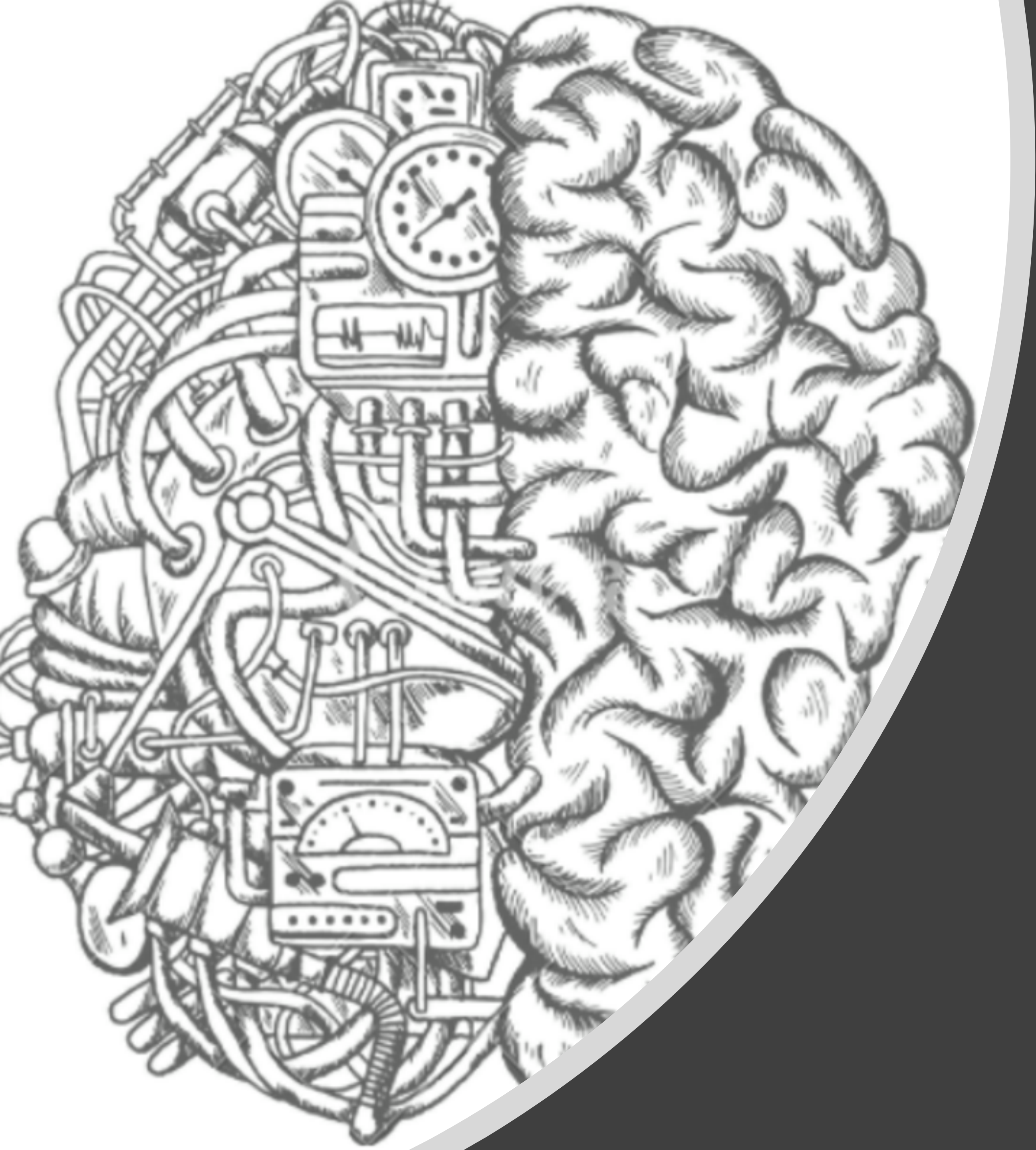


CS 559: Machine Learning Fundamentals and Applications

Lecture 3

Lecturer: Xinchao Wang

xinchao.wang@stevens.edu

Teaching Assistant: Yiding Yang

yyang99@stevens.edu

Overview

- Making Decisions
- Parameter Estimation
  - Frequentist or Maximum Likelihood approach

Expected Utility Example

- Utility: something you want to maximize
- You are asked if you wish to take a bet on the outcome of tossing a fair coin.
  - If you bet and win, you gain \$100. -> U(win, bet) = 100
  - If you bet and lose, you lose \$200. -> U(lose, bet) = -200
  - If you don't bet, the cost to you is zero. -> U(win, no bet) = 0, U(lose, no bet) = 0
- Your expected winnings/losses are:
  - $U(\text{bet}) = p(\text{win}) \times U(\text{win, bet}) + p(\text{lose}) \times U(\text{lose, bet}) = 0.5 \times 100 - 0.5 \times 200 = -50$
    $U(\text{no bet}) = 0$
- Based on making the decision which maximizes expected utility, you would therefore be advised not to bet.

Bayesian Decision Theory

Bayes' Rule

$$\underbrace{P(\omega_j \mid \mathbf{x})}_{\text{posterior}} = \frac{\overbrace{P(\omega_j)}^{\text{prior}}\,\overbrace{p(\mathbf{x} \mid \omega_j)}^{\text{likelihood}}}{\underbrace{p(\mathbf{x})}_{\text{evidence}}}$$

$$P(\omega_j = 0) + P(\omega_j = 1) = 1$$

$$p(\mathbf{x}) = p(\mathbf{x} \mid \omega_j = 1)P(\omega_j = 1) + p(\mathbf{x} \mid \omega_j = 0)P(\omega_j = 0)$$

$$p(\omega_j = 0 \mid \mathbf{x}) + p(\omega_j = 1 \mid \mathbf{x}) = 1$$

Bayes Rule - Intuition

- The essence of the Bayesian approach is to provide a mathematical rule explaining how you should **change your existing beliefs** in the light of **new evidence**.

