Lecture 3: Bayesian Decision Theory

Prof. Krishna R. Pattipati
Dept. of Electrical and Computer Engineering
University of Connecticut
Contact: krishna@engr.uconn.edu (860) 486-2890

Fall 2018
September 24, 2018

Lecture Outline

- Bayesian Approach to Decision making
- Classification Problem
- Bayesian Decision Theory for Classification
  - General Case with Reject (Unknown) Option
  - Gaussian Case
- Error Bounds
  - Chernoff and Bhattacharyya Bounds
- Extensions
  - Binary Features, Missing and Noisy Features

Reading List

- Section 1.5 of Bishop
- Section 5.7 of Murphy
- Chapter 2 of Duda, Hart and Sorkin
- Sections 7.1-7.7 of Theodoridis
- Lecture Notes

Supervised Learning Systems

Input	Output	Application
Voice Recording	Transcript	Speech Recognition (C)
Historical Market Data	Future Market Prediction	Trading Bots (R)
Photograph	Caption	Image Tagging (C)
Drug Chemical Properties	Treatment Efficiency	Pharma R&D (R)
Stored Transaction Details	Is the Transaction Fraudulent?	Fraud Detection (C)
Recipe Ingredients	Customer Reviews	Food Recommendations (C)
Purchase Histories	Future Purchase Behavior	Customer Retention (C/R)
Car Locations and Speed	Traffic Flow	Traffic Lights (R)
Faces	Names	Face Recognition (C)

Erik Brynjolfsson and Andre Mcafee, "The Business of Artificial Intelligence: What It Can – and Cannot - Do for Your Organization" HBR book on *AI and Machine Learning*, July 2018.

Bayesian Approach to Learning and Decision Making

Step 1: **Formulate knowledge about the situation probabilistically**

- *Define a model* that expresses qualitative aspects of our knowledge (e.g., forms of distributions, independence assumptions). *The model will have some unknown parameters*.
- *Specify a prior probability distribution for these unknown parameters* that expresses our beliefs about which values are more or less likely, before seeing the data.

Step 2: *Gather data. Consult experts, conduct surveys, open sources, research*

Step 3: *Compute the posterior probability distribution for the parameters, given the observed data*.

Step 4: *Use this posterior distribution* to:

- *Reach scientific conclusions, properly accounting for uncertainty.*
- *Make predictions by averaging over the posterior distribution.*
- *Make decisions so as to minimize posterior expected loss or maximize expected reward or any suitable objective function.*

Based on Radford Neal's tutorial on ``Bayesian Methods for Machine Learning'')

Classifiers

- Set of objects to be classified into C classes $z = \{1,2,....,C\}$

  (z is a hidden or latent discrete variable or category)
- Each object gives rise to a feature vector $\underline{x} \in R^P$

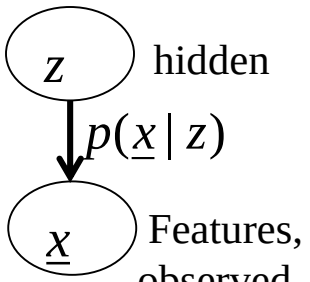
  or some discrete set $\boldsymbol{x_1}$ x $\boldsymbol{x_2}$ x.....x $\boldsymbol{x_p}$

  binary $[0,1]^p$

- A priori probability of each class $\{P(z=i)\}_{i=1}^{C}$ $z \in \{1,2,....,C\}$
- Class conditional probability densities $\{p(\underline{x} \mid z=i)\}_{i=1}^{C}$

- Key Relationship: Bayes Rule

$$P(z=i \mid \underline{x}) = \frac{p(\underline{x} \mid z=i)P(z=i)}{p(\underline{x})} = \frac{(Likelihood)\ (prior)}{Evidence}$$



$$p(\underline{x}) = \sum_{i=1}^{C} p(\underline{x}, z=i) = \sum_{i=1}^{C} p(\underline{x} \mid z=i)P(z=i)$$

Mixture Density

Total Probability Theorem

What are the Features?

"A classification partitions the feature space into decision regions that indicate to which class $\underline{x}$ belongs". Action $\alpha = i \Rightarrow$ Class/hidden variable is estimated as $\hat{z} = i$

- What are the features?

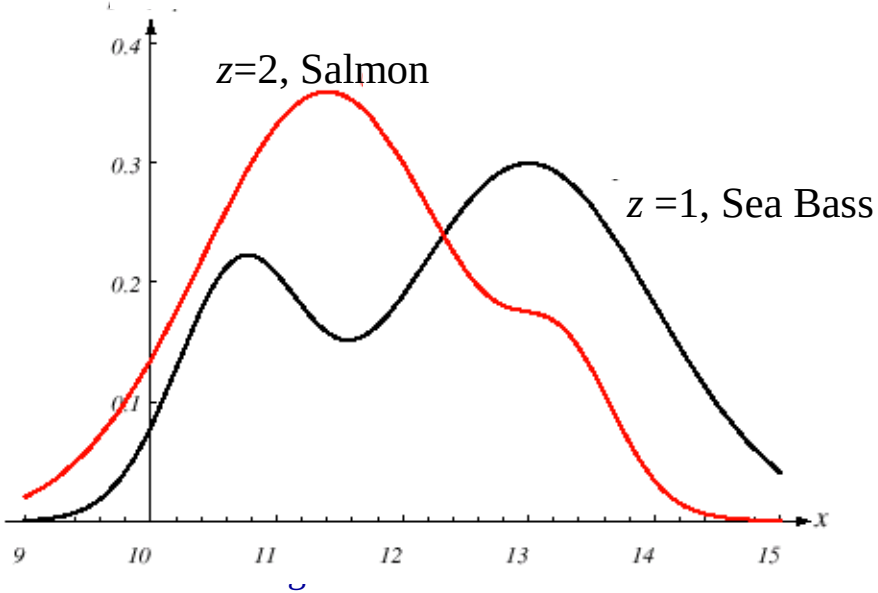
- Contain information to distinguish among classes
- Sensitive to variability in the data
- Small no. of them to enable fast implementation

Illustration of Bayes Rule

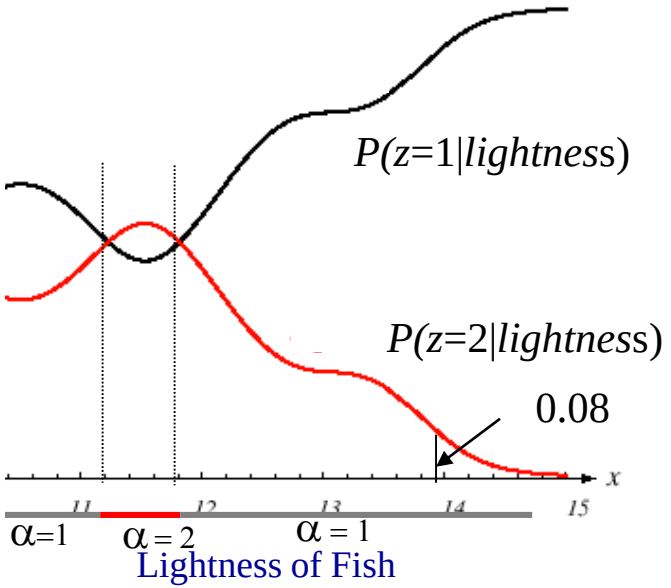
$$P(z=i \mid \underline{x}) = \frac{p(\underline{x} \mid z=i)P(z=i)}{p(\underline{x})} = \frac{(Likelihood)\ (prior)}{Evidence}$$

Likelihood Functions

$p(lightness \mid z)$



Posterior Probabilities
when $P(z=1)=2/3$ and $P(z=2)=1/3$



When x=14, the probability that the fish is category $z=2$, *Salmon* is 0.08

Picking the class with the highest posterior is a special case of class of Bayesian decision rules

Classification Criteria -1

- Costs of misclassifications (Risk or Loss matrix):
  - λ_{ij} = Cost of assigning a pattern $\underline{x}$ to class $\hat{z} = i$ (action $\alpha = i$) when in fact it belongs to class $z = j, j = 1,2,...,C$
  - typically $\lambda_{ii} = 0$ (no penalty for correct decisions)

Criterion 1: Minimize expected cost of misclassifications (ECM)

$$ECM = \sum_{j=1}^{C} P(z = j) \sum_{i=1}^{C} \lambda_{ij} P(\alpha = i \mid z = j)$$

$P(\alpha = i \mid z = j) = \text{prob}\{\text{assigning a pattern to class } i \text{ (action } \alpha = i) \mid \text{actual class is } z = j\}$

