INF 4300
17.10.12

Introduction to classifiction

Anne Solberg (anne@ifi.uio.no)

Based on Chapter 2 (2.1-2.6) in Duda and Hart: Pattern Classification

Introduction to classification

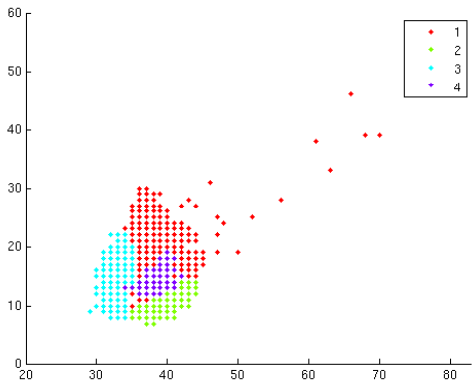
- One of the most challenging topics in image analysis is recognizing a specific object in an image. To do that, several steps are normally used. Some kind of segmentation can delimit the spatial extent of the foreground objects, then a set of features to describe the object characteristics are computed, before the recognition step that decides which object type this is.
- The focus of the next three lectures is the recognition step. The starting point is that we have a set of K features computed for an object. These features can either describe the shape or the gray level/color properties of the object. Based on these features we need to decide what kind of an object this is.
- Statistical classification is the process of estimating the probability that an object belongs to one of S object classes based on the observed value of K features. Classification can be done both unsupervised or supervised. In unsupervised classification the categories or classes are not known, and the classification process with be based on grouping similar objects together. In supervised classification the categories or classes are known, and the classification will of consist of estimating the probability that the object belongs to each of the S object classes. The object is assigned to the blass with highest probability. For supervised classifcation, training data is needed. Training data consists of a set of objects with know class type, and they are used to estimate the parameters of the classifier.
- The performance of a classifier is normally computed as the accuracy it gets when classifying a different set of objects with known class labels called the test set.

- Assume that each object *i* in the scene is represented by a feature vector $\mathbf{x}_i$.
- If we have computed K features, $\mathbf{x}_i$ will be a vector with K components:

$$\mathbf{x}_i = \begin{Bmatrix} x_i^1 \\ \vdots \\ x_i^K \end{Bmatrix}$$

- A classifier that is based on K features is called a multivariate classifier.
- The K features and the objects in the training data set will define a K-dimensional feature space.
- The training data set is a set of objects with known class labels.
- As we saw last week, we can visualize class separation on data with known class labels if we have one or two features, but visualizing multidimensional space is difficult.

- In the scatter plot, each object in the training data set is plotted in K-dimensional space at a position relative to the value of the K features.
- Each object is represented by a dot, and the color of the dot represents the class. In this example we have 4 classes visualize using 2 features.
- A scatter plot is a tool for visualizing features. We will later look at other class separability measures.



Concepts in classification

- In the following three lectures we will cover these topics related to classification:
  - Training set
  - Test set
  - Classifier accuracy/confusion matrices.
  - Computing the probability that an object belongs to a class.
    - Let each class be represented by a probability density function. In general many probability densities can be used but we use the multivariate normal distribution which is commonly used.
  - Bayes rule
  - Discriminant functions/Decision boundaries
  - Normal distribution, mean vector and covariance matrices
  - kNN classification
  - Unsupervised classification/clustering

Introduction to classification

- Supervised classification is related to thresholding
  - Divide the image into two classes: foreground and background
- Thresholding is a two-class classification problem based on a 1D feature vector
  - The feature vector consist of only the grey level f(x,y)
- How can we classify a feature vector of K shape features into correct character type?
- We will now study multivariate classification theory where we use N features to determine if an object belongs to a set of K object classes.
- Recommended additional reading:
  - Pattern Classification, R. Duda, P. Hart and D. Stork.
    - Chapter 1 Introduction
    - Chapter 2 Bayesian Decision Theory, 2.1-2.6
    - **See ~inf3300/www_docs/bilder/dudahart_chap2.pdf and dudahart-appendix.pdf**

Plan for this lecture:

- Explain the relation between thresholding and classification with 2 classes
- Background in probability theory
- Bayes rule
- Classification with a Gaussian density and a single feature
- Briefly: training and testing a classifier
  - We og deeper into this in two weeks.

