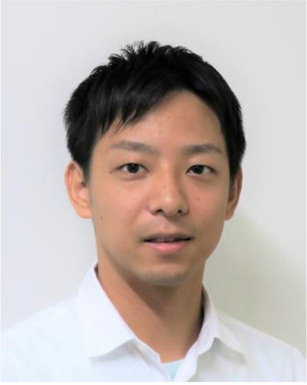
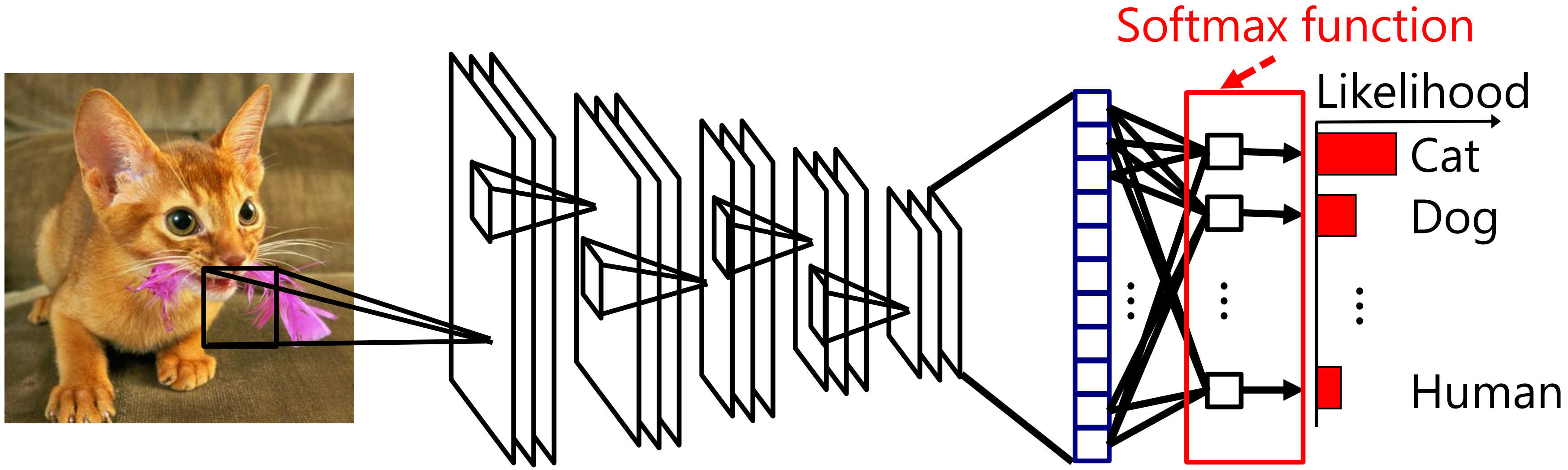
A Discriminative Gaussian Mixture Model with Sparsity

