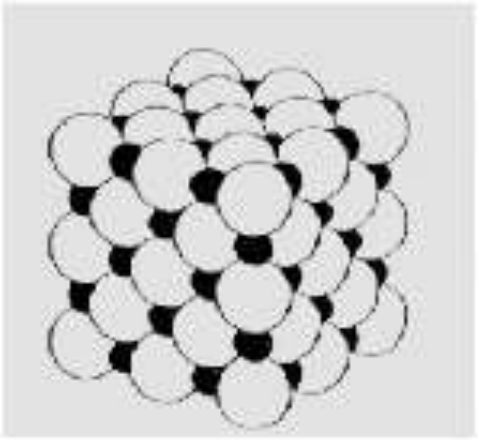
UNIT- 5
Pattern Recognition & Machine Learning

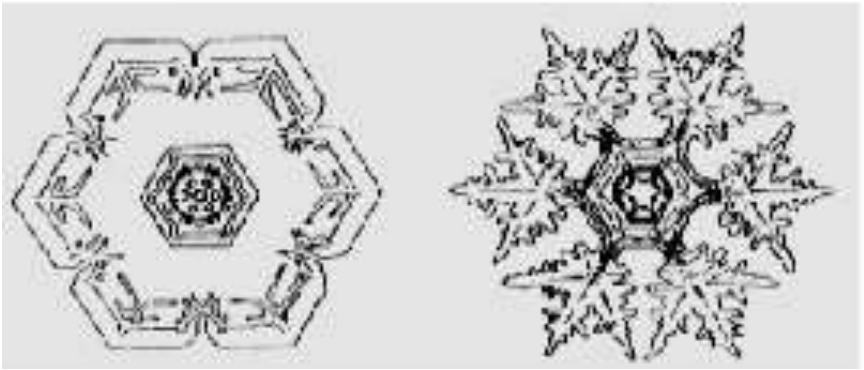
What is Machine Learning?

- It is very hard to write programs that solve problems like recognizing a face.
  - We don't know what program to write because we don't know how our brain does it.
  - Even if we had a good idea about how to do it, the program might be horrendously complicated.
- Instead of writing a program by hand, we collect lots of examples that specify the correct output for a given input.
- A machine learning algorithm then takes these examples and produces a program that does the job.
  - The program produced by the learning algorithm may look very different from a typical hand-written program. It may contain millions of numbers.
  - If we do it right, the program works for new cases as well as the ones we trained it on.

What are Patterns?

- Laws of Physics & Chemistry generate patterns.



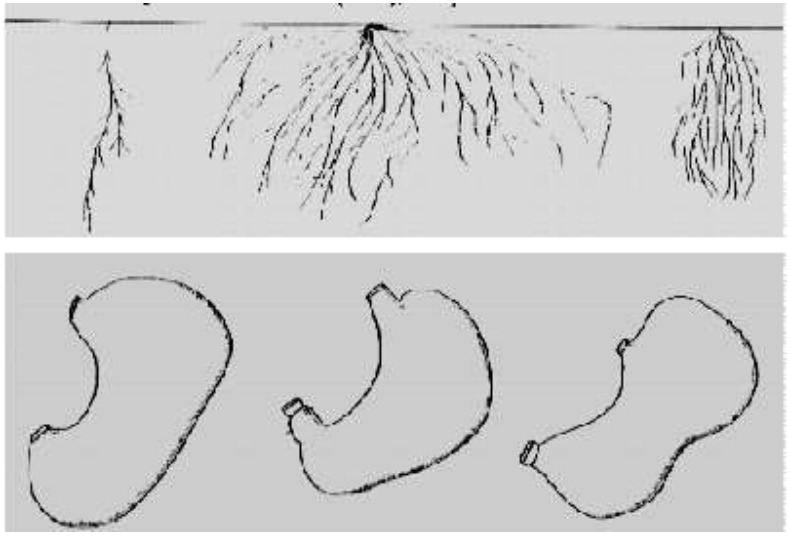


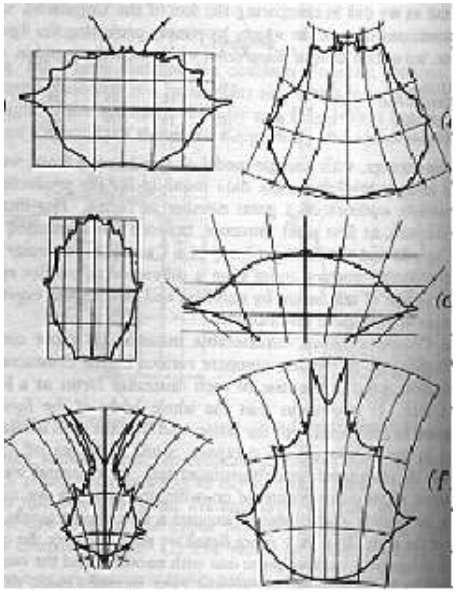
Patterns in Astronomy.

- Humans tend to see patterns everywhere.

Patterns in Biology.