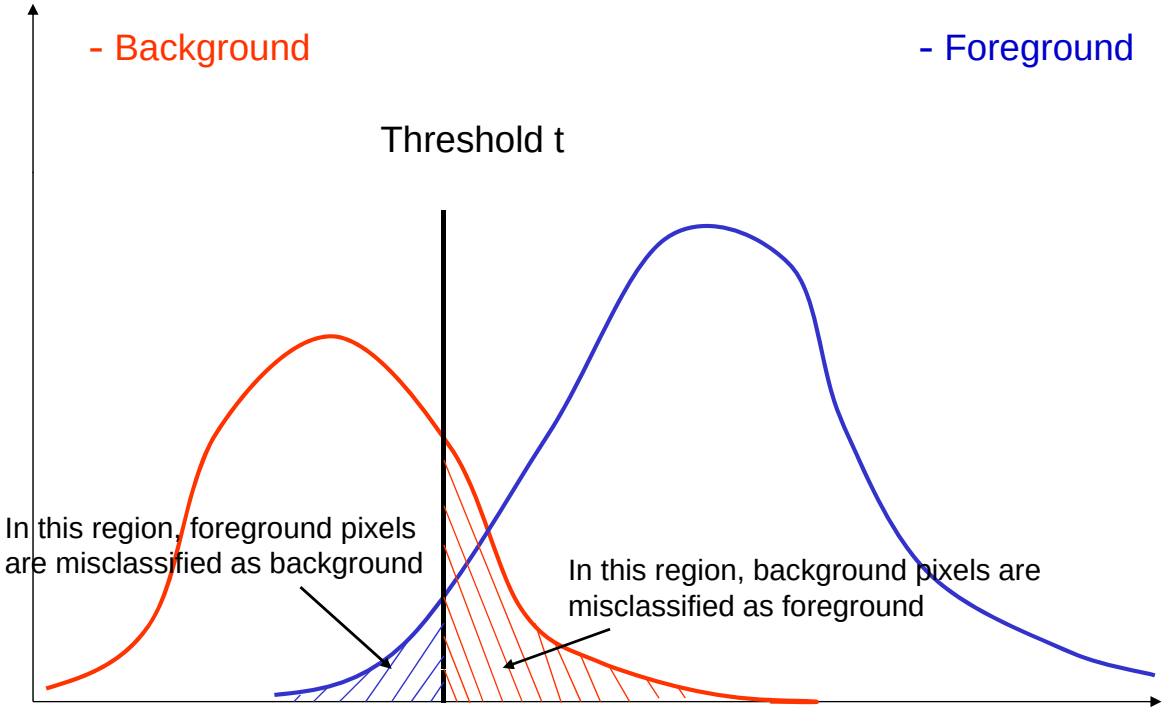
From INF2310: Thresholding

- Basic thresholding assigns all pixels in the image to one of 2 classes: foreground or background

$$g(x,y) = \begin{cases} 0 \text{ if } & f(x,y) \le T \\ 1 \text{ if } & f(x,y) > T \end{cases}$$

- This can be seen as a 2-class classification problem based on a single feature, the gray level.
- The 2 classes are background and foreground, and the threshold T defines the border between them.

Classification error for thresholding



Classification error for thresholding

- We assume that b(z) is the normalized histogram for background *b(z)* and *f(z)* is the histogram for foreground.
- The histograms are estimates of the probability distribution of the gray levels in the image.
- Let *F* and *B* be the prior probabilities for background and foreground(*B+F=1*)
- The normalized histogram for the image is then given by

$$p(z) = B \cdot b(z) + F \cdot f(z)$$

- The probability for misclassification given a treshold t is:

