INF 4300
26.09.16
Introduction to classifiction

Anne Solberg (anne@ifi.uio.no)

- From scatterplots to classification

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

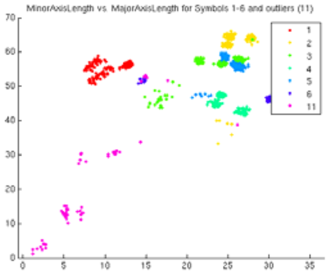
- Visualizing features using scatter plots
- Explain the relation between thresholding and classification with 2 classes
- Background in probability theory
- Bayes rule
- Classification with a Gaussian density and a single feature
  - Linear boundaries in feature space
- Briefly: training and testing a classifier

From features to discrimination between objects

- The following slides introduces simple tools as scatter plots to visualize how good a feature (or combination of 2-3 features) is in separating objects of different types/classes.
- To evaluate features, we use training data consisting of objects with KNOWN CLASS.

Scatter plots

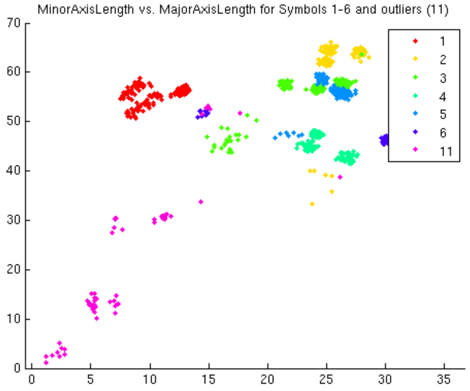
- A 2D scatter plot is a plot of feature values for two different features. Each object's feature values are plotted in the position given by the features values, and with a class label telling its object class.
- A scatter plott visualize the space spanned by 2 or more features: called the **feature space**
- Matlab: gscatter(feature1, feature2, labelvector)
- Classification is done based on *more than two features*, but this is difficult to visualize.
- Features with good class separation show clusters for each class, but different clusters should ideally be separated.



Feature 2: major axis length

Feature 1: minor axis length

Which numbers are well separated?



Aristotle and Occam

- Our search for models or hypotheses that describe the laws of nature is based on a ”minimum complexity principle”.
- Aristotle (384-322 BC), Physics, book I, chapter VI:
  ‘The more limited, if adequate, is always preferable’.
- William of Occam (1285-1349):
  ‘Pluralitas non est ponenda sine necessitate’.
- The simplest model that explains the data is the best.
- So far, “Occam’s Razor” has generally motivated
  the search and selection of reduced dimensionality feature sets.
- It should also motivate us to generate
  only a few but powerful features.
- Many practitioners have forgotten the minimum complexity principle.

The ”curse-of-dimensionality”

- Also called ”peaking phenomenon”.
- For a finite training sample size, the correct classification rate initially increases when adding new features, attains a maximum and then begins to decrease.
- The implication is that:
- For a high measurement complexity, we will need large amounts of training data in order to attain the best classification performance.
- => 5-10 samples
  per feature per class.

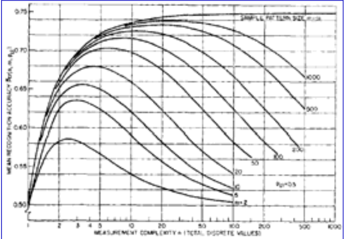


Fig. 3. Finite data set accuracy ($p_{c1} = \frac{1}{2}$).

Correct classification rate as function of feature dimensionality, for different amounts of training data. Equal prior probabilities of the two classes is assumed.

Illustration from G.F. Hughes (1968).

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 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/texture 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 class 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.
- Classification can be pixel-based or region-based.
- 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.

Concepts in classification

- In the following three lectures we will cover these topics related to classification:
  - Training set
  - Validation 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

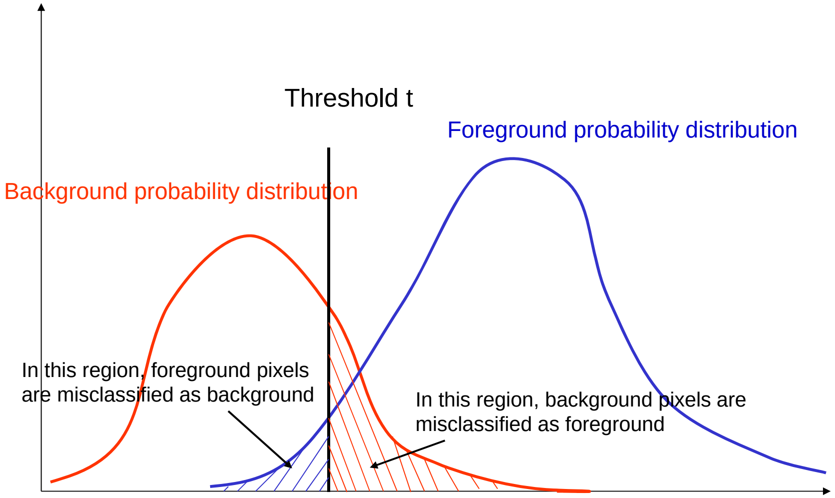
From INF2310: Thresholding

- 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


Threshold t
Foreground probability distribution
Background probability distribution
In this region, foreground pixels are misclassified as background
In this region, background pixels are misclassified as foreground

