No.4

Generative Models (I): Bayesian Decision Theory

Hui Jiang

Department of Electrical Engineering and Computer Science

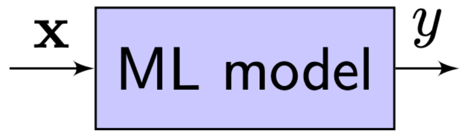
Lassonde School of Engineering

York University, Toronto, Canada

Outline

- Discriminative vs. Generative models
  - Generative modeling: a statistical perspective to ML
- Bayesian decision theory
  - Generative models for classification
  - Generative models for regression
- The Plug-in MAP rule
- Some probabilistic models for generative modeling

Discriminative Models in ML

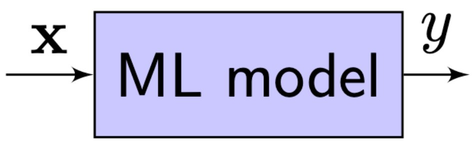


- Input $\mathbf{x}$ is a random vector, $\mathbf{x} \sim p(\mathbf{x})$
- Output y is generated by a *deterministic target* function $y = \bar{f}(\mathbf{x})$ for each $\mathbf{x}$
- Our goal: estimate $\bar{f}(\cdot)$ in a model space $\mathbb{H}$
- Training samples: $\mathcal{D}_N = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_N, y_N)\}$, where $\mathbf{x}_i \sim p(\mathbf{x})$ and $y_i = \bar{f}(\mathbf{x}_i)$
- Determine a loss function $l(l, l')$
- Empirical risk mininization (ERM):

$$f^* = \arg\min_{f \in \mathbb{H}} R_{\text{emp}}(f|\mathcal{D}_N) = \arg\min_{f \in \mathbb{H}} \sum_{i=1}^{N} l(y_i, f(\mathbf{x}_i))$$

- The performance depends on the generalization bound

Generative Models in ML



- Input $\mathbf{x}$ and output y are both random variables, $(\mathbf{x}, y) \sim p(\mathbf{x}, y)$
- The relation $\mathbf{x} \to y$ solely relies on $p(y|\mathbf{x})$
- Our goal: estimate $p(\mathbf{x}, y)$ using a probabilistic model $\hat{p}_\theta(\mathbf{x}, y)$
- Training samples: $\mathcal{D}_N = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_N, y_N)\}$, where $(\mathbf{x}_i, y_i) \sim p(\mathbf{x}, y)$
- The relation $\mathbf{x} \to y$ may be approximated by:

$$\hat{p}_\theta(y|\mathbf{x})$$

- The performance depends on the gap between $p(\mathbf{x}, y)$ and $\hat{p}_\theta(\mathbf{x}, y)$: $\mathrm{KL}\big(p(\cdot) \mid\mid \hat{p}_\theta(\cdot)\big)$

Generative Models for Classification

- Input $\mathbf{x}$: feature vectors (continuous or discrete)
- Output is discrete $y = \{\omega_1, \omega_2, \cdots, \omega_K\}$: class label
- The joint distribution $p(\mathbf{x}, y) = p(y)p(\mathbf{x}|y)$ breaks down to:
  - Prior probabilities: $p(y = \omega_k) \triangleq \Pr(\omega_k)$ $(\forall k = 1, 2, \cdots, K)$
  - Class-conditional distribution: $p(\mathbf{x}|y = \omega_k) \triangleq p(\mathbf{x}|\omega_k)$ $(\forall k = 1, 2, \cdots, K)$
- Probabilistic distribution constraints:
  - Priors satisfy $\sum_{k=1}^{K} Pr(\omega_k) = 1$
  - If $\mathbf{x}$ is continuous,

$$\int_{\mathbf{x}} p(\mathbf{x}|\omega_k) d\mathbf{x} = 1 \quad (\forall k = 1, 2, \cdots, K)$$

  - If $\mathbf{x}$ is discrete,

$$\sum_{\mathbf{x}} p(\mathbf{x}|\omega_k) = 1 \quad (\forall k = 1, 2, \cdots, K)$$

Example of class-conditional p.d.f.