$$E_B(t) = \int_{-\infty}^{t} f(z)dz$$

$$E_F(t) = \int_{t}^{\infty} b(z)dz$$

Find T that minimizes the error

$$E(t) = F\int_{-\infty}^{t} f(z)dz + B\int_{t}^{\infty} b(z)dz$$

$$\frac{dE(t)}{dt} = 0 \Rightarrow F \cdot f(T) = B \cdot b(T)$$

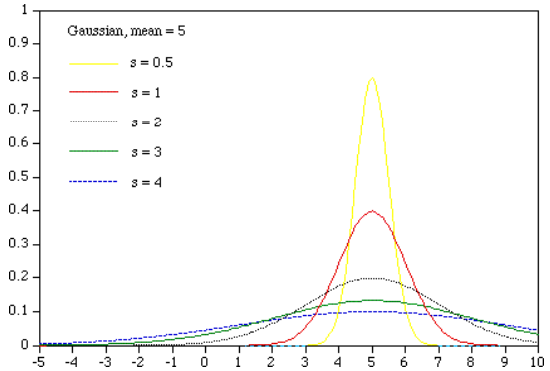
Minimum error is achieved by setting T equal to the point where the probabilities for foreground and background are equal.

Distributions, standard deviation and variance

- A Gaussian distribution (normal distribution) is specified given the mean value μ and the variance σ^2:

$$p(z) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- Variance σ^2, standard deviation σ



Two Gaussian distributions for a single feature

- Assume that *b(z)* and *f(z)* are Gaussian distributions, then

$$p(z) = \frac{B}{\sqrt{2\pi\sigma_B^{\,2}}} e^{-\frac{(x-\mu_B)^2}{2\sigma_B^{\,2}}} + \frac{F}{\sqrt{2\pi\sigma_F^{\,2}}} e^{-\frac{(x-\mu_F)^2}{2\sigma_F^{\,2}}}$$

- μ_B and μ_F are the mean values for background and foreground.
- σ_B^2 and σ_F^2 are the variance for background and foreground.

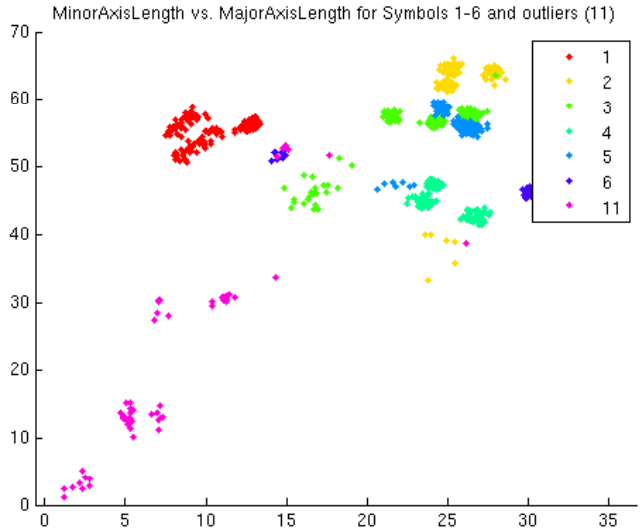
The 2-class classification problem summarized

- Given two Gaussian distributions *b(z)* and *f(z)*.
- The classes have prior probabilities F and B.
- Every pixel should be assigned to the class that minimizes the classification error.
- The classification error is minimized at the point where F f(z) = B b(z).

- What we will do now is to generalize to K classes and D features.

How do we find the best border beteen K classes with 2 features?

- We will find the theoretical answer and a geometrical interpretation of class means, variance, and the equivalent of a threshold.



The goal of classification

- We estimate the decision boundaries based on training data.
- Classification performance is always estimated on a separate ”test” data set.
  - We try to measure the generalization performance.
- The classifier should perform well when classifying new samples
  - Have lowest possible classification error.
- We often face a tradeoff between classification error on the training set and generalization ability when determining the complexity of the decision boundary.

Probability theory - Appendix A.4

- Let x be a discrete random variable that can assume any of a finite number of M different values.

