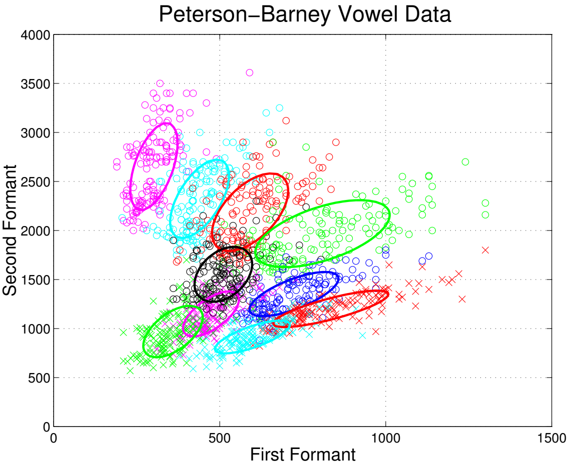
UNIVERSITY OF CAMBRIDGE

4F10: Probability of Error & Decision Boundaries

Mark Gales

Michaelmas 2021

Vowel Classification Using Formant Features



- No separation of classes given observation $\boldsymbol{x}^\star$ (uncertainty)
  - there is no “perfect” decision

Supervised Training and Evaluation Data

- Interested in supervised training data for classification

$$\mathcal{D} = \{\{\boldsymbol{x}_1, y_1\}, \ldots, \{\boldsymbol{x}_N, y_N\}\}$$

  - $\boldsymbol{x}_i$: the observation, feature vector
  - $y_i \in \{\omega_1, \ldots, \omega_K\}$: class label for observation $\boldsymbol{x}_i$

- Samples are "draws" from joint distribution $p(\omega, \boldsymbol{x})$
- "Standard" process for obtaining data for a task
  - obtain set of observations (features) $\boldsymbol{x}_i$ and labels y_i

$$\boldsymbol{x}_i \sim p(\boldsymbol{x}); \quad y_i \sim P(\omega|\boldsymbol{x}_i)$$

- BUT care about performance on unseen data: split data
  - training data: used to train the model parameters
  - evaluation data: used to evaluate model on unseen data

Expected Loss

- Given any decision, there will be an associated "Loss"
  - useful for any system if this value is known! Usually

$$\mathcal{L}_{\texttt{act}} = \int \left[\sum_{i=1}^{K} \mathcal{L}(f(\boldsymbol{x}, \boldsymbol{\theta}), \omega_i) P(\omega_i|\boldsymbol{x}) \right] p(\boldsymbol{x}) d\boldsymbol{x} \quad \geq \quad \mathcal{L}_{\texttt{emp}} = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(f(\boldsymbol{x}_i, \boldsymbol{\theta}), y_i)$$

  - $f(\boldsymbol{x}, \boldsymbol{\theta})$ is the prediction given model parameters $\boldsymbol{\theta}$
  - as $N \to \infty$ empirical and actual losses the same $\mathcal{L}_{\texttt{emp}} \to \mathcal{L}_{\texttt{act}}$

- Simple approach is a held-out evaluation set of $N^\star$ samples
  - observation $\boldsymbol{x}_i^\star$, and label, $y_i^\star \in \{\omega_1, \dots, \omega_K\}$, pairs
  - draws from $\boldsymbol{x}_i^\star \sim p(\boldsymbol{x})$ followed by $y_i^\star \sim P(\omega|\boldsymbol{x}_i^\star)$
  - use this data to compute held-out empirical loss $\mathcal{L}_{\texttt{eval}}$

$$\mathcal{L}_{\texttt{eval}} = \frac{1}{N^\star} \sum_{i=1}^{N^\star} \mathcal{L}(f(\boldsymbol{x}_i^\star, \boldsymbol{\theta}), y_i^\star) \approx \mathcal{L}_{\texttt{act}}$$