Classification error for thresholding

- We assume that b(z) is the normalized histogram for background *b(z)* and *f(z)* is the normalized 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

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

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

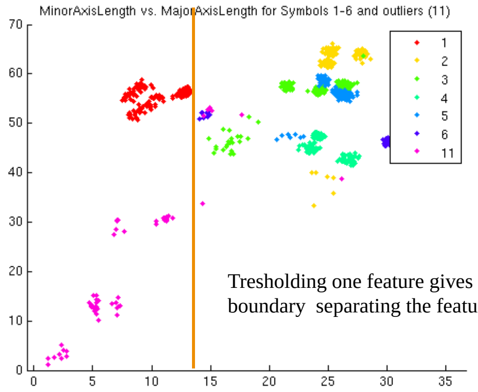
Minimum error is achieved by setting T equal to the point where the probabilities for foreground and background are equal.

The locations in feature space where the probabilities are equal is called decision boundary in classification.

The goal of classification is to find the boundaries.

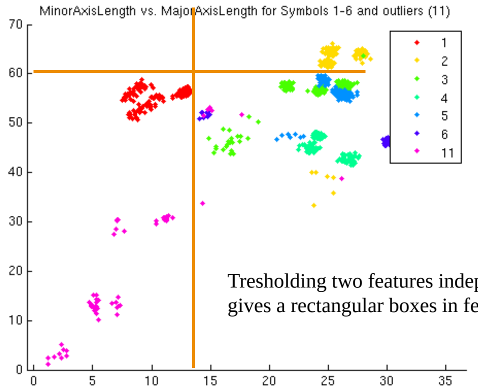
See

Partitioning the feature space using thresholding – 1 feature and 1 threshold



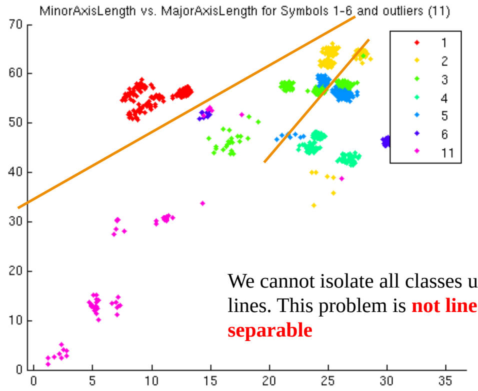
Tresholding one feature gives a linear boundary separating the feature space

Partitioning the feature space using thresholding – 2 features and 2 thresholds



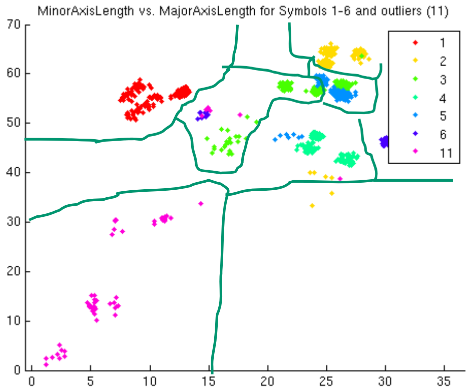
Tresholding two features independently gives a rectangular boxes in feature space

Can we find a line with better separation?



We cannot isolate all classes using straight lines. This problem is **not linearly separable**

The goal: partitioning the space with smooth boundaries

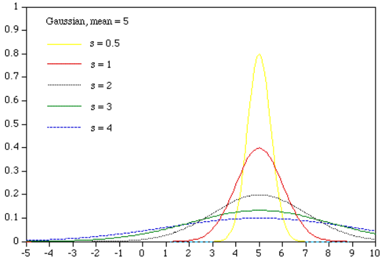


Distributions, standard deviation and variance

- A univariate (one feature) 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 $\sigma^{2,}$ standard deviation σ



Two Gaussian distributions for thresholding 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.