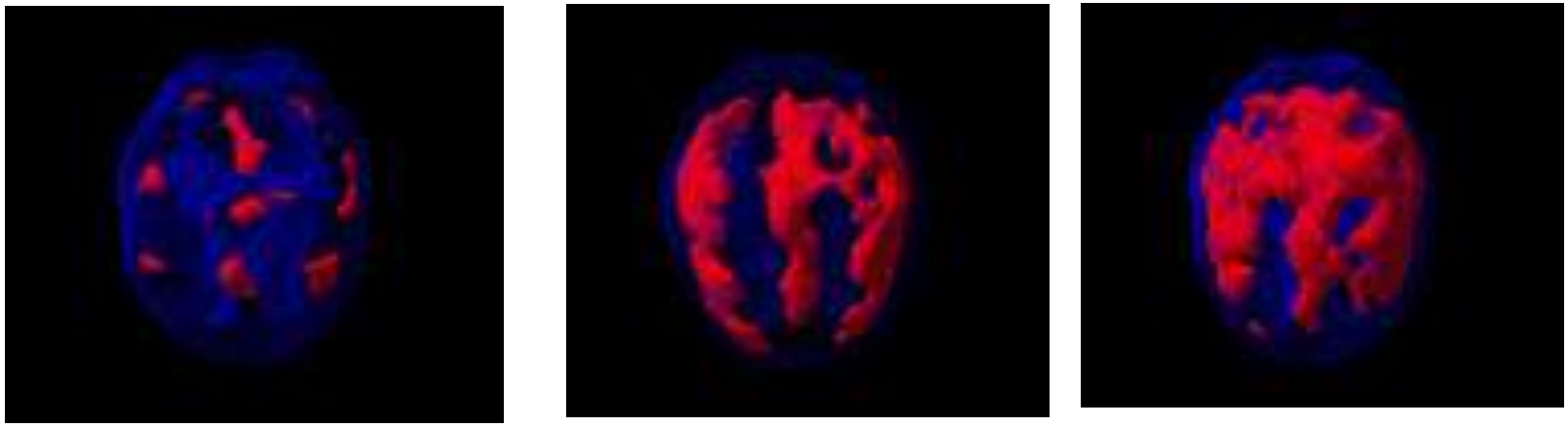
- Applications: Biometrics, Computational Anatomy, Brain Mapping.





Patterns of Brain Activity.

- Relations between brain activity, emotion, cognition, and behaviour.

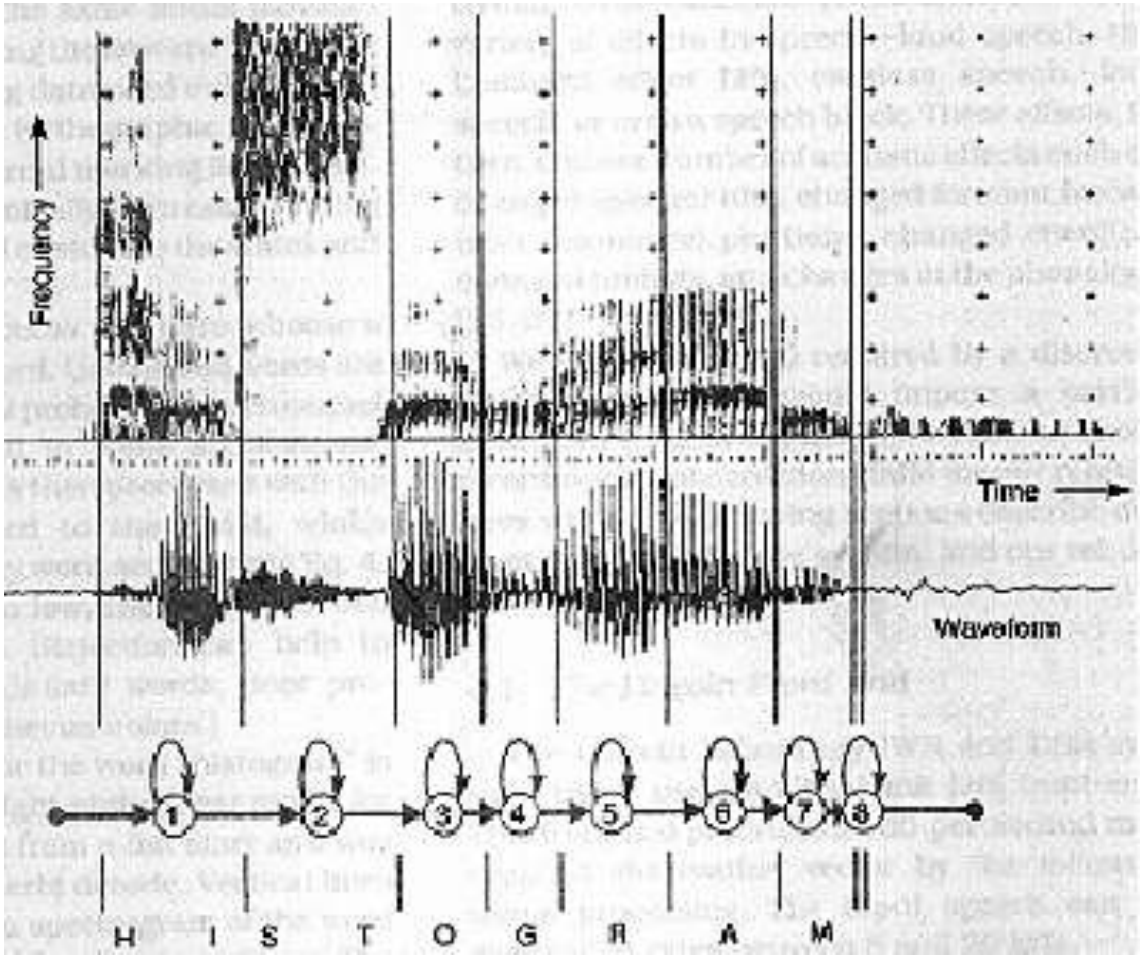


Variations of Patterns.