Element of confusion matrix

2x2 Contingency Table, Confusion Matrix

Decision/Truth	z=Bad	z=Good
$\hat{z} = \text{Bad}$	True Positive (TP), Hit, Detection, Success, Defines Sensitivity, Power, P_D	False Positive (FP), False Alarm (FA), DefinesType I Error, P_{FA}
$\hat{z} = \text{Good}$	Missed Detection, False Negative (FN), Defines Type II Error, 1-P_D	True Negative (TN), Correct Negative Correct Rejection, Defines Specificity, 1-P_{FA}

$$P_D = P(\hat{z} = Bad \mid z = Bad) = \text{Recall=Sensitivity=} \frac{TP}{TP + FN}$$

$$P_{FA} = P(\hat{z} = Bad \mid z = Good) = \frac{FP}{TN + FP} = 1 - \text{specifcity}$$

$$\text{Precision} = \text{Positive Prediction Value (PPV)} = P(z = Bad \mid \hat{z} = Bad)$$

$$= \frac{P_D P(z = Bad)}{P_D P(z = Bad) + P_{FA} P(z = Good)} = \frac{TP}{TP + FP}$$

$$F = \frac{2 \text{ x Precision x Recall}}{\text{Precision+Recall}} = \frac{2P_D P(z = Bad)}{P_D P(z = Bad) + P(z = Bad) + P_{FA} P(z = Good)}$$

$$= \frac{2TP}{2TP + FN + FP}$$

Classification Criteria - 2

Criterion 2: Minimize the probability of error (PE):

$$PE = \sum_{j=1}^{C} P(z = j)[1 - P(\alpha = j \mid z = j)] \quad \text{since} \sum_{i=1}^{C} P(\alpha = i \mid z = j) = 1$$

- This corresponds to: $\text{PE} = \sum_{j=1}^{C} P(z = j).P(Error \mid z = j)$

- ECM reduces to PE when the loss matrix has zero diagonal elements and the rest ones (*zero-one* or *symmetric loss function*)

$$\lambda_{ij} = 1 - \delta_{ij}$$

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

Kronecker Delta Function

Classification Criteria - 3

Criterion 3: Minimize the Maximum Cost of Misclassifications (*MINIMAX* Criterion)

$$CM_{\min\max} = \min_i \max_j \sum_{i=1}^{C} \lambda_{ij} P(\alpha = i \mid z = j)$$

- Insensitive to prior probabilities

Criterion 4: Neyman-Pearson Criterion

$$\min \sum_{j=1}^{C} P(z = j) \sum_{i=1}^{C} \lambda_{ij} P(\alpha = i \mid z = j)$$

$$s.t. \quad \sum_{i=1}^{C} \lambda_{ik} P(\alpha = i \mid z = k) \le \gamma \quad for\ some\ k$$

- Example: Maximize Detection Probability subject to a constraint on False Alarm Probability

Variant

Augment the decisions to include Action, $\alpha = 0$ (class z=0):

$\alpha = 0 \Rightarrow$ unknown, reject, doubt, cannot decide

$$\boxed{\alpha = \{\alpha = 0,\ \alpha = 1,\ \ldots\ldots \alpha = C\}}$$

λ_r = Reject Cost; λ_e= Cost for Error

$$\Lambda = \begin{bmatrix} \lambda_{01} & \lambda_{02} & . & . & \lambda_{0C} \\ \lambda_{11} & \lambda_{12} & . & . & \lambda_{1C} \\ . & . & . & . & . \\ . & . & . & . & . \\ \lambda_{C1} & \lambda_{C2} & . & . & \lambda_{CC} \end{bmatrix} \text{Zero-constant case} \Rightarrow \Lambda = \begin{bmatrix} \lambda_r & \lambda_r & \lambda_r & . & . & \lambda_r \\ 0 & \lambda_e & \lambda_e & . & . & \lambda_e \\ . & 0 & . & . & . & . \\ . & . & . & . & . & . \\ \lambda_e & \lambda_e & \lambda_e & . & . & 0 \end{bmatrix}$$

$$ECM = \sum_{j=1}^{C} P(z = j) \sum_{i=0}^{C} \lambda_{ij} P(\alpha = i \mid z = j)$$

PE is still the same equation

Optimal Classifier

What is the **optimal** classifier? (Answer: **Bayesian Classifier**)

$$ECM = \sum_{j=1}^{C} P(z=j) \sum_{i=0}^{c} \lambda_{ij} P(\alpha = i \mid z = j)$$

$$= \int_{\underline{x}} \sum_{j=1}^{C} \sum_{i=0}^{C} \lambda_{ij} P(\alpha = i, \underline{x} \mid z = j) P(z = j) d\underline{x}$$

$$= \int_{\underline{x}} \sum_{j=1}^{C} \sum_{i=0}^{C} \lambda_{ij} P(\alpha = i \mid \underline{x}, z = j).p(\underline{x} \mid z = j).P(z = j) d\underline{x}$$

Since $\alpha = i$ *is a mapping from* $\underline{x} \rightarrow \hat{z}$, $P(\alpha = \hat{z} = i \mid \underline{x}, z = j) = P(\alpha = i \mid \underline{x})$

$$so,\ ECM = \int_{\underline{x}} \sum_{i=0}^{C} P(\alpha = i \mid \underline{x}).\left[\sum_{j=1}^{C} \lambda_{ij}\, p(\underline{x} \mid z = j).P(z = j) \right] d\underline{x}$$

Optimal Classifier (contd.)

$$\min_{\{y_i\}} \sum_{i=0}^{C} a_i y_i \qquad y_i = P(\alpha = i \mid \underline{x})$$

This is an **LP** of the form: $s.t. \sum_{i=0}^{C} y_i = 1; y_i \in \{0,1\}$ $\Rightarrow$ $a_i = \sum_{j=1}^{C} \lambda_{ij} p(\underline{x} \mid z = j) P(z = j)$

Optimal solution: $pick\ y_k = 1\ if\ k = \arg \min_{i \in \{0,1,2,..,C\}} \{a_i\}$

$$\Rightarrow \text{Pick action } \alpha = k \text{ (class } \hat{z} = k), \text{if } k = \arg \min_{i \in \{0,1,2,...,C\}} \sum_{j=1}^{C} \lambda_{ij} p(\underline{x} \mid z = j) P(z = j)$$

Note: $P(z = j \mid \underline{x}) = \dfrac{p(\underline{x} \mid z = j) P(z = j)}{p(\underline{x})}$ → a constant $\Rightarrow k = \arg \min_{i \in \{0,1,2,..,C\}} \sum_{j=1}^{c} \lambda_{ij} P(z = j \mid \underline{x})$

Let: $\underline{p} = [P(z = 1 \mid \underline{x})\ P(z = 2 \mid \underline{x}) \ldots\ldots P(z = C \mid \underline{x})]^T$

Compute $\underline{r} = \Lambda\, \underline{p}$ $\quad (C+1)\ vector$

$$k = \arg \min_{i \in \{1,2,..,C\}} \{r_i\}$$

Special Case: Binary Case - 1

$$\Lambda = \begin{bmatrix} \lambda_{01} & \lambda_{02} \\ \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \end{bmatrix}; \lambda_{11} < \lambda_{12}; \lambda_{22} < \lambda_{21} \Rightarrow \text{correct decisions have less cost}$$

Since $P(z=2 \mid \underline{x}) = 1 - P(z=1 \mid \underline{x})$, the decision rule is:

$$k = \arg \min_{i \in \{0,1,2\}} \{(\lambda_{i1} - \lambda_{i2}) P(z=1 \mid \underline{x}) + \lambda_{i2}\}$$

Special case 1: $\lambda_{01} = \lambda_{02} = \lambda_r = \infty \Rightarrow$ Likelihood Ratio Rule

$$\frac{p(\underline{x} \mid z=1)}{p(\underline{x} \mid z=2)} \geq \frac{(\lambda_{12} - \lambda_{22}) P(z=2)}{(\lambda_{21} - \lambda_{11}) P(z=1)} \Rightarrow \hat{z} = 1; \text{otherwise } \hat{z} = 2 \text{ (Prove it)}$$

Special case 2: $\lambda_{01} = \lambda_{02} = \lambda_r$

Reject range for $P(z=1 \mid \underline{x})$ exists if $\frac{\lambda_r - \lambda_{12}}{\lambda_{11} - \lambda_{12}} > \frac{\lambda_r - \lambda_{22}}{\lambda_{21} - \lambda_{22}}$; else no reject decision (Prove it)

Special case 3: $\lambda_{01} = \lambda_{02} = \lambda_r; \lambda_{11} = \lambda_{22} = 0; \lambda_{12} > 0; \lambda_{21} > 0$

Reject range for $P(z=1 \mid \underline{x})$ exists if $\frac{1}{\lambda_r} > \frac{1}{\lambda_{12}} + \frac{1}{\lambda_{21}}$; else no reject decision (Prove it)

General case: Draw the decision ranges as a function of $P(z=1 \mid \underline{x}) \in [0,1]$ for the four cases:

$(i) 0 < (\lambda_{21} - \lambda_{22}) < (\lambda_{01} - \lambda_{02}); (ii) 0 > (\lambda_{01} - \lambda_{02}) > (\lambda_{11} - \lambda_{12});$

$(iii) 0 > (\lambda_{11} - \lambda_{12}) > (\lambda_{01} - \lambda_{02}); (iv) 0 < (\lambda_{01} - \lambda_{02}) < (\lambda_{21} - \lambda_{22})$