- The probability that x belongs to class i is
  p_i = Pr(x=i), i=1,...M
- A probability distribution must sum to 1 and probabilities must be positive so $p_i \geq 0$ and $\sum_{i=1}^{M} p_i = 1$

Expected values - definition

- The expected value or mean of a random variable x is:

$$E[x] = \mu = \sum_x xP(x) = \sum_{i=1}^{M} ip_i$$

- The variance or second order moment σ^2 is:

$$E[x^2] = \mu = \sum_x x^2 P(x)$$

$$Var[x] = \sigma^2 = E[(x-u)^2] = \sum_x (x-u)^2 P(x)$$

- These will be estimated from training data where we know the true class labels (foil 39 for the univariate case).

Pairs of random variables

- Let x and y be random variables.
- The joint probability of observing a pair of values (x=i,y=j) is p_{ij}.
- Alternatively we can define a joint probability distribution function P(x,y) for which

$$P(x,y) \geq 0, \quad \sum_x \sum_y P(x,y) = 1$$

- The marginal distributions for x and y (if we want to eliminate one of them) is:

$$P_x(x) = \sum_y P(x,y)$$

$$P_y(y) = \sum_x P(x,y)$$

Statistical independence
Expected values of two variables

- Variables x and y are statistical independent if and only if $P(x,y) = P_x(x)P_y(y)$
- Two variables are uncorrelated if

$$\sigma_{xy} = 0$$

- Expected values of two variables:

$$E(f(x,y)) = \sum_x \sum_y f(x,y)P(x,y)$$

$$\mu_x = E(x) = \sum_x \sum_y xP(x,y)$$

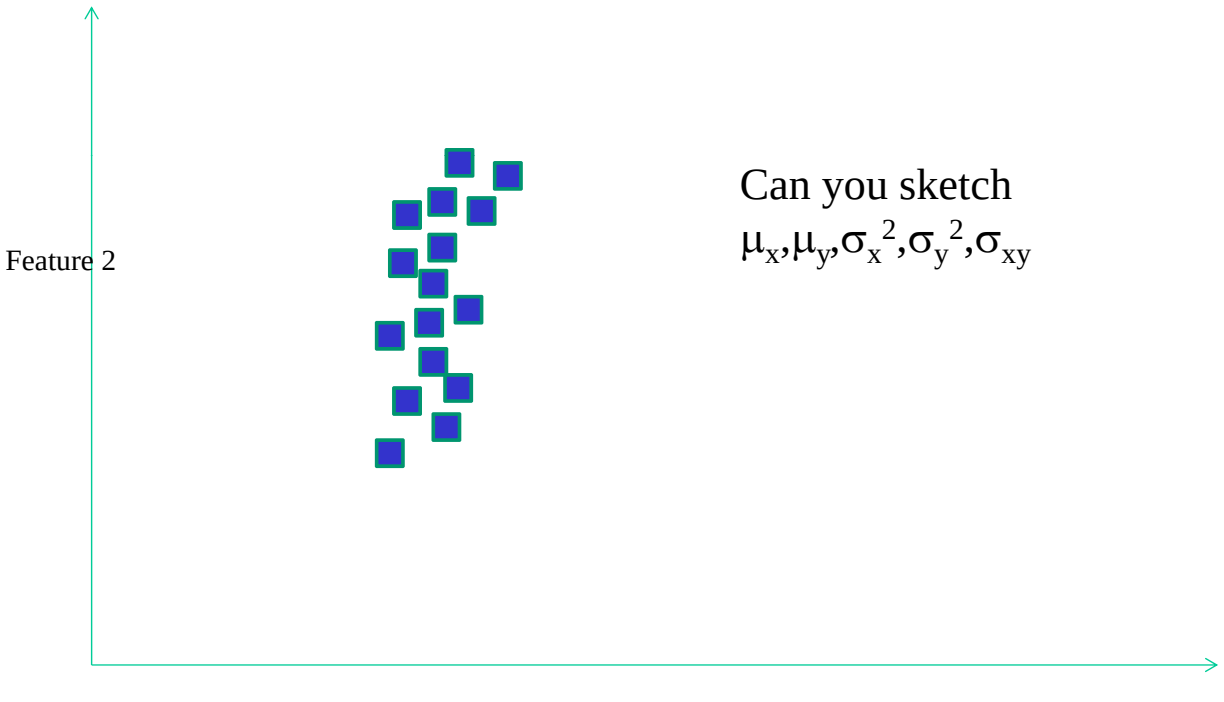
$$\mu_y = E(y) = \sum_x \sum_y yP(x,y)$$