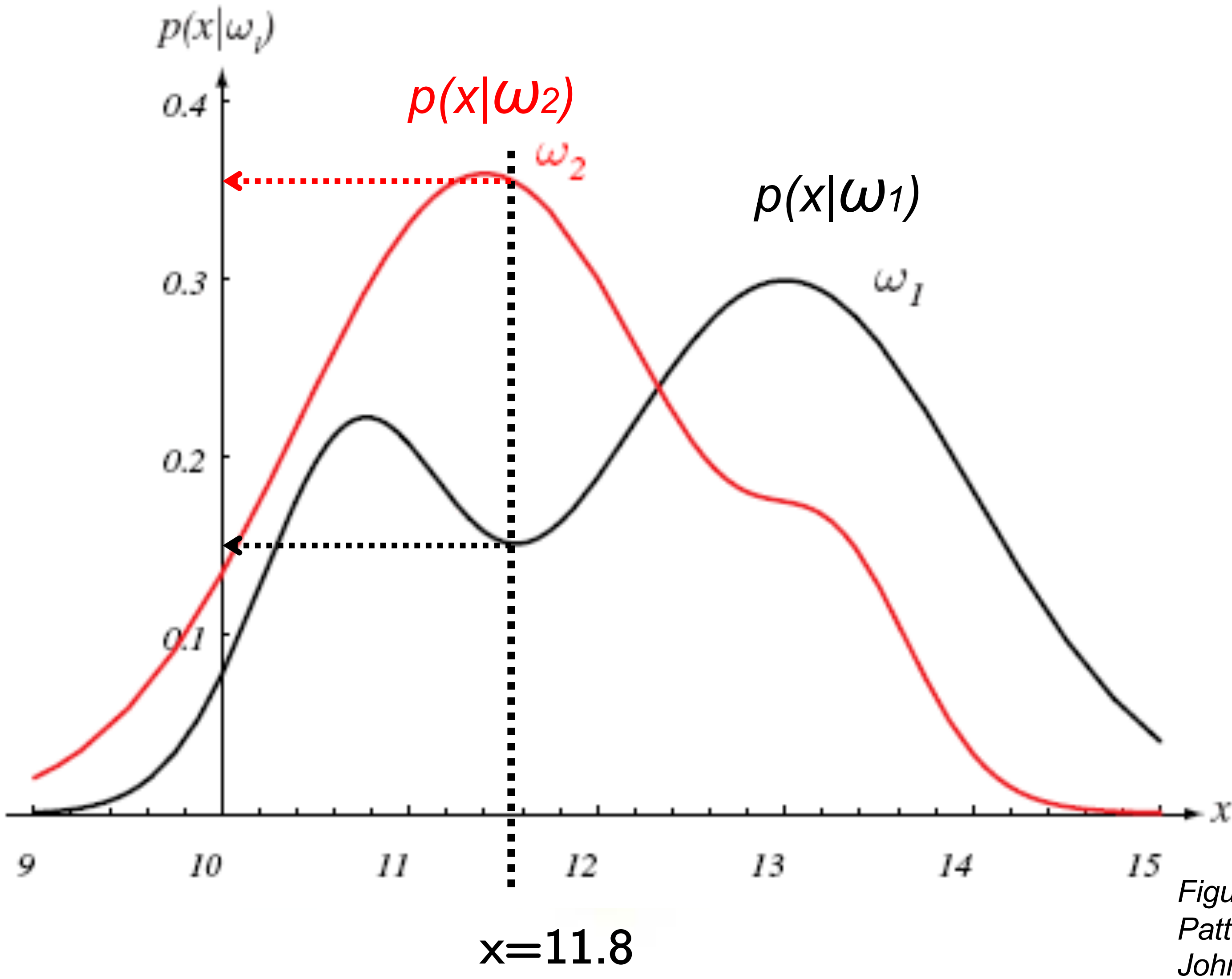


Figure from Duda et. al., Pattern classification John Wiley & Sons®, Inc.

Examples of pattern classification(I)

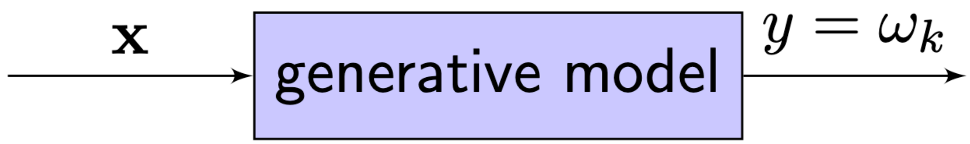
- Speech recognition:
  - Pattern: voice spoken by a human being
  - Classes: language words/sentences used by the speaker
  - Input features: speech signal characteristics measured by a microphone → a sequence of feature vectors
    - Each vector: continuous, high-dimensional, real-valued numbers