Special Case: Binary Case - 2

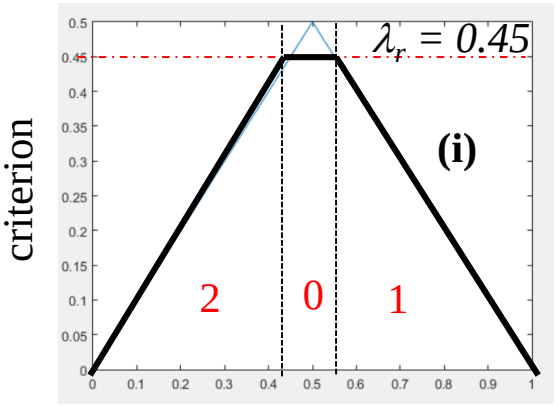
$$(1)\,\Lambda = \begin{bmatrix} \lambda_r & \lambda_r \\ 0 & 1 \\ 1 & 0 \end{bmatrix}; (2)\Lambda = \begin{bmatrix} \lambda_r & \lambda_r \\ 0 & 1000 \\ 1 & 0 \end{bmatrix}; (3)\Lambda = \begin{bmatrix} \lambda_r & \lambda_r \\ 0 & 1 \\ 10 & 0 \end{bmatrix}$$

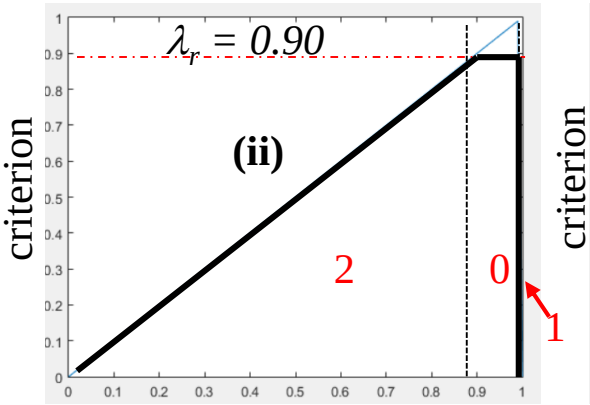
Decision Rules:

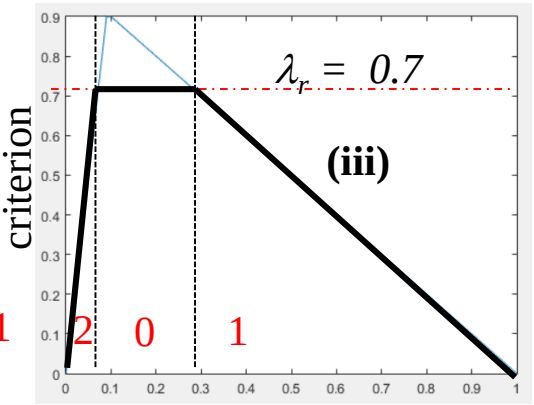
$(i)\, k = \arg \min_{i\in\{0,1,2\}} \{\lambda_r, 1 - P(z=1|\underline{x}), P(z=1|\underline{x})\}$; Reject option exists if $\lambda_r < 1/2$

$(ii) k = \arg \min_{i\in\{0,1,2\}} \{\lambda_r, 1000[1 - P(z=1|\underline{x})], P(z=1|\underline{x})\}$; Reject option exists if $\lambda_r < 1000/1001$

$(iii) k = \arg \min_{i\in\{0,1,2\}} \{\lambda_r, 1 - P(z=1|\underline{x}), 10P(z=1|\underline{x})\}$; Reject option exists if $\lambda_r < 10/11$







Special Case: Nearly Symmetric Loss Function

$$\lambda_{ij} = \begin{cases} 0 & \alpha = i \quad \text{Correct Decision} \\ \lambda_r & \alpha = 0 \quad \text{Reject} \\ \lambda_e & \alpha \neq i \quad \text{Error} \end{cases}$$

Nearly-Symmetric Loss function

$$\underline{r} = \Lambda \underline{p}$$

$$= \begin{bmatrix} \lambda_r & \lambda_r & \lambda_r & . & . & \lambda_r \\ 0 & \lambda_e & \lambda_e & . & . & \lambda_e \\ \lambda_e & 0 & \lambda_e & . & . & \lambda_e \\ \lambda_e & \lambda_e & 0 & . & . & \lambda_e \\ \lambda_e & \lambda_e & \lambda_e & . & . & . \\ \lambda_e & \lambda_e & \lambda_e & . & . & 0 \end{bmatrix} \begin{bmatrix} P(z=1|\underline{x}) \\ P(z=2|\underline{x}) \\ . \\ . \\ . \\ P(z=C|\underline{x}) \end{bmatrix} = \begin{bmatrix} \lambda_r \\ \lambda_e[1-P(z=1|\underline{x})] \\ \\ \\ \\ \lambda_e[1-P(z=C|\underline{x})] \end{bmatrix}$$

Special Case (contd.)

$$Pick\ action\ \alpha = k,\ k \in \{0,1,2,..,C\}$$

$$If\ \ k = \arg\min\{\lambda_r, \lambda_e[1-P(z=1|\underline{x})], ..., \lambda_e[1-P(z=C|\underline{x})]\}$$

$$= \arg\min\{\frac{\lambda_r}{\lambda_e}, [1-P(z=1|\underline{x})], ..., [1-P(z=C|\underline{x})]\} \because \lambda_e > 0$$

$$= \arg\min\{\frac{\lambda_r}{\lambda_e} - 1, -P(z=1|\underline{x}), ..., -P(z=C|\underline{x})\}$$

$$= \arg\max\{\underbrace{1-\frac{\lambda_r}{\lambda_e}}_{\beta}, P(z=1|\underline{x}), ..., P(z=C|x)\}$$

$$\text{Decide for } \alpha = k (class\ \hat{z} = k): \begin{cases} if\ P(z=k|\underline{x}) = \max\limits_{j\in\{1,2,..,C\}} P(z=j|\underline{x})\ \underline{and}\ P(z=k|\underline{x}) > \beta \\ otherwise \qquad\qquad reject\ (action\ \alpha = 0) \end{cases}$$

note that reject can only occur if $\beta > 1/C$, i.e., $1/C < \beta < 1$

Special Case (contd.)

Alternately,

$$\text{Decide for } \alpha = k: \begin{cases} \text{if } p(\underline{x} \mid z=k)P(z=k) = \max\limits_{j\in\{1,2,..,C\}} p(\underline{x} \mid z=j)P(z=j) \text{ and} \\ \qquad p(\underline{x} \mid z=k)P(z=k) > \beta p(\underline{x}) \\ \text{otherwise} \qquad \text{reject (action } \alpha = 0) \end{cases}$$

$$\beta = 0 \Rightarrow \lambda_r = \lambda_e \Rightarrow \arg \max_{j\in\{1,2,...,C\}} P(z=j \mid \underline{x})$$

MAP Classifier

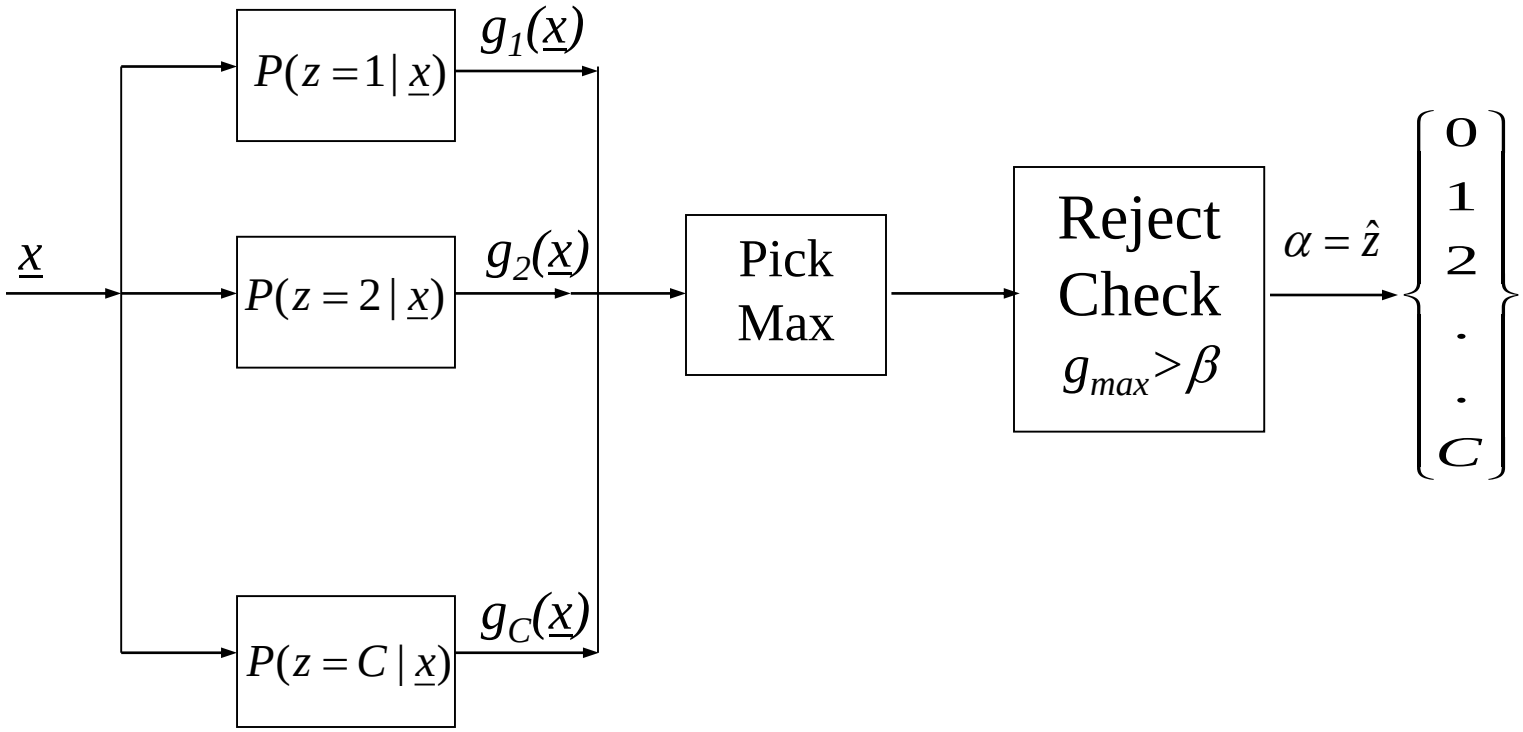
$$\Rightarrow \arg \max_{j\in\{1,2,..,C\}} [p(\underline{x} \mid z=j)P(z=j)]$$

$$\Rightarrow \arg \max_{j\in\{1,2,..,C\}} [\ln p(\underline{x} \mid z=j) + \ln P(z=j)]$$

$$\Rightarrow \text{If } p(z=j) = \frac{1}{C} \forall j \Rightarrow \arg \max_{j\in\{1,2,..,C\}} \ln p(\underline{x} \mid z=j)$$