- In other words, it allows scientists to combine **new data** with their **existing knowledge or expertise**.

From the Economist (2000)

Prior

- Prior comes from **prior knowledge**, no data have been seen yet
- If there is a reliable source of prior knowledge, it should be used
- Some problems cannot even be solved reliably without a good prior
- However prior alone is not enough, we still need likelihood

Decision Rule based on Priors

- Model state of nature (or “categories”) as a random variable, ω:
  - Possible outcomes $\{\omega_1, \omega_2\}$
  - $\omega = \omega_1$: the event that the next sample is from category 1
  - $P(\omega_1)$ = probability of category 1
  - $P(\omega_2)$ = probability of category 2
  - $P(\omega_1) + P(\omega_2) = 1$
- If all incorrect classifications have an equal cost:
  - Decide ω_1 if $P(\omega_1) > P(\omega_2)$; otherwise, decide ω_2

Posterior, Likelihood, Evidence

$$p(\omega_j \mid \mathbf{x}) = \frac{p(\omega_j)\, p(\mathbf{x} \mid \omega_j)}{p(\mathbf{x})}$$

- In the case of two categories

$$P(x) = \sum_{j=1}^{j=2} P(x \mid \omega_j) P(\omega_j)$$

- Posterior = (Likelihood * Prior) / Evidence

Decision using Posteriors

- Decision given the posterior probabilities

  X is an observation (therefore given and fixed) for which:

  if $P(\omega_1 \mid x) > P(\omega_2 \mid x)$ True state of nature = ω_1
  if $P(\omega_1 \mid x) < P(\omega_2 \mid x)$ True state of nature = ω_2

  Therefore:

  whenever we observe a particular x, the probability of error is :

  $P(error \mid x) = P(\omega_1 \mid x)$ if we decide ω_2
  $P(error \mid x) = P(\omega_2 \mid x)$ if we decide ω_1

Probability of Error

- Minimizing the probability of error

- Decide ω_1 if $P(\omega_1|x) > P(\omega_2|x)$;
  otherwise decide ω_2

  Therefore:

$$P(\text{error} \mid x) = \min\,[P(\omega_1 \mid x), P(\omega_2 \mid x)]$$

(Bayes decision)

Decision-Theoretic Classification

$\omega \in \Omega$: unknown class or category, finite set of options

$x \in \mathbb{X}$: observed data, can take values in any space

$a \in A$: action to choose one of the categories (or possibly to reject data)

$L(\omega, a)$: loss of action a given true class ω

Loss Function

- Loss: something you want to minimize
- The loss function states how costly each action taken is
  - Opposite of Utility function: L = - U
- Most common choice is the 0-1 loss

$$L(y, a) = \mathbb{I}(y \neq a) = \begin{cases} 0 & \text{if } a = y \\ 1 & \text{if } a \neq y \end{cases}$$

- In regression, square loss is the most common choice

$$L(y^{true}, y^{pred}) = (y^{true} - y^{pred})^2$$

More General Loss Function

- Allowing actions other than **classification**, which primarily allows the possibility of rejection

- Refusing to make a decision in close or bad cases!

- The loss function still states how costly each action taken is

Notation

- Let $\{\omega_1, \omega_2,\ldots, \omega_c\}$ be the set of c states of nature (or "categories")
- Let $\{\alpha_1, \alpha_2,\ldots, \alpha_a\}$ be the set of possible actions
- Let $\lambda(\alpha_i|\omega_j)$ be the loss incurred for taking action α_i when the state of nature is ω_j

Conditional Risk

- Conditional Risk $R(\alpha_i|x)$
  - Risk of taking action α_i given data X

- $$R(\alpha_i \mid x) = \sum_{j=1}^{j=c} \lambda(\alpha_i \mid \omega_j) P(\omega_j \mid x)$$

  for i = 1,…,a

Conditional Risk

- Two-category classification

α_1 : decide ω_1

α_2 : decide ω_2

$\lambda_{ij} = \lambda(\alpha_i \mid \omega_j)$

loss incurred for deciding ω_i when the true state of nature is ω_j

Conditional risk:

$$R(\alpha_i \mid x) = \sum_{j=1}^{j=c} \lambda(\alpha_i \mid \omega_j) P(\omega_j \mid x)$$

for i = 1,…,a

$R(\alpha_1 \mid x) = \lambda_{11}P(\omega_1 \mid x) + \lambda_{12}P(\omega_2 \mid x)$

$R(\alpha_2 \mid x) = \lambda_{21}P(\omega_1 \mid x) + \lambda_{22}P(\omega_2 \mid x)$

Decision Rule

Our rule is the following:

if $R(\alpha_1 \mid x) < R(\alpha_2 \mid x)$

action α_1: decide ω_1

Recall that:

$R(\alpha_1 \mid x) = \lambda_{11}P(\omega_1 \mid x) + \lambda_{12}P(\omega_2 \mid x)$

$R(\alpha_2 \mid x) = \lambda_{21}P(\omega_1 \mid x) + \lambda_{22}P(\omega_2 \mid x)$

Therefore, we decide ω_1 if:

$(\lambda_{21} - \lambda_{11})\, P(x \mid \omega_1)\, P(\omega_1) > (\lambda_{12} - \lambda_{22})\, P(x \mid \omega_2)\, P(\omega_2)$

and decide ω_2 otherwise

Likelihood Ratio

- The decision rule is equivalent to the following rule:

$$if\ \frac{P(x|\omega_1)}{P(x|\omega_2)} > \frac{\lambda_{12}-\lambda_{22}}{\lambda_{21}-\lambda_{11}} \cdot \frac{P(\omega_2)}{P(\omega_1)}$$