$$\sigma_x^2 = E\left[(x-\mu_x)^2\right] = \sum_x \sum_y (x-\mu_x)^2 P(x,y)$$

$$\sigma_y^2 = E\left[(y-\mu_y)^2\right] = \sum_x \sum_y (y-\mu_y)^2 P(x,y)$$

$$\sigma_{xy} = E\left[(x-\mu_x)(y-\mu_y)\right] = \sum_x \sum_y (x-\mu_x)(y-\mu_y)P(x,y)$$

Where (in this course) have you seen similar formulas?



Can you sketch
$\mu_x, \mu_y, \sigma_x^2, \sigma_y^2, \sigma_{xy}$

Conditional probability

- If two variables are statistically dependent, knowing the value of one of them lets us get a better estimate of the value of the other one.
- The conditional probability of *x* given *y* is:

$$\Pr[x = i \mid y = j] = \frac{\Pr[x = i, y + j]}{\Pr[y = j]}$$

and for distributions:

$$P(x \mid y) = \frac{P(x, y)}{P(y)}$$

- Example: Threshold a page with dark text on white background
- x is the grey level of a pixel, and y is its class (F or B).
- If we consider which grey levels x can have - we expect small values if x is text (y=F), and large values if x is background (y=B).

Expected values of M variables

- Using vector notation:

$$\boldsymbol{\mu} = E[\mathbf{x}] = \sum_{x} \mathbf{x} P(\mathbf{x})$$

$$\boldsymbol{\Sigma} = E\left[(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T\right]$$

Bayesian decision theory

- A fundamental statistical approach to pattern classification.
- Named after Thomas Bayes (1702-1761), an english priest and matematician.
- It combines prior knowledge about the problem with a probability distribution function.
- The most central concept is Bayes decision rule.

Bayes rule in general

- The equation:

$$P(x \mid y) = \frac{P(y \mid x)P(x)}{\sum_x P(y \mid x)P(x)} = \frac{P(y \mid x)P(x)}{P(y)}$$

- In words:

$$posterior = \frac{likelihood \times prior}{evidence}$$

- y are observations, x is the unknown class labels.
- We want to find the most probable class x given the observations y.
- To be explained for the classification problem later :-)

Mean vectors and covariance matrices in N dimensions

- If f(**x**) is a n-dimensional feature vector, we can formulate its mean vector and covariance matrix as:

$$f(\mathbf{x}) = \begin{bmatrix} f_1(x) \\ f_2(x) \\ . \\ . \\ f_n(x) \end{bmatrix} \qquad \boldsymbol{\mu} = E[\mathbf{x}] = \begin{bmatrix} E(x_1) \\ E(x_2) \\ . \\ . \\ E(x_n) \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ . \\ . \\ \mu_n \end{bmatrix}$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & . & . & \sigma_{1n} \\ \sigma_{21} & \sigma_{22} & . & . & \sigma_{2n} \\ . & . & . & . & . \\ . & . & . & . & . \\ \sigma_{n1} & \sigma_{n2} & . & . & \sigma_{nn} \end{bmatrix} = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & . & . & \sigma_{1n} \\ \sigma_{21} & \sigma_2^2 & . & . & \sigma_{2n} \\ . & . & . & . & . \\ . & . & . & . & . \\ \sigma_{n1} & \sigma_{n2} & . & . & \sigma_n^2 \end{bmatrix}$$

with n features, the mean vector μ will be of size 1xn and Σ or size nxn.

