Computer Vision

Pattern Recognition Concepts – Part I

Luis F. Teixeira
MAP-i | 2012/13

Pattern Recognition

- **What is it?**
- Many definitions in the literature
  - "The **assignment** of a physical object or event to one of several prespecified **categories**" – Duda and Hart
  - "A problem of estimating density functions in a high-dimensional space and **dividing the space** into the **regions of categories or classes**" – Fukunaga
  - "Given some examples of complex signals and the correct decisions for them, **make decisions automatically** for a stream of future examples" – Ripley
  - "The science that concerns the **description or classification** (recognition) of **measurements**" – Schalkoff
  - "The process of **giving names** ω to **observations** x", – Schürmann
  - Pattern Recognition is concerned with answering the question "What is this?" – Morse

Pattern Recognition

Related fields

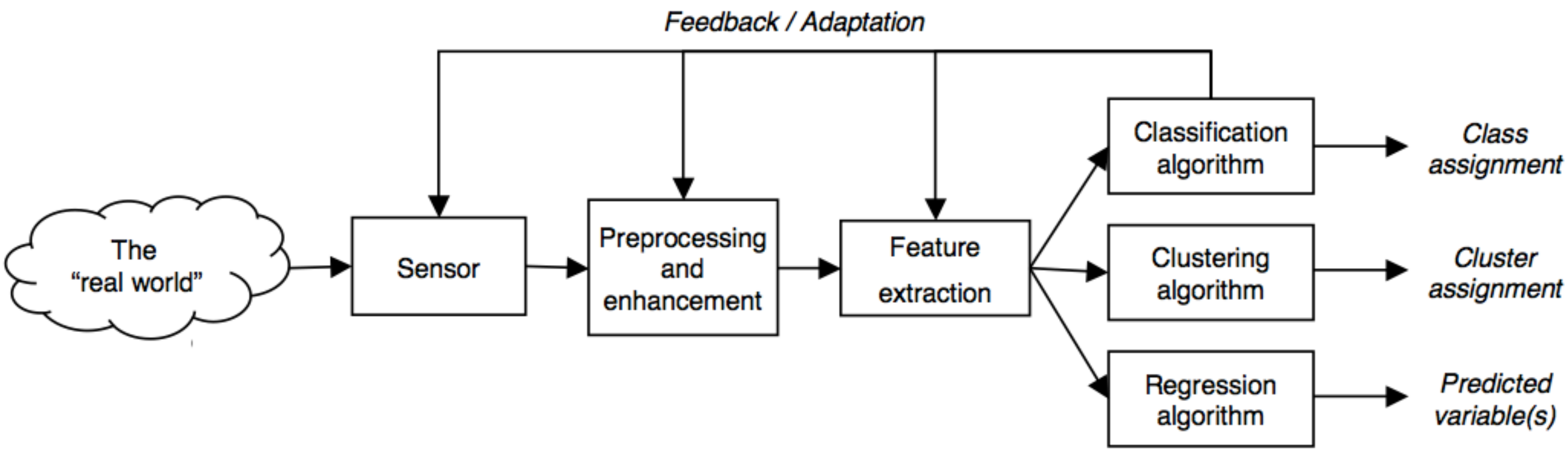
- Adaptive signal processing
- Machine learning
- Robotics and vision
- Cognitive sciences
- Mathematical statistics
- Nonlinear optimization
- Exploratory data analysis
- Fuzzy and genetic systems
- Detection and estimation theory
- Formal languages
- Structural modeling
- Biological cybernetics
- Computational neuroscience

Applications

- Image processing
- Computer vision
- Speech recognition
- Scene understanding
- Search, retrieval and visualization
- Computational photography
- Human-computer interaction
- Biometrics
- Document analysis
- Industrial inspection
- Financial forecast
- Medical diagnosis
- Surveillance and security
- Art, cultural heritage and entertainment

Pattern Recognition System

- **A typical pattern recognition system contains**
  - A sensor
  - A preprocessing mechanism
  - A feature extraction mechanism (manual or automated)
  - A classification or description algorithm
  - A set of examples (training set) already classified or described



Algorithms

- Classification
  - **Supervised**, **categorical** labels
  - Bayesian classifier, KNN, SVM, Decision Tree, Neural Network, etc.
- Clustering
  - **Unsupervised**, **categorical** labels
  - Mixture models, K-means clustering, Hierarchical clustering, etc.
- Regression
  - **Supervised or Unsupervised**, **real**-valued labels

Algorithms

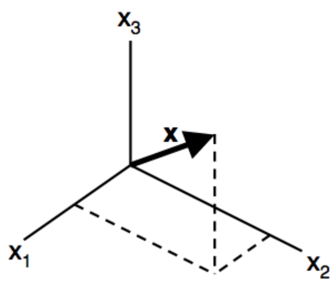
- Classification
  - **Supervised**, **categorical** labels
  - Bayesian classifier, KNN, SVM, Decision Tree, Neural Network, etc.
- Clustering
  - Unsupervised, categorical labels
  - Mixture models, K-means clustering, Hierarchical clustering, etc.
- Regression
  - -Supervised or Unsupervised, real-valued labels

Concepts

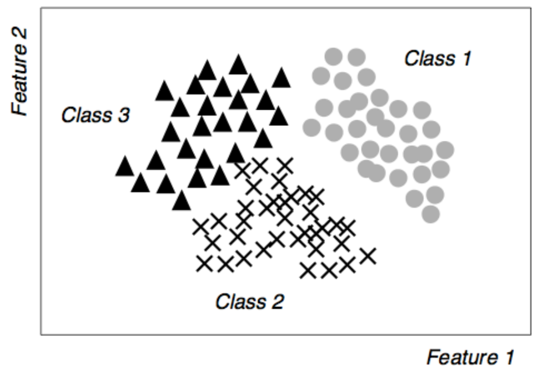
- **Feature**
  - A feature is any distinctive aspect, quality or characteristic. Features may be symbolic (i.e., color) or numeric (i.e., height)
  - The combination of *d* features is represented as a *d*-dimensional column vector called a **feature vector**
    - The d-dimensional space defined by the feature vector is called **feature space**
    - Objects are represented as points in a feature space. This representation is called a **scatter plot**