Bayes' Decision Rule

- Consider making a decision with classifier $f(\boldsymbol{x}^\star, \boldsymbol{\theta})$
  - let $f(\boldsymbol{x}^\star, \boldsymbol{\theta}) = \hat{\omega}$, and assume followig loss function

$$\mathcal{L}(\hat{\omega}, \omega_i) = \begin{cases} 0, & \hat{\omega} = \omega_i \\ 1, & \text{otherwise} \end{cases}$$

- Apply Bayes decision rule
  - make decision that minimises loss (probability of error)

$$\begin{aligned} \hat{\omega} &= f(\boldsymbol{x}^\star, \boldsymbol{\theta}) \\ &= \arg\min_{\omega} \left\{ \sum_{i=1}^{K} \mathcal{L}(\omega, \omega_i) P(\omega_i | \boldsymbol{x}^\star; \boldsymbol{\theta}) \right\} \\ &= \arg\max_{\omega} \left\{ P(\omega | \boldsymbol{x}^\star; \boldsymbol{\theta}) \right\} \end{aligned}$$

  - select the class with the highest posterior
  - but need to train $\boldsymbol{\theta}$ to obtain $P(\omega | \boldsymbol{x}^\star; \boldsymbol{\theta})$

Two-Class (Binary) Classification

- Consider a simple binary classification task
  - class ω_1 has label 1, class ω_2 has label -1
  - prediction from $f(\boldsymbol{x}_i^\star, \boldsymbol{\theta})$, $\hat{\omega}$, is either 1 or -1
  - 1/0 loss function

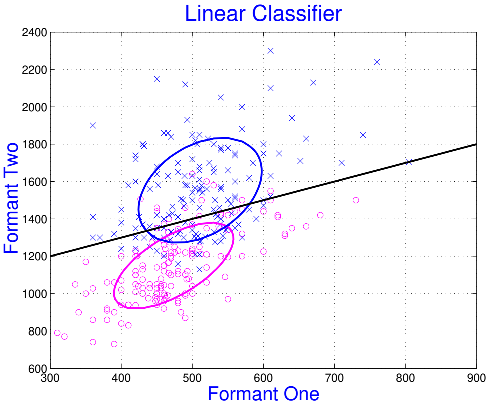
$$\mathcal{L}(\hat{\omega}, \omega_j) = \begin{cases} 0, & \hat{\omega} = \omega_j \\ 1, & \text{otherwise} \end{cases}$$

- Estimate of probability of error (equal "loss") becomes

$$\begin{aligned} P(\text{error}) \approx \mathcal{L}_{\texttt{eval}} &= \frac{1}{N^\star} \sum_{i=1}^{N^\star} \mathcal{L}(f(\boldsymbol{x}_i^\star, \boldsymbol{\theta}), y_i^\star) \\ &= \frac{1}{2N^\star} \sum_{i=1}^{N^\star} |f(\boldsymbol{x}_i^\star, \boldsymbol{\theta}) - y_i^\star| \end{aligned}$$

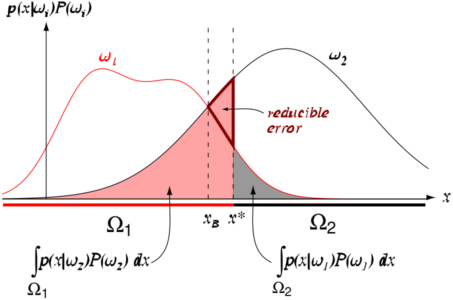
  - count the number of errors, divides by total number of samples

Binary Classification



- Classifier partitions feature space into regions
  - for each region there is an associated label
  - simple linear decision boundary illustrated above

