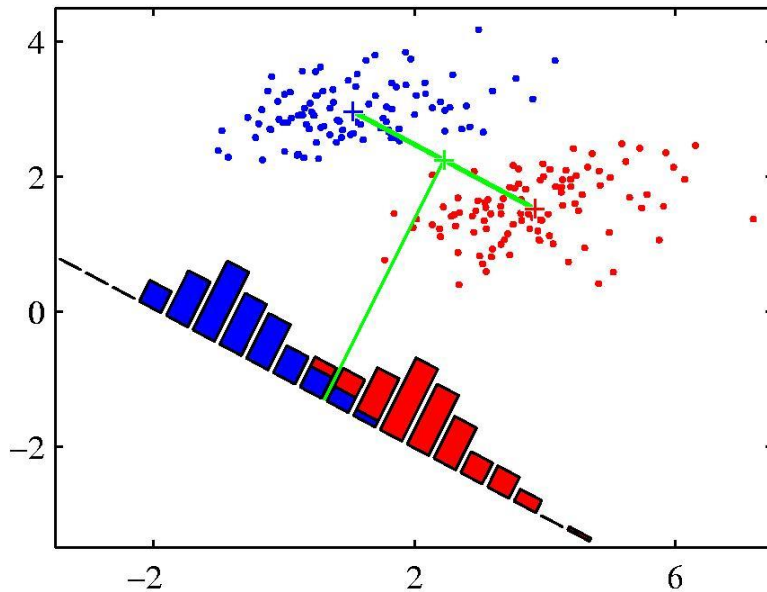
# Fisher's Linear Discriminant

---

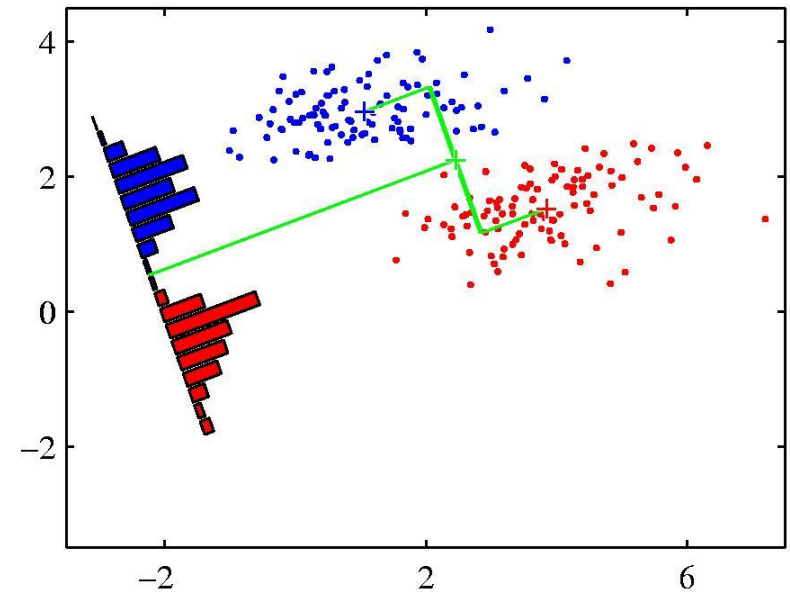
- ❑ **A simple linear discriminant function** is a projection of the data down to 1-D.
    - ✓ So choose the projection that gives the best separation of the classes. [What do we mean by “best separation”?](#)
  
  - ❑ **An obvious direction to choose** is the direction of the line joining the class means.
    - ✓ But if the main direction of variance in each class is not orthogonal to this line, this will not give good separation ([see the next figure](#)).
  
  - ❑ **Fisher's method chooses the direction** that maximizes the ratio of [between](#) class variance to [within](#) class variance.
    - ✓ This is the direction in which the projected points contain the most information about class membership (under Gaussian assumptions)
-

# Fisher's Linear Discriminant Function

---



When projected onto the line joining the class means, the classes are not well separated.



Fisher chooses a direction that makes the projected classes much tighter, even though their projected means are less far apart.

---

# Fisher's Linear Discriminants (I)

---

- What linear transformation is best for discrimination?

$$y = \mathbf{w}^T \mathbf{x}$$

- The projection onto the vector separating the class means seems sensible:

$$\mathbf{w} \propto \mathbf{m}_2 - \mathbf{m}_1$$

- But we also want small variance within each class:

$$s_1^2 = \sum_{n \in C_1} (y_n - m_1)^2$$

$$s_2^2 = \sum_{n \in C_2} (y_n - m_2)^2$$

- Fisher's objective function is:

$$J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2}$$

← between  
← within

# Fisher's Linear Discriminants (II)

---

$$J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} = \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}}$$

$$\mathbf{S}_B = (\mathbf{m}_2 - \mathbf{m}_1) (\mathbf{m}_2 - \mathbf{m}_1)^T$$

$$\mathbf{S}_W = \sum_{n \in C_1} (\mathbf{x}_n - \mathbf{m}_1) (\mathbf{x}_n - \mathbf{m}_1)^T + \sum_{n \in C_2} (\mathbf{x}_n - \mathbf{m}_2) (\mathbf{x}_n - \mathbf{m}_2)^T$$

*Optimal solution:*  $\mathbf{w} \propto \mathbf{S}_W^{-1} (\mathbf{m}_2 - \mathbf{m}_1)$

$$(\mathbf{w}^T \mathbf{S}_B \mathbf{w}) \mathbf{S}_W \mathbf{w} = (\mathbf{w}^T \mathbf{S}_W \mathbf{w}) \mathbf{S}_B \mathbf{w}$$

---

# Fisher's Linear Discriminants (III)

---

$$E = \frac{1}{2} \sum_{n=1}^N (\mathbf{w}^T \mathbf{x}_n + w_0 - t_n)^2 \quad \sum_{n=1}^N t_n = N_1 \frac{N}{N_1} - N_2 \frac{N}{N_2} = 0$$

Set its derivatives w.r.t.  $\mathbf{w}$  and  $\mathbf{x}_0$  to 0, then we will have

for two classes

$$\sum_{n=1}^N (\mathbf{w}^T \mathbf{x}_n + w_0 - t_n) = 0$$

$$\sum_{n=1}^N (\mathbf{w}^T \mathbf{x}_n + w_0 - t_n) \mathbf{x}_n = 0$$

$$\mathbf{m} = \frac{1}{N} \sum_{n=1}^N \mathbf{x}_n = \frac{1}{N} (N_1 \mathbf{m}_1 + N_2 \mathbf{m}_2) \quad \left( \mathbf{S}_W + \frac{N_1 N_2}{N} \mathbf{S}_B \right) \mathbf{w} = N (\mathbf{m}_1 - \mathbf{m}_2)$$

$$w_0 = -\mathbf{w}^T \mathbf{m}$$

$$\mathbf{w} \propto \mathbf{S}_W^{-1} (\mathbf{m}_2 - \mathbf{m}_1)$$

---

# K-Case Classification

---

- **Fisher's linear discriminants** can be extended to K-case classification.

$$J(\mathbf{w}) = \text{Tr} \left\{ (\mathbf{W} \mathbf{S}_W \mathbf{W}^T)^{-1} (\mathbf{W} \mathbf{S}_B \mathbf{W}^T) \right\}$$