Hideaki Hayashi and Seiichi Uchida

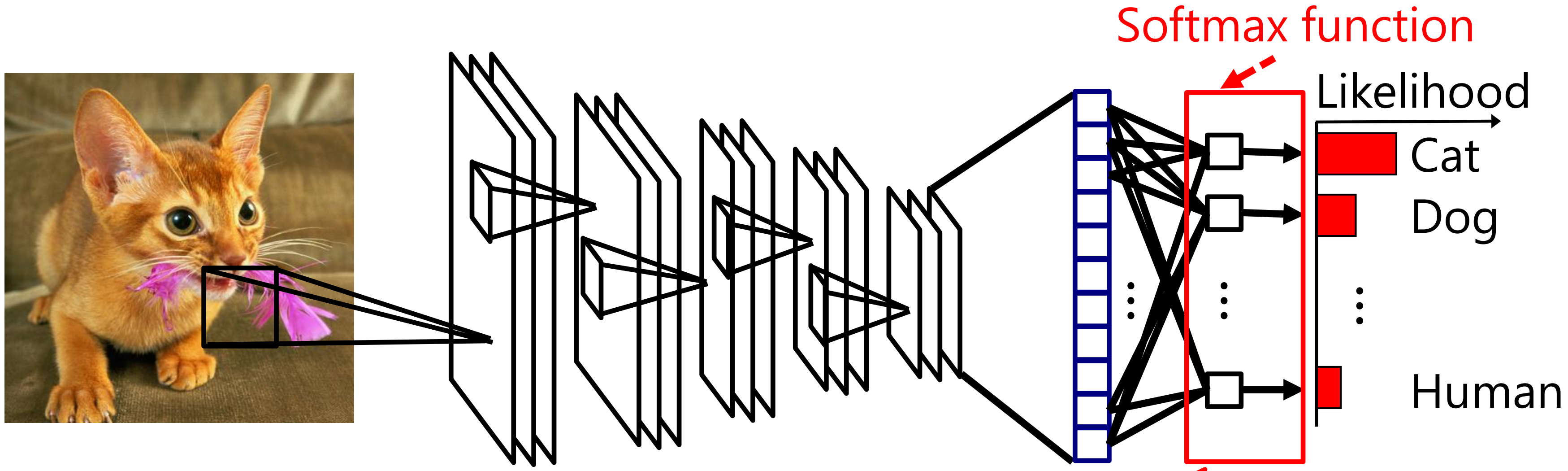
Kyushu University



Why do you use softmax?



Why do you use softmax?



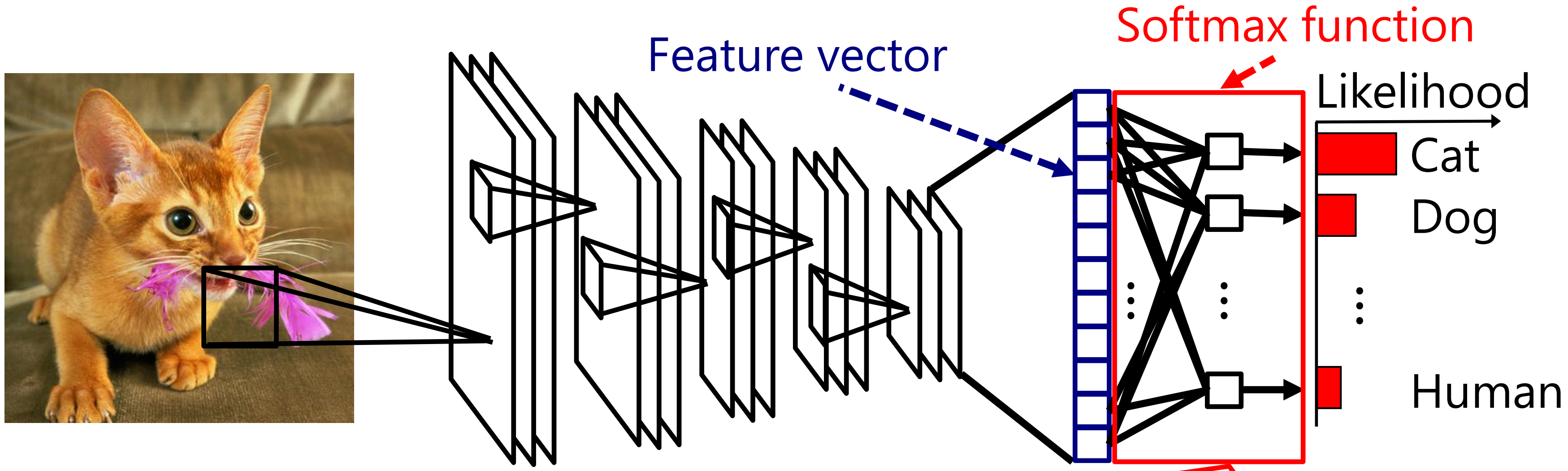
Normalization of likelihoods

Input		Output
y_c	⇒	$f(y_c)$
can be negative		range of [0, 1]