- Patterns vary with *expression*, *lighting*, *occlusions*.

Speech Patterns.

- Acoustic signals.



Preamble

PATTERN RECOGNITION $\Rightarrow$ Pattern + Recognition

PATTERN : Pattern is a set of objects or phenomena or concepts where the elements of the set are similar to one another in certain ways/aspects. The Pattern are described by certain quantities, qualities, traits, notable features and so on.

Example : Humans, Radar Signals, insects, Animals, sonar signals. Fossil records, Micro organisms signals, clouds etc.

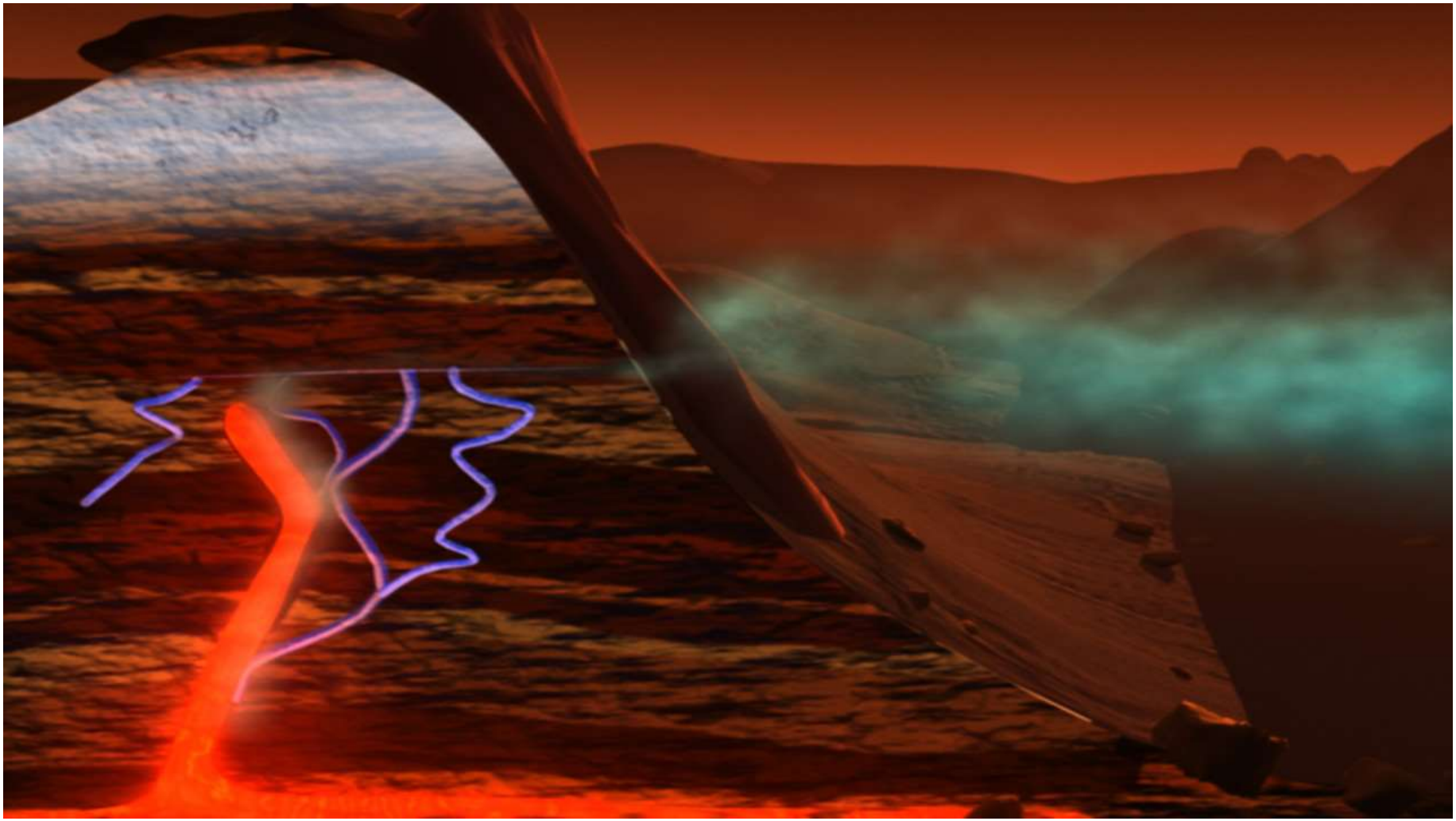
Humans have a pattern which is different from the pattern of animals. Each individuals has a pattern which is different from the patterns of others.