Then take action α_1 (decide ω_1)

Otherwise take action α_2 (decide ω_2)

Minimum-Error-Rate Classification

- Actions are decisions on classes

  If action α_i is taken and the true state of nature is ω_j then:

  the decision is correct if $i = j$ and in error if $i \neq j$

- Seek a decision rule that minimizes the probability of error which is called the error rate

The Zero-One Loss Function

- Zero-One Loss Function

$$\lambda(\alpha_i,\omega_j)=\begin{cases}0 & i=j\\ 1 & i\neq j\end{cases} \qquad i,j=1,\dots,c$$

- Therefore, the conditional risk is

$$R(\alpha_i \mid x)=\sum_{j=1}^{j=c}\lambda(\alpha_i \mid \omega_j)P(\omega_j \mid x)$$

$$=\sum_{j\neq i}P(\omega_j \mid x)=1-P(\omega_i \mid x)$$

- The risk corresponding to this loss function is the average probability of error

Minimum Error Rate Decision Rule

- Minimizing the risk requires maximizing $P(\omega_i \mid x)$
  since $R(\alpha_i \mid x) = 1 - P(\omega_i \mid x)$

- For Minimum error rate
  - Decide ω_i if $P(\omega_i \mid x) > P(\omega_j \mid x) \; \forall j \neq i$

Minimum Error Rate Decision Rule

- Given the likelihood ratio and the zero-one loss function:

$$\textit{Let } \frac{\lambda_{12}-\lambda_{22}}{\lambda_{21}-\lambda_{11}} \cdot \frac{P(\omega_2)}{P(\omega_1)} = \theta_\lambda \textit{ then decide } \omega_1 \textit{ if : } \frac{P(x\,|\,\omega_1)}{P(x\,|\,\omega_2)} > \theta_\lambda$$

- If λ is the zero-one loss function which means:

$$\lambda = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$\textit{then } \theta_\lambda = \frac{P(\omega_2)}{P(\omega_1)} = \theta_a$$

$$\textit{if } \lambda = \begin{pmatrix} 0 & 2 \\ 1 & 0 \end{pmatrix} \textit{ then } \theta_\lambda = \frac{2P(\omega_2)}{P(\omega_1)} = \theta_b$$

Classifiers, Discriminant Functions and Decision Surfaces

- The multi-category case

  - Set of discriminant functions $g_i(x)$, $i = 1,\ldots, c$

  - The classifier assigns a feature vector x to class ω_i if:

$$g_i(x) > g_j(x) \ \forall j \neq i$$

Max Discriminant Functions

- Let $g_i(x) = - R(\alpha_i \mid x)$
  (max. discriminant corresponds to min. risk)

- For the minimum error rate, we take $g_i(x) = P(\omega_i \mid x)$

  (max. discriminant corresponds to max. posterior)

$$g_i(x) \equiv P(x \mid \omega_i)\, P(\omega_i)$$

$$g_i(x) = \ln P(x \mid \omega_i) + \ln P(\omega_i)$$

(ln: natural logarithm)

Decision Regions

- Feature space divided into c decision regions

$$\text{if } g_i(x) > g_j(x)\ \forall j \neq i \text{ then x is in } \mathcal{R}_i$$

$$(\mathcal{R}_i \text{ means assign x to } \omega_i)$$

- The two-category case
  - A classifier has two discriminant functions g_1 and g_2

Let $g(x) \equiv g_1(x) - g_2(x)$

Decide ω_1 if $g(x) > 0$; Otherwise decide ω_2

Computation of g(x)

$$g(x) = P(\omega_1 \mid x) - P(\omega_2 \mid x)$$

$$g(x) = \ln \frac{P(x \mid \omega_1)}{P(x \mid \omega_2)} + \ln \frac{P(\omega_1)}{P(\omega_2)}$$

Discriminant Functions for the Normal Density

- Minimum error-rate classification can be achieved by the discriminant function

$g_i(x) = \ln P(x \mid \omega_i) + \ln P(\omega_i)$

- Case of multivariate normal

$$g_i(x) = -\frac{1}{2}(x-\mu_i)^t \sum_i^{-1}(x-\mu_i) - \frac{d}{2}\ln 2\pi - \frac{1}{2}\ln|\Sigma_i| + \ln P(\omega_i)$$

- Case $\Sigma_i = \sigma^2 I$ (I is the identity matrix)

$$g_i(x) = w_i^t x + w_{i0} \text{ (linear discriminant function)}$$

where:

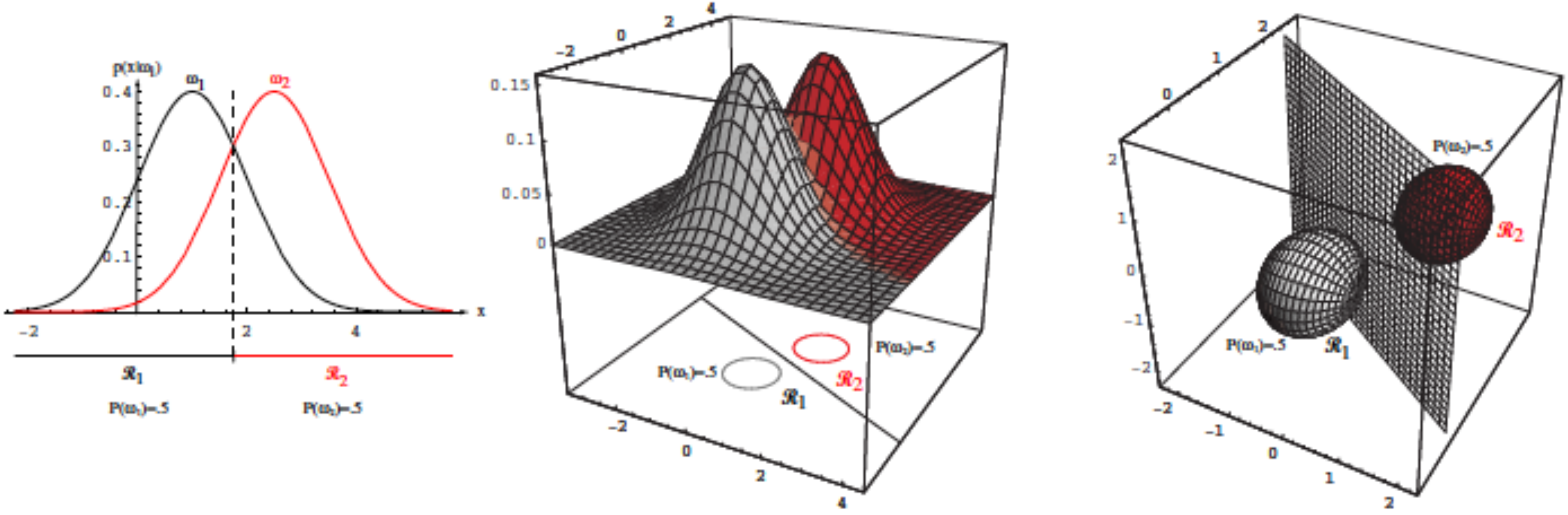
$$w_i = \frac{\mu_i}{\sigma^2};\; w_{i0} = -\frac{1}{2\sigma^2}\mu_i^t \mu_i + \ln P(\omega_i)$$

(w_{i0} is called the threshold for the ***i***th category)

Linear Machines

- A classifier that uses linear discriminant functions is called a "linear machine"

- The decision surfaces for a linear machine are pieces of hyperplanes defined by: $g_i(x) = g_j(x)$