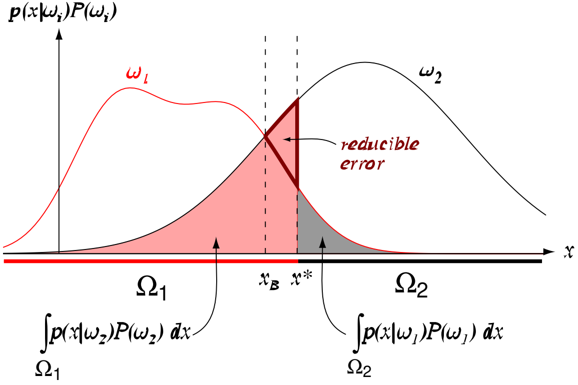
True Probability of Error



- $f(\boldsymbol{x}, \boldsymbol{\theta})$ yields two regions $\Omega_1 \to \omega_1$, and $\Omega_2 \to \omega_2$,

$$\begin{aligned}
P(\text{error}) &= P(\boldsymbol{x} \in \Omega_2, \omega_1) + P(\boldsymbol{x} \in \Omega_1, \omega_2) \\
&= P(\boldsymbol{x} \in \Omega_2|\omega_1)P(\omega_1) + P(\boldsymbol{x} \in \Omega_1|\omega_2)P(\omega_2) \\
&= \int_{\Omega_2} p(\boldsymbol{x}|\omega_1)P(\omega_1)d\boldsymbol{x} + \int_{\Omega_1} p(\boldsymbol{x}|\omega_2)P(\omega_2)d\boldsymbol{x}
\end{aligned}$$

Probability of Error Visualisation (from DHS)



- Optimal decision boundary at $\boldsymbol{x}_{\mathrm{B}}$ (suboptimal at $\boldsymbol{x}^{\star}$)
  - no lower probability of error can be obtained
  - same decision boundary as Bayes' decision rule

Nature of classifier

- Need to decide the form of classifier (with parameters $\boldsymbol{\theta}$)
- Classifiers for making these decisions can be broadly split as:
  - Generative models: model the joint distribution of observations and classes is trained, $p(\boldsymbol{x}, \omega; \boldsymbol{\theta})$.
    The posterior of class ω_i is then obtained from Bayes' rule

$$P(\omega_i|\boldsymbol{x}^\star; \boldsymbol{\theta}) = \frac{p(\boldsymbol{x}^\star, \omega_i; \boldsymbol{\theta})}{\sum_{j=1}^{K} p(\boldsymbol{x}^\star, \omega_j; \boldsymbol{\theta})}$$

  - Discriminative models: a model of the posterior distribution of the class given the observation is trained, $P(\omega|\boldsymbol{x}^\star; \boldsymbol{\theta})$.
  - Discriminant functions: a mapping from an observation $\boldsymbol{x}^\star$ to a class ω is directly trained. No posterior probability, $P(\omega|\boldsymbol{x})$, generated just class label.

See Bishop for a discussion of the merits of these.

Probability of Error Visualisation

- Probability of error marginalises over joint distribution

$$P(\text{error}) = \int_{\Omega_2} p(\boldsymbol{x}, \omega_1) d\boldsymbol{x} + \int_{\Omega_1} p(\boldsymbol{x}, \omega_2) d\boldsymbol{x}$$

  - can be expressed as (generative model)

$$P(\text{error}) = \int_{\Omega_2} p(\boldsymbol{x}|\omega_1) P(\omega_1) d\boldsymbol{x} + \int_{\Omega_1} p(\boldsymbol{x}|\omega_2) P(\omega_2) d\boldsymbol{x}$$

  - can also be expressed as (discriminative model)

$$P(\text{error}) = \int_{\Omega_2} P(\omega_1|\boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x} + \int_{\Omega_1} P(\omega_2|\boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x}$$

Unequal loss function

- Loss function may be unequal for particular applications
  - consider speaker verification for bank account access
- Consider the following loss for a binary task
  - here correct classification has zero loss

$$\mathcal{L}(f(\boldsymbol{x}^\star,\boldsymbol{\theta}),\omega_1) = \begin{cases} 0, & f(\boldsymbol{x}^\star,\boldsymbol{\theta}) = \omega_1 \\ \mathcal{C}_{21}, & f(\boldsymbol{x}^\star,\boldsymbol{\theta}) = \omega_2 \end{cases}$$

$$\mathcal{L}(f(\boldsymbol{x}^\star,\boldsymbol{\theta}),\omega_2) = \begin{cases} \mathcal{C}_{12}, & f(\boldsymbol{x}^\star,\boldsymbol{\theta}) = \omega_1 \\ 0, & f(\boldsymbol{x}^\star,\boldsymbol{\theta}) = \omega_2 \end{cases}$$