Cloud Patterns

Forest and Cultivated Land

Coal Mine Detection

Natural Gas Detection



PATTERN RECOGNITION

- It is said that each thief has his own patterns. Some enter through windows, some through doors and so on. Some do only ‘pick-pocketing’, some steal cycles, some steal cars and so on.
- The body pattern of human beings has not changed since millions of years. But pattern of computers and other machines continuously change. Because of the fixed pattern of human bodies, the work of medical doctors is easier compared to the work of engineers who deal with machines whose patterns continuously change.

PATTERN RECOGNITION

- **RECOGNITION**

  Recognition $\Rightarrow$ Re + Cognition

- **COGNITION:-** To become acquainted with, to come to know the act, or the process of knowing an entity (the process of knowing).
- Recognition : The knowledge or feeling that the present object has been met before (the process of knowing again).
- Recognition & acquire knowledge through sender perception are very much related.

POPULAR DEFINITIONS OF PATTERN RECOGNITION

Pattern Recognition consists of recognizing a pattern using a machine (computer). It can be defined in several ways.

- DEFINITION.1.:- It is a study of ideas and algorithms that provide computers with a perceptual capability to put abstract objects, or patterns into categories in a simple and reliable way.

- DEFINITION.2.:- It is an ambitious endeavor of mechanization of the most fundamental function of cognition.

IMPLICATION OF PATTERN RECOGNITION

Pattern Recognition implies following three things. It has been perceived

- The object has been cognized earlier or the picture/description of the object has been cognized earlier.
- The earlier details of cognition are stored.
- The object is encountered again at which time it is to be recognized.

Applications of Pattern Recognition.

Handwritten digit/letter recognition
Biometrics: voice, iris, fingerprint, face, and gait recognition
Speech recognition
Smell recognition (e-nose, sensor networks)
Defect detection in chip manufacturing
Interpreting DNA sequences
Fruit/vegetable recognition
Medical diagnosis
Terrorist Detection
Credit Fraud Detection
Credit Applications.
… …

COVERAGE OF PATTERN RECOGNITION

Pattern Recognition covers a wide spectrum of disciplines such as

1. Cybernetics
2. Computer Science
3. System Science
4. Communication Sciences
5. Electronics
6. Mathematics
7. Logic
8. Psychology
9. Physiology
10. Philosophy

APPLICATION OF PATTERN RECOGNITION

1. Medical diagnosis
2. Life form analysis
3. Sonar detection
4. Radar detection
5. Image processing
6. Process control
7. Information Management systems
8. Aerial photo interpretation.
9. Weather prediction
10. Sensing of life on remote planets.
11. Behavior analysis
12. Character recognition
13. Speech and Speaker recognition etc.

METHODOLOGY OF PATTERN RECOGNITIONS OF PR

It consists of the following:

1. We observe patterns
2. We study the relationships between the various patterns.
3. We study the relationships between patterns and ourselves and thus arrive at situations.
4. We study the changes in situations and come to know about the events.
5. We study events and thus understand the law behind the events.
6. Using the law, we can predict future events.

What is pattern recognition?

"The assignment of a physical object or event to one of several prespecified categeries"

- A **pattern** is an object, process or event that can be given a name.
- A **pattern class** (or category) is a set of patterns sharing common attributes and usually originating from the same source.
- During **recognition** (or **classification**) given objects are assigned to prescribed classes.
- A **classifier** is a machine which performs classification.

Examples of applications

- **Optical Character Recognition (OCR)**
  - Handwritten: sorting letters by postal code, input device for PDA‘s.
  - Printed texts: reading machines for blind people, digitalization of text documents.
- **Biometrics**
  - Face recognition, verification, retrieval.
  - Finger prints recognition.
  - Speech recognition.
- **Diagnostic systems**
  - Medical diagnosis: X-Ray, EKG analysis.
  - Machine diagnostics, waster detection.
- **Military applications**
  - Automated Target Recognition (ATR).
  - Image segmentation and analysis (recognition from aerial or satelite photographs).