p(x|ω_i)
ω_1
ω_2
x
ℛ_1
ℛ_2
P(ω_1)=.5
P(ω_2)=.5

- The hyperplane separating $\mathcal{R}_i$ and $\mathcal{R}_j$

$$g_i(x) = w_i^t x + w_{i0} \text{ and } g_j(x) = w_j^t x + w_{j0}$$

$$\text{Decision boundary}: g_i(x) = g_j(x)$$

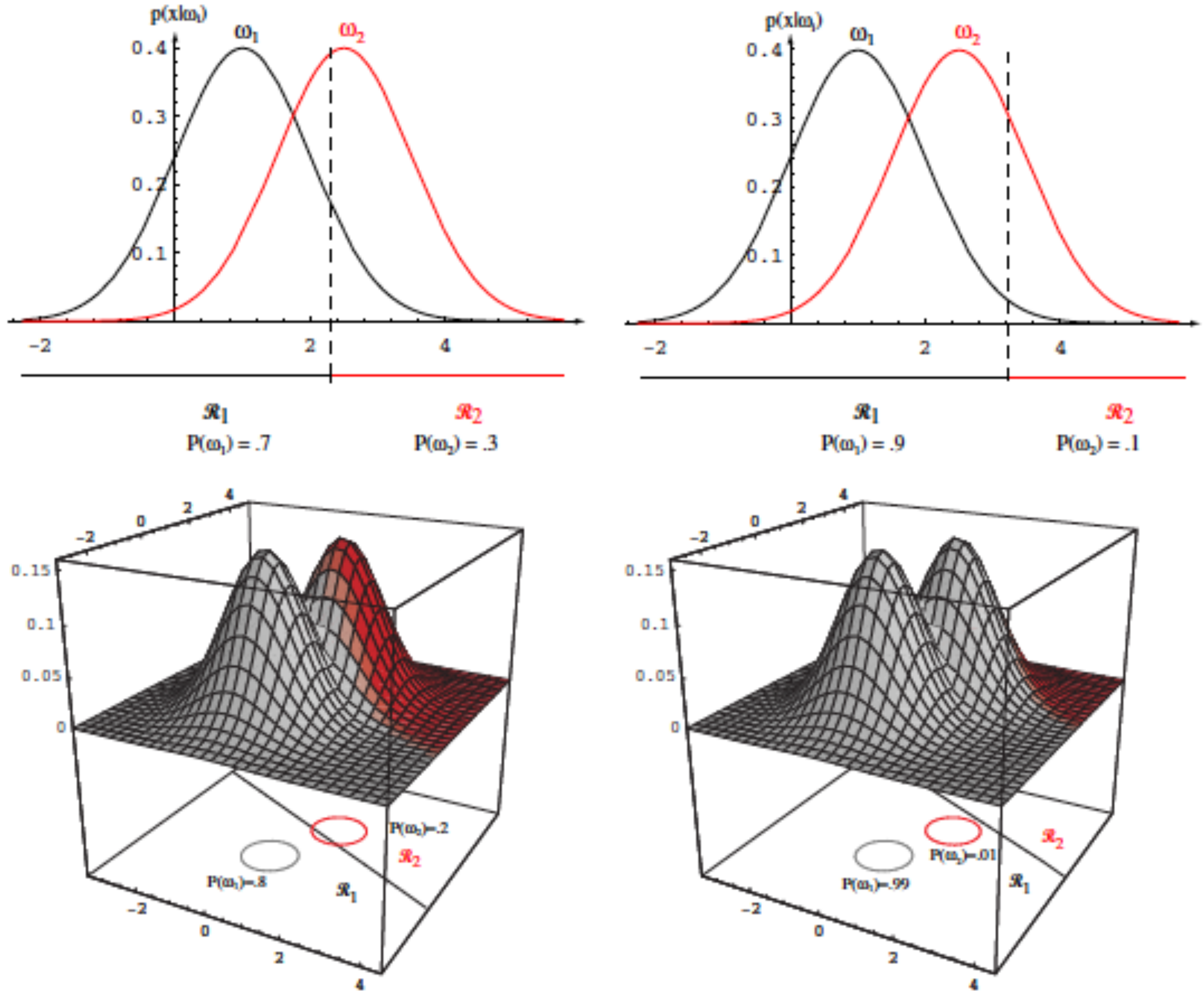
$$w^t(x - x_0) = 0$$

$$w = \mu_i - \mu_j$$

$$x_0 = \frac{1}{2}(\mu_i + \mu_j) - \frac{\sigma^2}{\|\mu_i - \mu_j\|^2} \ln \frac{P(\omega_i)}{P(\omega_j)} (\mu_i - \mu_j)$$

always orthogonal to the line linking the means

$$\textit{if } P(\omega_i) = P(\omega_j) \textit{ then } x_0 = \frac{1}{2}(\mu_i + \mu_j)$$



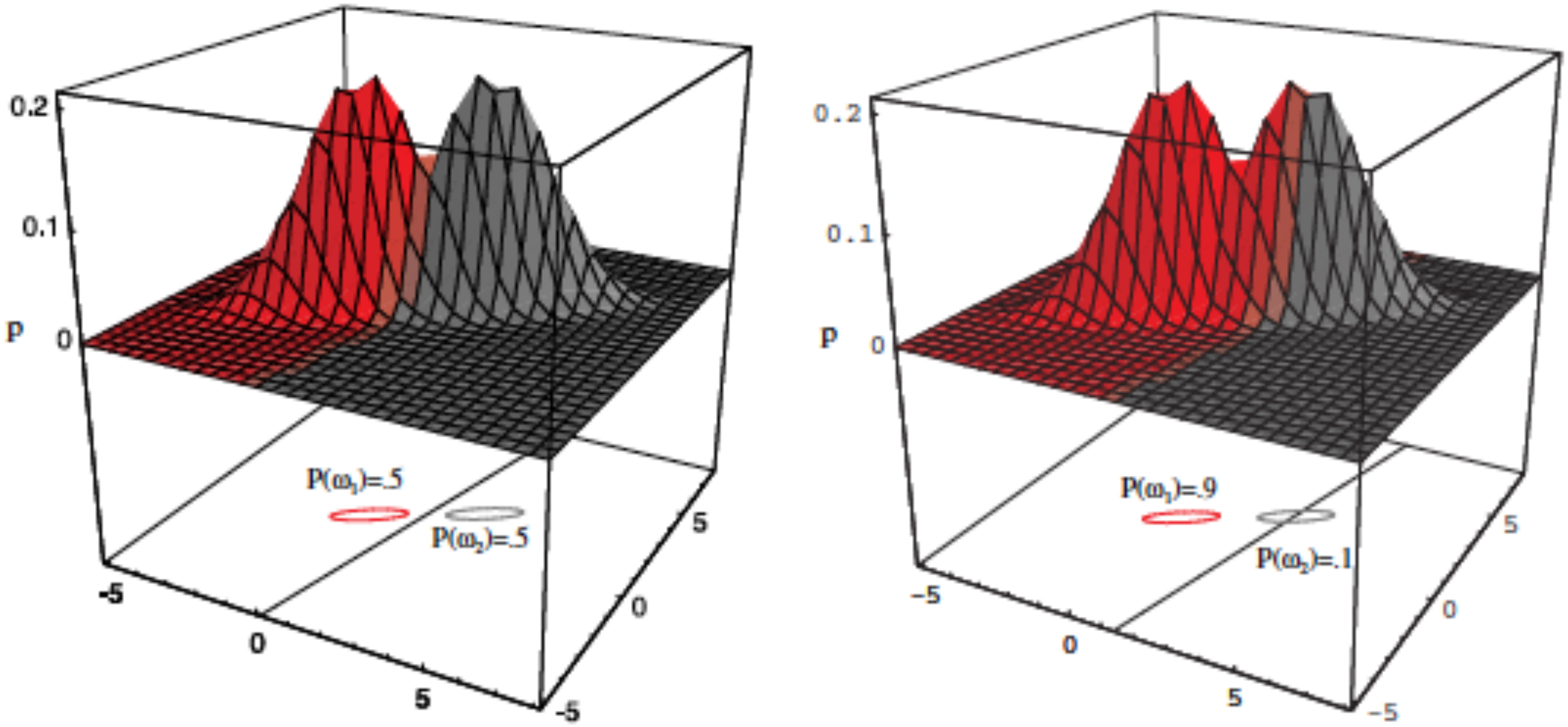
p(x|ωi)
ω1
ω2
ℛ1
P(ω1) = .7
ℛ2
P(ω2) = .3
ℛ1
P(ω1) = .9
ℛ2
P(ω2) = .1
P(ω1)=.8
P(ω2)=.2
P(ω1)=.99
P(ω2)=.01

- Case $\Sigma_i = \Sigma$ (covariances of all classes are identical but arbitrary!)
  - Hyperplane separating $\mathcal{R}_i$ and $\mathcal{R}_j$.

$$w_i = \Sigma^{-1}\mu_i$$

$$x_0 = \frac{1}{2}(\mu_i + \mu_j) - \frac{\ln\left[P(\omega_i)/P(\omega_j)\right]}{(\mu_i - \mu_j)^t\Sigma^{-1}(\mu_i - \mu_j)}.(\mu_i - \mu_j)$$

  - The hyperplane separating $\mathcal{R}_i$ and $\mathcal{R}_j$ is generally not orthogonal to the line between the means.