---

# Outlines

---

- Three Approaches to Linear Classification
    - Approach I: Discriminant Functions
      - Least Square Classification
      - Fisher's Linear Discriminants
      - Perceptrons
    - Approach II: Probabilistic Generative Models
    - Approach III: Probabilistic Discriminative Models
  - Bayesian Information Criterion
-



# Perceptrons

---

□ “Perceptrons” describes a whole family of learning machines

- ✓ a layer of fixed non-linear basis functions followed by a simple linear discriminant function.
- ✓ introduced in the late 1950's
- ✓ a simple online learning procedure

$$y(\mathbf{x}) = f(\mathbf{w}^T \phi(\mathbf{x})) \quad f(a) = \begin{cases} +1, & a \geq 0 \\ -1, & a < 0 \end{cases}$$

---

# Perceptron Training

---

Perceptron criterion:

$$E_P(\mathbf{w}) = - \sum_{n \in \mathcal{M}} \mathbf{w}^T \phi_n t_n \quad t_n = \{1, -1\}:$$

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E_P(\mathbf{w}) = \mathbf{w}^{(\tau)} + \eta \phi_n t_n$$

Set the learning rate  $\eta$  as 1, then we will have

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

which indicates the convergence of perceptron training

---

# Simplified Perceptron Training

---

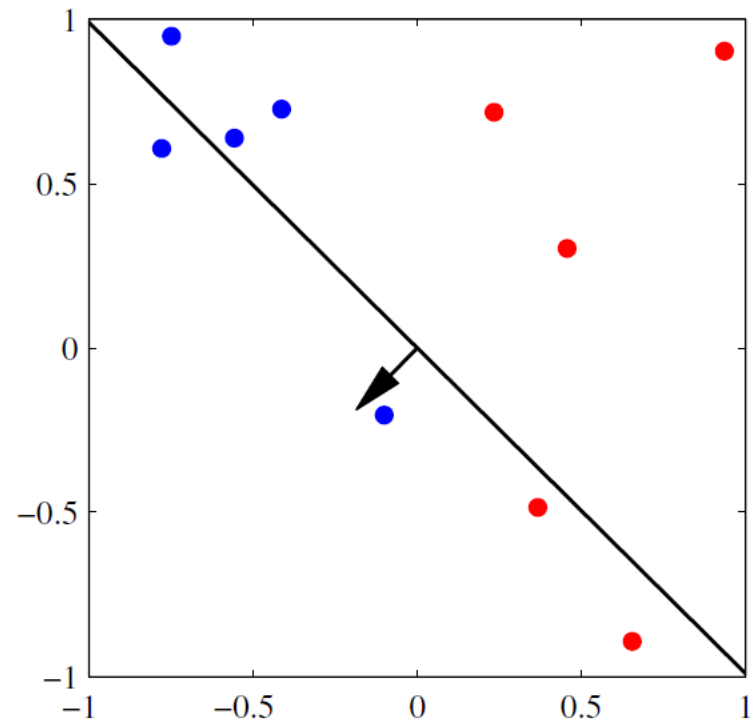
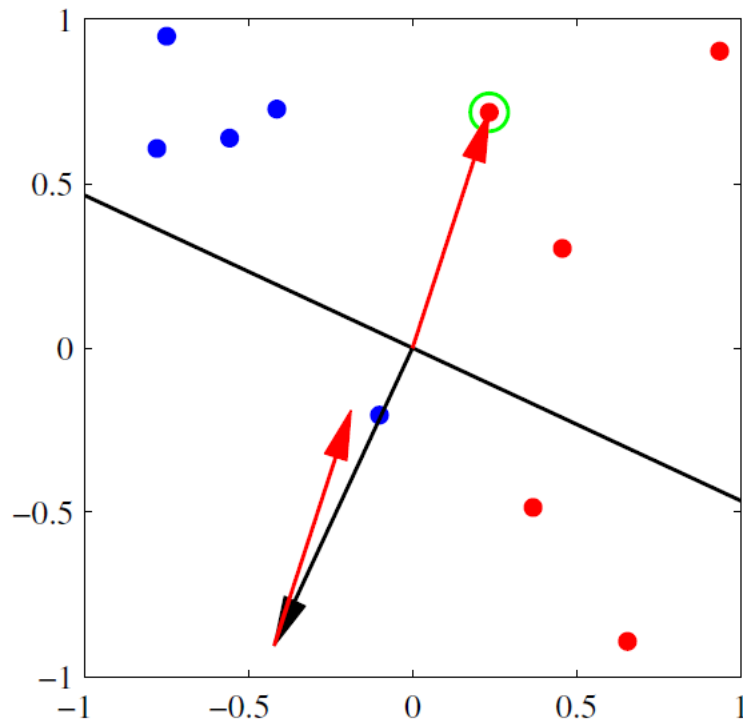
- ❑ Pick training cases using any policy that ensures that every training case will keep getting picked
  - ✓ If the output is correct, leave its weights alone.
  - ✓ If the output is -1 but should be 1, add the feature vector to the weight vector.
  - ✓ If the output is 1 but should be -1, subtract the feature vector from the weight vector

$$\mathbf{w}^{new} = \mathbf{w}^{old} - 0.5(y_n - t_n)\mathbf{x}_n$$

- ❑ This is guaranteed to find a set of weights that gets the right answer on the whole training set **if any such a set exists**. There is no need to choose a learning rate.
-