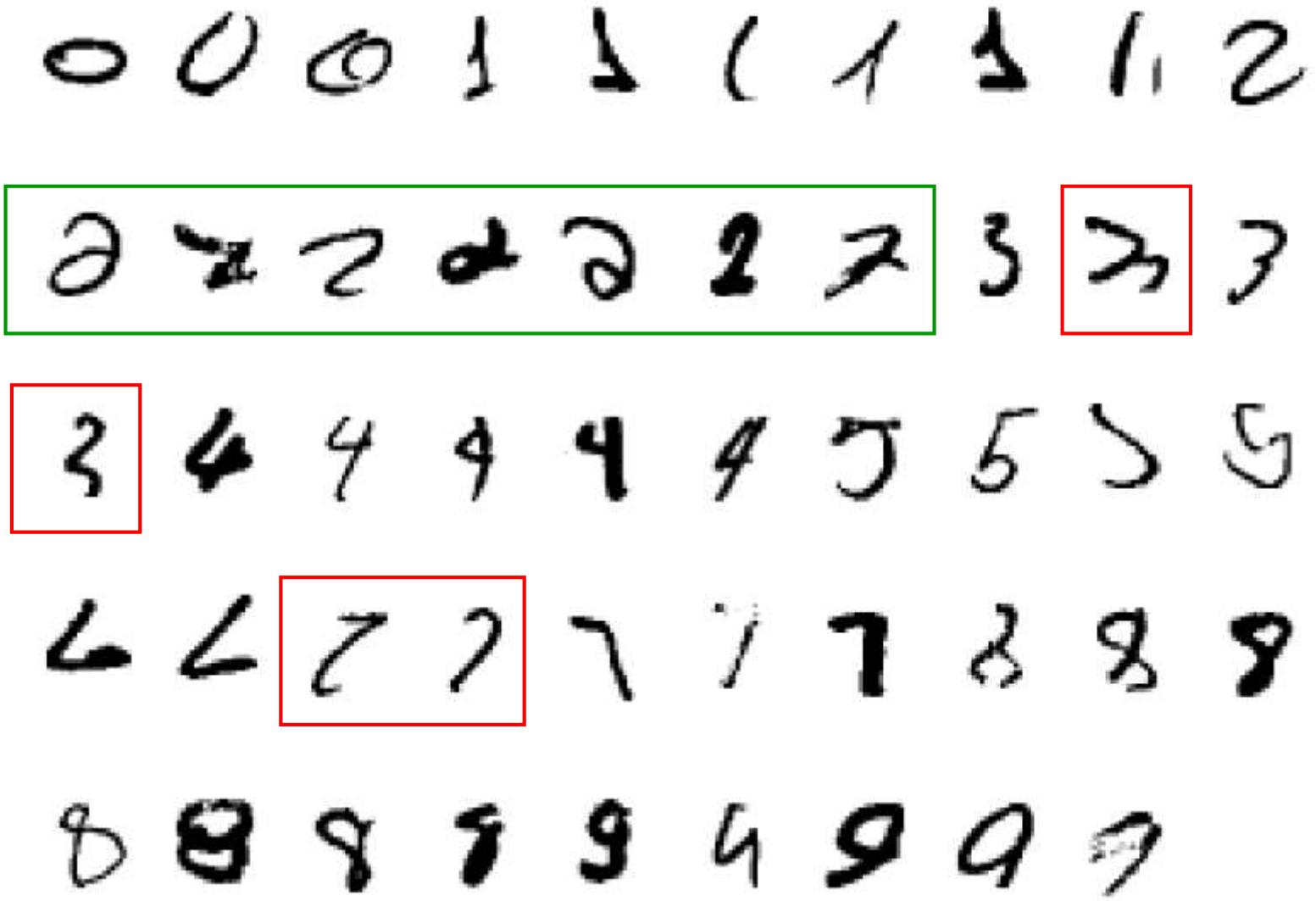
Approaches

- **Statistical PR:** based on underlying statistical model of patterns and pattern classes.
- **Structural (or syntactic) PR:** pattern classes represented by means of formal structures as grammars, automata, strings, etc.
- **Neural networks:** classifier is represented as a network of cells modeling neurons of the human brain (connectionist approach).

Goal of Pattern Recognition.

- Recognize Patterns. Make decisions about patterns.
- Visual Example – is this person happy or sad?
- Speech Example – did the speaker say “Yes” or “No”?
- Physics Example – is this an atom or a molecule?

A classic example of a task that requires machine learning: It is very hard to say what makes a 2



Some more examples of tasks that are best solved by using a learning algorithm

- Recognizing patterns:
  - Facial identities or facial expressions
  - Handwritten or spoken words
  - Medical images
- Generating patterns:
  - Generating images or motion sequences (demo)
- Recognizing anomalies:
  - Unusual sequences of credit card transactions
  - Unusual patterns of sensor readings in a nuclear power plant or unusual sound in your car engine.
- Prediction:
  - Future stock prices or currency exchange rates

Some web-based examples of machine learning

- The web contains a lot of data. Tasks with very big datasets often use machine learning
  - especially if the data is noisy or non-stationary.
- Spam filtering, fraud detection:
  - The enemy adapts so we must adapt too.
- Recommendation systems:
  - Lots of noisy data. Million dollar prize!
- Information retrieval:
  - Find documents or images with similar content.
- Data Visualization:
  - Display a huge database in a revealing way (demo)

Parameter Estimation

- We have a number of design samples or training data.
- The problem is to **find some way to use this information to design or train the classifier.**
- One approach:
  - **Use the samples to estimate** the unknown probabilities and probability densities,
  - And then use the resulting estimates as if they **were the true values.**

Parameter Estimation

- If we know the **number parameters** in advance and our general **knowledge about the problem** permits us to **parameterize the conditional desities** then severity of the problem can be reduced significantly.

Parameter Estimation

- For example:
  - We can reasonably assume that the **p(x|wi) is a normal density** with mean **μ** and covariance matrix ∑,
  - We do not know the exact values of these quantities,
  - However, this knowledge simplifies the problem **from one of estimating an unknown function p(x|wi) to one of estimating the parameters the mean μ_i and covariance matrix $\sum_i$**
  - **Estimating p(x|wi) → estimating μ_i and $\sum_i$**

Parameter Estimation

- Data availability in a Bayesian framework
  - We could design an optimal classifier if we knew:
    - $P(\omega_i)$ (priors)
    - $P(x \mid \omega_i)$ (class-conditional densities)

  Unfortunately, we rarely have this complete information!