- Natural language understanding:
  - Pattern: written or spoken languages of human
  - Classes: all possible semantic meanings or intentions
  - Input features: the used words or word-sequences (sentences)
    - Discrete, scalars or vector

Examples of pattern classification(II)

- Image understanding:
  - Pattern: given images
  - Classes: all known object categories
  - Input features: color or gray scales in all pixels
    - Continuous, multiple vectors/matrix
  - Examples: face recognition, OCR (optical character recognition).
- Gene finding in bioinformatics:
  - Pattern: a newly sequenced DNA sequence
  - Classes: all known genes
  - Input features: all nucleotides in the sequence
    - Discrete; 4 types (adenine, guanine, cytosine, thymine)
- Protein classification in bioinformatics:
  - Pattern: protein primary 1-D sequence
  - Classes: all known protein families or domains
  - Input features: all amino acids in the sequence: discrete; 20 types

Bayesian Decision Theory (I): Classification



- Given any $\mathbf{x}$, determine the best $g(\mathbf{x}) \in \{\omega_1, \cdots \omega_K\}$
- The decision rule: $\mathbf{x} \Rightarrow \omega_k \quad (\forall k = 1, 2, \cdots, K)$
- Bayesian Decision Theory: the best decision is

$$
\begin{aligned}
g^*(\mathbf{x}) &= \arg\max_k \; {\color{red} p(\omega_k|\mathbf{x})} = \arg\max_k \; \frac{\Pr(\omega_k) p(\mathbf{x}|\omega_k)}{p(\mathbf{x})} \\
&= \arg\max_k \; \Pr(\omega_k) \cdot p(\mathbf{x}|\omega_k)
\end{aligned}
$$

  which is called **maximum a posterior (MAP) rule** or Bayes decision rule.
- Proof: why this is optimal?

Optimality of the MAP rule (I)

Theorem 1

Assume $p(\mathbf{x}, \omega)$ is known, when $\mathbf{x}$ is used to predict ω, the MAP rule leads to the lowest expected risk (using 0-1 loss).

Proof:

- The 0-1 loss function: $l(\omega, \omega') = \begin{cases} 0 & \text{when } \omega = \omega' \\ 1 & \text{otherwise} \end{cases}$
- The expected risk of any rule $\mathbf{x} \to g(\mathbf{x}) \in \{\omega_1, \cdots \omega_K\}$:

$$
\begin{aligned}
R(g) &= \mathbb{E}_{p(\mathbf{x},\omega)}\Big[l\big(\omega, g(\mathbf{x})\big)\Big] = \int_{\mathbf{x}} \sum_{k=1}^{N} l\big(\omega_k, g(\mathbf{x})\big) p(\mathbf{x}, \omega_k) d\mathbf{x} \\
&= \int_{\mathbf{x}} \underbrace{\left[\sum_{k=1}^{N} l\big(\omega_k, g(\mathbf{x})\big) p(\omega_k|\mathbf{x})\right]}_{\sum_{\omega_k \neq g(\mathbf{x})} p(\omega_k|\mathbf{x})} p(\mathbf{x}) d\mathbf{x}
\end{aligned}
$$

Optimality of the MAP rule (II)

- Due to $\sum_{k=1}^{N} p(\omega_k|\mathbf{x}) = 1$, we have

$$\sum_{\omega_k \neq g(\mathbf{x})} p(\omega_k|\mathbf{x}) = 1 - p\big(g(\mathbf{x})|\mathbf{x}\big)$$

- We have

$$R(g) \downarrow \implies \forall \mathbf{x}, \Big[1 - p\big(g(\mathbf{x})|\mathbf{x}\big)\Big] \downarrow \implies \forall \mathbf{x}, p\big(g(\mathbf{x})|\mathbf{x}\big) \uparrow$$