- The matrix will be symmetric as $\sigma_{kl} = \sigma_{lk}$

Bayes rule for a classification problem

- Suppose we have J, j=1,...J classes. ω is the class label for a pixel, and x is the observed gray level (or feature vector).
- We can use Bayes rule to find an expression for the class with the highest probability:

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

$$\text{posterior probability} = \frac{likelihood \times \text{prior probability}}{\text{normalizing factor}}$$

- For thresholding, $P(\omega_j)$ is the prior probability for background or foreground. If we don't have special knowledge that one of the classes occur more frequent than other classes, we set them equal for all classes. ($P(\omega_j)=1/J$, j=1.,,,J).
- Small p means a probability distribution
- Capital P means a probability (scalar value between 0 and 1)

Bayes rule explained

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

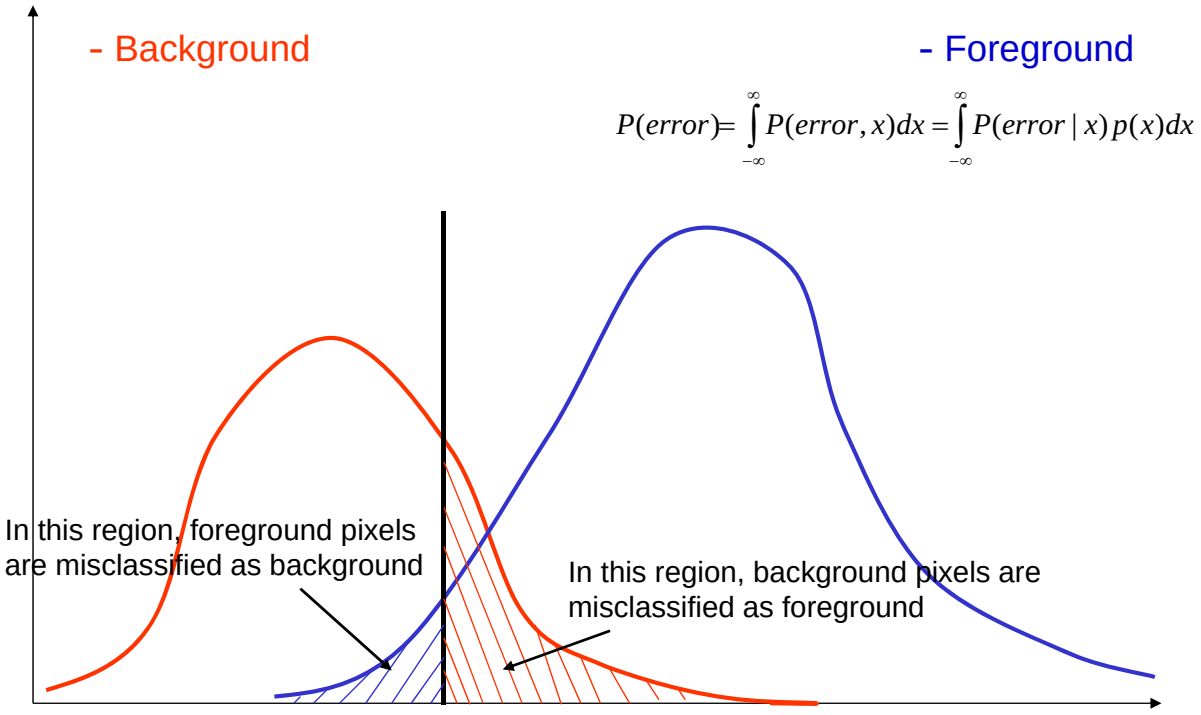
- $p(x|\omega_j)$ is the probability density function that models the likelihood for observing gray level x if the pixel belongs to class ω_j.
  - Typically we assume a type of distribution, e.g. Gaussian, and the mean and covariance of that distribution is fitted to some data that we know belong to that class. This fitting is called classifier training.
- $P(\omega_j|x)$ is the posterior probability that the pixel actually belongs to class ω_j. We will soon se that the the classifier that achieves the minimum error is a classifier that assigns each pixel to the class ω_j that has the highest posterior probability.
- p(x) is just a scaling factor that assures that the probabilities sum to 1.