- Design a classifier from a training sample
  - No problem with prior estimation
  - Samples are often too small for class-conditional estimation (large dimension of feature space!)

Parameter Estimation

- Given a bunch of data from each class how to estimate the parameters of class conditional densities, $P(x \mid \omega_i)$?
- Ex: $P(x \mid \omega_i) = N(\mu_j, \Sigma_j)$ is Normal.

  Parameters $\theta_j = (\mu j, \Sigma j)$

Two major approaches

- **Maximum-Likelihood Method**
- **Bayesian Method**

  – Use $P(\omega_i \mid x)$ for our classification rule!
  – Results are nearly identical, but the approaches are different

Maximum-Likelihood vs. Bayesian:

- Maximum Likelihood
- Parameters are fixed but unknown!
- Best parameters are obtained by maximizing the probability of obtaining the samples observed

- Bayes
- Parameters are random variables having some known distribution
- Best parameters are obtained by estimating them given the data

Major assumptions

- A priori information $P(\omega_i)$ for each category is available
- Samples are i.d. and $P(x \mid \omega_i)$ is Normal

$$P(x \mid \omega_i) \sim N(\mu_i, \Sigma_i)$$

- Note: Characterized by 2 parameters

Maximum-Likelihood Estimation

- Has good convergence properties as the sample size increases
- Simpler than any other alternative techniques

Maximum-Likelihood Estimation

We assume that samples are collected randomly from a given form of distribution, whose parameters are unknown.Unknown parameters are denoted by the vector **θ.**

We partition our training data **X** into c class specific subsets $D_1 \ldots D_c$, assuming that the data in D_j have been drawn randomly according to the distribution $p(\mathbf{x}|\omega_j)$

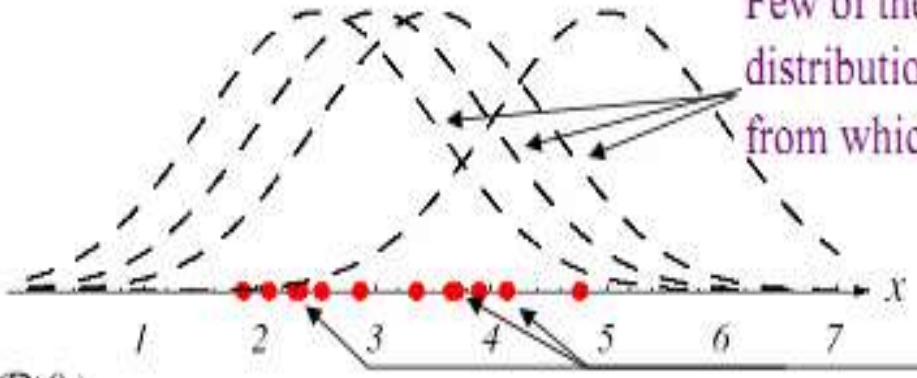
If , for example, we know each distribution is normal, $p(\mathbf{x}|\omega_j) \sim N(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$ but we do not know the mean and covariance, then our problem is to estimate

$$\boldsymbol{\theta}_j = \begin{pmatrix} \boldsymbol{\mu}_j \\ \boldsymbol{\Sigma}_j \end{pmatrix}$$

To make the dependence of $p(\mathbf{x}|\omega_j)$ on $\boldsymbol{\theta}_j$ more specific, we can write $p(\mathbf{x}|\omega_j, \boldsymbol{\theta}_j)$. Furthermore, to make our life easier – and why not – we assume that all $\boldsymbol{\theta}_j$ are independent, i.e., knowing one does not tell us anything about the others. This will allow us to work for each $\boldsymbol{\theta}_j$ separately, without worrying about interdependence of the parameters. We may therefore drop the class subscripts j

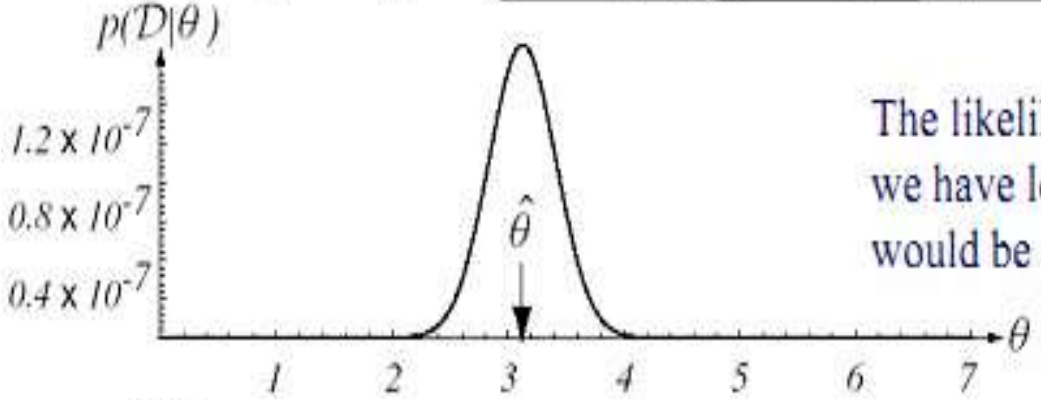
Maximum-Likelihood Estimation

- Our problem is to **use the information provided by the training samples** to **obtain good estimates for the unknown parameter** vectors associated with each category.