- Since $g(\mathbf{x}) \in \{\omega_1, \cdots \omega_K\}$, we choose:

$$g^*(\mathbf{x}) = \arg\max_k \; p(\omega_k|\mathbf{x})$$

∎

The MAP decision rule example

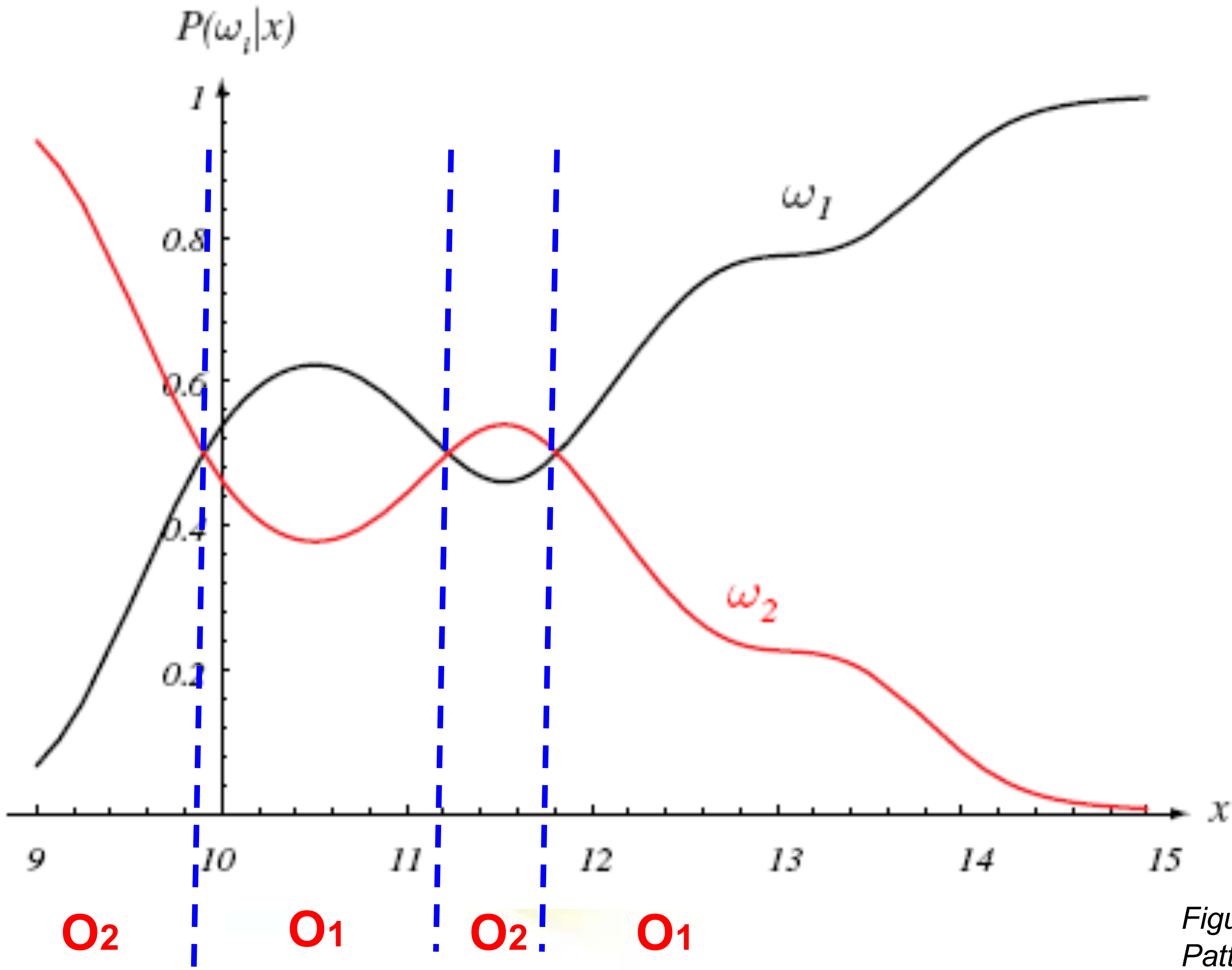
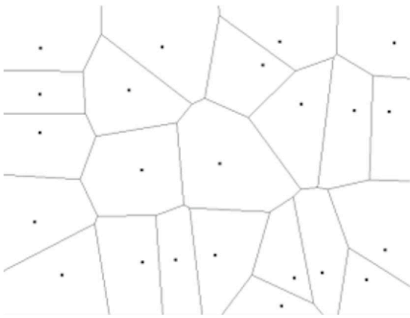
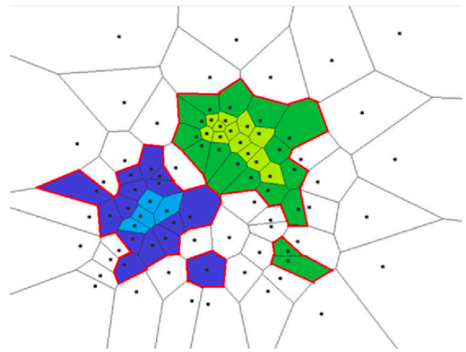


Figure from Duda et. al.,
Pattern classification
John Wiley & Sons®, Inc.