$$f(y_c) = \frac{\exp[y_c]}{\sum_{c=1}^{C} \exp[y_c]}$$

y_c: input for class c
C: number of classes

What softmax really does



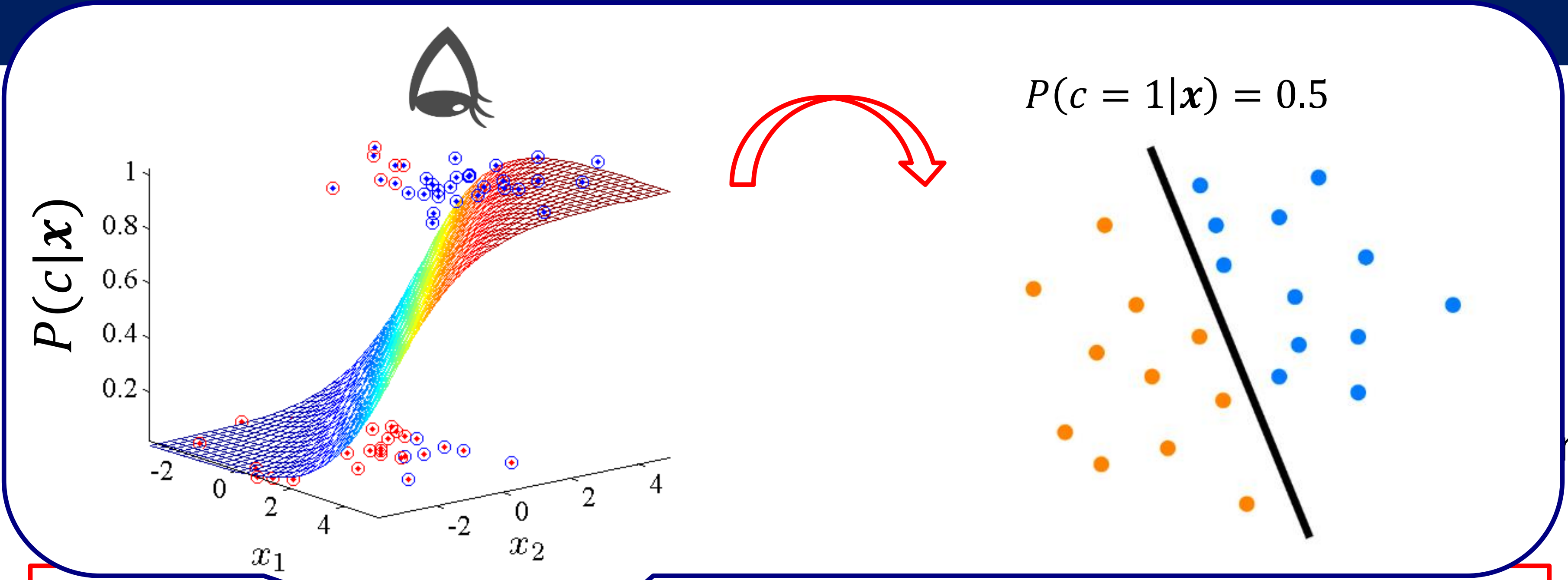
Fully connected (FC) layer + softmax function is a **dense** and **linear** classifier

$$P(c|\boldsymbol{x}) = \frac{\exp[\boldsymbol{w}_c^{\mathrm{T}}\boldsymbol{x}]}{\sum_{c=1}^{C}\exp\left[\boldsymbol{w}_c^{\mathrm{T}}\boldsymbol{x}\right]}$$

$\boldsymbol{x}$: feature vector

$\boldsymbol{w}_c$: weights for class c

What softmax really does



Fully connec C) layer + softmax function is a **dense** and **linear** classifier

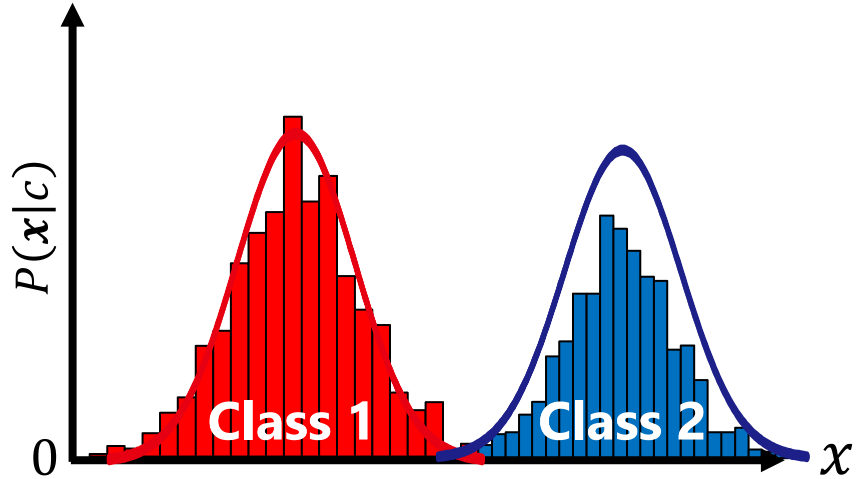
$$P(c|\boldsymbol{x}) = \frac{\exp[\boldsymbol{w}_c^{\mathrm{T}}\boldsymbol{x}]}{\sum_{c=1}^{C}\exp[\boldsymbol{w}_c^{\mathrm{T}}\boldsymbol{x}]}$$