0.2
0.1
P
0
P(ω1)=.5
P(ω2)=.5
-5
0
5
5
0
-5
0.2
0.1
P
0
P(ω1)=.9
P(ω2)=.1
-5
0
5
5
0
-5

- Case Σ_i =arbitrary
  - The covariance matrices are different for each category

$$g_i(x) = x^t W_i x + w_i^t x + w_{i0}$$

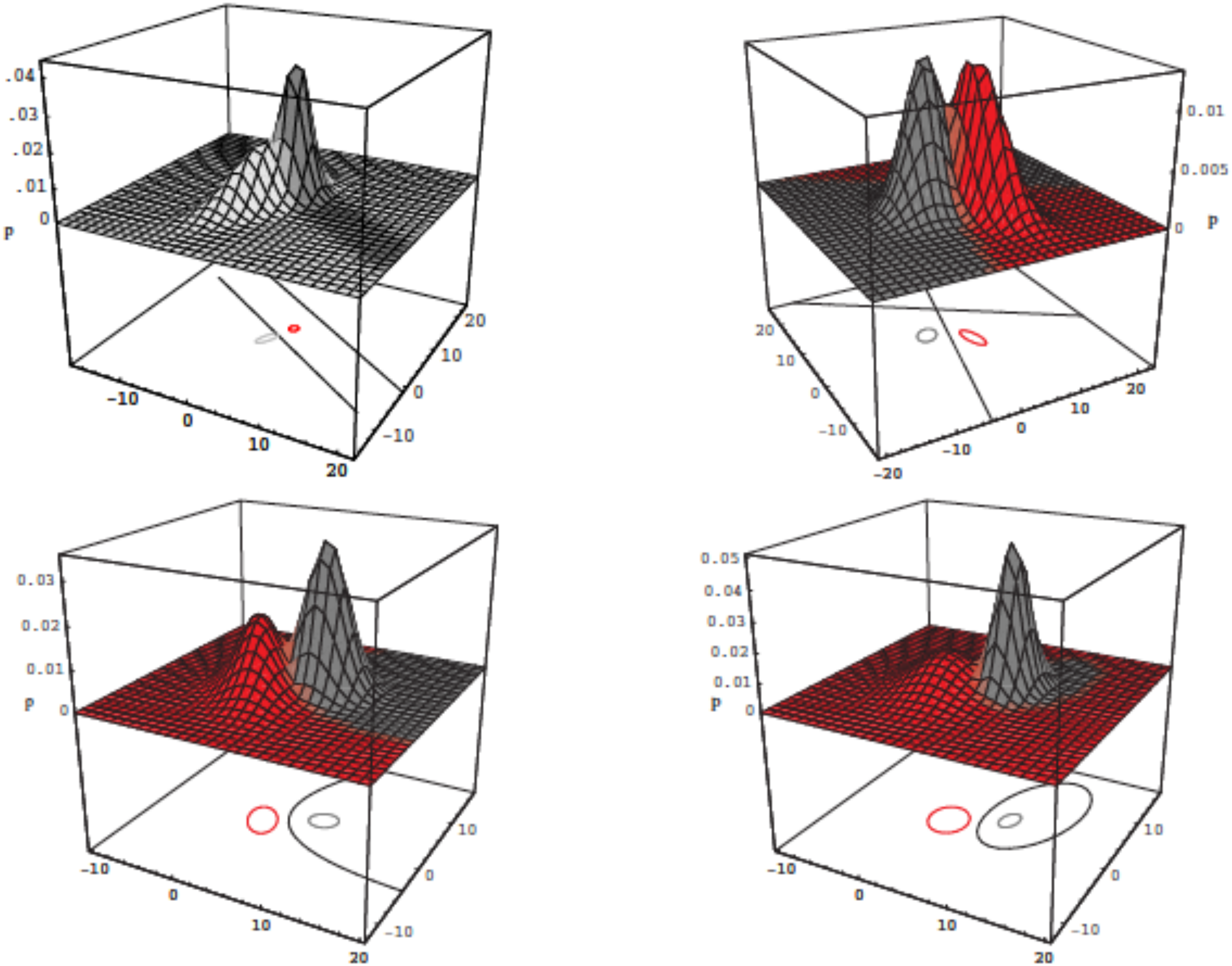
where :

$$W_i = -\frac{1}{2}\Sigma_i^{-1}$$

$$w_i = \Sigma_i^{-1}\mu_i$$

$$w_{i0} = -\frac{1}{2}\mu_i^t\Sigma_i^{-1}\mu_i - \frac{1}{2}\ln|\Sigma_i| + \ln P(\omega_i)$$

  - Hyperquadrics which are: hyperplanes, pairs of hyperplanes, hyperspheres, etc.