# Perceptron Training Procedure

---

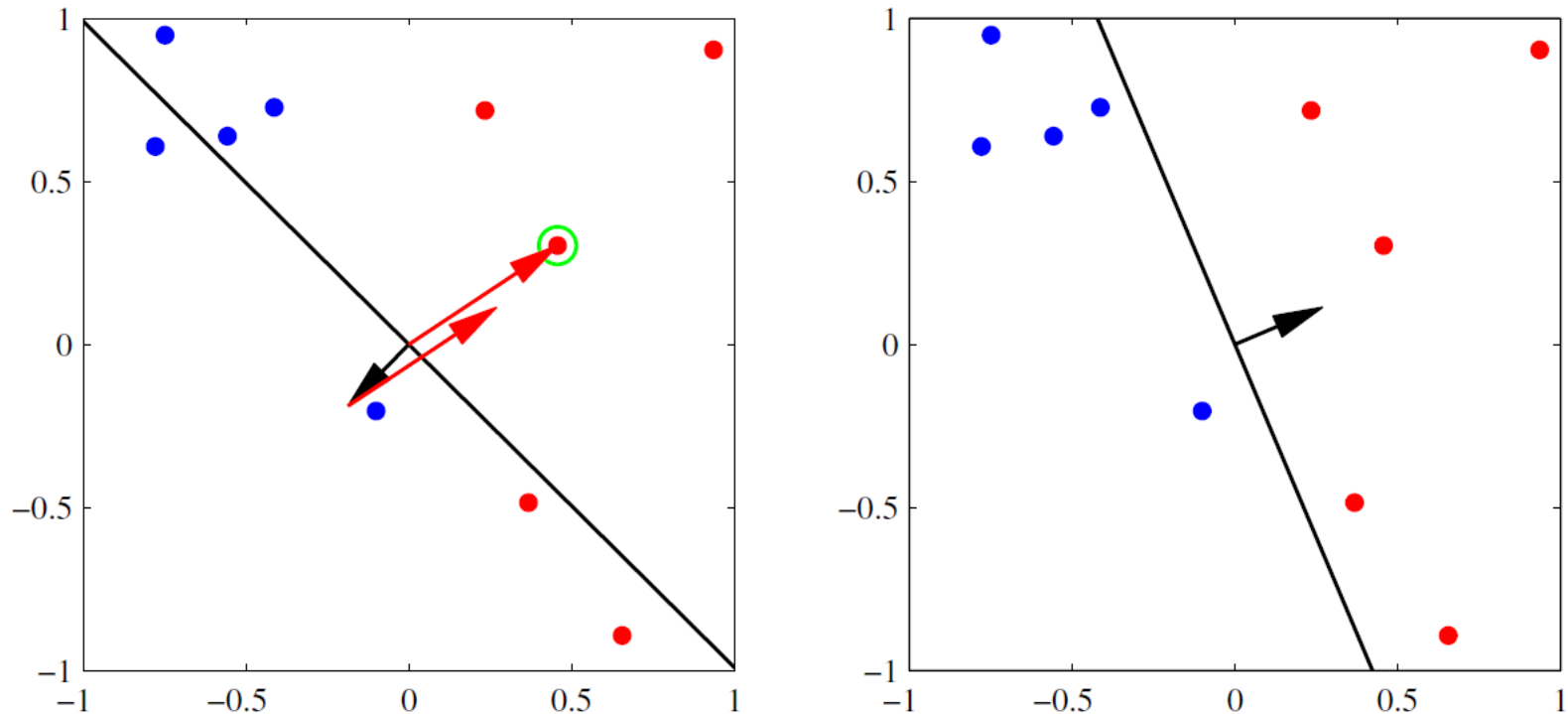


decision boundary: black line;  $w$ : black arrow; mismatching data: green circle  
 $\Delta w$ : red arrow;

---

# Perceptron Training Procedure

---



decision boundary: black line;  $w$ : black arrow; mismatching data: green circle  
 $\Delta w$ : red arrow;

---

# What Perceptrons Cannot Learn

---

The adaptive part of a perceptron cannot even tell if two single bit features have the same value!

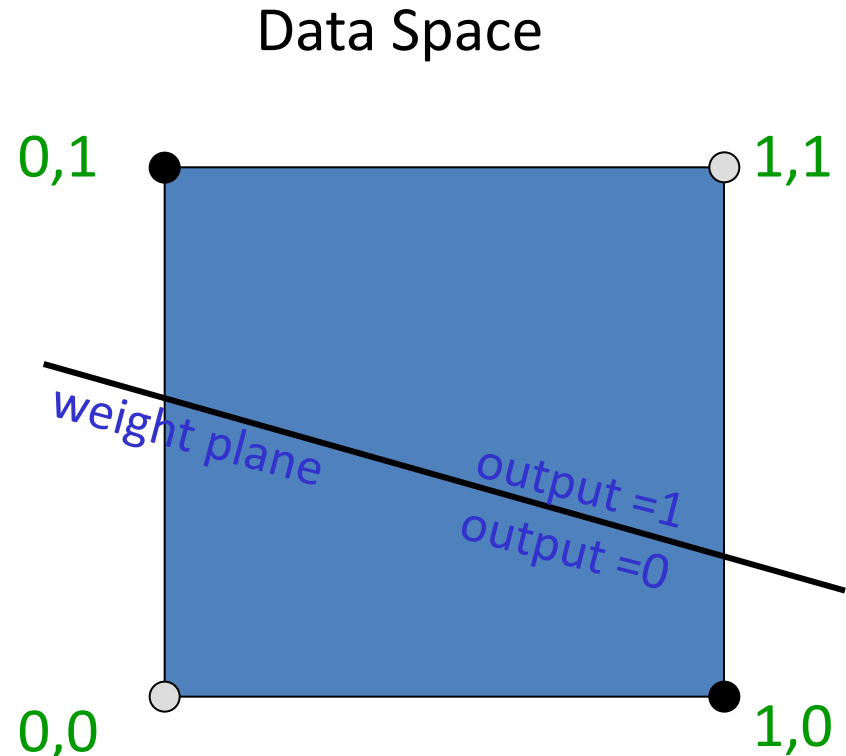
Same:  $(1,1) \rightarrow 1$ ;  $(0,0) \rightarrow 1$

Different:  $(1,0) \rightarrow 0$ ;  $(0,1) \rightarrow 0$

The four feature-output pairs give four inequalities that are impossible to satisfy:

$$w_1 + w_2 \geq \theta, \quad 0 \geq \theta$$

$$w_1 < \theta, \quad w_2 < \theta$$



The positive and negative cases cannot be separated by a plane

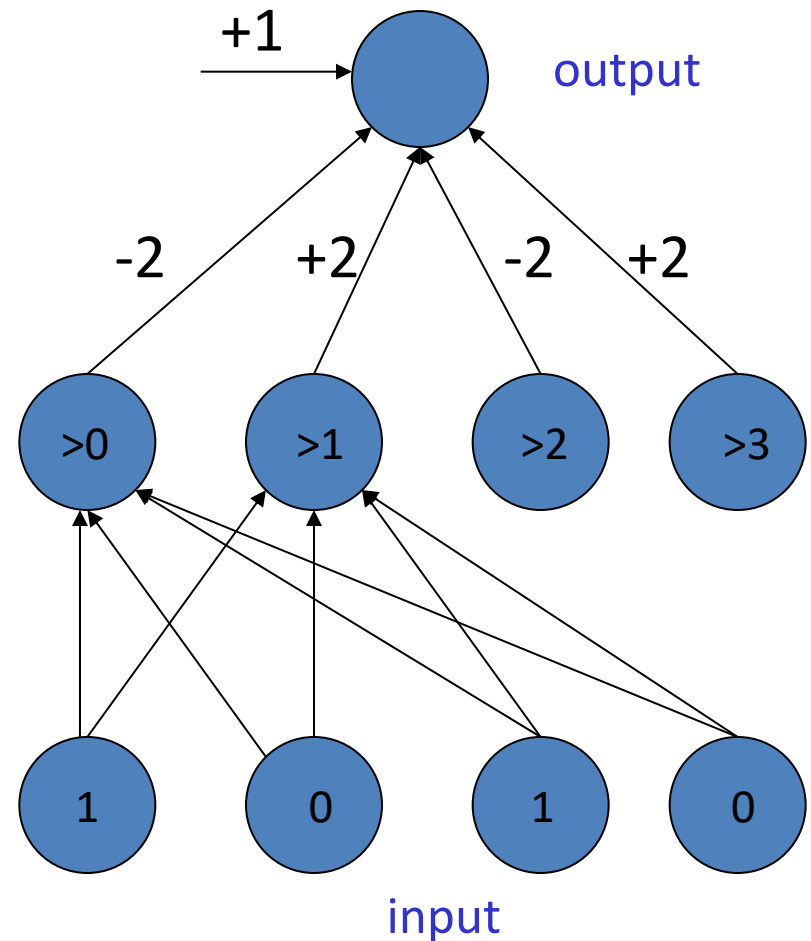
# The N-bit Even Parity Task

---

□ There is a simple solution that requires N hidden units.