$\boldsymbol{x}$: feature vector

$\boldsymbol{w}_c$: weights for class c

Discriminative model based on Gaussian distribution

One Gaussian for each class



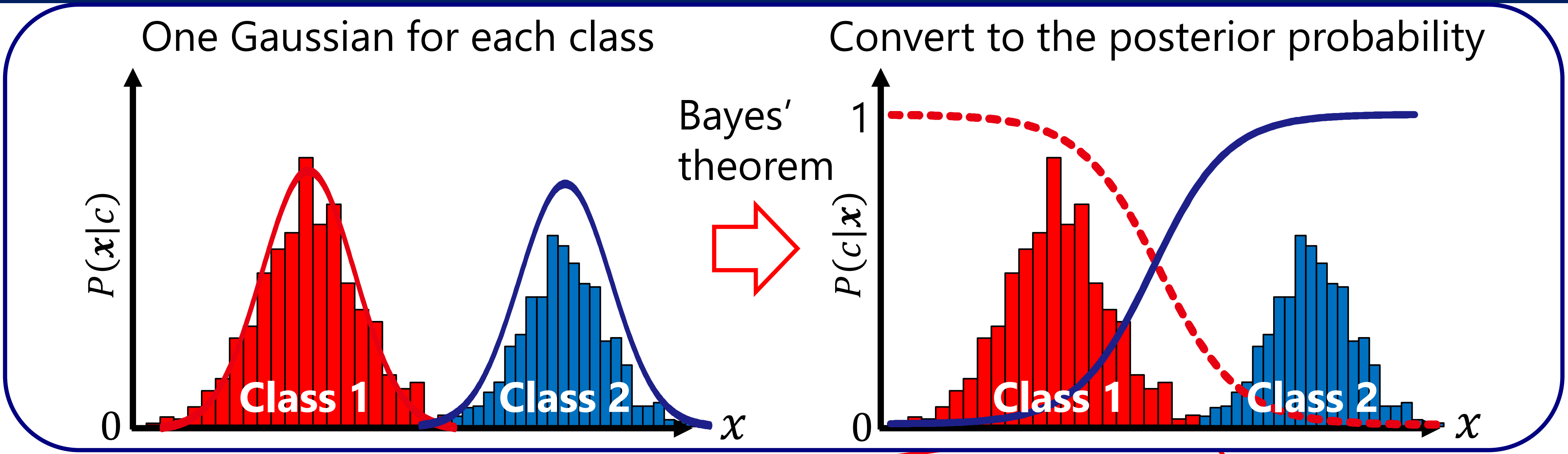
Fully connected (FC) layer + softmax function is a **dense** and **linear** classifier

$$P(c|\boldsymbol{x}) = \frac{\exp[\boldsymbol{w}_c^{\mathrm{T}}\boldsymbol{x}]}{\sum_{c=1}^{C}\exp[\boldsymbol{w}_c^{\mathrm{T}}\boldsymbol{x}]}$$

$\boldsymbol{x}$: feature vector

$\boldsymbol{w}_c$: weights for class c

Discriminative model based on Gaussian distribution



One Gaussian for each class

Convert to the posterior probability

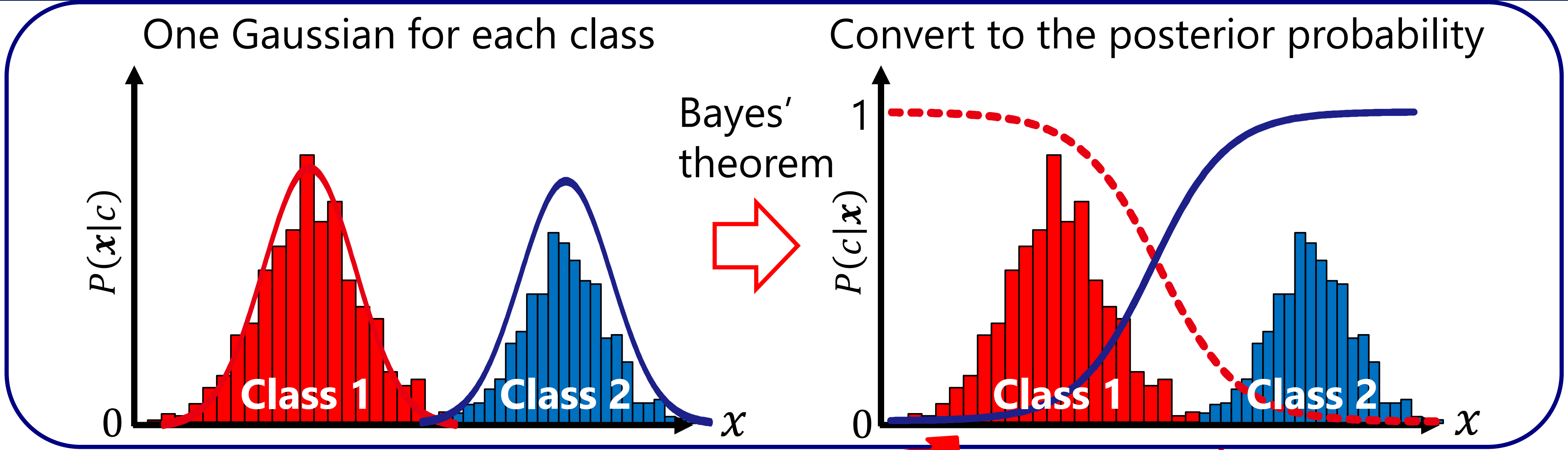
Fully connected (FC) layer + softmax function is a **dense** and **linear** classifier