.04
.03
.02
.01
0
P
-10
0
10
20
20
10
0
-10
0.01
0.005
0
P
20
10
0
-10
-20
-10
0
10
20
0.03
0.02
0.01
P
0
-10
0
10
20
10
0
-10
0.05
0.04
0.03
0.02
0.01
P
0
-10
0
10
20
10
0
-10

Bayes Decision Theory – Discrete Features

- Components of x are binary or integer valued, x can take only one of m discrete values

$$v1, v2, \ldots, vm$$

- Case of independent binary features in 2 category problem
- Let $x = [x_1, x_2, \ldots, x_d]^t$ where each x_i is either 0 or 1, with probabilities:

$$p_i = P(x_i = 1 \mid \omega_1)$$

$$q_i = P(x_i = 1 \mid \omega_2)$$

$$P(x \mid \omega_1) = \prod_{i=1}^{d} p_i^{x_i} (1 - p_i)^{1-x_i}$$

$$P(x \mid \omega_2) = \prod_{i=1}^{d} q_i^{x_i} (1 - q_i)^{1-x_i}$$

Likelihood ratio :

$$\frac{P(x \mid \omega_1)}{P(x \mid \omega_2)} = \prod_{i=1}^{d} \left(\frac{p_i}{q_i}\right)^{x_i} \left(\frac{1 - p_i}{1 - q_i}\right)^{1-x_i}$$

Discriminant function :

$$\mathrm{g(x)} = \sum_{i=1}^{d} \left[x_i \ln \frac{p_i}{q_i} + (1 - x_i) \ln \frac{1 - p_i}{1 - q_i} \right] + \ln \frac{P(\omega_1)}{P(\omega_2)}$$

$$g(x) = \sum_{i=1}^{d} w_i x_i + w_0$$

$$where:$$

$$w_{\mathrm{i}} = \ln \frac{p_i(1-q_i)}{q_i(1-p_i)} \qquad i = 1,\ldots,d$$

$$and:$$

$$w_0 = \sum_{i=1}^{d} \ln \frac{1-p_i}{1-q_i} + \ln \frac{P(\omega_1)}{P(\omega_2)}$$

$$decide\ \omega_1 \text{ if } \mathrm{g(x)} > 0 \text{ and } \omega_2 \text{ if } \mathrm{g(x)} \leq 0$$

Maximum-Likelihood & Bayesian Parameter Estimation

Adapted from:
Duda, Hart and Stork, Pattern Classification textbook

O. Veksler

E. Sudderth
D. Batra

Introduction

- We could design an optimal classifier if we knew:
  - $p(\omega_i)$ (priors)
  - $p(x \mid \omega_i)$ (class-conditional densities)
  - Unfortunately, we rarely have this complete information!

- Design a classifier from training data

Supervised Learning in a Nutshell

- Training Stage:
  - Raw Data → x (Feature Extraction)
  - Training Data { (x,y) } → f (Learning)

- Testing Stage
  - Raw Data → x (Feature Extraction)
  - Test Data x → f(x) (Apply function, Evaluate error)

Statistical Estimation View

- Probabilities to the rescue:
  - x and y are *random variables*
  - $D = (x_1,y_1), (x_2,y_2), \ldots, (x_N,y_N) \sim P(X,Y)$

- IID: Independent Identically Distributed
  - Both training & testing data sampled IID from P(X,Y)
  - Learn on training set
  - Have some hope of *generalizing* to test set

Parameter Estimation

- Use a priori information about the problem

- E.g.: Normality of $p(x \mid \omega_i)$

$$p(x \mid \omega_i) \sim N(\mu_i, \Sigma_i)$$

- Simplify problem
  - From estimating unknown distribution function
  - To estimating parameters

Why Gaussians?

- Why does the entire world seem to always be harping on about Gaussians?
  - Central Limit Theorem!
  - They’re easy (and we like easy)
  - Closely related to squared loss (for regression)
  - Mixture of Gaussians is sufficient to approximate many distributions

Some properties of Gaussians

- Affine transformation
  - multiplying by scalar and adding a constant
  - $X \sim N(\mu,\sigma^2)$
  - $Y = aX + b \rightarrow Y \sim N(a\mu+b,a^2\sigma^2)$

- Sum of Independent Gaussians
  - $X \sim N(\mu_X,\sigma^2_X)$
  - $Y \sim N(\mu_Y,\sigma^2_Y)$
  - $Z = X+Y \quad \rightarrow Z \sim N(\mu_X+\mu_Y, \sigma^2_X+\sigma^2_Y)$

Estimation techniques

- Maximum-Likelihood (ML) and Bayesian estimation
- Results are often identical, but the approaches are fundamentally different

- Frequentist View
  - limit $N \rightarrow \infty$ #(A is true)/N
  - limiting frequency of a repeating non-deterministic event

- Bayesian View
  - P(A) is your “belief” about A

Parameter Estimation

- Parameters in ML estimation are fixed but unknown!
- Best parameters are obtained by maximizing the probability of obtaining the samples observed
- Bayesian methods view the parameters as random variables having some known distribution
- In either approach, we use $p(\omega_i \mid x)$ for our classification rule

Independence Across Classes

- For each class ω_i we have a proposed density $p_i(x| \omega_i)$ with unknown parameters θ_i which we need to estimate
- Since we assumed independence of data across the classes, estimation is an identical procedure for all classes
- To simplify notation, we drop sub-indexes and say that we need to estimate parameters θ for density $\mathbf{p(x)}$

Maximum-Likelihood Estimation

- General principle
  - Assume **c** datasets (classes) D_1, D_2, …, D_c drawn independently according to $p(x|\ \omega_j)$

Maximum-Likelihood Estimation

- Assume that $p(x|\ \omega_j)$ has known parametric form determined by parameter vector θ_j
- Further assume that D_i gives no information about θ_j if $i \neq j$
  - Drop subscripts in remainder

Likelihood

- Use set of independent samples to estimate $p(D \mid \theta)$
  - Let $D = \{x_1, x_2, \dots, x_n\}$
  - $p(x_1, \dots, x_n \mid \theta) = \prod_{i=1}^{n} p(xi \mid \theta)$; $|D| = n$

  Our goal is to determine:
  value of θ that best agrees with observed training data