- ✓ Each hidden unit computes whether more than M of the inputs are on.
- ✓ This is a linearly separable problem.
- There are many variants of this solution.
  - ✓ It can be learned.
  - ✓ It generalizes well if:

$$2^N \gg N^2$$



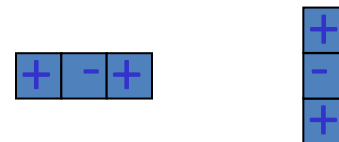
# Distinguishing Patterns

---

- ❑ What kind of features are required to distinguish two different patterns of 5 pixels independent of position and orientation?
  - ✓ Do we need to replicate T and C templates across all positions and orientations?
  - ✓ Looking at pairs of pixels will not work
  - ✓ Looking at triples will work if we assume that each input image only contains one object.



Replicate the following two feature detectors in all positions



If any of these equal their threshold of 2, it's a C. If not, it's a T.



# Outlines

---

- Three Approaches to Linear Classification Models
    - Approach I: Discriminant Functions
      - Least Square Classification
      - Fisher Discriminant Function
      - Perceptrons
    - Approach II: Probabilistic Generative Models
    - Approach III: Probabilistic Discriminative Models
      - Bayesian Information Criterion
-

# Probabilistic Generative Models

---

- Use a separate generative model of the input vectors for each class, and see which model makes a test input vector most probable.
- The posterior probability of class 1 is given by:

$$p(C_1 | \mathbf{x}) = \frac{p(C_1)p(\mathbf{x} | C_1)}{p(C_1)p(\mathbf{x} | C_1) + p(C_0)p(\mathbf{x} | C_0)} = \frac{1}{1 + e^{-z}} = \sigma(z)$$

$$\text{where } z = \ln \frac{p(C_1)p(\mathbf{x} | C_1)}{p(C_0)p(\mathbf{x} | C_0)} = \boxed{\ln \frac{p(C_1 | \mathbf{x})}{1 - p(C_1 | \mathbf{x})}}$$



*z* is called the logit and is given by the log odds

---

# A Simple Example

---

- Assume that the input vectors for each class are from a Gaussian distribution, and all classes have the same covariance matrix.

$$p(\mathbf{x} | C_k) = \overset{\substack{\text{normalizing} \\ \text{constant}}}{a} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_k)^T \overset{\substack{\text{inverse} \\ \text{covariance matrix}}}{\Sigma^{-1}} (\mathbf{x} - \boldsymbol{\mu}_k) \right\}$$

- For two classes, C1 and C0, the posterior is a logistic:

$$p(C_1 | \mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x} + w_0)$$

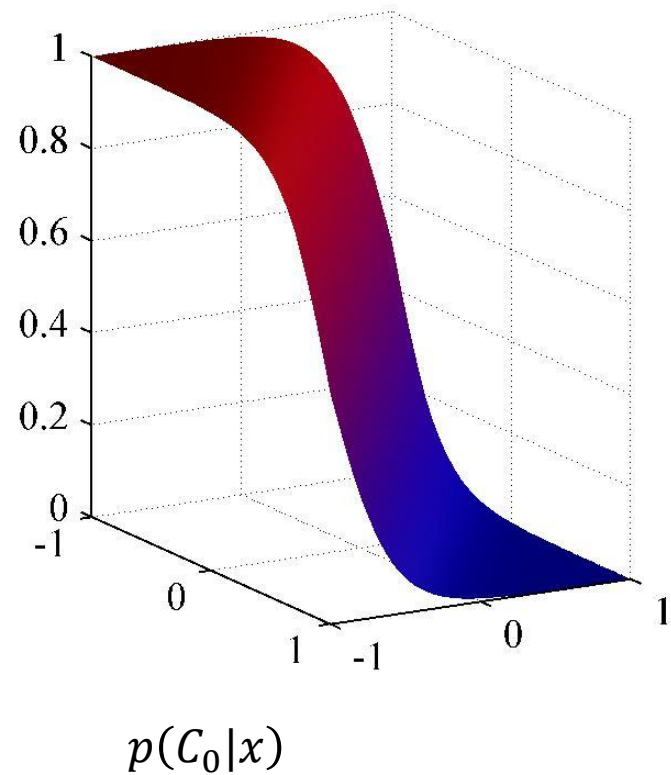
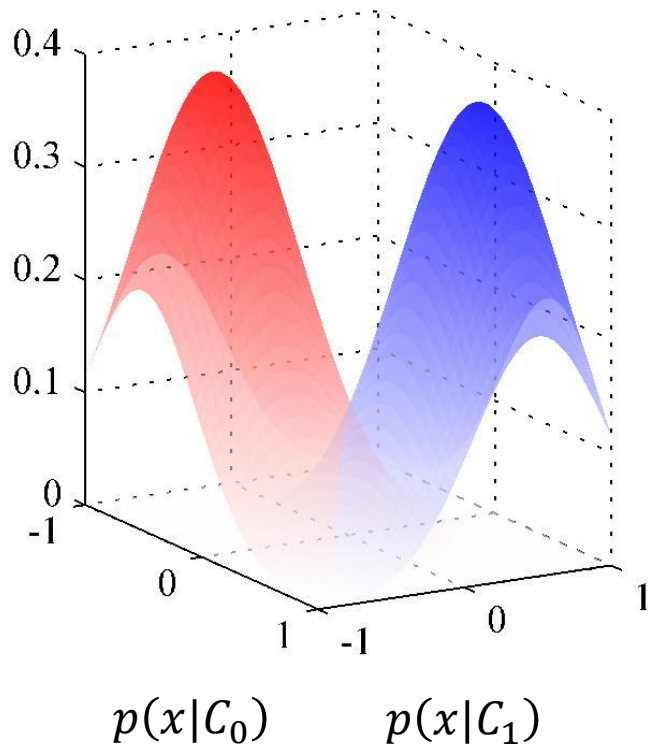
$$\mathbf{w} = \Sigma^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)$$

$$w_0 = -\frac{1}{2} \boldsymbol{\mu}_1^T \Sigma^{-1} \boldsymbol{\mu}_1 + \frac{1}{2} \boldsymbol{\mu}_0^T \Sigma^{-1} \boldsymbol{\mu}_0 + \ln \frac{p(C_1)}{p(C_0)}$$

---

# Likelihood and Posterior

---



# K-Case Classification

---

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

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

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

$$\mathbf{w}_k = \Sigma^{-1} \boldsymbol{\mu}_k$$

$$w_{k0} = -\frac{1}{2} \boldsymbol{\mu}_k^T \Sigma^{-1} \boldsymbol{\mu}_k + \ln p(\mathcal{C}_k)$$

---

# Inverse Covariance Matrix

---

- ❑ If the Gaussian is spherical we don't need to worry about the covariance matrix.
- ❑ So we could start by transforming the data space to make the Gaussian spherical
  - ✓ This is called “whitening” the data.
  - ✓ It pre-multiplies by the matrix square root of the inverse covariance matrix.
- ❑ In the transformed space, the weight vector is just the difference between the transformed means.