Probability of error

- If we have 2 classes, we make an error either if we decide ω_1 if the true class is ω_2 if we decide ω_2 if the true class is ω_1.
- If $P(\omega_1|x) > P(\omega_2|x)$ we have more belief that x belongs to ω_1, and we decide ω_1.
- The probability of error is then:

$$P(error \mid x) = \begin{cases} P(\omega_1 \mid x) \text{ if we decide } \omega_2 \\ P(\omega_2 \mid x) \text{ if we decide } \omega_1 \end{cases}$$

Back to classification error for thresholding



Minimizing the error

$$P(error) = \int_{-\infty}^{\infty} P(error, x)dx = \int_{-\infty}^{\infty} P(error \mid x)\, p(x)dx$$

- When we derived the optimal threshold, we showed that the minimum error was achieved for placing the threshold (or *decision border* as we will call it now) at the point where

  $P(\omega_1|x) = P(\omega_2|x)$

- This is still valid.

Bayes decision rule

- In the 2 class case, our goal of minimizing the error implies a decision rule:

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

- For *J* classes, the rule analogously extends to choose the class with *maximum a posteriori* probability
- The *decision boundary* is the"border" between classes *i* and *j*, simply where $P(\omega_i|x) = P(\omega_j|x)$
  - Exactly where the threshold was set in minimum error thresholding!

Bayes classification with J classes and D features

- How do we generalize:
  - To more the one feature at a time
  - To J classes
  - To consider loss functions (that some errors are more costly than others)

Bayes rule with c classes and d features

- If we measure d features, $\boldsymbol{x}$ will be a d-dimensional feature vector.
- Let $\{\omega_1,....,\omega_C\}$ be a set of c classes.
- The posterior probability for class c is now computed as

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

$$p(x) = \sum_{j=1}^{c} p(\mathbf{x} \mid \omega_j)P(\omega_j)$$

- Still, we assign a pixel with feature vector x to the class that has the highest posterior probability:

$$\text{Decide } \omega_1 \text{ if } P(\omega_1 \mid \mathbf{x}) \geq P(\omega_j \mid \mathbf{x}), \text{ for all } j \neq i$$

Discriminant functions

- The decision rule

Decide ω_1 if $P(\omega_1 \mid \mathbf{x}) > P(\omega_j \mid \mathbf{x})$, for all $j \neq i$

can be written as assign **x** to ω_1 if

$$g_i(\mathbf{x}) > g_j(\mathbf{x})$$

- The classifier computes J discriminant functions $g_i(\mathbf{x})$ and selects the class corresponding to the largest value of the discriminant function.
- Since classification consists of choosing the class that has the largest value, a scaling of the discriminant function $g_i(\mathbf{x})$ by $f(g_i(\mathbf{x}))$ will not effect the decision if f is a monotonically increasing function.
- This can lead to simplifications as we will soon see.

Equivalent discriminant functions

- The following choices of discriminant functions give equivalent decisions:

$$g_i(\mathbf{x}) = P(\omega_i \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid \omega_i)P(\omega_i)}{p(\mathbf{x})}$$

$$g_i(\mathbf{x}) = p(\mathbf{x} \mid \omega_i)P(\omega_i)$$

$$g_i(\mathbf{x}) = \ln p(\mathbf{x} \mid \omega_i) + \ln P(\omega_i)$$

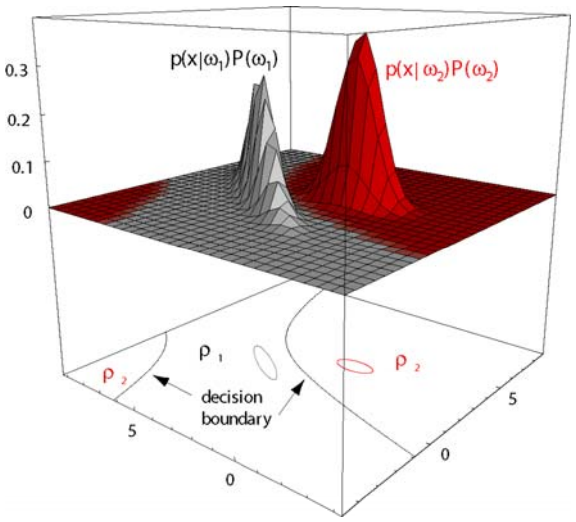
- The effect of the decision rules is to divide the feature space into c decision regions $R_1, \ldots\ldots R_c$.
- If $g_i(\mathbf{x}) > g_j(\mathbf{x})$ for all $j \neq i$, then **x** is in region R_i.
- The regions are separated by decision boundaries, surfaces in features space where the discriminant functions for two classes are equal