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$$

Feature vector



Feature space (3D)



Scatter plot (2D)

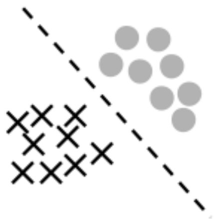
Concepts

- **Pattern**
  - Pattern is a composite of traits or features characteristic of an individual
  - In **classification**, a pattern is a pair of variables {**x**,ω} where
    - **x** is a collection of observations or features (feature vector)
    - ω is the concept behind the observation (label)

- **What makes a "good" feature vector?**
  - The quality of a feature vector is related to its ability to discriminate examples from different classes
    - Examples from the same class should have similar feature values
    - Examples from different classes have different feature values

Concepts

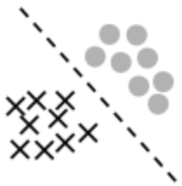
- “Good” features?



“Good” features

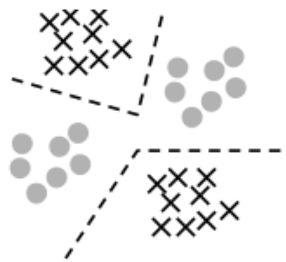
“Bad” features

- Feature properties

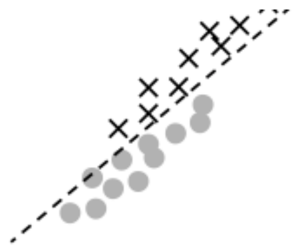


Linear separability

Non-linear separability



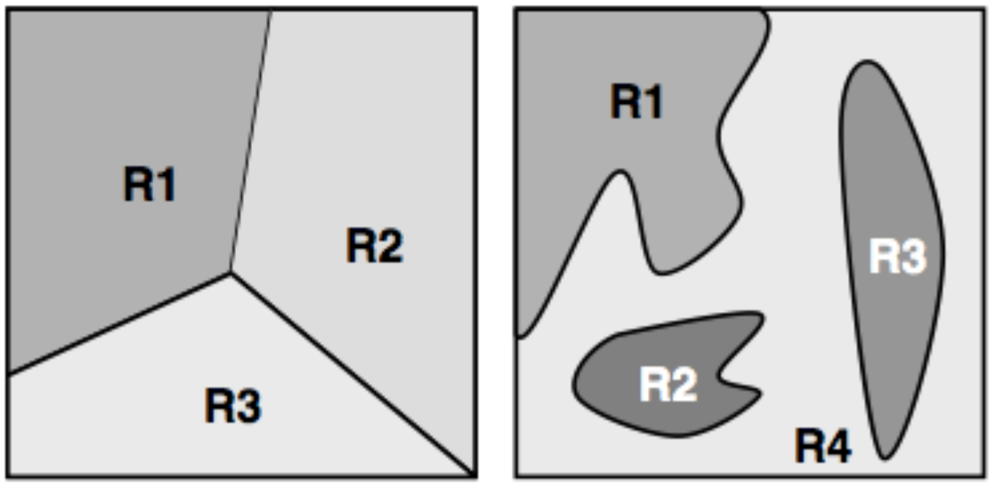
Multi-modal



Highly correlated features

Concepts

- **Classifiers**
  - The goal of a classifier is to partition the feature space into class-labeled **decision regions**
  - Borders between decision regions are called **decision boundaries**



Classification

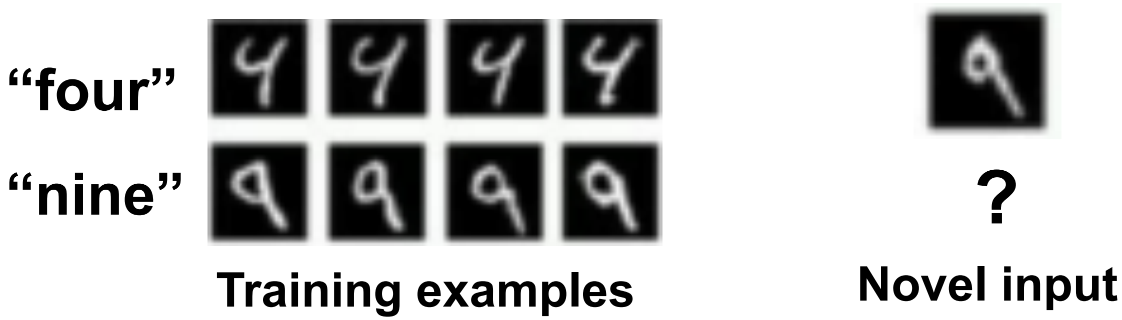
$$y = f(\mathbf{x})$$

output **prediction function** **feature vector**

- **Training:** given a *training set* of labeled examples $\{(\mathbf{x}_1,y_1), \ldots, (\mathbf{x}_N,y_N)\}$, estimate the prediction function f by minimizing the prediction error on the training set
- **Testing:** apply f to a never before seen *test example* **x** and output the predicted value $y = f(\mathbf{x})$

Classification

- Given a collection of *labeled* examples, come up with a function that will predict the labels of new examples.



- How good is some function we come up with to do the classification?
- Depends on
  - Mistakes made
  - Cost associated with the mistakes

An example*

- **Problem**: sorting incoming fish on a conveyor belt according to species
- Assume that we have only two kinds of fish:
  - Salmon
  - Sea bass

Picture taken with a camera

Adapted from Duda, Hart and Stork, Pattern Classification, 2nd Ed.