$$\mathbf{w} = \Sigma^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)$$

*gives the same value  
for  $\mathbf{w}^T \mathbf{x}$  as :*

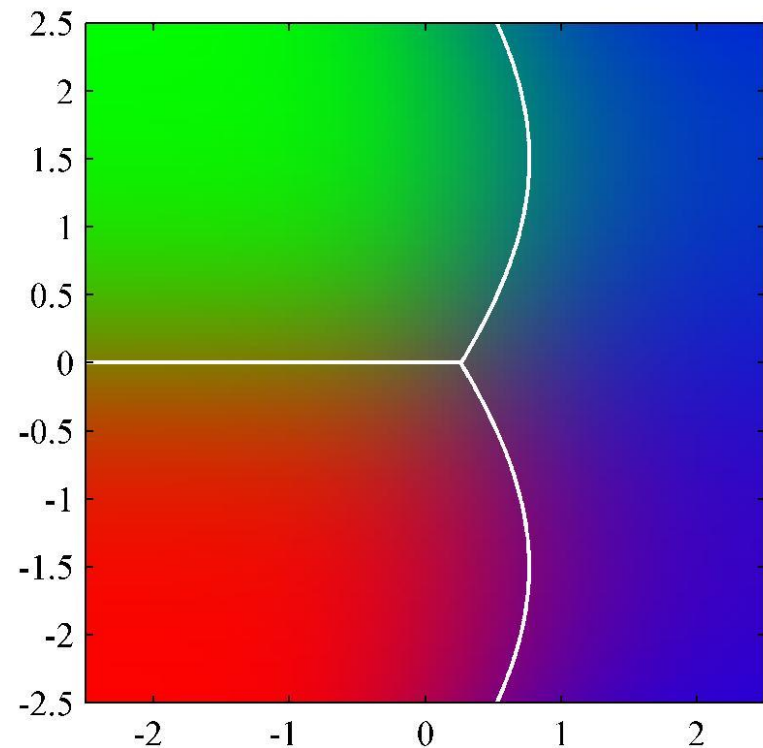
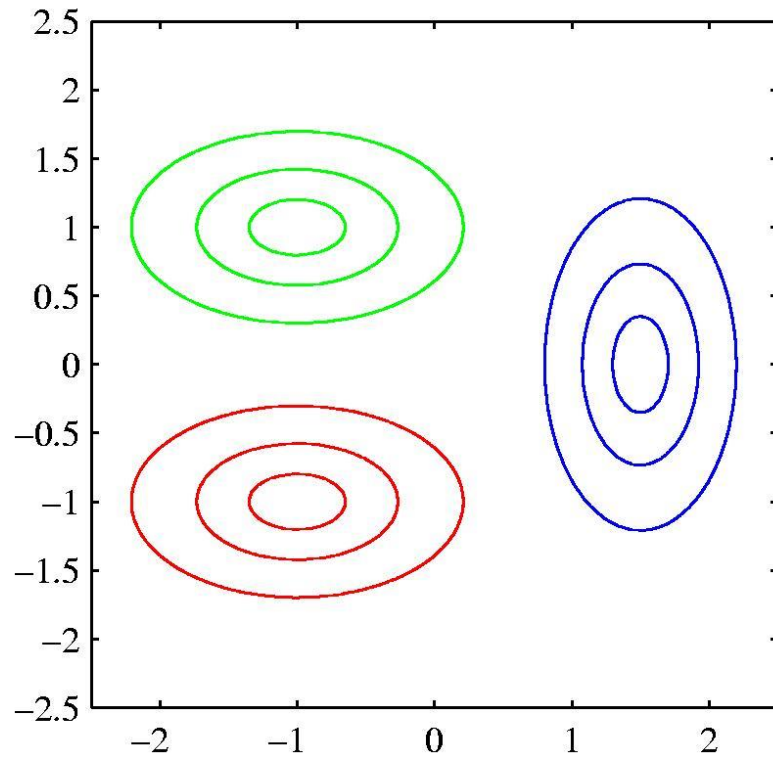
$$\mathbf{w}_{aff} = \Sigma^{-\frac{1}{2}} \boldsymbol{\mu}_1 - \Sigma^{-\frac{1}{2}} \boldsymbol{\mu}_0$$

$$\text{and } \mathbf{x}_{aff} = \Sigma^{-\frac{1}{2}} \mathbf{x}$$

*gives for  $\mathbf{w}_{aff}^T \mathbf{x}_{aff}$*

# Different Covariance Matrices

---



The decision surface is planar when the covariance matrices are the same; the decision surface is quadratic when they are not.

---

# Generative: ML Gaussian Mixtures

---

$$p(x, C_1) = p(C_1)p(x|C_1) = \pi N(x|\mu_1, \Sigma)$$

$$p(x, C_2) = p(C_2)p(x|C_2) = (1 - \pi)N(x|\mu_2, \Sigma)$$

Likelihood

$$p(\mathbf{t}, \mathbf{X}|\pi, \mu_1, \mu_2, \Sigma) = \prod_{n=1}^N [\pi N(x_n|\mu_1, \Sigma)]^{t_n} [(1 - \pi)N(x_n|\mu_2, \Sigma)]^{1-t_n}$$

$$\Rightarrow \pi_{ML} = \frac{1}{N} \sum_{n=1}^N t_n = \frac{N_1}{N} = \frac{N_1}{N_1 + N_2} \quad \mu_{1ML} = \frac{1}{N_1} \sum_{n=1}^N t_n x_n \quad \mu_{2ML} = \frac{1}{N_2} \sum_{n=1}^N (1 - t_n) x_n$$

$$\Sigma = \pi \Sigma_1 + (1 - \pi) \Sigma_2 \quad \Sigma_{iML} = \frac{1}{N_i} \sum_{x_n \in C_i} (x_n - \mu_i)(x_n - \mu_i)^T \quad i=1,2$$

---



# Generative: MAP Gaussian Mixtures

---

$$\pi_0 = \frac{N_{10}}{N_{10} + N_{20}} \quad x \in \mathcal{C}_i \sim \mathcal{N}(x | \mu_{i0}, \Sigma_{i0})$$

$$\pi_{MAP} = \frac{N_1 + N_{10}}{N + N_0} = \frac{N_1 + N_{10}}{N_1 + N_2 + N_{10} + N_{20}}$$

$$\begin{cases} \Sigma_{iMAP}^{-1} &= \Sigma_{iML}^{-1} + \Sigma_{i0}^{-1} \\ \Sigma_{iMAP}^{-1} \mu_{iMAP} &= \Sigma_{iML}^{-1} \mu_{iML} + \Sigma_{i0}^{-1} \mu_{i0} \end{cases}$$

$$\Sigma = \pi \Sigma_1 + (1 - \pi) \Sigma_2$$

---

# Outlines

---

- Three Approaches to Linear Classification Models
    - Approach I: Discriminant Functions
      - Least Square Classification
      - Fisher Discriminant Function
      - Perceptrons
    - Approach II: Probabilistic Generative Models
    - Approach III: Probabilistic Discriminative Models
      - Bayesian Information Criterion
-

# Probabilistic Discriminative Models

---

- ❑ *Discriminative training*: we can maximize the likelihood function defined through the conditional distribution  $p(\mathcal{C}_k|\mathbf{x})$
  - ❑ *Advantages of discriminative approaches*: fewer parameters to be determined
-

# Logistic Regression

---

- When there are only two classes we can model the conditional probability of the positive class as

$$p(C_1 | \mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x} + w_0) \quad \text{where} \quad \sigma(z) = \frac{1}{1 + \exp(-z)}$$

- If we use the right error function, something nice happens: The gradient of the logistic and the gradient of the error function cancel each other:

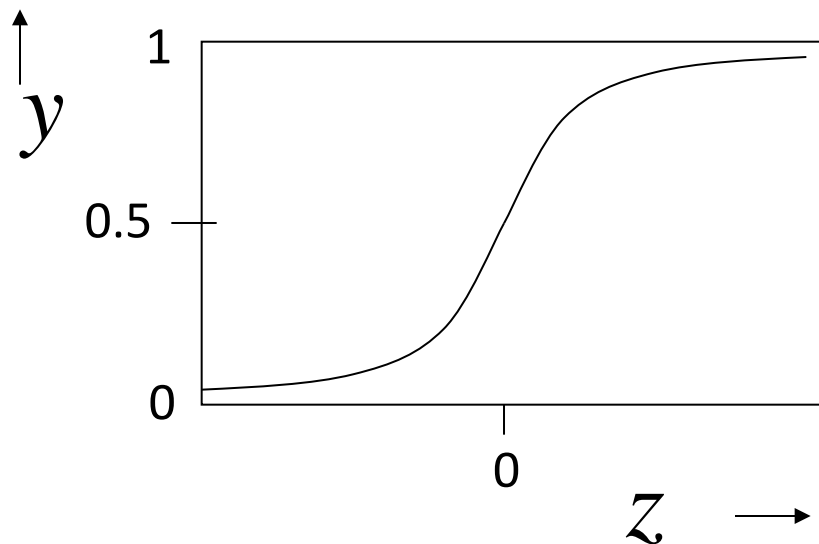
$$E(\mathbf{w}) = -\ln p(\mathbf{t} | \mathbf{w}), \quad \nabla E(\mathbf{w}) = \sum_{n=1}^N (y_n - t_n) \mathbf{x}_n$$

---

# The Logistic Function

---

- The output is a smooth function of the inputs and the weights.



$$z = \mathbf{w}^T \mathbf{x} + w_0$$

$$y = \sigma(z) = \frac{1}{1 + e^{-z}}$$

$$\frac{\partial z}{\partial w_i} = x_i \quad \frac{\partial z}{\partial x_i} = w_i$$

$$\frac{dy}{dz} = y(1 - y)$$




It is odd to express it in terms of  $y$ .

# The Natural Error Function

---


- To fit a logistic model using maximum likelihood, we need to minimize the negative log probability of the correct answer summed over the training set.

$$\begin{aligned} E &= - \sum_{n=1}^N \ln p(t_n | y_n) \quad \leftarrow \boxed{\text{cross-entropy}} \\ &= - \sum_{n=1}^N t_n \ln y_n + (1 - t_n) \ln (1 - y_n) \end{aligned}$$



if  $t = 1$                       if  $t = 0$

error derivative on  
training case  $n$



$$\begin{aligned} \frac{\partial E_n}{\partial y_n} &= - \frac{t_n}{y_n} + \frac{1 - t_n}{1 - y_n} \\ &= \frac{y_n - t_n}{y_n (1 - y_n)} \end{aligned}$$

# The Chain Rule for Error Derivatives

---

$$z_n = \mathbf{w}^T \mathbf{x}_n + w_0, \quad \frac{\partial z_n}{\partial \mathbf{w}} = \mathbf{x}_n$$

$$\frac{\partial E_n}{\partial y_n} = \frac{y_n - t_n}{y_n(1 - y_n)}, \quad \frac{dy_n}{dz_n} = y_n(1 - y_n)$$

$$\frac{\partial E_n}{\partial \mathbf{w}} = \frac{\partial E_n}{\partial y_n} \frac{dy_n}{dz_n} \frac{\partial z_n}{\partial \mathbf{w}} = (y_n - t_n) \mathbf{x}_n$$

$$\mathbf{w}^{new} = \mathbf{w}^{old} - (y_n - t_n) \mathbf{x}_n \quad \leftarrow \boxed{\text{If the step size is taken as 1}}$$

---

# Softmax for Two Classes

---

$$y_1 = \frac{e^{z_1}}{e^{z_1} + e^{z_0}} = \frac{1}{1 + e^{-(z_1 - z_0)}}$$

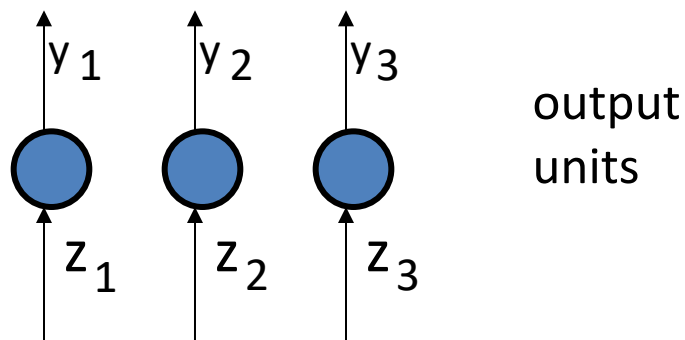
- ❑ So the logistic is just a special case that avoids using redundant parameters:
    - ✓ Adding the same constant to both  $z_1$  and  $z_0$  has no effect.
    - ✓ The over-parameterization of the softmax is because the probabilities must add to 1.
-



# Softmax for Multiple Classes

---

The output units use a non-local non-linearity:



The natural cost function is the negative log prob of the right answer

The steepness of  $E$  exactly balances the flatness of the softmax.