Classification Error Probability

- Any rule $\mathbf{x} \rightarrow g(\mathbf{x}) \in \{\omega_1, \cdots \omega_K\}$ partitions input space into K regions: O_1, O_2, $\cdots$,O_K: if $\mathbf{x} \in O_k$, implies $g(\mathbf{x}) = \omega_k$.



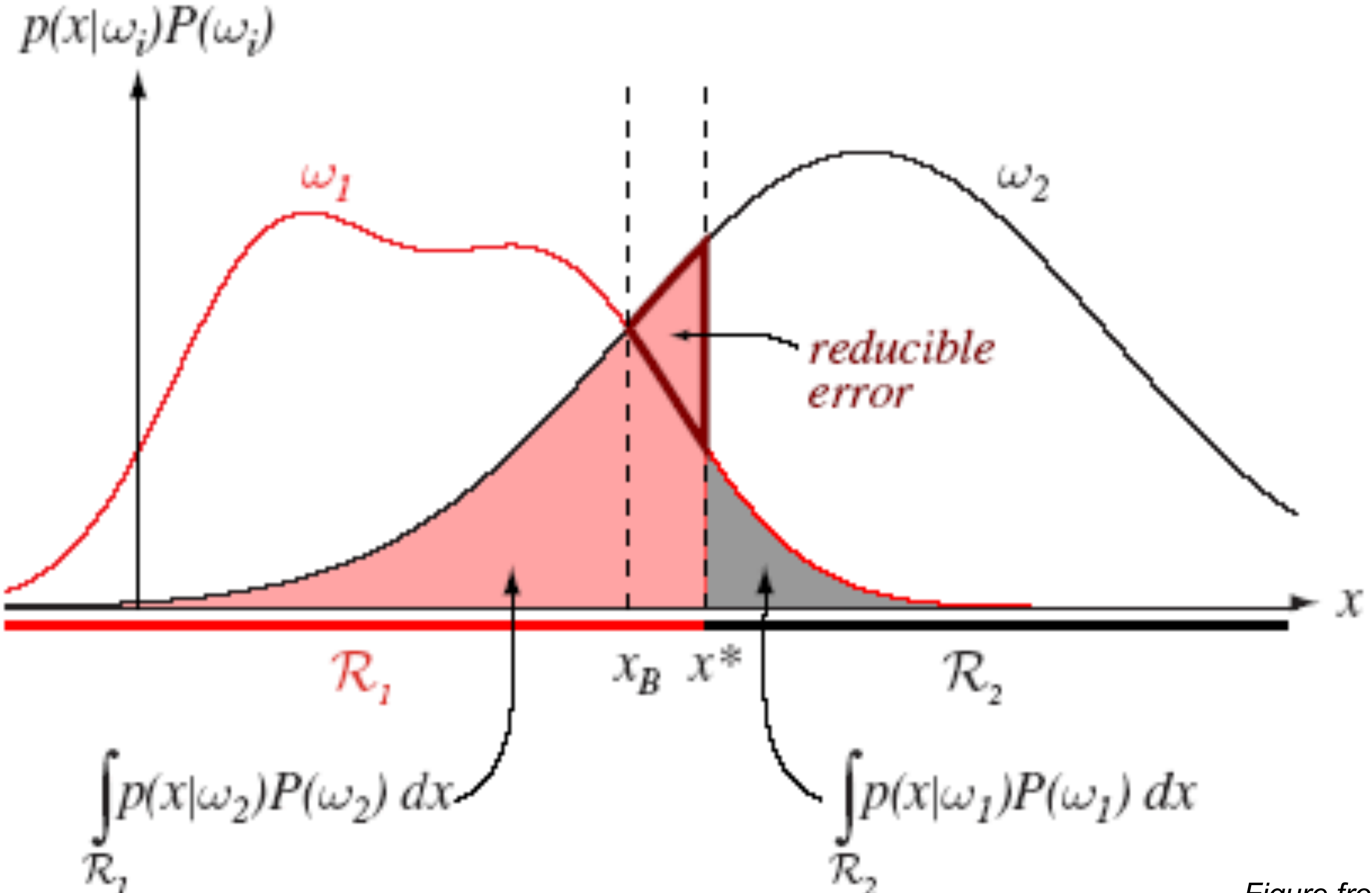


- The expected risk is the probability of classification error:

$$
\begin{aligned}
R(g) &= \Pr(\text{error}) = 1 - \Pr(\text{correct}) = 1 - \sum_{k=1}^{K} \Pr(\mathbf{x} \in O_k, \omega_k) \\
&= 1 - \sum_{k=1}^{K} \Pr(\omega_k) \int_{\mathbf{x} \in O_k} p(\mathbf{x}|\omega_k) d\mathbf{x}
\end{aligned}
$$