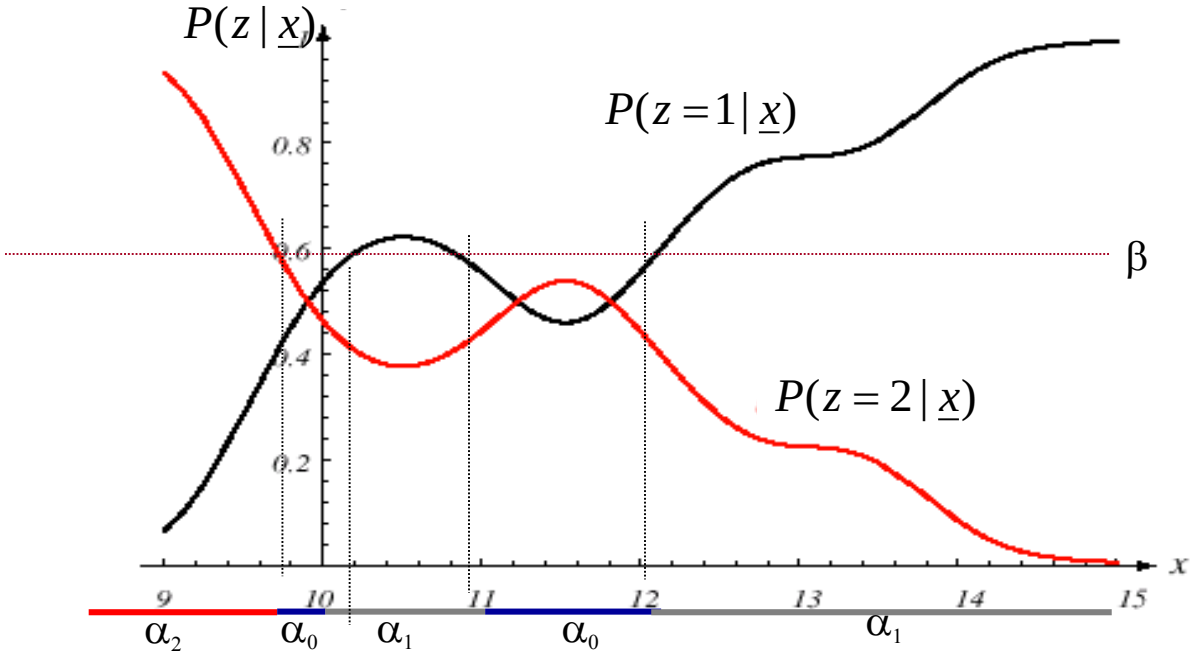
ML Classifier

Structure of the Classifier



$g_i(\underline{x})$ or monotonic transformations of $g_i(\underline{x})$ are called **discriminant functions**

Classifier with Reject Option
(in the feature space)



Key: *P(z=k|x)* also provide a measure of confidence in the decision

Other reject criteria: $g_{max} - g_{second\ max} < \delta \Rightarrow$ reject / doubt

Gaussian Classes

- Gaussian Classes $\Rightarrow$ $p(\underline{x} \mid z = i) = N(\underline{\mu}_i, \Sigma_i)$ $\Rightarrow$ $p(\underline{x})$ is a Gaussian mixture!

$$p(\underline{x} \mid z = i) = \frac{1}{(2\pi)^{p/2} \mid \Sigma_i \mid^{1/2}} \exp\left[-\frac{1}{2} (\underline{x} - \underline{\mu}_i)^T \Sigma_i^{-1} (\underline{x} - \underline{\mu}_i) \right]$$

$\underline{\mu}_i$= p-component *mean vector* $\underline{\mu}_i = E(\underline{x}) = \int \underline{x}\, p(\underline{x} \mid z = i) d\,\underline{x}$

Σ_i= p-by-p *covariance matrix*, $\Sigma_i = E[(\underline{x} - \underline{\mu}_i)(\underline{x} - \underline{\mu}_i)^T)$

$$= \int (\underline{x} - \underline{\mu}_i)(\underline{x} - \underline{\mu}_i)^T\, p(\underline{x} \mid z = i) d\,\underline{x}$$

Σ_i is symmetric and positive definite

$\Downarrow$

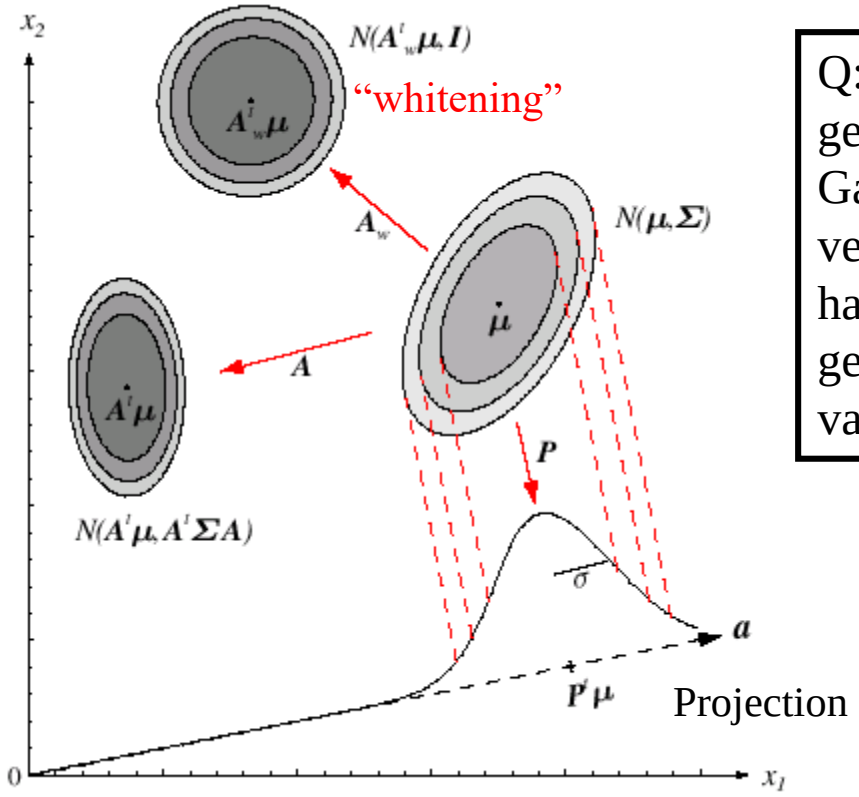
Eigenvalues are positive and eigenvectors are orthogonal

$$\Sigma_i = Q \Lambda Q^T ;\ Q^{-1} = Q^T$$

Transformations of Gaussian Density

$$p(\underline{x}) = N(\underline{\mu}, \Sigma) \; and \; \underline{y} = A^T \underline{x} \Rightarrow p(\underline{y}) = N(A^T \underline{\mu}, A^T \Sigma A)$$

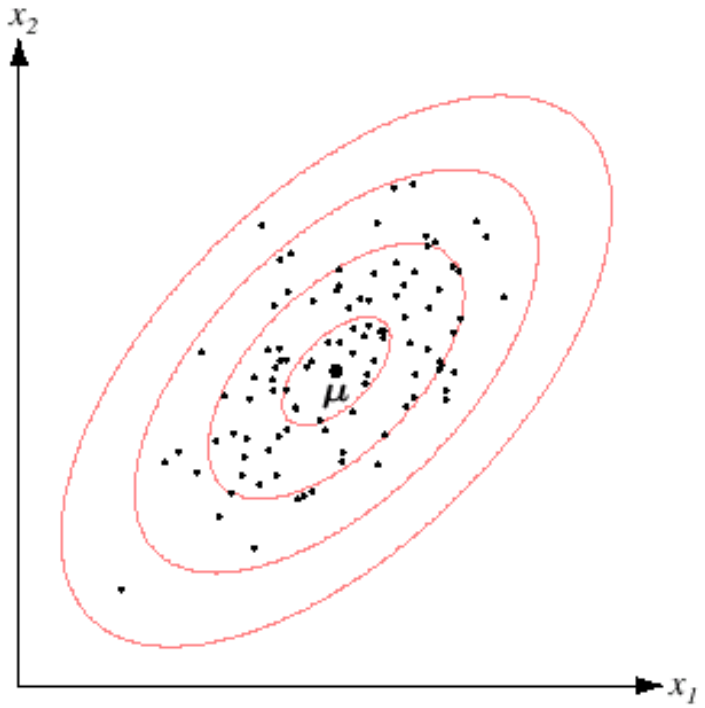
$$A = A_w = Q\Lambda^{-1/2} \Rightarrow p(\underline{y}) = N(\Lambda^{-1/2} Q^T \underline{\mu}, I) \Rightarrow "whitening"$$



Q: How do you generate multi-variate Gaussian random vector $N(\underline{\mu}, \Sigma)$ if you have a routine to generate N(0,1) variable?