$$P(c|\boldsymbol{x}) = \frac{\exp[\boldsymbol{w}_c^{\mathrm{T}}\boldsymbol{x}]}{\sum_{c=1}^{C}\exp\left[\boldsymbol{w}_c^{\mathrm{T}}\boldsymbol{x}\right]}$$

$\boldsymbol{x}$: feature vector

$\boldsymbol{w}_c$: weights for class c

Discriminative model based on Gaussian distribution



Same function

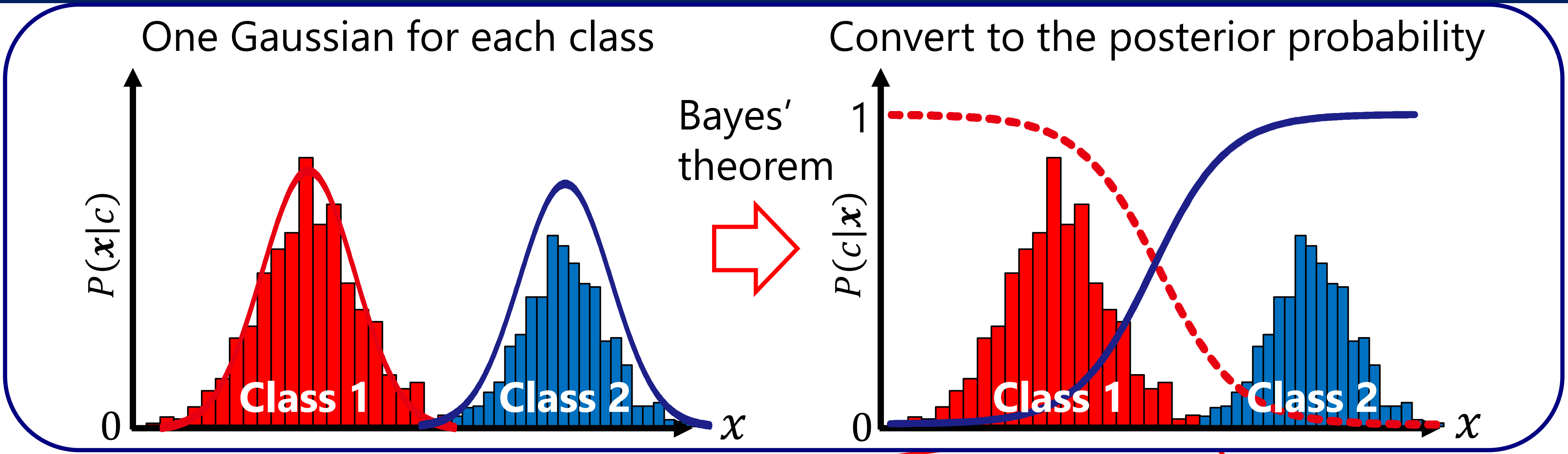
Fully connected (FC) layer + soft
a **dense** and **linear** classifier

$$P(c|\boldsymbol{x}) = \frac{\exp[\boldsymbol{w}_c^{\mathrm{T}}\boldsymbol{x}]}{\sum_{c=1}^{C}\exp[\boldsymbol{w}_c^{\mathrm{T}}\boldsymbol{x}]}$$