- Apply Bayes' decision rule to classification minimising loss

$$\hat{\omega} = \arg\min_{\omega}\left\{\sum_{i=1}^{2}\mathcal{L}(\omega,\omega_i)P(\omega_i|\boldsymbol{x}^\star;\boldsymbol{\theta})\right\}$$

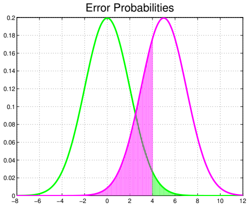
Receiver Operating Characteristics Curve

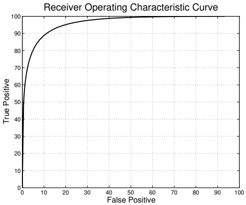
- This decision rule can be expressed as classifying $\boldsymbol{x}^\star$ using

$$\frac{\mathcal{C}_{21} P(\omega_1|\boldsymbol{x}^\star;\boldsymbol{\theta})}{\mathcal{C}_{12} P(\omega_2|\boldsymbol{x}^\star;\boldsymbol{\theta})} \overset{\omega_1}{\underset{\omega_2}{\gtrless}} 1, \qquad \frac{P(\omega_1|\boldsymbol{x}^\star;\boldsymbol{\theta})}{P(\omega_2|\boldsymbol{x}^\star;\boldsymbol{\theta})} \overset{\omega_1}{\underset{\omega_2}{\gtrless}} \frac{\mathcal{C}_{12}}{\mathcal{C}_{21}}$$

  - the "cost" ratio $\mathcal{C}_{12}/\mathcal{C}_{21}$ is effectively an operating threshold

- Possible to produce curves by changing this threshold
  - ω_1 = positive, ω_2 = negative, equal priors $P(\omega_1) = P(\omega_2)$





Decision Boundaries

Decision Process

- From Bayes decision rule for point $\boldsymbol{x}^\star$
  - assuming equal loss

$$\hat{\omega} = \arg\max_{\omega} \left\{ P(\omega|\boldsymbol{x}^\star;\boldsymbol{\theta}) \right\}$$

  - we can select the form of distribution
  - but we don't know the model parameters $\boldsymbol{\theta}$ ...

- Need to find $\boldsymbol{\theta}$
  - this is the model that we train - use supervised training

$$\mathcal{D} = \left\{ \{\boldsymbol{x}_1, y_1\}, \ldots, \{\boldsymbol{x}_N, y_N\} \right\}$$

  - draws from $\boldsymbol{x}_i \sim p(\boldsymbol{x})$ followed by $y_i \sim P(\omega|\boldsymbol{x}_i)$
  - use this data to estimate $\boldsymbol{\theta}$

Maximum Likelihood Estimation

- Need to get the "best" set of model parameters
  - find the parameters most-likely to generate the training labels

(Conditional) Maximum Likelihood Estimation (MLE)

$$\begin{aligned}\hat{\boldsymbol{\theta}} &= \arg\max_{\boldsymbol{\theta}} \left\{\log\left(P(y_1,\ldots,y_N|\boldsymbol{x}_1,\ldots,\boldsymbol{x}_N;\boldsymbol{\theta})\right)\right\} \\ &= \arg\max_{\boldsymbol{\theta}} \left\{\sum_{i=1}^{N}\log\left(P(y_i|\boldsymbol{x}_i;\boldsymbol{\theta})\right)\right\}\end{aligned}$$