$$y_i = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

$$\frac{\partial y_i}{\partial z_i} = y_i (1 - y_i)$$

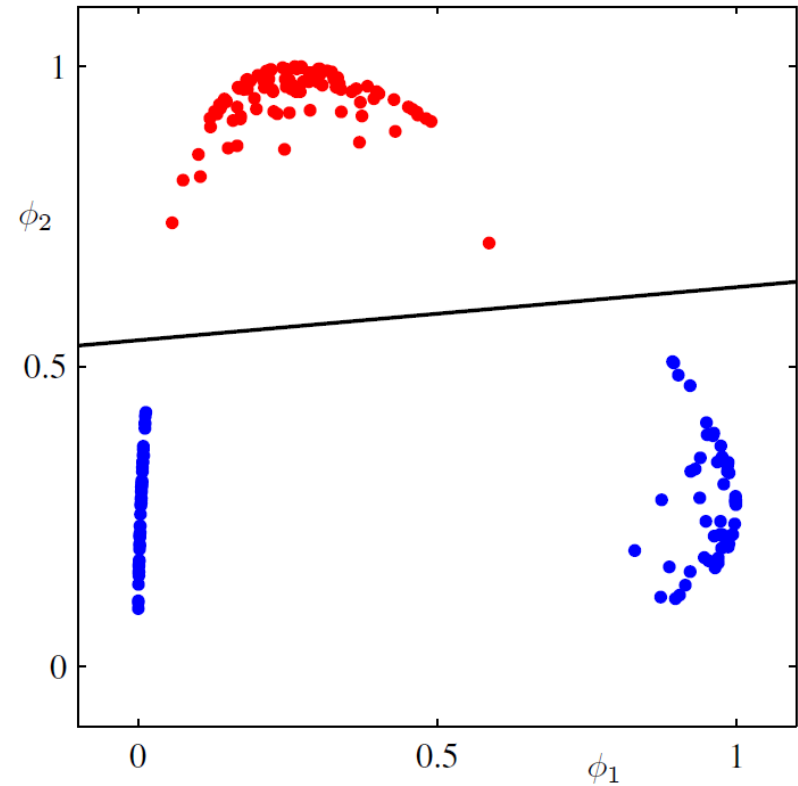
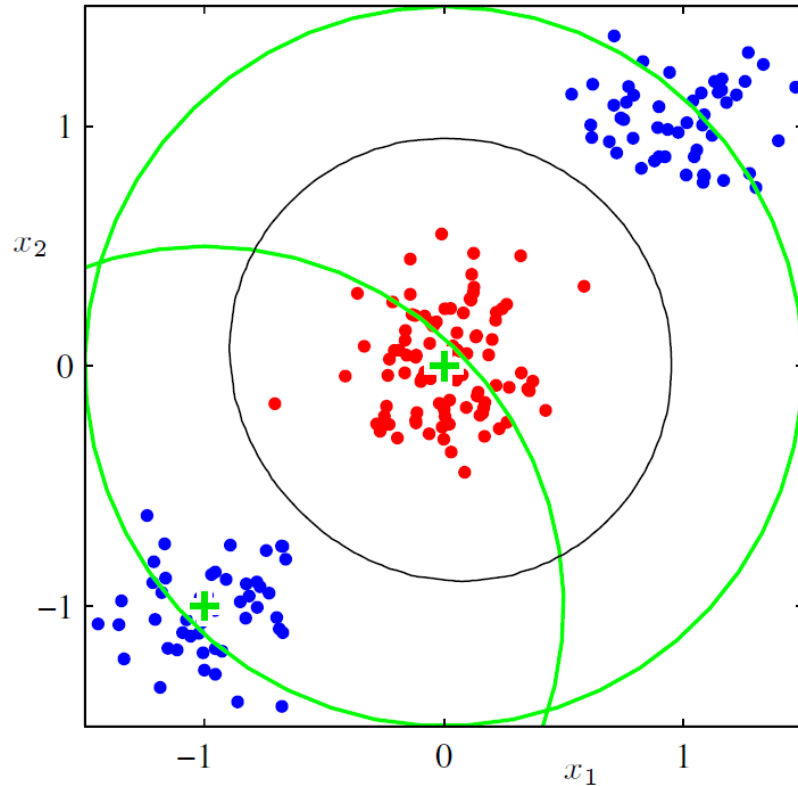
target value

$$E = - \sum_j \overset{\downarrow}{t_j} \ln y_j$$

$$\frac{\partial E}{\partial z_i} = \sum_j \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial z_i} = y_i - t_i$$

# Fixed Basis Functions

---



Using Gaussian basis functions to achieve “linearly separable” cases

---

# Discriminative: ML Logistic Regression

---

$$p(C_0|\phi) = y(\phi) = \sigma(w^T\phi) \quad p(C_1|\phi) = 1 - p(C_0|\phi)$$

where  $\frac{d\sigma(a)}{da} = \sigma(1 - \sigma)$

$$p(t|w) = \prod_{n=1}^N y_n^{t_n} (1 - y_n)^{1-t_n}$$

$$E(w) = -\ln p(t|w) = -\sum_{n=1}^N [t_n \ln y_n + (1 - t_n) \ln(1 - y_n)] \quad \text{Likelihood}$$

$$\nabla E(w) = \sum_{n=1}^N (y_n - t_n) \phi_n \quad H = \nabla \nabla E(w) = \sum_{n=1}^N y_n (1 - y_n) \phi_n \phi_n^T$$

$$w_{ML} \longleftarrow w^{new} = w^{old} - H^{-1} \nabla E(w) \quad q(w) = N(w|w_{ML}, H^{-1})$$

← step size taken as  $H^{-1}$ : Gauss-Newton Method

# Discriminative: ML Predictive Distribution

---

$$\mathbb{E}[t] = y$$

$$\text{var}[t] = y(1 - y)$$

$t$ : Bernoulli distribution with the parameter of  $y$

$$\mathbb{E}[w^T \phi] = w_{ML}^T \phi$$

$$\text{var}[w^T \phi] = \phi^T H^{-1} \phi$$

---

# Discriminative: MAP Logistic Regression

---

$$p(w) = N(w|m_0, S_0) \quad p(w|t) \propto p(w)p(t|w)$$

$$E(w) = -\ln p(w|t) = \frac{1}{2}(w - m_0)^T S_0^{-1} (w - m_0) - \sum_{n=1}^N [t_n \ln y_n + (1 - t_n) \ln(1 - y_n)]$$

$$\nabla E(w) = S_0^{-1}(w - m_0) + \sum_{n=1}^N (y_n - t_n) \phi_n$$

$$H = \nabla \nabla E(w) = S_0^{-1} + \sum_{n=1}^N y_n(1 - y_n) \phi_n \phi_n^T$$

$$w_{MAP} \longleftarrow w^{new} = w^{old} - H^{-1} \nabla E(w) \quad q(w) = N(w|w_{MAP}, H^{-1})$$

← step size taken as  $H^{-1}$ : Gauss-Newton Method

# MAP Predictive Distribution

---

$$\mathbb{E}[t] = y$$

$$\text{var}[t] = y(1 - y)$$

$t$ : Bernoulli distribution with the parameter of  $y$

$$\mathbb{E}[w^T \phi] = w_{MAP} \phi$$

$$\text{var}[w^T \phi] = \phi^T H^{-1} \phi$$

---

# Comparison of Two Approaches

---

□ **Generative approach:** train each model separately to fit the input vectors of that class

- ✓ Different models can be trained on different cores
- ✓ It is easy to add a new class without retraining all the other classes

□ There are significant advantages when the linear models are harder to train

□ **Gaussian Mixture Model**

□ **Discriminative approach:** train both models to maximize the probability of getting the labels right

- ✓ Emphasize the boundary among different classes
- ✓ Fewer parameters to be determined

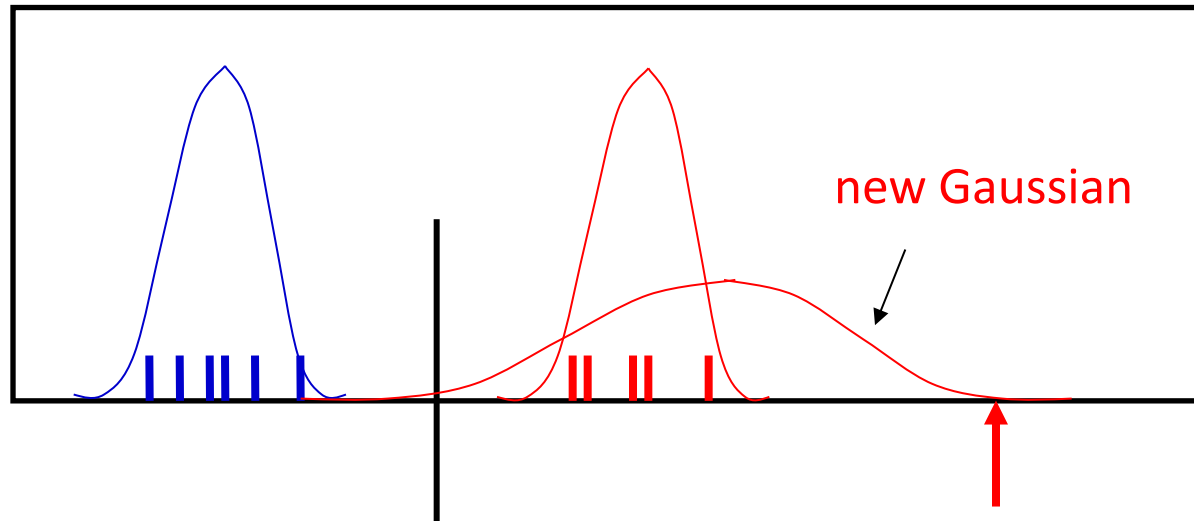
□ There are significant advantages when the linear models are easy to train

□ **Logistic Regression Model**

---

# Comparison of Two Approaches

---



decision  
boundary

What happens to the  
decision boundary if we add  
a new red point here?