- Bayes error: $R(g^*)$ of the MAP rule (the lowest possible error)

Example of Error Probability in 2-class case



Error ω2 → ω1

Error ω1 → ω2

Figure from Duda et. al., Pattern classification John Wiley & Sons®, Inc.

Bayes Error

- Bayes error: error probability of the Bayes (MAP) decision rule.

- Since Bayes decision rule guarantees the minimum error, the Bayes error is the lower bound of all possible error probabilities.

- It is difficult to calculate the Bayes error, even for the very simple cases because of discontinuous nature of the decision regions in the integral, especially in high dimensions.

- Some approximation methods to estimate an upper bound.
  - Chernoff bound
  - Bhattacharyya bound

- Evaluate on an independent test set.

Example: the MAP rule for independent binary features

- 2-class (ω_1 and ω_2) classification: $\Pr(\omega_1)$ and $\Pr(\omega_2)$
- Using n independent binary features $\mathbf{x} = \left[x_1, x_2, \cdots, x_n\right]^\mathsf{T}$, $x_i \in \{0, 1\} \quad i = 1, 2, \cdots, n$
- Denote $p_i \triangleq \Pr(x_i = 1|\omega_1)$ and $q_i \triangleq \Pr(x_i = 1|\omega_2)$, we have:

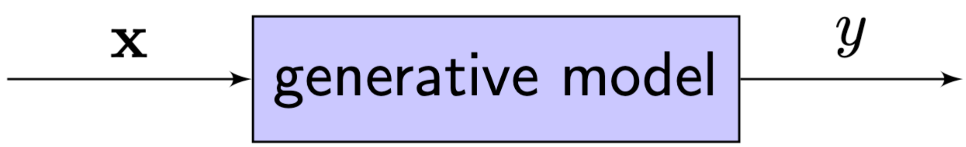
$$p(\mathbf{x}|\omega_1) = \prod_{i=1}^{n} p_i^{x_i}(1 - p_i)^{1-x_i} \quad p(\mathbf{x}|\omega_2) = \prod_{i=1}^{n} q_i^{x_i}(1 - q_i)^{1-x_i}$$

- The MAP rule: given $\mathbf{x}$, classify as ω_1 if $\Pr(\omega_1) \cdot p(\mathbf{x}|\omega_1) \geq \Pr(\omega_2) \cdot p(\mathbf{x}|\omega_2)$, otherwise ω_2.
- Take logarithm to derive a **linear** decision boundary:

$$g(\mathbf{x}) = \sum_{i=1}^{n} \lambda_i x_i + \lambda_0 = \begin{cases} \geq 0 & \Longrightarrow \quad \omega_1 \\ < 0 & \Longrightarrow \quad \omega_2 \end{cases}$$

where $\lambda_i = \ln \frac{p_i(1-q_i)}{q_i(1-p_i)}$ and $\lambda_0 = \sum_{i=1}^{n} \ln \frac{1-p_i}{1-q_i} + \ln \frac{\Pr(\omega_1)}{\Pr(\omega_2)}$

Generative Models for Regression



- Input: n-dimensional vector $\mathbf{x}$, output: $y \in \mathbb{R}$
- The joint distribution $p(\mathbf{x}, y)$ is know, $\mathbf{x}$ is used to predict y.
- What is the best decision rule for $\mathbf{x} \to y = g(\mathbf{x})$?

$$g^*(\mathbf{x}) = \mathbb{E}(y|\mathbf{x}) = \int_{\mathbf{x}} y \cdot p(y|\mathbf{x}) dy$$

Theorem 2

Assume $p(\mathbf{x}, y)$ is known, the conditional mean $\mathbb{E}(y|\mathbf{x})$ leads to the lowest expected risk (using mean square loss).