Contours of Gaussian Density

- Loci of points of constant density are hyper-ellipsoids with *eigenvectors of* Σ *(columns of Q) as the principal axes*



- Squared Mahalanobis distance

$$r^2 = (\underline{x} - \underline{\mu})^T \Sigma^{-1} (\underline{x} - \underline{\mu})$$

- Contours of constant density are hyper-ellipsoids of constant Mahalanobis distance
- Volume of hyper-ellipsoid for distance r

$$V = \begin{cases} |\Sigma|^{1/2} \, r^p \pi^{p/2} / (p/2)!; & p \text{ even} \\ |\Sigma|^{1/2} \, (2r)^p \pi^{(p-1)/2} \dfrac{(\frac{p-1}{2})!}{p!}; & p \text{ odd} \end{cases}$$

Discriminants for Gaussian Classes - 1

- General form: $g_i(\underline{x}) = \ln p(\underline{x} \mid z = i) + \ln P(z = i)$

$$g_i(\underline{x}) = -\frac{1}{2}(\underline{x} - \underline{\mu}_i)^T \Sigma_i^{-1}(\underline{x} - \underline{\mu}_i) - \frac{1}{2}\ln|\Sigma_i| + \ln P(z = i) - \frac{p}{2}\ln(2\pi)$$ **(hyper-quadric)**

$$k = \arg\max_i \left\{ -\frac{1}{2}\ln|\Sigma_i| - \frac{1}{2}(\underline{x} - \underline{\mu}_i)^T \Sigma_i^{-1}(\underline{x} - \underline{\mu}_i) + \ln P(z = i) \right\}$$ **(quadratic rule)**

- Case 1: $\Sigma_i = \sigma^2 I_p \Rightarrow$ spherical Gaussian densities

$$g_i(\underline{x}) = -\frac{\| \underline{x} - \underline{\mu}_i \|^2}{2\sigma^2} + \ln P(z = i)$$

$$= -\frac{1}{2\sigma^2}[\underline{x}^T \underline{x} - 2\underline{\mu}_i^T \underline{x} + \underline{\mu}_i^T \underline{\mu}_i] + \ln P(z = i)$$

Discriminants for Gaussian Classes - 2

- Case 1: $\Sigma_i = \sigma^2 I_p \Rightarrow$ *spherical Gaussian densities (cont'd)*

  equivalently, ignoring additive constants

$$g_i(\underline{x}) = \frac{1}{\sigma^2}\underline{\mu}_i^T \underline{x} - \left(\frac{1}{2\sigma^2}\underline{\mu}_i^T \underline{\mu}_i - \ln P(z=i)\right)$$
$$= \underline{w}_i^T \underline{x} - w_{io}$$

(linear rule)

- In the two category case, simplifies to a **single hyperplane equation**

$$g(\underline{x}) = \sigma^2[g_1(\underline{x}) - g_2(\underline{x})]$$
$$= (\underline{\mu}_1 - \underline{\mu}_2)^T \underline{x} - \left[\frac{1}{2}(\underline{\mu}_1^T \underline{\mu}_1 - \underline{\mu}_2^T \underline{\mu}_2) - \sigma^2 \ln \frac{P(z=1)}{P(z=2)}\right]$$
$$= \underline{w}^T \underline{x} - w_o = \underline{w}^T(\underline{x} - \underline{x}_o) = 0;\ \underline{x}_o = \frac{\underline{w}}{\underline{w}^T \underline{w}} w_0$$

Insights from Spherical & Binary Case

- Equal priors

$$g(\underline{x}) = \underline{w}^T(\underline{x} - \underline{x}_o) = 0;\ \underline{x}_o = \frac{(\underline{\mu}_1 + \underline{\mu}_2)}{2}$$

Hyperplane passes through the average of two means

In one dimension, discriminant is $x = x_0 = (\mu_1 + \mu_2)/2$
In general, minimum distance or nearest neighbor classifier (assign x to the class of nearest mean)

- Unequal priors

In one dimension, discriminant is $x = x_0 = \frac{(\mu_1 + \mu_2)}{2} + \frac{\sigma^2}{(\mu_2 - \mu_1)} \ln \frac{P(z=1)}{P(z=2)}$

$\underline{x}_0$ moves away from the more likely mean

- Small variance $\Rightarrow$ ***$\underline{x}_0$ insensitive to priors***
- Large variance $\Rightarrow$ ***$\underline{x}_0$ sensitive to priors***

Discriminants for Gaussian Classes - 3

- General form: $$g_i(\underline{x}) = -\frac{1}{2}(\underline{x}-\underline{\mu}_i)^T \Sigma_i^{-1}(\underline{x}-\underline{\mu}_i) - \frac{1}{2}\ln|\Sigma_i| + \ln P(z=i)$$

- Case 2: $\Sigma_i = \Sigma$ $\Rightarrow$ data falls in hyperellipsoidal clusters of equal size and shape

$$g_i(\underline{x}) = -\frac{1}{2}(\underline{x}-\underline{\mu}_i)^T \Sigma^{-1}(\underline{x}-\underline{\mu}_i) + \ln P(z=i)$$

$$= -\frac{1}{2}\underline{x}^T\Sigma^{-1}\underline{x} + \underline{\mu}_i^T\Sigma^{-1}\underline{x} - [\frac{1}{2}\underline{\mu}_i^T\Sigma^{-1}\underline{\mu}_i - \ln P(z=i)]$$

equal priors $\Rightarrow$ minimum Mahalanobis distance or weighted nearest neighbor classifier

equivalently, ignoring additive constant term involving $\underline{x}^T\Sigma^{-1}\underline{x}$

$$g_i(\underline{x}) = \underline{\mu}_i^T\Sigma^{-1}\underline{x} - [\frac{1}{2}\underline{\mu}_i^T\Sigma^{-1}\underline{\mu}_i - \ln P(z=i)]$$

$$= \underline{w}_i^T\underline{x} - w_{i0}$$

(linear rule)

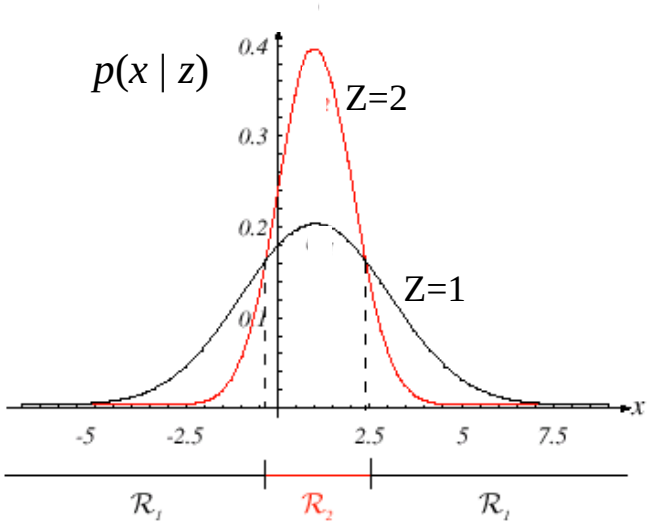
Discriminants for Gaussian Classes - 4

- Case 3: Σ_i is arbitrary $\Rightarrow g_i(\underline{x})$ is a hyperquadric

  $\Rightarrow$ decision surfaces can be hyperplanes, hyperspheres, hyperellipsoids, hyperparaboloids, or hyperhyperboloids

$$g_i(\underline{x}) = -\frac{1}{2}(\underline{x}-\underline{\mu}_i)^T\Sigma_i^{-1}(\underline{x}-\underline{\mu}_i) - \frac{1}{2}\ln|\Sigma_i| + \ln P(z=i)$$

$$= -\frac{1}{2}\underline{x}^T\Sigma_i^{-1}\underline{x} + \underline{\mu}_i^T\Sigma_i^{-1}\underline{x} - [\frac{1}{2}\underline{\mu}_i^T\Sigma_i^{-1}\underline{\mu}_i + \frac{1}{2}\ln|\Sigma_i| - \ln P(z=i)]$$