For generative fitting, the red mean moves rightwards but the decision boundary moves leftwards!

---



# Outlines

---

- Three Approaches to Linear Classification
    - Approach I: Discriminant Functions
      - Least Square Classification
      - Fisher's Discriminants
      - Perceptrons
    - Approach II: Probabilistic Generative Models
    - Approach III: Probabilistic Discriminative Models
  - Bayesian Information Criterion
-

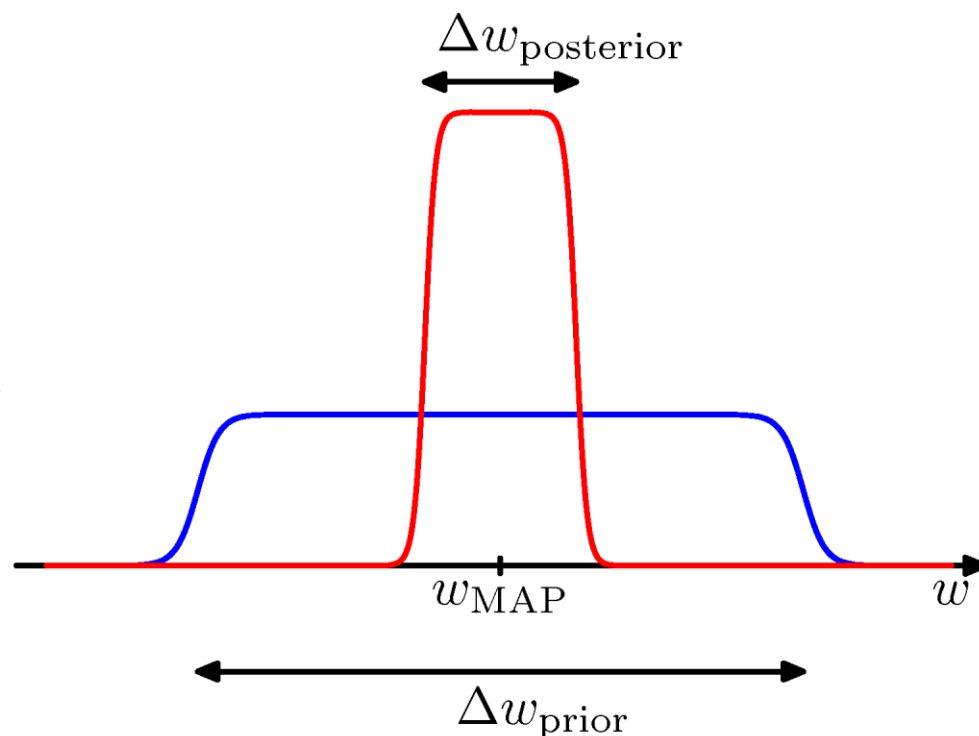
# Bayesian Model Comparison (1)

---

For a given model with a single parameter,  $w$ , consider the approximation

$$p(\mathcal{D}) = \int p(\mathcal{D}|w)p(w) dw$$
$$\simeq p(\mathcal{D}|w_{\text{MAP}}) \frac{\Delta w_{\text{posterior}}}{\Delta w_{\text{prior}}}$$

where the posterior is assumed to be sharply peaked.



# Bayesian Model Comparison (2)

---

Taking logarithms, we obtain

$$\ln p(\mathcal{D}) \simeq \ln p(\mathcal{D}|w_{\text{MAP}}) + \underbrace{\ln \left( \frac{\Delta w_{\text{posterior}}}{\Delta w_{\text{prior}}} \right)}_{\text{Negative}}.$$

With  $M$  parameters, all assumed to have the same ratio  $\Delta w_{\text{posterior}}/\Delta w_{\text{prior}}$ , we get

$$\ln p(\mathcal{D}) \simeq \ln p(\mathcal{D}|\mathbf{w}_{\text{MAP}}) + \underbrace{M \ln \left( \frac{\Delta w_{\text{posterior}}}{\Delta w_{\text{prior}}} \right)}_{\text{Negative and linear in } M}.$$

# Bayesian Information Criterion

---

Akaike Information Criterion (AIC)

$$\ln p(\mathcal{D}|\mathbf{w}_{\text{ML}}) - M$$

Bayesian Information Criterion (BIC)

$$\ln p(\mathcal{D}) \simeq \ln p(\mathcal{D}|\boldsymbol{\theta}_{\text{MAP}}) - \frac{1}{2}M \ln N$$

M: model order; N: data number

---

# Laplace Approximation

---

$$\ln f(\mathbf{z}) \simeq \ln f(\mathbf{z}_0) - \frac{1}{2}(\mathbf{z} - \mathbf{z}_0)^T \mathbf{A}(\mathbf{z} - \mathbf{z}_0)$$

where  $\mathbf{A} = - \nabla \nabla \ln f(\mathbf{z})|_{\mathbf{z}=\mathbf{z}_0}$

$$\begin{aligned} Z &= \int f(\mathbf{z}) \, d\mathbf{z} \\ &\simeq f(\mathbf{z}_0) \int \exp \left\{ -\frac{1}{2}(\mathbf{z} - \mathbf{z}_0)^T \mathbf{A}(\mathbf{z} - \mathbf{z}_0) \right\} \, d\mathbf{z} \\ &= f(\mathbf{z}_0) \frac{(2\pi)^{M/2}}{|\mathbf{A}|^{1/2}} \end{aligned}$$

---

# Model Evaluation

---

Let  $Z = p(\mathcal{D})$   $f(\boldsymbol{\theta}) = p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})$

$$p(\mathcal{D}) = \int p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta}$$

Then, the evidence is given by

$$\ln p(\mathcal{D}) \simeq \ln p(\mathcal{D}|\boldsymbol{\theta}_{\text{MAP}}) + \underbrace{\ln p(\boldsymbol{\theta}_{\text{MAP}}) + \frac{M}{2} \ln(2\pi) - \frac{1}{2} \ln |\mathbf{A}|}_{\text{Occam factor}}$$

where

penalizes model complexity

$$\mathbf{A} = -\nabla \nabla \ln p(\mathcal{D}|\boldsymbol{\theta}_{\text{MAP}})p(\boldsymbol{\theta}_{\text{MAP}}) = -\nabla \nabla \ln p(\boldsymbol{\theta}_{\text{MAP}}|\mathcal{D})$$

---

# Summary

---

- Three Approaches to Linear Classification
    - Approach I: Discriminant Functions
      - Least Square Classification
      - Fisher's Discriminants
      - Perceptrons
    - Approach II: Probabilistic Generative Models
    - Approach III: Probabilistic Discriminative Models
      - Bayesian Information Criterion
-