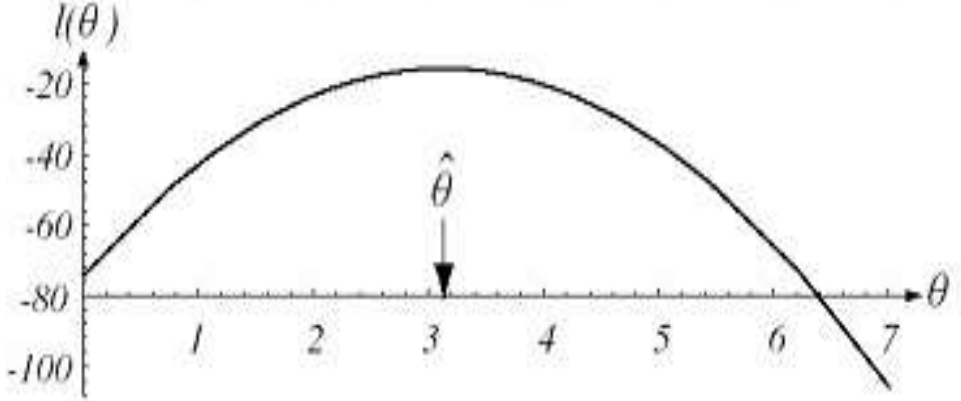


Few of the infinite number of possible Gaussian distributions of some variance and unknown mean θ from which the observed values might be drawn

Observed data points (1D)



The likelihood $p(D \mid \boldsymbol{\theta})$ as a function of the mean. If we have lots of training points, this likelihood would be very narrow.

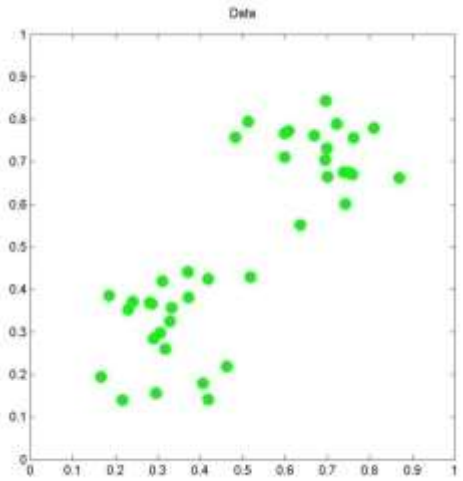


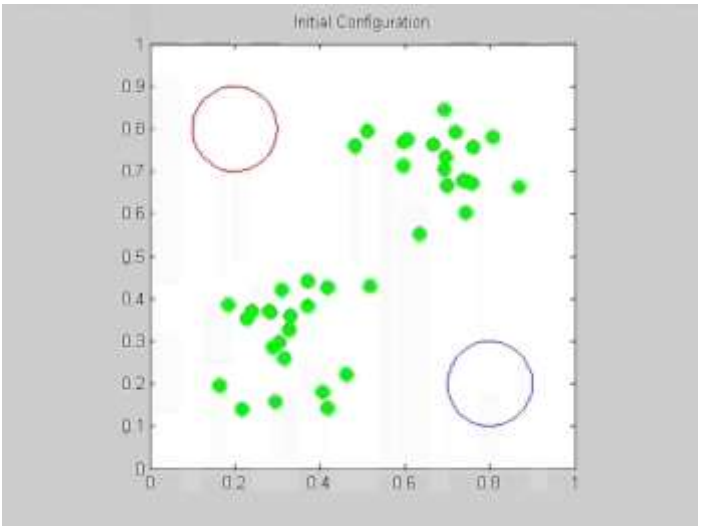
The **θ** that maximizes this likelihood – the value we are looking for – also maximizes the logarithm of the likelihood $l(\boldsymbol{\theta})$. Since log is a monotonically increasing function

K-means clustering

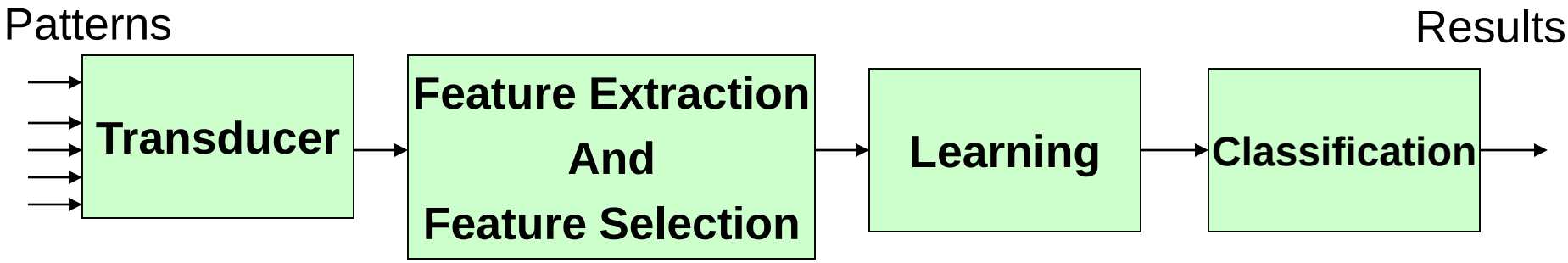
1. Guess number of clusters, K
2. Guess initial cluster centers, μ_1, μ_2
3. Assign data points x_i to nearest cluster center
4. Re-compute cluster centers based on assignments

Reiterate





STATISTICAL APPROACH



Block diagram representation of statistical approach

Transducer : It is used for making measurements for various attributes of the pattern.
Feature Extractor: From the measurements, it extracts, number of features which are required for describing the pattern and classifying.
Feature selector : Depending on the problem the feature selector selects minimum number of features that are sufficient to classify the pattern.