An example: the problem

What ***humans*** see

0	3	2	5	4	7	6	9	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	2	3	0	1	2	3	4	5
4	3	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0

What ***computers*** see

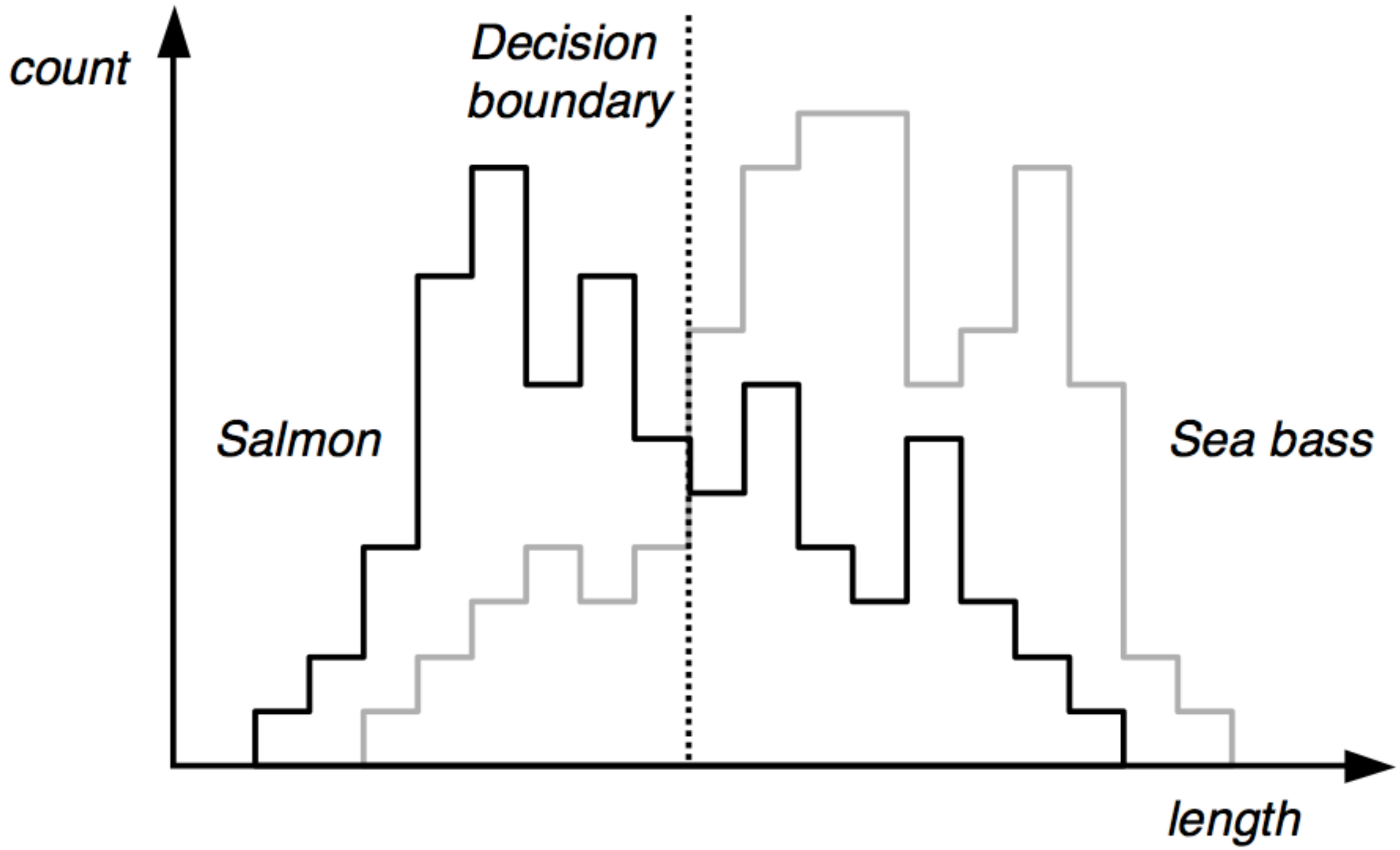
An example: decision process

- What kind of information can distinguish one species from the other?
  - Length, width, weight, number and shape of fins, tail shape, etc.
- What can cause problems during sensing?
  - Lighting conditions, position of fish on the conveyor belt, camera noise, etc.
- What are the steps in the process?
  - Capture image -> isolate fish -> take measurements -> make decision

An example: our system

- **Sensor**
  - The camera captures an image as a new fish enters the sorting area
- **Preprocessing**
  - Adjustments for average intensity levels
  - Segmentation to separate fish from background
- **Feature Extraction**
  - Assume a fisherman told us that a sea bass is generally longer than a salmon. We can use **length** as a feature and decide between sea bass and salmon according to a threshold on length.
- **Classification**
  - Collect a set of examples from both species
    - Plot a distribution of lengths for both classes
  - Determine a decision boundary (threshold) that minimizes the classification error