Optimality of Conditional Mean for Regression

Proof:

- The expected risk of any rule $\mathbf{x} \to g(\mathbf{x}) \in \mathbb{R}$:

$$
\begin{aligned}
R(g) &= \mathbb{E}_{p(\mathbf{x},y)}\Big[l\big(\omega, g(\mathbf{x})\big)\Big] = \int_{\mathbf{x}}\int_{y}\Big(y - g(\mathbf{x})\Big)^2 p(\mathbf{x}, y)d\mathbf{x}dy \\
&= \int_{\mathbf{x}} \underbrace{\left[\int_{y}\Big(y - g(\mathbf{x})\Big)^2 p(y|\mathbf{x})dy\right]}_{Q(g|\mathbf{x})} p(\mathbf{x})d\mathbf{x}
\end{aligned}
$$

- Functional derivative:

$$
\frac{\partial Q(g|\mathbf{x})}{\partial g(\cdot)} = 0 \implies \int_{y}\Big(g(\mathbf{x}) - y\Big)p(y|\mathbf{x})dy = 0
$$

$$
\implies g^*(\mathbf{x}) = \int_{y} y \cdot p(y|\mathbf{x})dy = \mathbb{E}(y|\mathbf{x}) \qquad \blacksquare
$$

Plug-in MAP Decision Rule for classification

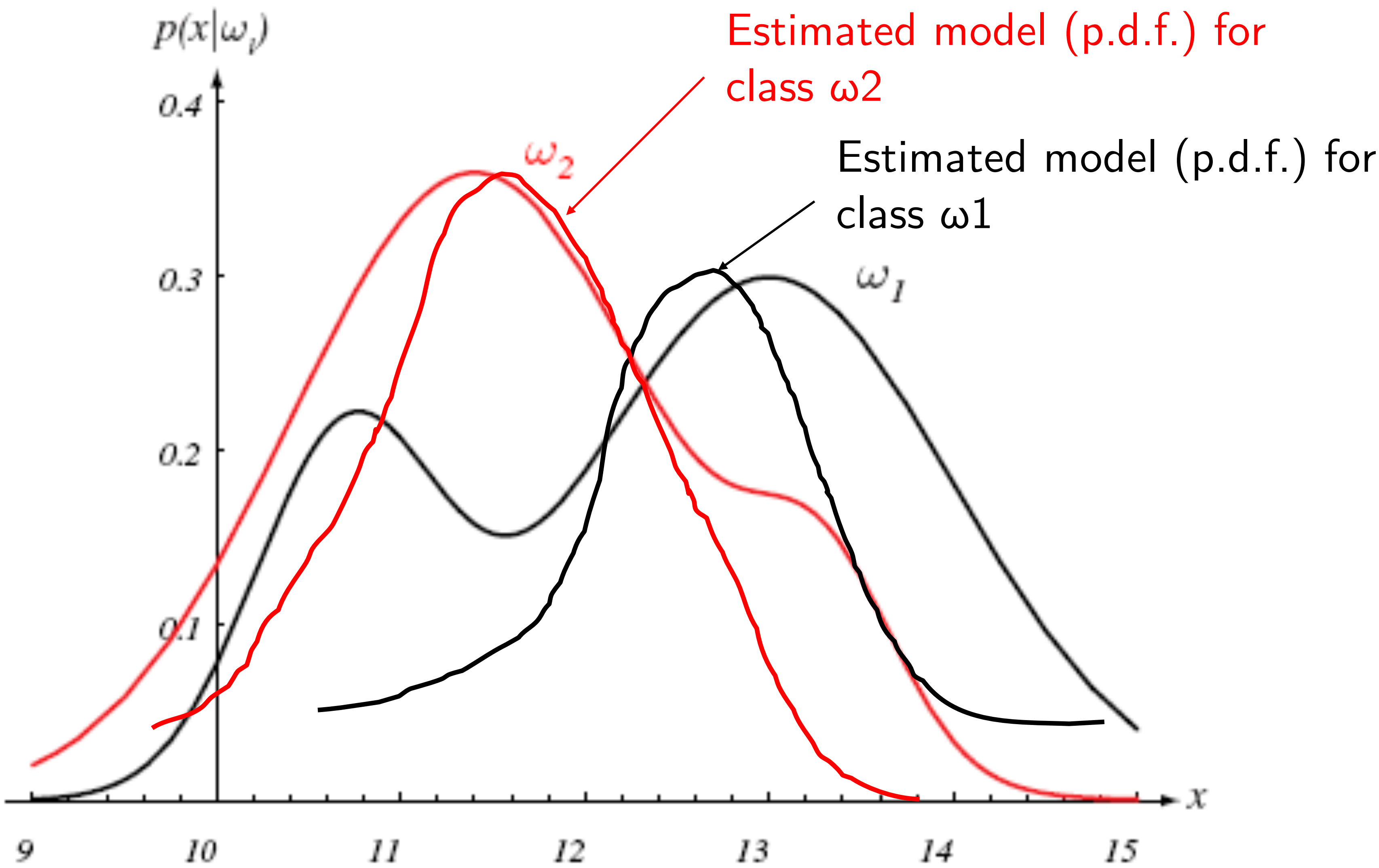
- The true distributions $\Pr(\omega_k)$ and $p(\mathbf{x}|\omega_k)$ are unknown.
- Training data: $\mathcal{D}_N = \Big\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_N, y_N)\Big\}$
- Choose two probabilistic models:
  - $\hat{p}_\lambda(\omega_k)$ to approximate $\Pr(\omega_k)$
  - $\hat{p}_{\theta_k}(\mathbf{x})$ to approximate $p(\mathbf{x} \mid \omega_k) \quad (\forall k = 1, 2, \cdots, K)$
- Parameter estimation: estimate $\{\lambda, \theta_1, \cdots, \theta_K\}$ using $\mathcal{D}_N$
- The optimal MAP rule:

$$\omega^* = \arg\max_k \; \Pr(\omega_k) \cdot p(\mathbf{x}|\omega_k)$$

- The Plug-in MAP decision rule:

$$\omega^* = \arg\max_k \; \hat{p}_\lambda(\omega_k) \cdot \hat{p}_{\theta_k}(\mathbf{x})$$

Data modeling



Useful generative models (I)

- A proper generative model must be chosen based on the nature of observation data (the underlying structure of data).
- Some useful generative models for a variety of data types:
  - Normal (Gaussian) distribution

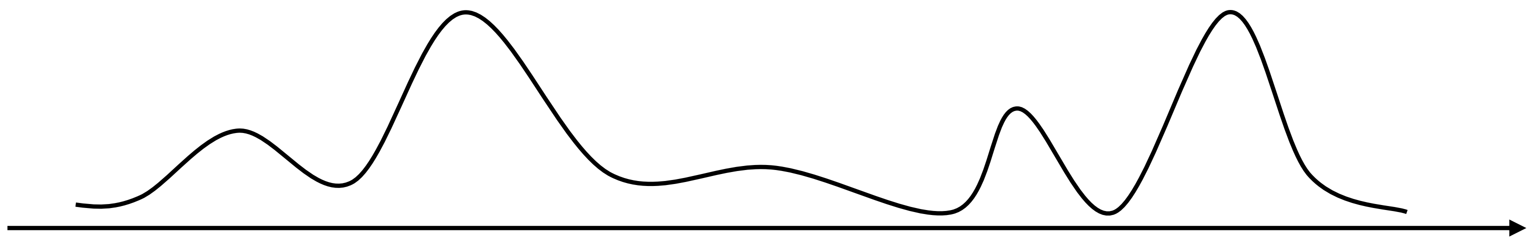
    ➔ uni-modal continuous feature scalars
  - Multivariate normal (Gaussian) distribution

    ➔ uni-modal continuous feature vectors
  - Gaussian Mixture models (GMM)

    ➔ continuous feature scalars/vectors with multi-modal distribution nature

    ➔ For speaker recognition/verification

    distribution of speech features over a large population



Useful generative models (II)

- Some useful generative models (cont'd)
  - Markov chain model: discrete sequential data
    - N-gram model in language modeling
  - Hidden Markov Models (HMM): ideal for various kinds of sequential observation data; provides better modeling capability than simple Markov chain model.
    - Model speech signals for recognition (one of the most successful story of data modeling)
    - Model language/text data for part-of-speech tagging, shallow language understanding, etc.
    - Model biological data (DNA & protein sequence): profile HMM.
    - Lots of other application domains.

Useful generative models (III)

- Some useful generative models (cont'd)
  - Markov Random Field (a.k.a. undirected graphical model):
    - multi-dimensional spatial data
    - Conditional random fields (CRF)
  - Bayesian networks (a.k.a. directed graphical model)
    - High-dimensional data (discrete or continuous)
    - Latent Dirichlet allocation (LDA)
    - Automatically learn dependency from data