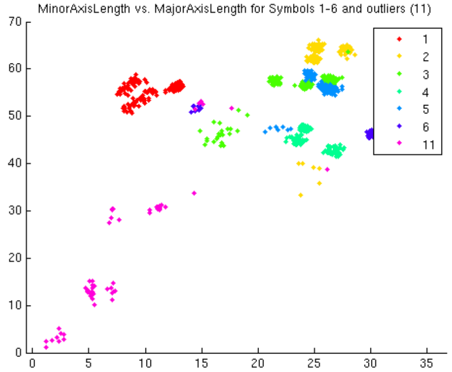
The 2-class classification problem summarized

- 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 (equivalent to the threshold for multivariate data) 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:

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

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

- These will be estimated from training data where we know the true class labels: .

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

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

Pairs of random variables - definitions

- Let *x* and *y* be two 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 - definitions

- Variables *x* and *y* are statistical independent if and only if

$$P(x,y)=P_x(x)P_y(y)$$

- In words: two variables are indepentent if the occurrence of one does not affect the other.
- If two variables are not independent, they are dependent.
- If two variables are independent, they are also uncorrelated.
- For more than two variables: all pairs must be independent.
- Two variables are uncorrelated if

$$\sigma_{xy}=0$$

- If Cov[X ,Y] = E[X Y] – E[X]E[Y] =0, we must have E[X Y] = E[X]E[Y]
- If two variables are uncorrelated, they *can* still be dependent.

Expected values of two variables

- Expected values of two variables:

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

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

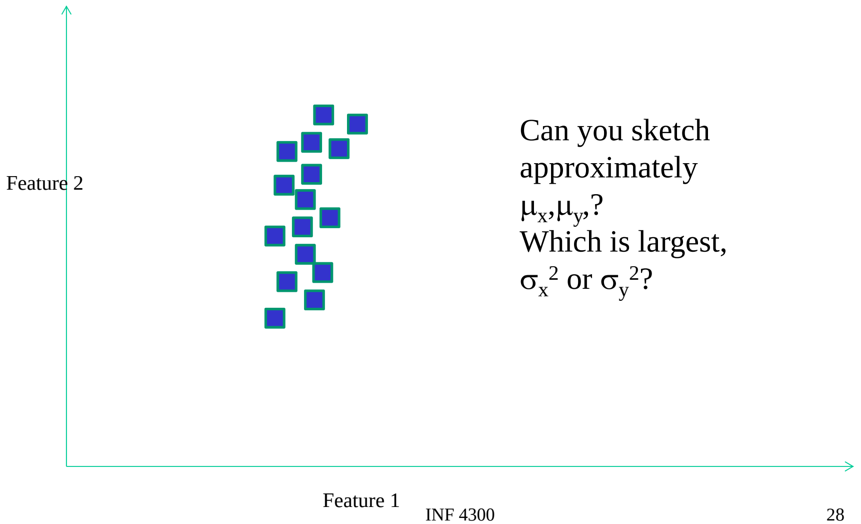
$$\mu_y=E(y)=\sum_x\sum_y yP(x,y)$$

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

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

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

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



Can you sketch approximately μ_x,μ_y,?
Which is largest, σ_x^2 or σ_y^2?

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. We need to consider their covariance.
- The conditional probability of *x* given *y* is defined:

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

and for distributions:

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

- Example: Draw two cards from a deck. Drawing a king in the first draw has probability 4/52. Drawing a king in the secong draw (given that the first draw gave a king) is 3/51.

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 (for us) is Bayes decision rule.

Bayes rule in general

- The equation:

$$P(\omega \mid x) = \frac{P(x \mid \omega)P(\omega)}{\sum_{\omega} P(x \mid \omega)P(\omega)} = \frac{P(x \mid \omega)P(\omega)}{P(x)}$$

- In words:

$$\text{Probability for class } \omega \text{ given feature vector x} = \frac{\text{Probability for value x if x is from class } \omega \times \text{Prior probability for class c}}{\text{Scaling factor}}$$

- x are observations/feature vector, ω is the unknown class labels.
- We want to find the most probable class ω given the observed feature vector x.

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:

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

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

- 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

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

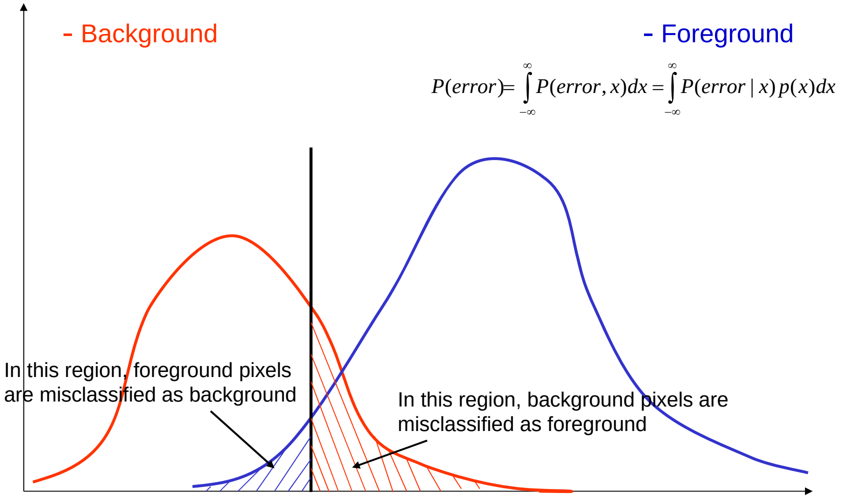
- p(x|ω_j) is the probability density function that models the likelihood for observing gray level x if the pixel belongs to class $\omega_{j.}$
  - Typically we assume a type of distribution, e.g. Gaussian, and the mean and covariance of that distribution is fitted to some data that we know belong to that class. This fitting is called classifier training.
- P(ω_j|x) is the posterior probability that the pixel actually belongs to class $\omega_{j.}$ 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 $\omega_{1.}$
- If P(ω_1|x) > P(ω_2|x) we have more belief that x belongs to ω_1, and we decide $\omega_{1.}$
- 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



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

In this region, foreground pixels are misclassified as background

In this region, background pixels are misclassified as foreground

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 boundary* as we will call it now) at the point where

  P(ω_1|x) = P(ω_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 analogusly 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 J classes and d features

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

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

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

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

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

Discriminant functions

- The decision rule

$$\text{Decide } \omega_1 \text{ if } \mathrm{P}(\omega_1|\mathbf{x}) > P(\omega_\mathrm{j}|\mathbf{x}), \text{ for all } j \neq i$$

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,......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

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:

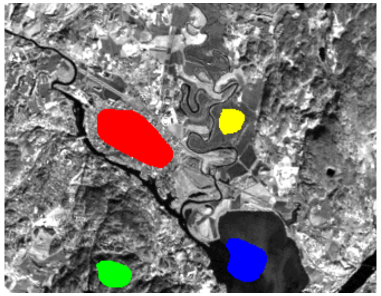
$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]$$

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

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

Example: image and training masks

The masks contain labels for the training data.
If label=1, then the pixel belongs to class 1 (red), and so on.
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.
The mask is often visualized in pseudo-colors on top of the input image, where each class is assign a color.
We should have a similar mask for the test data.



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. The mask has N_k pixels for each class.
- 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

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

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

Training

```
for i=1:N
  for j=i:M
    if mask(i,j)>==K
      increment nof. Samples in class K
      store the feature vector f(i,j) in a vector of training samples from class K
   end
end
end

For class k=1:K
    compute mean(k) and sigma(k)
```

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

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

How do to classification with a univariate Gaussian (1 feature)

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

  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 C with the highest value of $P(\omega_j|x)$ by setting label_image(x,y)= C

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.

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 different 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.

Validating classifier performance

- Classification performance is evaluated on a different set of samples with known class - the test set.
- The training set and the test set must be independent!
- Normally, the set of ground truth pixels (with known class) is partionioned into a set of training pixels and a set of test pixels of approximately the same size.
- This can be repeated several times to compute more robust estimates as average test accuracy over several different partitions of test set and training set.
  - By selecting e.g. 10 random partitions of the set of samples into a training set and a test set.

Confusion matrices

- A matrix with the true class label versus the estimated class labels for each class

Estimated class labels

True class labels

	Class 1	Class 2	Class 3	Total #samples
Class 1	80	15	5	100
Class 2	5	140	5	150
Class 3	25	50	125	200
Total	110	205	135	450

Confusion matrix - cont.

Alternatives:

•Report nof. correctly classified pixels for each class.

•Report the percentage of correctly classified pixels for each class.

•Report the percentage of correctly classified pixels in total.

•Why is this not a good measure if the number of test pixels from each class varies between classes?

	Class 1	Class 2	Class 3	Total #samples
Class 1	80	15	5	100
Class 2	5	140	5	150
Class 3	25	50	125	200
Total	110	205	135	450

Using more than one feature

- The power of the computer lies in deciding based on more than 1 feature at a time
- A simple trick to do this is to assume that features i and j and independent, then p(i,j|c)=p(i|c)p(j|c)
- The joint decision based on D independent features is then:

$$P(\omega_j \mid x_1, x_2, \ldots x_D) = \frac{p(x_1 \mid \omega_j)\, p(x_2 \mid \omega_j) \ldots p(x_D \mid \omega_j) P(\omega_j)}{p(x_1, x_2, \ldots x_D)}$$

Upcoming lectures

- Multivariate Gaussian
- Classifier evaluation
- Feature selection/feature transforms
- Knn-classification
- Unsupervised classification