An example: features

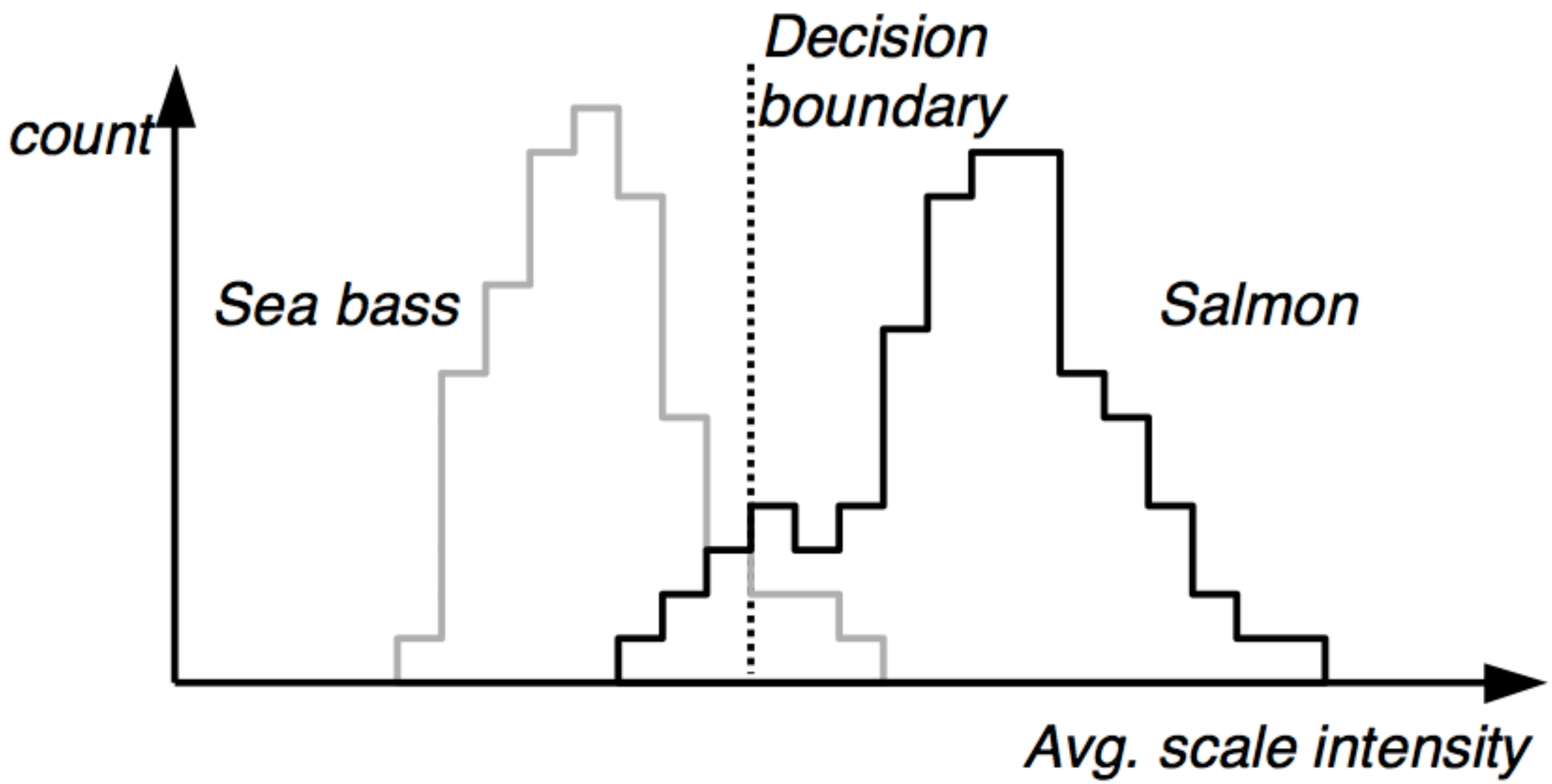


We estimate the system's probability of error and obtain a discouraging result of 40%. Can we improve this result?

An example: features

- Even though sea bass is longer than salmon on the average, there are many examples of fish where this observation does not hold
- Committed to achieve a higher recognition rate, we try a number of features
  - Width, Area, Position of the eyes w.r.t. mouth...
  - only to find out that these features contain no discriminatory information
- Finally we find a “good” feature: **average intensity of the scales**

An example: features



Histogram of the lightness feature for two types of fish in **training samples**. It looks easier to choose the threshold but we still can not make a perfect decision.