  - assumes that all training samples are independent

- Can also be applied to distribution estimation
  - unsupervised training case

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \left\{\sum_{i=1}^{N}\log\left(p(\boldsymbol{x}_i;\boldsymbol{\theta})\right)\right\}$$

Generative Models

- Use generative model to approximate $P(\omega|\boldsymbol{x}^\star)$
  - generative model, parameters $\boldsymbol{\theta}$, joint distribution $p(\boldsymbol{x}, \omega)$
  - posterior distribution of class ω_i for unseen observation $\boldsymbol{x}^\star$

$$P(\omega_i|\boldsymbol{x}^\star) \approx \frac{p(\boldsymbol{x}^\star, \omega_i; \boldsymbol{\theta})}{\sum_{j=1}^{K} p(\boldsymbol{x}^\star, \omega_j; \boldsymbol{\theta})} = \frac{p(\boldsymbol{x}^\star|\omega_i; \boldsymbol{\theta}) P(\omega_i)}{\sum_{j=1}^{K} p(\boldsymbol{x}^\star|\omega_j; \boldsymbol{\theta}) P(\omega_j)}$$

  - $P(\omega_i)$ is the prior for class ω_i
  - $p(\boldsymbol{x}^\star|\omega_i; \boldsymbol{\theta})$ is the likelihood of the observation given class ω_i

- Use maximum likelihood (ML) training to estimate $\boldsymbol{\theta}$
  - separate model trained for each class ω_i, $\boldsymbol{\theta}_i$

$$\hat{\boldsymbol{\theta}}_i = \arg\max_{\boldsymbol{\theta}} \left\{ \sum_{j: y_j = \omega_i} \log\left(p(\boldsymbol{x}_j|\omega_i; \boldsymbol{\theta})\right) \right\}$$

Multivariate Gaussian Class Conditional PDFs

- Need to decide on form and parameters of $p(\boldsymbol{x}|\omega_i;\boldsymbol{\theta})$
  - use multivariate Gaussian distribution as class-conditional PDF

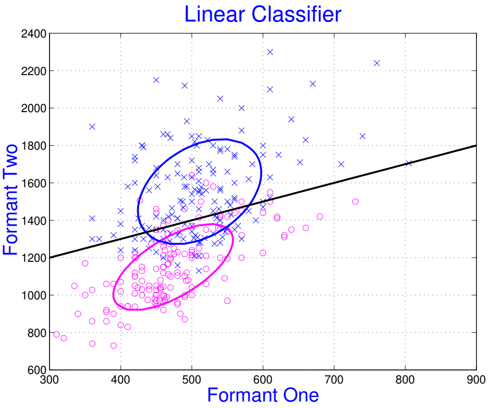
$$p(\boldsymbol{x}|\omega_i;\boldsymbol{\theta}_i) = \mathcal{N}(\boldsymbol{x};\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i) = \frac{1}{(2\pi)^{\frac{d}{2}}|\boldsymbol{\Sigma}_i|^{\frac{1}{2}}}\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^{\mathsf{T}}\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)\right)$$

  - $\boldsymbol{x}$: the d-dimensional observation (input) vector
  - $\boldsymbol{\mu}_i$: the d-dimensional mean vector for class ω_i
  - $\boldsymbol{\Sigma}_i$: the $d \times d$ covariance matrix for class ω_i

- Maximum likelihood estimates of the parameters given by

$$\begin{aligned} \hat{\boldsymbol{\mu}}_i &= \frac{\sum_{j:y_j=\omega_i}\boldsymbol{x}_j}{\sum_{j:y_j=\omega_i}1} \\ \hat{\boldsymbol{\Sigma}}_i &= \frac{\sum_{j:y_j=\omega_i}(\boldsymbol{x}_j-\hat{\boldsymbol{\mu}}_i)(\boldsymbol{x}_j-\hat{\boldsymbol{\mu}}_i)^{\mathsf{T}}}{\sum_{j:y_j=\omega_i}1} \end{aligned}$$