Decision functions - two classes

- If we have only two classes, assigning **x** to ω_1 if $g_1 > g_2$ is equivalent to using a single discriminant function: $g(\mathbf{x}) = g_1(\mathbf{x}) - g_2(\mathbf{x})$
  and decide ω_1 if $g(\mathbf{x}) > 0$
- The following functions are equivalent:

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

$$g(\mathbf{x}) = \ln \frac{p(\mathbf{x} \mid \omega_1)}{p(\mathbf{x} \mid \omega_2)}$$



The Gaussian density - univariate case (a single feature)

- To use a classifier we need to select a probability density function $p(x|\omega_i)$.
- The most commonly used probability density is the normal (Gaussian) distribution:

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]$$

with expected value (or mean) $\mu = \mathrm{E}[x] = \int_{-\infty}^{\infty} x p(x) dx$

and variance $\sigma^2 = E\left[(x-\mu)^2\right] = \int_{-\infty}^{\infty} (x-\mu)^2 p(x) dx$

Training a univariate Gaussian classifier

- To be able to compute the value of the discriminant function, we need to have an estimate of μ_j and σ_j^2 for each class j.
- Assume that we know the true class labels for some pixels and that this is given in a mask image.
- Training the classifier then consists of computing μ_j and σ_j^2 for all pixels with class label j in the mask file.
- They are computed from training data as:
- For all pixels x_i with label k in the training mask, compute

$$\mu_k = \frac{1}{N_k}\sum_{i=i}^{N_k} x_i$$

$$\sigma_k^2 = \frac{1}{N_k}\sum_{i=i}^{N_k} (x_i - \mu_k)^2$$

Classification with a univariate Gaussian

- Decide on values for the prior probabilities, $P(\omega_j)$. If we have no prior information, assume that all classes are equally probable and $P(\omega_j)=1/c$.
- Estimate μ_j and σ_j^2 based on training data based on the formulae on the previous slide.
- For class j=1,….J, compute the discriminant function

$$P(\omega_j \mid x) = p(x \mid \omega_j)P(\omega_j) = \frac{1}{\sqrt{2\pi}\sigma_j}\exp\left[-\frac{1}{2}\left(\frac{x-\mu_j}{\sigma_j}\right)^2\right]P(\omega_j)$$

- Assign pixel x to the class with the highest value of $P(\omega_j|x)$

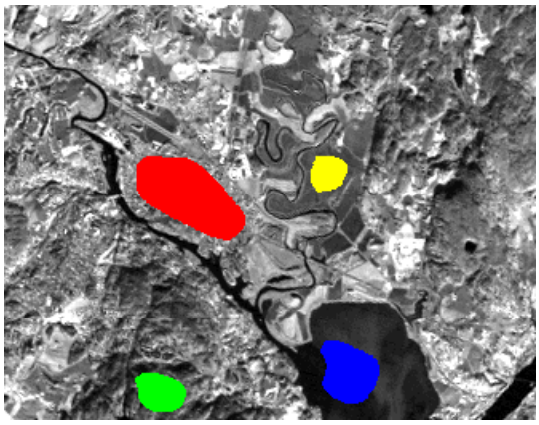
The result after classification is an image with class labels corresponding to the most probable class for each pixel.

We compute the classification error rate from an independent test mask.

Example: image and training masks

The masks contain labels for the training data.
If a pixel is not part of the training data, it will have label 0.
A pixel belonging to class k will have value k in the mask image.

We should have a similar mask for the test data.



Estimating classification error

- A simple measure of classification accuracy can be to count the percentage of correctly classified pixels overall (averaged for all classes), or per. class. If a pixel has true class label k, it is correctly classified if ω_j=k.
- Normally we use a pixels to train and test a classifier, so we have a **disjoint training mask and test mask.**
- Estimate the classification error by classifying all pixels in the test set and count the percentage of wrongly classified pixels.