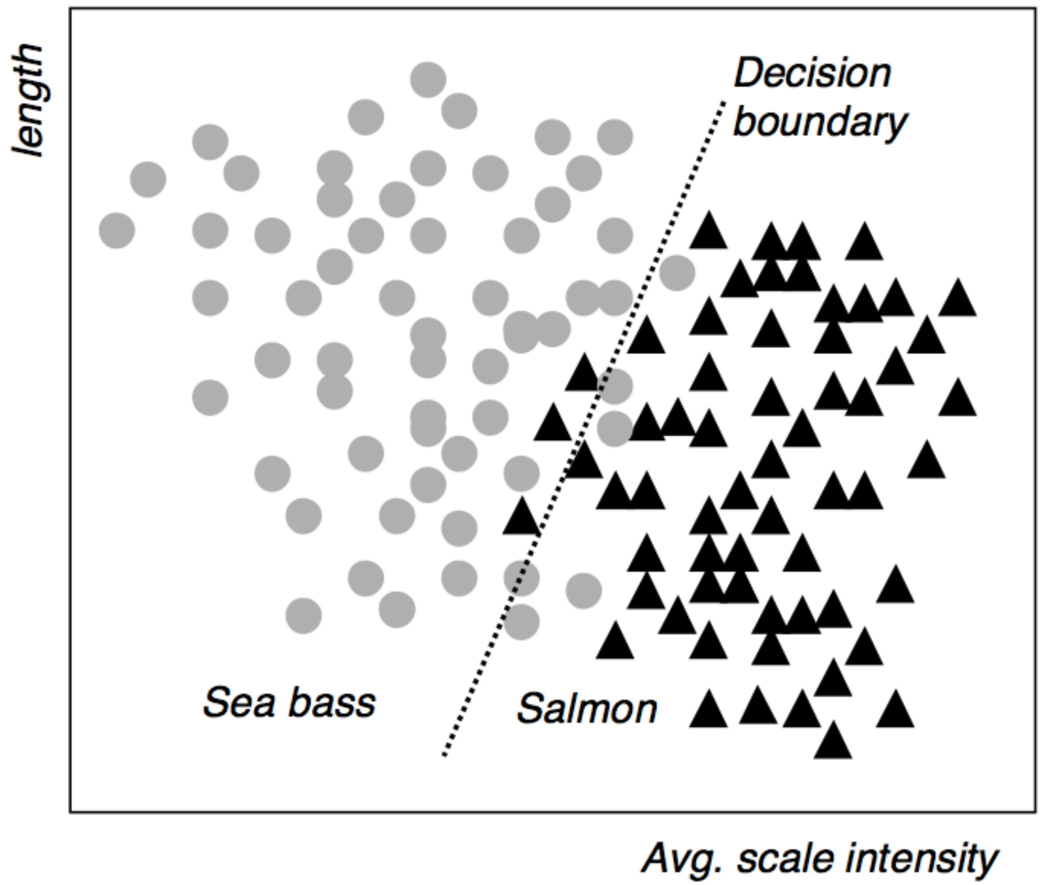
An example: multiple features

- We can use two features in our decision:
  - lightness: $\boldsymbol{x}_1$
  - length: $\boldsymbol{x}_2$
- Each fish image is now represented as a point (feature vector)

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

in a two-dimensional **feature space**.

An example: multiple features

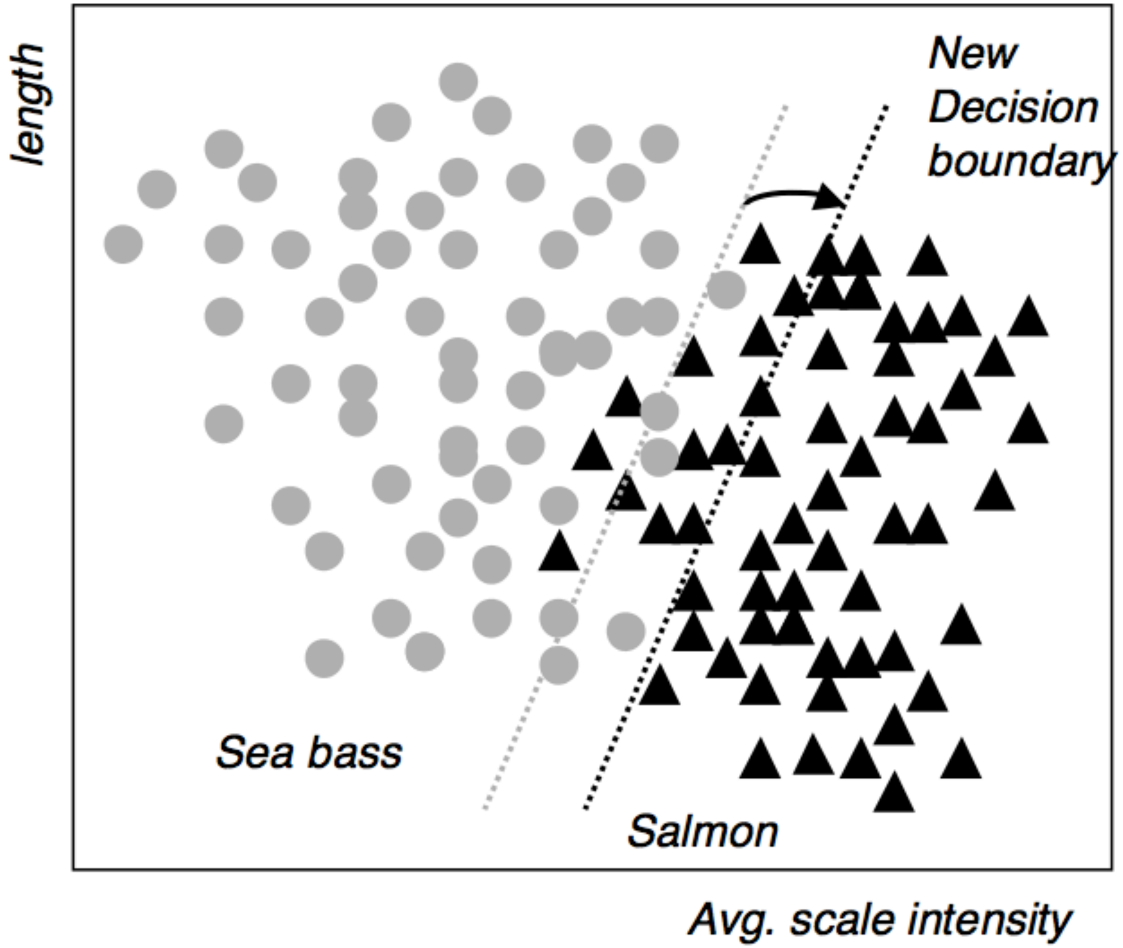


Scatter plot of lightness and length features for training samples. We can compute a **decision boundary** to divide the feature space into two regions with a classification rate of 95.7%.

An example: cost of error

- We should also consider **costs of different errors** we make in our decisions.
- For example, if the fish packing company knows that:
  - Customers who buy salmon will object vigorously if they see sea bass in their cans.
  - Customers who buy sea bass will not be unhappy if they occasionally see some expensive salmon in their cans.
- How does this knowledge affect our decision?