- ***Non-simply connected*** *regions*

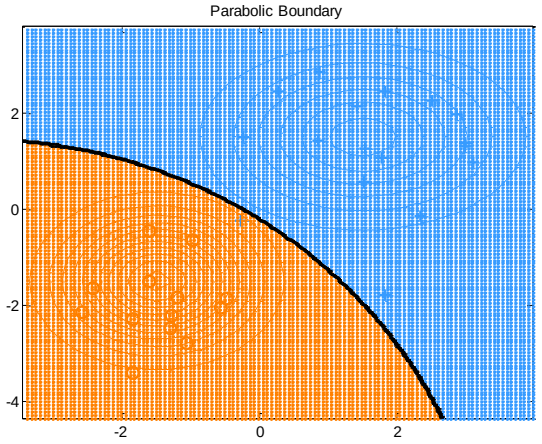
  Even in one dimensional case

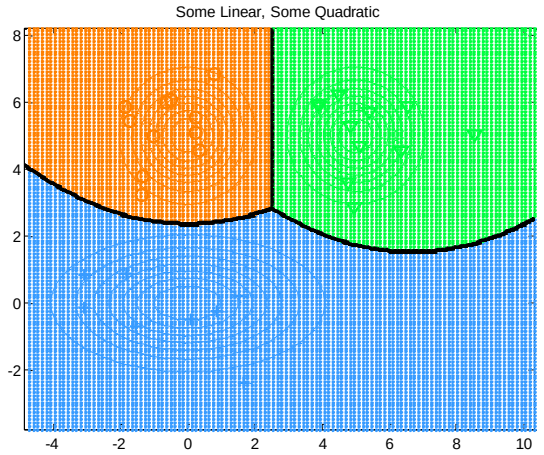
$$Note: P(z=1|\underline{x}) = \frac{e^{g_1(\underline{x})}}{e^{g_1(\underline{x})}+e^{g_2(\underline{x})}} = \frac{1}{1+e^{g_2(\underline{x})-g_1(\underline{x})}}$$

$$= \frac{1}{1+e^{g(\underline{x})}} = \sigma(x) \text{ when } \Sigma_1 = \Sigma_2$$

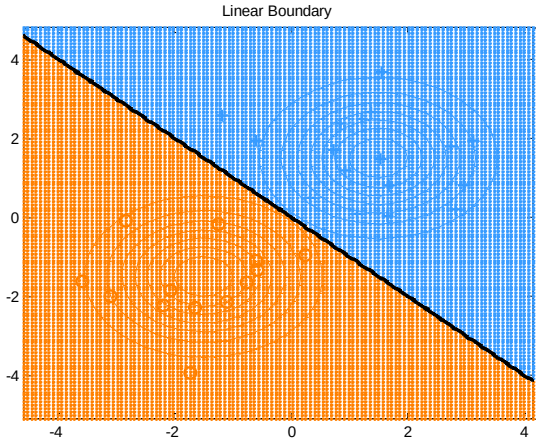
Discriminants for Gaussian Classes - 5

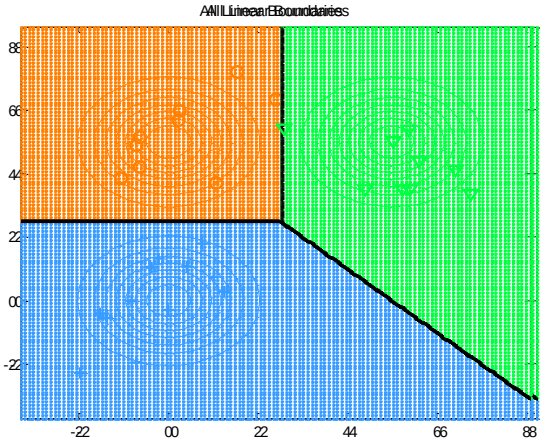
- Quadratic Decision Boundaries for 2-class and 3-class cases





- Linear Decision Boundaries for 2-class and 3-class cases





discrimAnalysisDboundariesDemo from Murphy

Error Probabilities: Two Class Case-1

- Two ways in which classification error can occur
  - Observation $\underline{x}$ falls in R_2 and the true class is $z=1$, or $\underline{x}$ falls in R_1 and the true class is $z=2$

$$P(error) = P(\underline{x} \in R_2, z=1) + P(\underline{x} \in R_1, z=2)$$
$$= \int_{R_2} p(\underline{x} \mid z=1)P(z=1)d\underline{x} + \int_{R_1} p(\underline{x} \mid z=2)P(z=2)d\underline{x}$$

- Chernoff Bound

know $\min(a,b) \le a^{\beta}b^{1-\beta}$ *for* $a,b \ge 0$ *and* $0 \le \beta \le 1$

If $a \ge b$ *then* $(a/b)^{\beta} \ge 1$
$\Rightarrow (a/b)^{\beta} b \ge b$
$\Rightarrow a^{\beta}b^{1-\beta} \ge b$

$$P(error) = \int_{\underline{x}} P(error \mid \underline{x})\, p(\underline{x})d\underline{x}$$
$$= \int_{\underline{x}} \min[P(z=1 \mid \underline{x}), P(z=2 \mid \underline{x})]p(\underline{x})d\underline{x}$$
$$\le [P(z=1)]^{\beta}[P(z=2)]^{1-\beta} \int_{\underline{x}} [p(\underline{x} \mid z=1)]^{\beta}[p(\underline{x} \mid z=2)]^{1-\beta} d\underline{x}$$

Error Probabilities: Two Class Case-2

- Gaussian case

$$P(error) \le [P(z=1)]^{\beta}[P(z=2)]^{1-\beta} e^{-k(\beta)} = e^{-k(\beta)+\beta \ln P(z=1)+(1-\beta)\ln[1-P(z=1)]}$$

$$where \quad k(\beta) = \frac{\beta(1-\beta)}{2} \| \underline{\mu}_2 - \underline{\mu}_1 \|^2_{[\beta\Sigma_1+(1-\beta)\Sigma_2]^{-1}} + \frac{1}{2}\ln \frac{|\beta\Sigma_1+(1-\beta)\Sigma_2|}{|\Sigma_1|^{\beta}|\Sigma_2|^{1-\beta}}$$

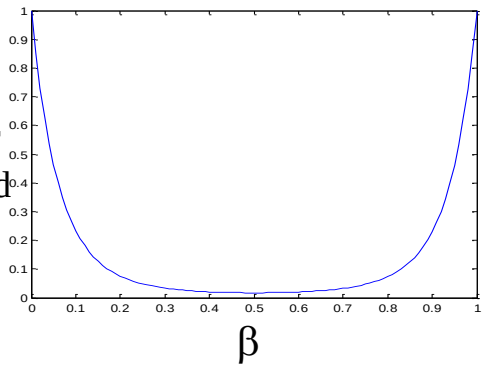
- Bhattacharyya Bound $\Rightarrow$ **β=1/2**

$$P(error) \le \sqrt{P(z=1)P(z=2)}\, e^{-k(1/2)}$$

$$\underline{\mu}_1 = \begin{bmatrix} 3 \\ 6 \end{bmatrix}; \Sigma_1 = Diag(1/2, 2)$$

$$\underline{\mu}_2 = \begin{bmatrix} 3 \\ -2 \end{bmatrix}; \Sigma_2 = Diag(2, 2)$$

$$where \quad k(1/2) = \frac{1}{8} \| \underline{\mu}_2 - \underline{\mu}_1 \|^2_{[\frac{\Sigma_1+\Sigma_2}{2}]^{-1}} + \frac{1}{2}\ln \frac{|\frac{\Sigma_1+\Sigma_2}{2}|}{\sqrt{|\Sigma_1||\Sigma_2|}}$$



In this case, minimum at
β = 0.5
k(β) = 4.0558
P(error) ≤ 0.00865
(assuming equal priors)

Discriminants for Gaussian Classes - 5

- Let us revisit two class case with equal covariance. In this case, simplifies to a single hyperplane equation (linear)

$$g(\underline{x}) = g_1(\underline{x}) - g_2(\underline{x})$$

$$= (\underline{\mu}_1 - \underline{\mu}_2)^T \Sigma^{-1} \underline{x} - [\frac{1}{2}(\underline{\mu}_1^T \Sigma^{-1} \underline{\mu}_1 - \underline{\mu}_2^T \Sigma^{-1} \underline{\mu}_2) - \ln \frac{P(z=1)}{P(z=2)}]$$

$$= \underline{w}^T \underline{x} - w_o = \underline{w}^T (\underline{x} - \underline{x}_o) = 0; \underline{x}_o = \frac{\underline{w}}{\underline{w}^T \underline{w}} w_o$$

where

$$\underline{x}_0 = \frac{1}{2}(\underline{\mu}_1 + \underline{\mu}_2) - \frac{\ln[P(z=1)/P(z=2)]}{\underbrace{(\underline{\mu}_1 - \underline{\mu}_2)^T \Sigma^{-1} (\underline{\mu}_1 - \underline{\mu}_2)}_{d'^2}} (\underline{\mu}_1 - \underline{\mu}_2)$$

d'	P_e
4.7	0.00939
3.3	0.04947
2.6	0.0968

$$P(z=1) = P(z=2) = \frac{1}{2} \Rightarrow \underline{x}_0 = \frac{1}{2}(\underline{\mu}_1 + \underline{\mu}_2)$$