  ^

- Note if D is fixed $p(D \mid \theta)$ is not a density

Example: Gaussian case

- Assume we have c classes and
  $p(x \mid \omega_j) \sim N(\mu_j, \Sigma_j)$
  $p(x \mid \omega_j) \equiv p(x \mid \omega_j, \theta_j)$ where:

$$\theta = (\mu_j, \Sigma_j) = (\mu_j^1, \mu_j^2, \ldots, \sigma_j^{11}, \sigma_j^{22}, \mathrm{cov}(x_j^m, x_j^n) \ldots)$$

- Use the information provided by the training samples to estimate

  $\theta = (\theta_1, \theta_2, \ldots, \theta_c)$, each θ_i ($i = 1, 2, \ldots, c$) is associated with each category

- Suppose that D contains n samples, $x_1, x_2, \ldots, x_n$

$$p(D \mid \theta) = \prod_{k=1}^{k=n} p(x_k \mid \theta)$$

- $p(D \mid \theta)$ is called the likelihood of θ w.r.t the set of samples
- ML estimate of θ is, by definition the value $\hat{\theta}$ that maximizes $p(D \mid \theta)$

  "It is the value of θ that best agrees with the actually observed training sample"

- Optimal estimation
  - Let $\theta = (\theta_1, \theta_2, \ldots, \theta_p)^t$ and let ∇_θ be the gradient operator

$$\nabla_\theta = \left[\frac{\partial}{\partial\theta_1}, \frac{\partial}{\partial\theta_2}, \ldots, \frac{\partial}{\partial\theta_p}\right]^t$$

  - We define $l(\theta)$ as the log-likelihood function

$$l(\theta) = \ln p(D \mid \theta)$$

  - New problem statement:
    determine θ that maximizes the log-likelihood

$$\hat{\theta} = \arg\max_{\theta} l(\theta)$$

Necessary conditions for an optimum:

$$\nabla_\theta l = \sum_{k=1}^{k=n} \nabla_\theta \ln p(x_k \mid \theta)$$

$$\nabla_\theta l = 0$$

- Local or global maximum
- Local or global minimum
- Saddle point

Example of ML estimation: unknown μ

- $p(x_i \mid \mu) \sim N(\mu, \Sigma)$
  (Samples are drawn from a multivariate normal population)

$$\ln p(x_k \mid \mu) = -\frac{1}{2}\ln\left[(2\pi)^d |\Sigma|\right] - \frac{1}{2}(x_k - \mu)^t \Sigma^{-1}(x_k - \mu)$$

$$\nabla_\theta \ln p(x_k \mid \mu) = \Sigma^{-1}(x_k - \mu)$$

$\theta = \mu$ therefore:

- The ML estimate for μ must satisfy:

$$\sum_{k=1}^{k=n} \Sigma^{-1}(x_k - \hat{\mu}) = 0$$

- Multiplying by Σ and rearranging, we obtain:

$$\hat{\mu} = \frac{1}{n}\sum_{k=1}^{k=n} x_k$$

Just the arithmetic average of the samples of the training samples!

Conclusion:

If $p(x_k \mid \omega_j)$ ($j = 1, 2, \ldots, c$) is assumed to be Gaussian in a *d*-dimensional feature space, then we can estimate the vector

$\theta = (\theta_1, \theta_2, \ldots, \theta_c)^t$ and perform optimal classification!

- Example of ML estimation: unknown μ *and* *σ* (univariate)

$$\theta = (\theta_1, \theta_2) = (\mu, \sigma^2)$$

$$l = \ln p(x_k \mid \theta) = -\frac{1}{2}\ln 2\pi\theta_2 - \frac{1}{2\theta_2}(x_k - \theta_1)^2$$

$$\nabla_\theta l = \begin{pmatrix} \dfrac{\partial}{\partial\theta_1}(\ln p(x_k \mid \theta)) \\ \dfrac{\partial}{\partial\theta_2}(\ln p(x_k \mid \theta)) \end{pmatrix} = 0$$

$$\begin{cases} \dfrac{1}{\theta_2}(x_k - \theta_1) = 0 \\ -\dfrac{1}{2\theta_2} + \dfrac{(x_k - \theta_1)^2}{2\theta_2^2} = 0 \end{cases}$$

Summation:

$$\begin{cases} \displaystyle\sum_{k=1}^{k=n} \frac{1}{\hat{\theta}_2}(x_k - \theta_1) = 0 & (1) \\ \displaystyle -\sum_{k=1}^{k=n} \frac{1}{\hat{\theta}_2} + \sum_{k=1}^{k=n} \frac{(x_k - \hat{\theta}_1)^2}{\hat{\theta}_2^2} = 0 & (2) \end{cases}$$

Combining (1) and (2), one obtains:

$$\mu = \sum_{k=1}^{k=n} \frac{x_k}{n} \qquad \sigma^2 = \frac{\displaystyle\sum_{k=1}^{k=n} (x_k - \mu)^2}{n}$$

Bias

- ML estimate for σ^2 is biased

$$E\left[\frac{1}{n}\Sigma(x_i-\bar{x})^2\right]=\frac{n-1}{n}\sigma^2\neq\sigma^2$$

- For one sample, the estimated variance is always zero => under-estimate
- An elementary unbiased estimator for Σ is:

$$\underbrace{C=\frac{1}{n-1}\sum_{k=1}^{k=n}(x_k-\mu)(x_k-\hat{\mu})^t}_{\text{Sample covariance matrix}}$$

- Ultimately, interested in estimate that maximizes classification performance

Model Error

- What if we assume class distribution to be $N(\mu,1)$, but true distribution is $N(\mu,10)$?
  - ML estimate: $\theta=\mu$ is the correct mean
- Will this θ result in best classifier performance?
  - NO