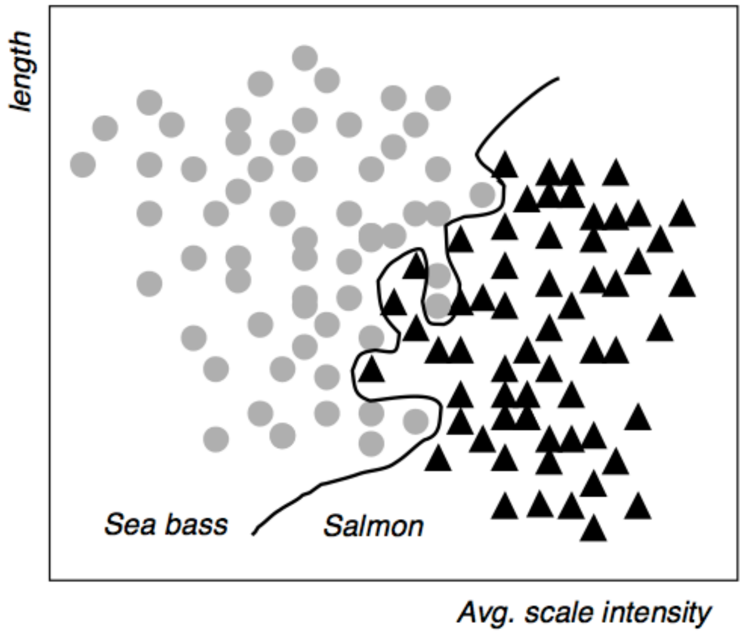
An example: cost of error



We could intuitively shift the decision boundary to minimize an alternative cost function

An example: generalization

- **The issue of generalization**
  - The recognition rate of our linear classifier (95.7%) met the design specifications, but we still think we can improve the performance of the system
  - We then design a über-classifier that obtains an impressive classification rate of 99.9975% with the following decision boundary



An example: generalization

- **The issue of generalization**
  - Satisfied with our classifier, we integrate the system and deploy it to the fish processing plant
  - A few days later the plant manager calls to complain that the system is misclassifying an average of 25% of the fish

- **What went wrong?**

Design cycle

- **Data collection**
  - Probably the most time-intensive component of a pattern recognition problem
  - Data acquisition and sensing
    - Measurements of physical variables
    - Important issues: bandwidth, resolution, sensitivity, distortion, SNR, latency, etc.
  - Collecting training and testing data
    - How can we know when we have adequately large and representative set of samples?

Design cycle

- **Feature choice**
  - Critical to the success of pattern recognition
  - Finding a new **representation** in terms of features
  - Discriminative features
    - Similar values for similar patterns
    - Different values for different patterns

- **Model learning and estimation**
  - Learning a mapping between features and pattern groups and categories

Design cycle

- **Model selection**
  - Definition of design criteria
  - Domain dependence and prior information
  - Computational cost and feasibility
  - Parametric vs. non-parametric models
  - Types of models: templates, decision-theoretic or statistical, syntactic or structural, neural, and hybrid

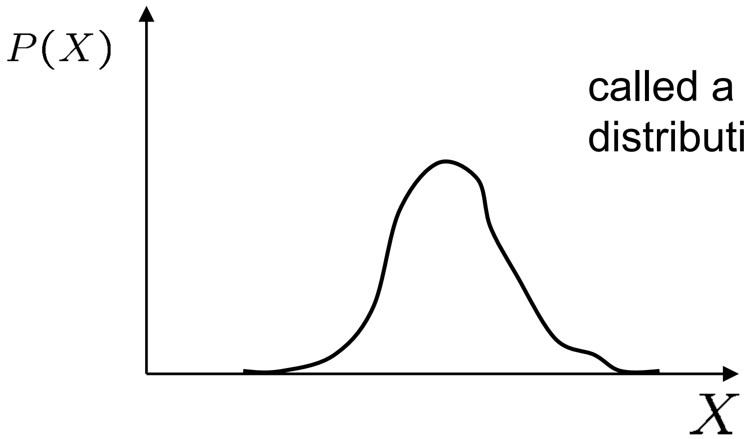
- **Model Training**
  - How can we learn the rule from data?
  - Given a feature set and a "blank" model, adapt the model to explain the data
  - Supervised, unsupervised and reinforcement learning

Design cycle

- **Predicting**
  - Using features and learned models to assign a pattern to a category
- **Evaluation**
  - How can we estimate the performance with training samples?
  - How can we predict the performance with future data?
  - Problems of overfitting and generalization

Review of probability theory

- Basic probability
  - X is a random variable
  - P(X) is the probability that X achieves a certain value



called a probability distribution/density function (PDF)

$$0 \leq P(X) \leq 1$$

$$\int_{-\infty}^{\infty} P(X)dX = 1$$

continuous X

$$\sum P(X) = 1$$

discrete X

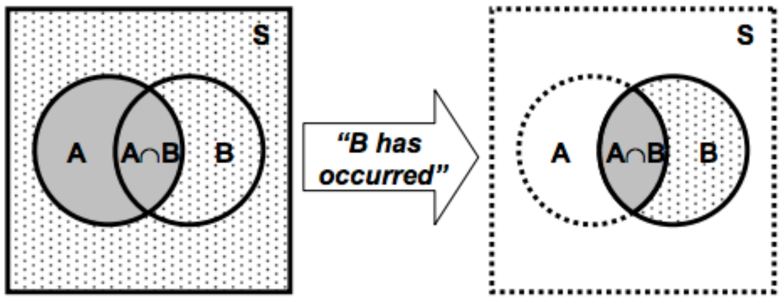
Conditional probability

- If A and B are two events, the probability of event A when we already know that event B has occurred P[A|B] is defined by the relation

$$P[A \mid B] = \frac{P[A \cap B]}{P[B]} \text{ for } P[B] > 0$$

- P[A|B] is read as the "conditional probability of A conditioned on B", or simply the "probability of A given B

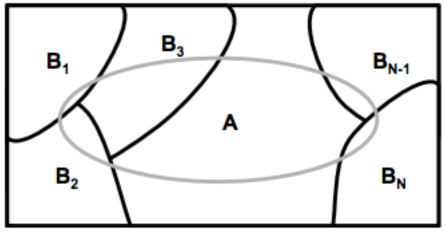
- Graphical interpretation



Conditional probability

- Theorem of Total Probability
  - Let B_1, B_2, …, B_N be mutually exclusive events, then

$$P[A] = P[A \mid B_1]P[B_1] + \ldots + P[A \mid B_N]P[B_N] = \sum_{k=1}^{N} P[A \mid B_k]P[B_k]$$