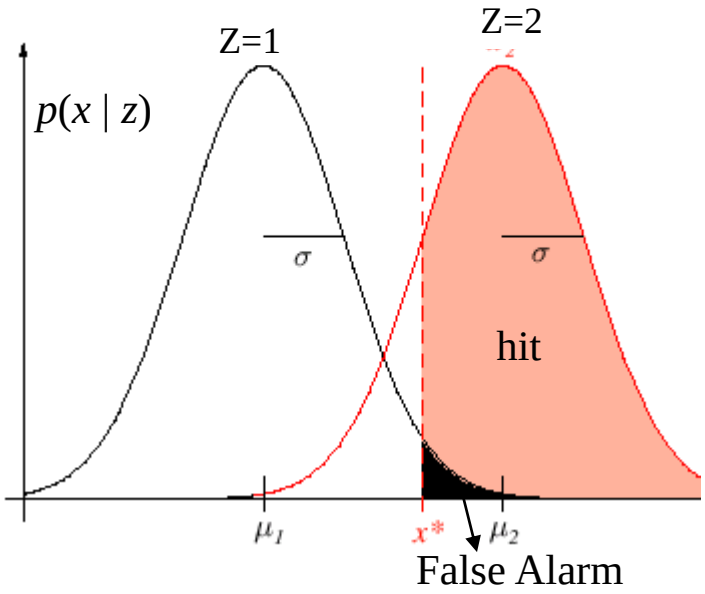
$$p(g(\underline{x}) \mid z=1) = N(\underline{w}^T \frac{\underline{\mu}_1 - \underline{\mu}_2}{2}, \underline{w}^T \Sigma \underline{w}) = N(\frac{d'^2}{2}, d'^2)$$

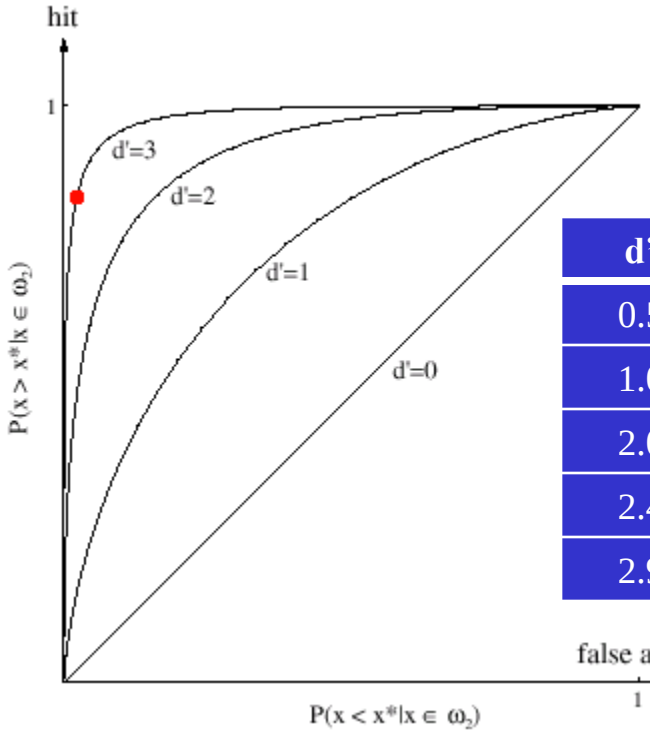
$$p(g(\underline{x}) \mid z=2) = N(\underline{w}^T \frac{\underline{\mu}_2 - \underline{\mu}_1}{2}, \underline{w}^T \Sigma \underline{w}) = N(-\frac{d'^2}{2}, d'^2)$$

$$P_e = \Phi(-\frac{|d'|}{2}) \Rightarrow \text{large } |d'| \text{ gives small } P_e$$

Error Probabilities: Two Class Case-3

- Receiver Operating Characteristic (ROC) Curve





d'	P_e
0.5	0.40129
1.0	0.30854
2.0	0.15866
2.4	0.11507
2.9	0.07353

- Some Common d' values:
  - Detecting liars with polygraph: 0.5-1.0
  - Detecting prostate cancer with PSA tests: 2.0
  - Detecting brain lesions with CT scans: 2.4-2.9

Error Probability: Multiclass Case

- *P(error)=1-P(correct)*
- There are more ways to be wrong than to be right. It is easier to compute *P(correct)*

$$P(correct) = \sum_{j=1}^{C} P(\underline{x} \in R_j, z = j)$$

$$= \sum_{j=1}^{C} P(\underline{x} \in R_j \mid z = j) P(z = j)$$

$$= \sum_{j=1}^{C} P(z = j) \int_{R_j} p(\underline{x} \mid z = j) d\underline{x}$$

Bayesian partitioning of feature space minimizes *P(error)*

Binary Feature Case

$$P(x_i \mid z=k) = \begin{cases} 1-\mu_{ik} \text{ for } x_i = 0 \\ \mu_{ik} \text{ for } x_i = 1 \end{cases}$$

$$P(\underline{x} \mid z=k) = \prod_{i=1}^{p} \mu_{ik}^{\,x_i} (1-\mu_{ik})^{1-x_i}$$

$$x_i = \begin{cases} 0 & \text{false} \\ 1 & \text{true} \end{cases}$$

$$\underline{x} = [x_1, x_2, \ldots, x_p]^T$$

$$P(z=k \mid \underline{x}) \;\propto\; P(\underline{x} \mid z=k) P(z=k)$$

$$= \underbrace{P(z=k)}_{\pi_k} \prod_{i=1}^{p} \mu_{ik}^{\,x_i} (1-\mu_{ik})^{1-x_i} = \pi_k . \prod_{i=1}^{p} \mu_{ik}^{\,x_i} (1-\mu_{ik})^{1-x_i}$$

$$g_k(\underline{x}) = \ln \pi_k + \sum_{i=1}^{p} [x_i \ln(\mu_{ik}) + (1-x_i)\ln(1-\mu_{ik})]$$

(linear rule)

$$= \sum_{i=1}^{p} x_i \underbrace{\ln \frac{\mu_{ik}}{1-\mu_{ik}}}_{\eta_{ik}} + \underbrace{\ln \pi_k + \sum_{i=1}^{p} \ln(1-\mu_{ik})}_{-w_{k0}} = \underline{w}_k^T \underline{x} - w_{ko}$$

$$Note: \mu_{ik} = \frac{\exp(\eta_{ik})}{1+\exp(\eta_{ik})};\ sigmoid$$

Easily extends to the case when x_i is multinomial, i.e., takes values $\{1,2,\ldots, M_i\}$, $M_i > 2$

Multinomial Feature Case

$$P(x_i = j \mid z = k) = P(x_{ij} = 1 \mid z = k\} = \mu_{ijk};\ j = 1,2,..,M_i$$

$$\sum_{j=1}^{M_i} \mu_{ijk} = 1 \forall i,k$$

$$P(\underline{x} \mid z = k) = \prod_{i=1}^{p} \left[\left(\prod_{j=1}^{M_i - 1} \mu_{ijk}^{\ x_{ij}} \right) (1 - \sum_{j=1}^{M_i - 1} \mu_{ijk})^{\left(1 - \sum_{j=1}^{M_i - 1} x_{ij}\right)} \right]$$

$x_i \in \{1, 2, ..., M_i\}$

$\underline{x} = [x_1, x_2, ..., x_p]^T$

$$Let\ x_{ij} = \begin{cases} 1\ if\ x_i = j \\ 0;\ otherwsie \end{cases};\ j = 1, 2, ..., M_i$$

$$\sum_{j=1}^{M_i} x_{ij} = 1$$

$$P(z = k \mid \underline{x}) \ \propto\ \mathrm{P}(\underline{x} \mid z = k) P(z = k) = \pi_k . \prod_{i=1}^{p} \left[\left(\prod_{j=1}^{M_i - 1} \mu_{ijk}^{\ x_{ij}} \right) (1 - \sum_{j=1}^{M_i - 1} \mu_{ijk})^{\left(1 - \sum_{j=1}^{M_i - 1} x_{ij}\right)} \right]$$

(linear rule)

$$g_k(\underline{x}) = \ln \pi_k + \sum_{i=1}^{p} \left\{ \left(\sum_{j=1}^{M_i - 1} x_{ij}\ \eta_{ijk} \right) + \ln \left(1 - \sum_{j=1}^{M_i - 1} \mu_{ijk} \right) \right\}; \eta_{ijk} = \ln \left(\frac{\mu_{ijk}}{1 - \sum_{l=1}^{M_i - 1} \mu_{ilk}} \right)$$

$$Note: \mu_{ijk} = \frac{\exp(\eta_{ijk})}{1 + \sum_{l=1}^{M_i - 1} \exp(\eta_{ilk})};\ soft \max\ or\ normalized\ \exp onential$$

Missing Features

- Partition $\underline{x}=[\underline{x}_v, \underline{x}_h]$
  - $\underline{x}_v$ represents the *available* or *visible* features
  - $\underline{x}_h$ represents the *missing* or *hidden* features
- MAP rule with missing features: $\arg \max_{j \in \{1,2,...,C\}} P(z = j \mid \underline{x}_v)$

$$P(z = j \mid \underline{x}_v) = \frac{p(z = j, \underline{x}_v)}{p(\underline{x}_v)}$$

$$= \frac{\int_{\underline{x}_h} P(z = j \mid \underline{x})\, p(\underline{x}) d\underline{x}_h}{\int_{\underline{x}_h} p(\underline{x}) d\underline{x}_h}$$

$$= \frac{\int_{\underline{x}_h} g_j(\underline{x})\, p(\underline{x}) d\underline{x}_h}{\int_{\underline{x}_h} p(\underline{x}) d\underline{x}_h}$$

Gaussian case:

$$P(z = j \mid \underline{x}_v) = \frac{p(\underline{x}_v \mid z = j) P(z = j)}{p(\underline{x}_v)}$$

Since $p(\underline{x}_v \mid z = j) = N(\underline{\mu}_{vj}, \Sigma_{vj})$, *we simply ignore missing features. With reject option, use* $\beta\, p(\underline{x}_v)$ *in place of* $\beta\, p(\underline{x})$.