Two-Class (Binary) Decision Boundaries



- Decision boundary is (hyper-)plane between class labels
  - for two-class task when the class probabilities the same

Decision Boundaries Equation

- Boundary occurs when (log) class posteriors are the same
  - a point $\boldsymbol{x}$ on the decision boundary (2 class) satisfies

$$\log(P(\omega_1|\boldsymbol{x};\boldsymbol{\theta})) = \log(P(\omega_2|\boldsymbol{x};\boldsymbol{\theta}))$$

  - for a generative classifier (denominator cancels) this yields

$$\log(P(\omega_1)p(\boldsymbol{x}|\boldsymbol{\omega}_1;\boldsymbol{\theta}_1)) = \log(P(\omega_2)p(\boldsymbol{x}|\boldsymbol{\omega}_2;\boldsymbol{\theta}_2))$$

- Consider multivariate Gaussian class-conditional PDFs
  - Gaussian parameters: Class $\omega_1 : \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1$, Class $\omega_2 : \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2$
  - prior parameters: Class $\omega_1 : P(\omega_1)$, Class $\omega_2 : P(\omega_2)$

What form does the decision boundary take?

Multivariate Gaussian Decision Boundaries (cont)

- Substituting for class-conditional PDFs and simplifying yields

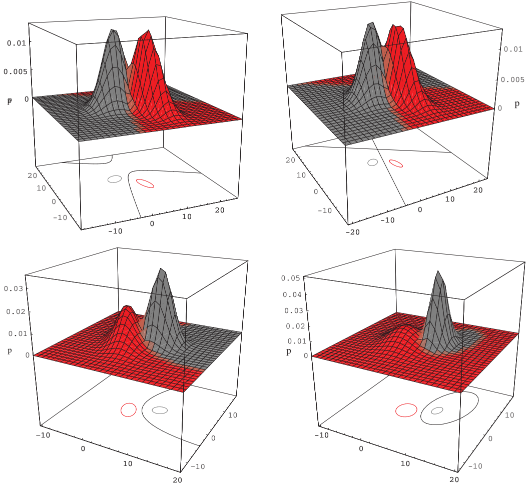
$$\boldsymbol{x}^{\mathsf{T}}\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}^{\mathsf{T}}\boldsymbol{x} + c = 0$$

  where

  - $\boldsymbol{A} = \boldsymbol{\Sigma}_1^{-1} - \boldsymbol{\Sigma}_2^{-1}$
  - $\boldsymbol{b} = 2\left(\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2 - \boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1\right)$
  - $c = \boldsymbol{\mu}_1^{\mathsf{T}}\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2^{\mathsf{T}}\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2 - \log\left(\frac{|\boldsymbol{\Sigma}_2|}{|\boldsymbol{\Sigma}_1|}\right) - 2\log\left(\frac{P(\omega_1)}{P(\omega_2)}\right)$

- General form yields hyper-quadratic decision boundaries
- Constrain covariance matrices to be the same $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}$
  - results in a linear decision boundary

$$\boldsymbol{b}^{\mathsf{T}}\boldsymbol{x} + c = 0$$


0.01
0.005
0
p
20
10
0
-10
-10
0
10
20
0.01
0.005
0
p
20
10
0
-10
-20
-10
0
10
20
0.03
0.02
0.01
0
p
-10
0
10
20
10
0
-10
0.05
0.04
0.03
0.02
0.01
0
p
-10
0
10
20
10
0
-10

How good is the Classifier?

- Bayes' decision rule minimises the loss (probability of error)
  - but relies on the estimates of class posteriors $P(\omega|\boldsymbol{x}^\star;\boldsymbol{\theta})$
  - how close to optimal depends on the accuracy of this estimate
- Class posterior estimates depend on:
  - training data ($\mathcal{D}$): quantity (infinite!), "matched" to test data
  - form of model: is the model "correct"?
  - optimisation: have the global optimal parameters been found
- In practice none of these are usually true ...
  - many design decisions - good engineering/maths required