- Bayes Theorem
  - Given B_1, B_2, …, B_N, a partition of the sample space S. Suppose that event A occurs; what is the probability of event B_j?
  - Using the definition of conditional probability and the Theorem of total probability we obtain

$$P[B_j \mid A] = \frac{P[A \cap B_j]}{P[A]} = \frac{P[A \mid B_j] \cdot P[B_j]}{\sum_{k=1}^{N} P[A \mid B_k] \cdot P[B_k]}$$

Bayes theorem

- For pattern recognition, Bayes Theorem can be expressed as

$$P(\omega_j \mid \mathrm{x}) = \frac{P(\mathrm{x} \mid \omega_j) \cdot P(\omega_j)}{\sum_{k=1}^{N} P(\mathrm{x} \mid \omega_k) \cdot P(\omega_k)} = \frac{P(\mathrm{x} \mid \omega_j) \cdot P(\omega_j)}{P(\mathrm{x})}$$

where ω_j is the jth class and **x** is the feature vector

- Each term in the Bayes Theorem has a special name
  - $P(\omega_j)$ **Prior** probability (of class ω_j)
  - $P(\omega_j|x)$ **Posterior** probability (of class ω_j given the observation x)
  - $P(x|\omega_j)$ **Likelihood** (conditional prob. of x given class ω_j)
  - P(x) **Evidence** (normalization constant that does not affect the decision)
- Two commonly used decision rules are
  - Maximum A Posteriori (**MAP**): choose the class ω_j with highest $P(\omega_j|x)$
  - Maximum Likelihood (**ML**): choose the class ω_j with highest $P(x|\omega_j)$
  - ML and MAP are equivalent for non-informative priors ($P(\omega_j)$ constant)

Bayesian decision theory

- Bayesian Decision Theory is a statistical approach that quantifies the **tradeoffs** between various decisions using **probabilities and costs** that accompany such decisions.

- Fish sorting example:
  - define C, the type of fish we observe (state of nature), as a random variable where
    - $C = C_1$ for sea bass
    - $C = C_2$ for salmon
  - $P(C_1)$ is the **a priori probability** that the next fish is a sea bass
  - $P(C_2)$ is the **a priori probability** that the next fish is a salmon

Prior probabilities

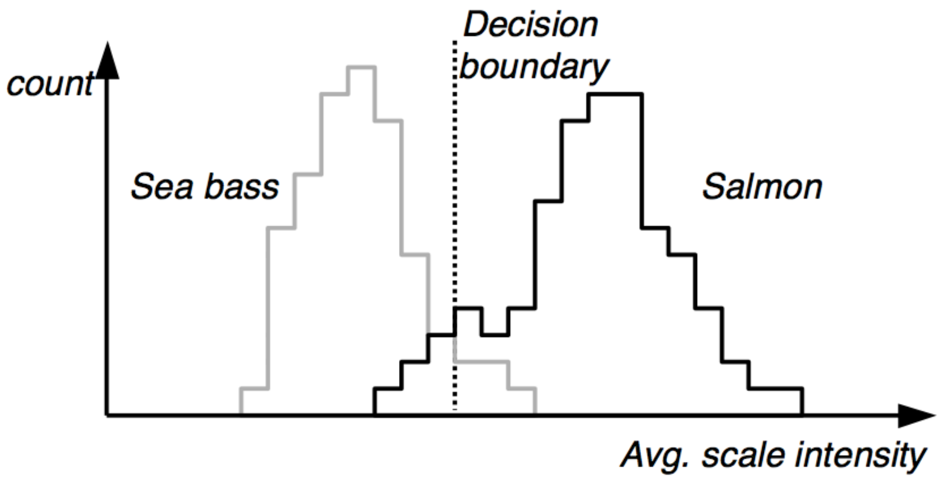
- Prior probabilities reflect our knowledge of how likely each type of fish will appear **before** we actually see it.
- How can we choose $P(C_1)$ and $P(C_2)$?
  - Set $P(C_1) = P(C_2)$ if they are equiprobable (uniform priors).
  - May use different values depending on the fishing area, time of the year, etc.
- Assume there are no other types of fish
  - $P(C_1) + P(C_2) = 1$

- In a general classification problem with K classes, **prior probabilities reflect prior expectations** of observing each class and

$$\sum_{i=1}^{K} P(C_i) = 1$$

Class-conditional probabilities

- Let x be a continuous random variable, representing the lightness measurement
- Define $p(x|C_j)$ as the class-conditional probability density (probability of x given that the state of nature is C_j for j = 1, 2).
- $p(x|C_1)$ and $p(x|C_2)$ describe the difference in lightness between populations of sea bass and salmon.



Posterior probabilities

- Suppose we know P(C_j) and P(x|C_j) for j = 1, 2, and measure the lightness of a fish as the value x.
- Define P(C_j|x) as the **a posteriori probability** (probability of the type being C_j, given the measurement of feature value x).
- We can use the **Bayes formula** to convert the prior probability to the posterior probability

$$P(C_j \mid x) = \frac{P(x \mid C_j)P(C_j)}{P(x)}$$

$$\text{where } P(x) = \sum_{j=1}^{2} P(x \mid C_j)P(C_j)$$

Making a decision

- How can we make a decision after observing the value of x?

$$\text{Decide}\begin{cases} C_1 & \text{if } \; P(C_1 \mid x) > P(C_2 \mid x) \\ C_2 & \textit{otherwise} \end{cases}$$

- Rewriting the rule gives

$$\text{Decide}\begin{cases} C_1 & \text{if } \; \dfrac{P(x \mid C_1)}{P(x \mid C_2)} > \dfrac{P(C_2)}{P(C_1)} \\ C_2 & \textit{otherwise} \end{cases}$$