$\boldsymbol{x}$: feature vector

$\boldsymbol{w}_c$: weights for class c

Discriminative model based on Gaussian distribution



Fully connected (FC) layer + softmax function is a **dense**, **linear**, and **unimodal** classifier

$$P(c|\boldsymbol{x}) = \frac{\exp[\boldsymbol{w}_c^{\mathrm{T}}\boldsymbol{x}]}{\sum_{c=1}^{C}\exp[\boldsymbol{w}_c^{\mathrm{T}}\boldsymbol{x}]}$$

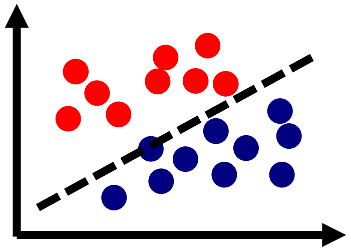
$\boldsymbol{x}$: feature vector

$\boldsymbol{w}_c$: weights for class c

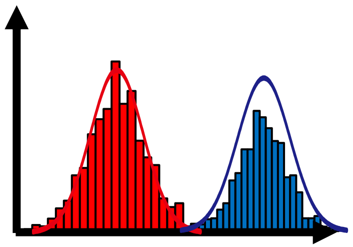
Sparse Discriminative Gaussian Mixture (SDGM)

FC layer + softmax

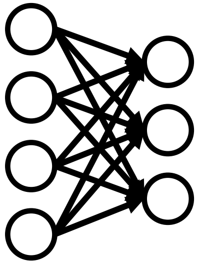
- Linear



- Unimodal

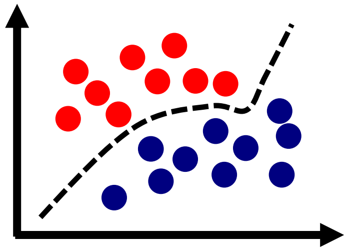


- Dense

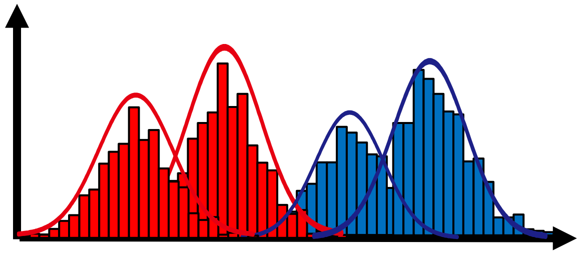


SDGM (proposed method)