Marginalize the Discriminant functions over missing features

Noisy Features

- Suppose *all* features are observed with *noise*
  - Let $\underline{x}$ represent the *true* features
  - Let $\underline{x}_n$ represent the *observed noisy* features
- MAP rule with noisy features: $\arg \max_{j \in \{1,2,\ldots,C\}} P(z = j \mid \underline{x}_n)$

$$P(z = j \mid \underline{x}_n) = \frac{\int_{\underline{x}} P(z = j \mid \underline{x}) p(\underline{x}_n \mid \underline{x}) p(\underline{x}) d\underline{x}}{\int_{\underline{x}} p(\underline{x}_n \mid \underline{x}) p(\underline{x}) d\underline{x}}$$

$$= \frac{\int_{\underline{x}} g_j(\underline{x}) p(\underline{x}_n \mid \underline{x}) p(\underline{x}) d\underline{x}}{\int_{\underline{x}} p(\underline{x}_n \mid \underline{x}) p(\underline{x}) d\underline{x}}$$

Marginalize the Discriminant functions over noise model. Need to do Monte Carlo approximation because the posterior probability is a softmax. Probit approximations are possible for binary case.

Noisy Features : Gaussian Case

- Suppose all features are observed with noise
  - Let $\underline{x}$ represent the *true* features
  - Let $\underline{x}_n$ represent the *observed noisy* features

$$\underline{x}_n = \underline{x} + \underline{v}_n \; ; \; \underline{v}_n \sim N(\underline{0}, \Sigma_v)$$
$$\Rightarrow p(\underline{x}_n \mid z = i) \sim N(\underline{\mu}_i, \Sigma_i + \Sigma_v)$$

- Discriminant functions with noisy features:
  - Quadratic Discriminant

$$g_i(\underline{x}_n) = -\frac{1}{2}(\underline{x}_n - \underline{\mu}_i)^T [\Sigma_i + \Sigma_v]^{-1} (\underline{x}_n - \underline{\mu}_i) - \frac{1}{2} \ln |\Sigma_i + \Sigma_v| + \ln P(z = i)$$

  - Linear Discriminant

$$g_i(\underline{x}_n) = \underline{\mu}_i^T (\Sigma + \Sigma_v)^{-1} \underline{x}_n - [\frac{1}{2} \underline{\mu}_i^T (\Sigma + \Sigma_v)^{-1} \underline{\mu}_i - \ln P(z = i)]$$

  - Binary case

$$g(\underline{x}_n) = g_1(\underline{x}_n) - g_2(\underline{x}_n) = (\underline{\mu}_1 - \underline{\mu}_2)^T (\Sigma + \Sigma_v)^{-1} \underline{x}_n - [\frac{1}{2}(\underline{\mu}_1^T (\Sigma + \Sigma_v)^{-1} \underline{\mu}_1 - \underline{\mu}_2^T (\Sigma + \Sigma_v)^{-1} \underline{\mu}_2) - \ln \frac{P(z = 1)}{P(z = 2)}]$$

Bayesian Classifier: Summary

- Bayesian Classifier:

$$k = \arg \min_{i \in \{0,1,2,..,C\}} \sum_{j=1}^{C} \lambda_{ij}\, p(\underline{x} \mid z = j) P(z = j) = \arg \min_{i \in \{0,1,2,..,C\}} \sum_{j=1}^{C} \lambda_{ij} P(z = j \mid \underline{x})$$

- Special Case: λ_r reject, λ_e for error

$$k = \begin{cases} if\ P(z = k \mid \underline{x}) is\ \max\ and\ P(z = k \mid \underline{x}) > \beta = 1 - \dfrac{\lambda_r}{\lambda_e} \\ \quad reject\ otherwise \end{cases}$$

- $\beta = 0 \Rightarrow$ MAP classifier

$$k = \arg \max_{i \in \{1,2,..,C\}} P(z = i \mid \underline{x})$$
$$= \arg \max_{i \in \{1,2,..,C\}} p(\underline{x} \mid z = i) P(z = i)$$
$$= \arg \max_{i \in \{1,2,..,C\}} \left(\ln p(\underline{x} \mid z = i) + \ln P(z = i) \right)$$

$$P(z = j) = \frac{1}{C} \forall j \Rightarrow ML\ Classifier$$

$$k = \arg \max_{i \in \{1,2,..,C\}} p(\underline{x} \mid z = i)$$
$$= \arg \max_{i \in \{1,2,..,C\}} \ln p(\underline{x} \mid z = i)$$

Gaussian Case: Summary

- Gaussian Classes: General Case

$$k = \arg \max_{i \in \{1,2,..,C\}} \left\{ -\frac{1}{2} \ln |\Sigma_i| - \frac{1}{2} (\underline{x} - \underline{\mu}_i)^T \Sigma_i^{-1} (\underline{x} - \underline{\mu}_i) + \ln P(z = i) \right\}$$ (quadratic rule)

- Gaussian Classes (when covariance matrices for all classes are taken to be equal, i.e., $\Sigma_i = \Sigma$)

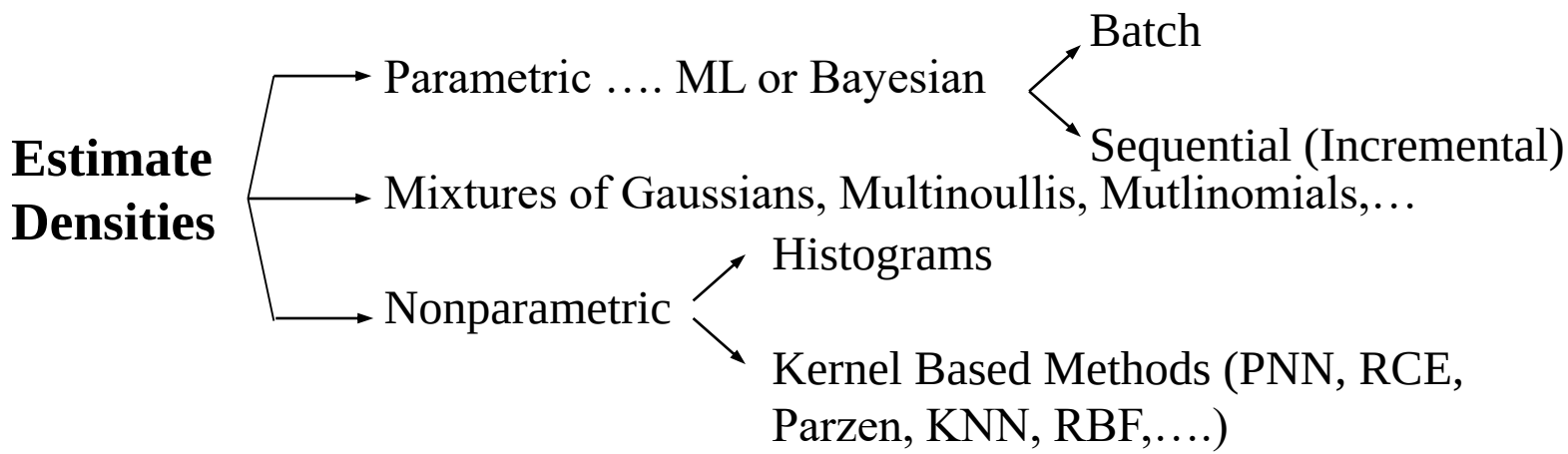
$$k = \arg \max_{i \in \{1,2,..,C\}} \left\{ \underline{\mu}_i^T \Sigma^{-1} \underline{x} - [\frac{1}{2} \underline{\mu}_i^T \Sigma^{-1} \underline{\mu}_i - \ln P(z = i)] \right\}$$ (linear rule)

- Gaussian Classes (when $\Sigma_i = \Sigma = \sigma^2 I_p$)

$$k = \arg \max_{i \in \{1,2,..,C\}} [\frac{1}{\sigma^2} \underline{\mu}_i^T \underline{x} - (\frac{1}{2\sigma^2} \underline{\mu}_i^T \underline{\mu}_i - \ln P(z = i))]$$ (simplified linear rule)

Real life: Don't Know: $P(z)$, $p(x|z=i)$, λ_{ij}

- We don't know $\{P(z = j), p(\underline{x} \mid z = j), \lambda_{ij}\}$?
- We usually estimate or infer them from data.
- Density estimation
  - Estimation of $p(\underline{x} \mid z=i)$ is difficult for $p \geq 2$ due to the so-called curse of dimensionality



Real life: Don't Know: $P(z)$, $p(x|z=i)$, λ_{ij}

- Generative versus Discriminative?
- Can we exploit & **generalize the forms of discriminant functions**? Can we estimate discriminants directly?
- **Different types of learning** (unsupervised, supervised, semi-supervised, RL)
- How to handle **missing data**?
- How do we **select features** $\underline{x}$ for best classification/regression accuracy?
- Can we **exploit dependency structure** among features?
- What happens if classes and features change **dynamically**?
- How do we **validate data driven models**? How do we **select the best model**?