- Bayes decision rule **minimizes** the error of this decision

Making a decision

- Confusion matrix
  - For C_1 we have:

		Assigned	
		C_1	C_2
True	C_1	correct detection	mis-detection
	C_2	false alarm	correct rejection

  - The two types of errors (false alarm and mis-detection) can have **distinct costs**

Minimum-error-rate classification

- Let $\{C_{1,\dots,} C_K\}$ be the finite set of K states of nature (**classes**, **categories**).
- Let **x** be the D-component vector-valued random variable (**feature vector)**.
- If all errors are equally costly, the minimum-error decision rule is defined as

$$\text{Decide } C_i \;\text{ if }\; P(C_i \mid x) > P(C_j \mid x) \quad \forall j \neq i$$

- The resulting error is called the **Bayes error** and is the best performance that can be achieved.

Bayesian decision theory

- Bayesian decision theory gives the **optimal decision** rule under the assumption that the "true" values of the probabilities are **known**.
- But, how can we estimate (learn) the unknown $p(\boldsymbol{x}|C_j)$, j = 1, …, K ?

- **Parametric models**: assume that the form of the density functions is known
- **Non-parametric models**: no assumption about the form

Bayesian decision theory

- Parametric models
  - Density models (e.g., Gaussian)
  - Mixture models (e.g., mixture of Gaussians)
  - Hidden Markov Models
  - Bayesian Belief Networks

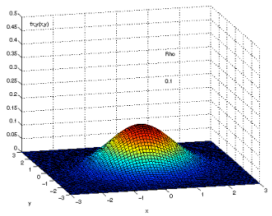
- Non-parametric models
  - Histogram-based estimation
  - Parzen window estimation
  - Nearest neighbour estimation

Gaussian density

- **Gaussian** can be considered as a model where the feature vectors for a given class are continuous-valued, randomly corrupted versions of a single typical or prototype vector.

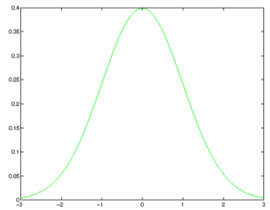
For $\mathbf{x} \in \boldsymbol{R}^D$

$$p(\mathbf{x}) = N(\boldsymbol{\mu}, \Sigma) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x}-\boldsymbol{\mu})}$$



For $x \in \boldsymbol{R}$

$$p(\mathbf{x}) = N(\boldsymbol{\mu}, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



- Some **properties** of the Gaussian:
  - Analytically tractable
  - Completely specified by the 1st and 2nd moments
  - Has the maximum entropy of all distributions with a given mean and variance
  - Many processes are asymptotically Gaussian (Central Limit Theorem)
  - “Uncorrelatedness” implies independence

Bayes linear classifier

- Let us assume that the **class-conditional densities** are **Gaussian** and then explore the resulting form for the posterior probabilities.
- Assume that all classes share the same covariance matrix, thus the density for class C_k is given by

$$p(\mathbf{x} \mid C_k) = \frac{1}{(2\pi)^{D/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T \Sigma^{-1} (\mathbf{x}-\boldsymbol{\mu}_k)^T}$$

- We then model the class-conditional densities p(**x**|C_k) and class priors p(C_k) and use these to compute **posterior probabilities** p(C_k|**x**) through Bayes' theorem:

$$p(C_k \mid \mathrm{x}) = \frac{p(\mathrm{x} \mid C_k) p(C_k)}{\sum_{j=1}^{K} p(\mathrm{x} \mid C_j) p(C_j)}$$

- Assuming only **2 classes** the **decision boundary is linear**

Bayesian decision theory

- Bayesian Decision Theory shows us how to design an optimal classifier if we know the prior probabilities P(C_k) and the class-conditional densities P(**x**|C_k).
- Unfortunately, we rarely have complete knowledge of the probabilistic structure.
- However, we can often find design samples or **training data** that include particular representatives of the patterns we want to classify.

- The **maximum likelihood** estimates of a Gaussian are

$$\hat{\boldsymbol{\mu}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i \text{ and } \hat{\Sigma} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \hat{\boldsymbol{\mu}})(\mathbf{x}_i - \hat{\boldsymbol{\mu}})^T$$

Supervised classification

- In practice there are two general strategies
  - Use the training data to build representative probability model; separately model class-conditional densities and priors (*generative*)
  - Directly construct a good decision boundary, and model the posterior (*discriminative*)

Discriminative classifiers

- In general, discriminant functions can be defined **independently** of the Bayesian rule. They lead to **suboptimal** solutions, yet, if chosen appropriately, they can be computationally more tractable.
- Moreover, in practice, they may also lead to better solutions. This, for example, may be case if the nature of the underlying pdf's are unknown.

Discriminative classifiers

- Non-Bayesian Classifiers
  - Distance-based classifiers:
    - Minimum (mean) distance classifier
    - Nearest neighbour classifier
  - Decision boundary-based classifiers:
    - Linear discriminant functions
    - Support vector machines
    - Neural networks
    - Decision trees

References

- Selim Aksoy, Introduction to Pattern Recognition, Part I, http://retina.cs.bilkent.edu.tr/papers/patrec_tutorial1.pdf
- Ricardo Gutierrez-Osuna, Introduction to Pattern Recognition, http://research.cs.tamu.edu/prism/lectures/pr/pr_l1.pdf
- Jaime S. Cardoso, Classification Concepts, http://www.dcc.fc.up.pt/~mcoimbra/lectures/MAPI_1112/CV_1112_5b_ClassificationConcepts.pdf
- Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
- Richard O. Duda, Peter E. Hart, David G. Stork, Pattern Classification, John Wiley & Sons, 2001