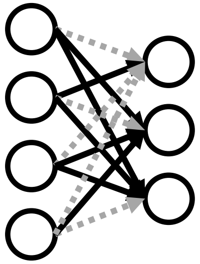
- Nonlinear



- Multimodal

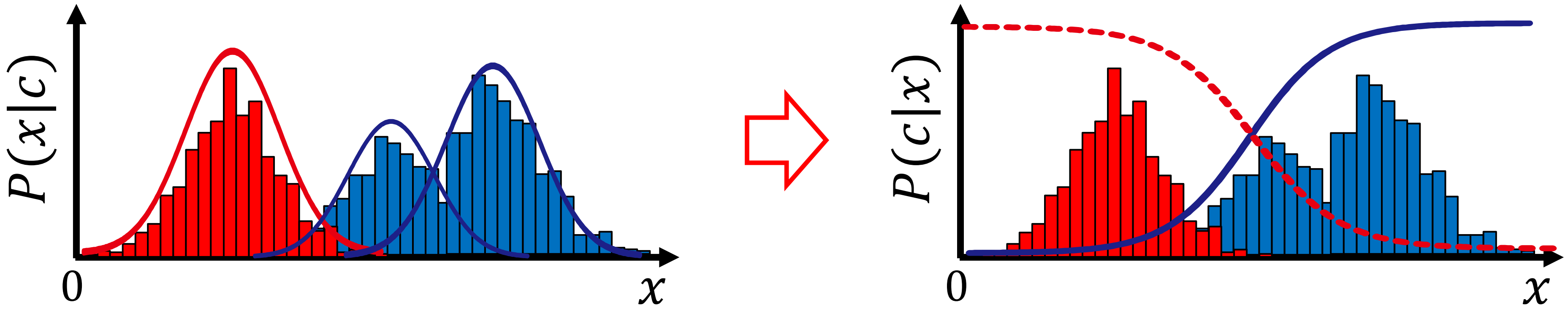


- Sparse

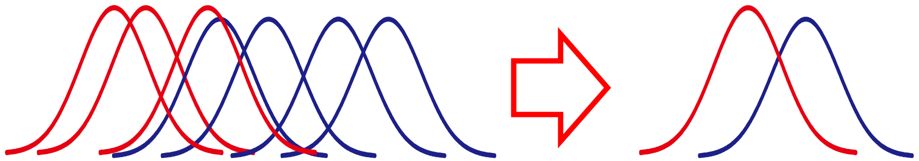


Technical highlights

- **Discriminative model based on Gaussian mixture models (GMMs)**



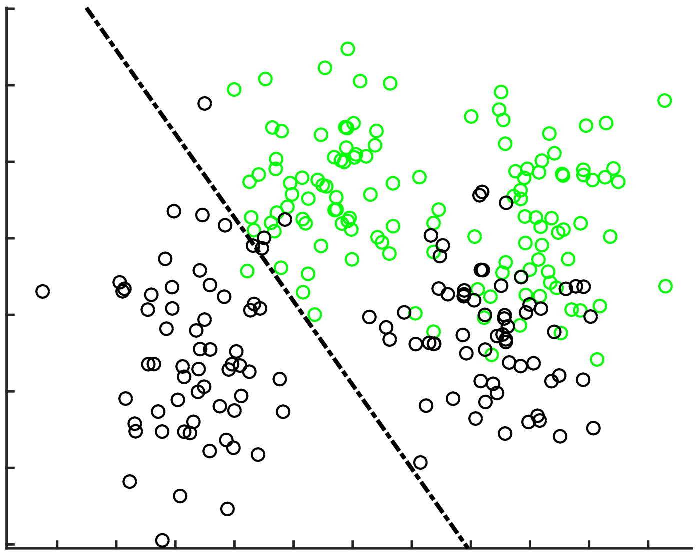
- **Sparse Bayesian learning**
  - Removes redundant weights
  - Determines the number of Gaussian components automatically



Better accuracy while reducing parameters

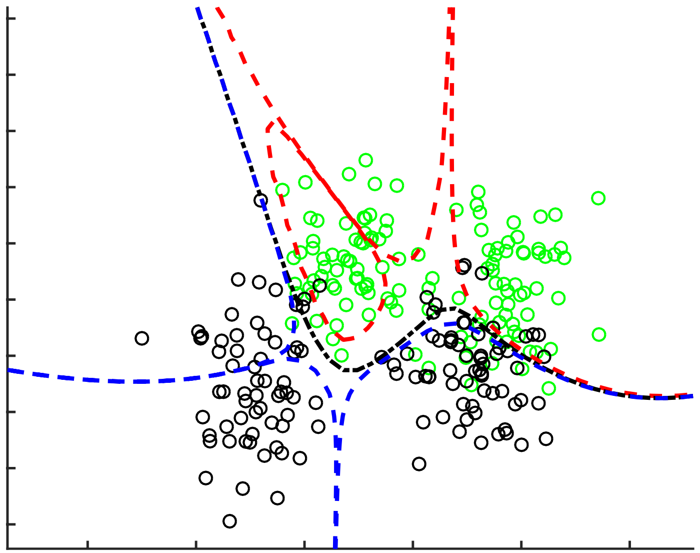
FC layer + softmax

Error rate: 11.4



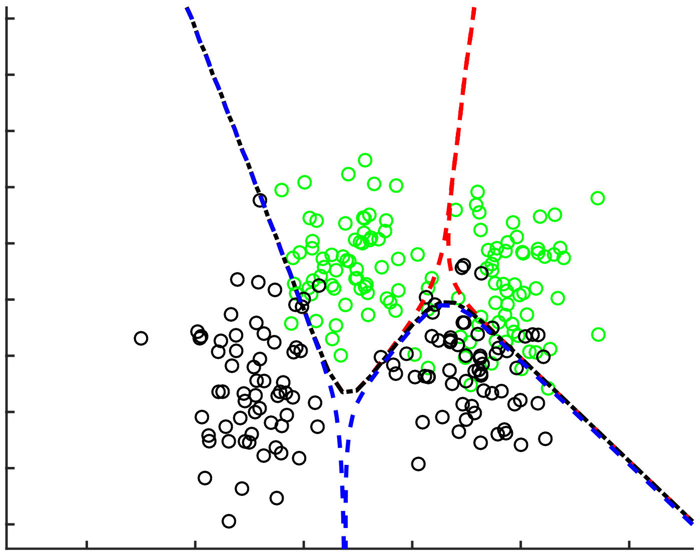
GMM-based classifier

Error rate: 9.3



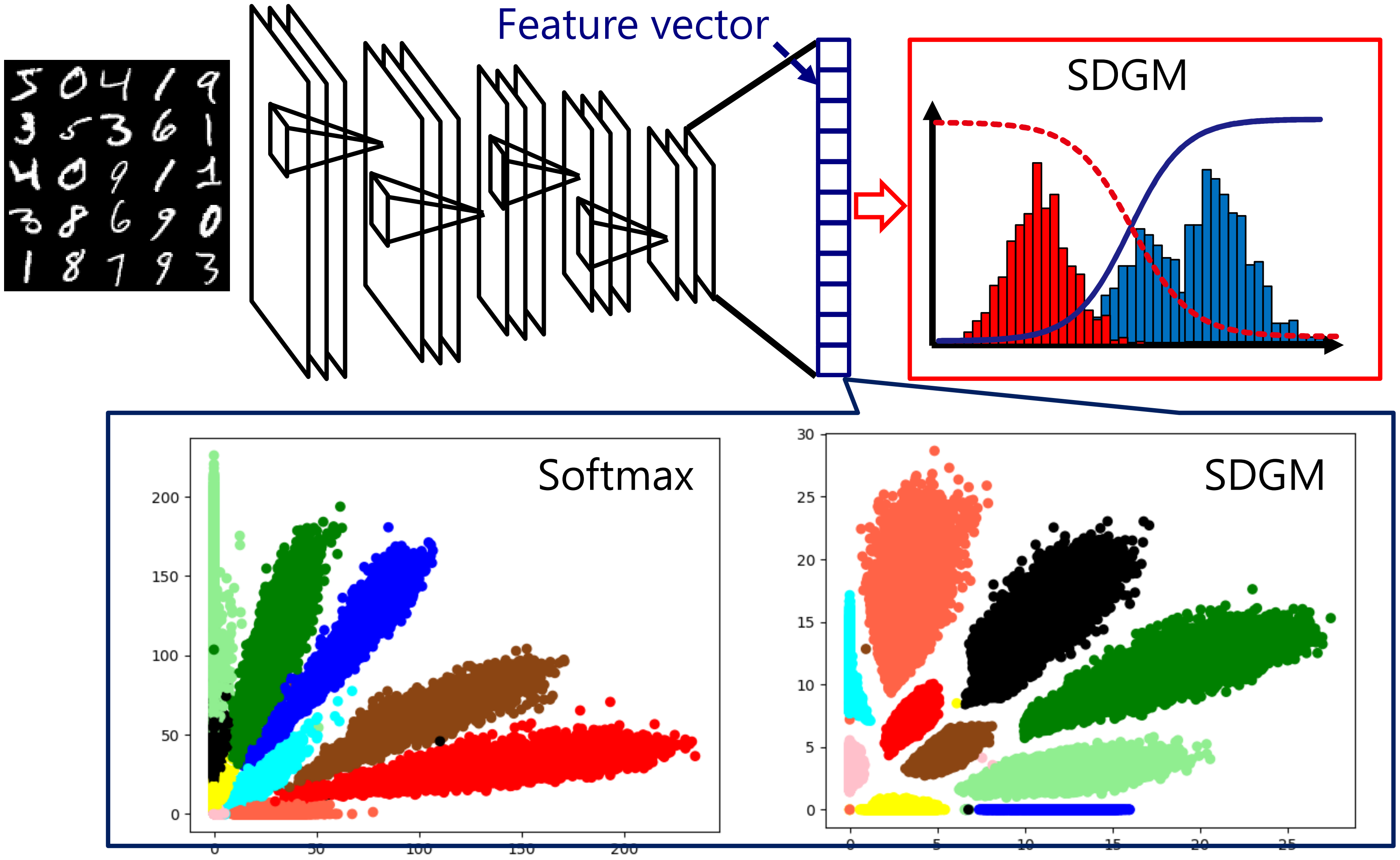
SDGM

Error rate: **9.1**



# Gaussian components:	6		**4**
# Parameters:	1250	99% reduced	**6**

Features were pushed into a Gaussian shape with margins


Feature vector
SDGM
Softmax
0
50
100
150
200
0
50
100
150
200
SDGM
0
5
10
15
20
25
30
0
5
10
15
20
25