STATISTICAL APPROACH

There are two feature selector methods.

1.Transformation Method :

Here we reduce the features by considering the linear or nonlinear combinations of original features. This is also called as aggregation method.

Eg:- let us assume originally we have four features f1,f2,f3,f4.

One method of selecting two features is

f5 = f1 + f2

f6 = f3 + f4.

2.Subsetting or filtering Method:

Here we select a subset of the original features.

Eg:- Original features are f1,f2,f3,f4.

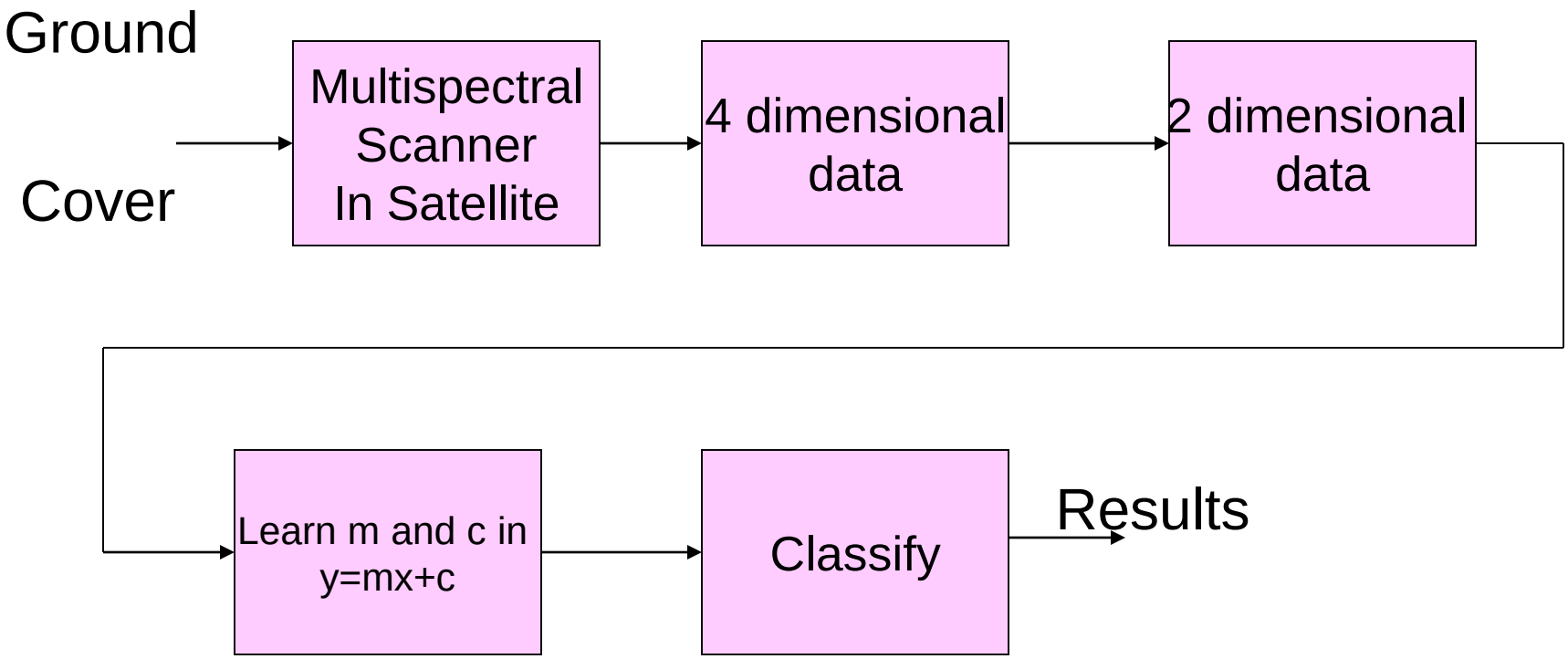
We can select a subset like

f5 = f1 and f6 = f3.

Learning : It is a process of determining useful parameters which are required for classifying the patterns efficiently.

Classifying: Here the patterns are assigned to different classes using a suitable classification method as shown in the fig 1.1

Classification Method



A Classification Method

SYNTACTIC APPROACH

Here we use the analogy between the structures of a pattern and the structure of sentence, written using a grammar.

E.g.: Rama was a very good king.

Here we decompose the pattern into sub-patterns called primitives when primitives are combined together using a certain syntax rule, we get the original pattern. So this method consists of parsing the pattern using a syntax rule.

Advantages : It classifies the pattern.

It describes the pattern.

MACHINE PERCEPTION

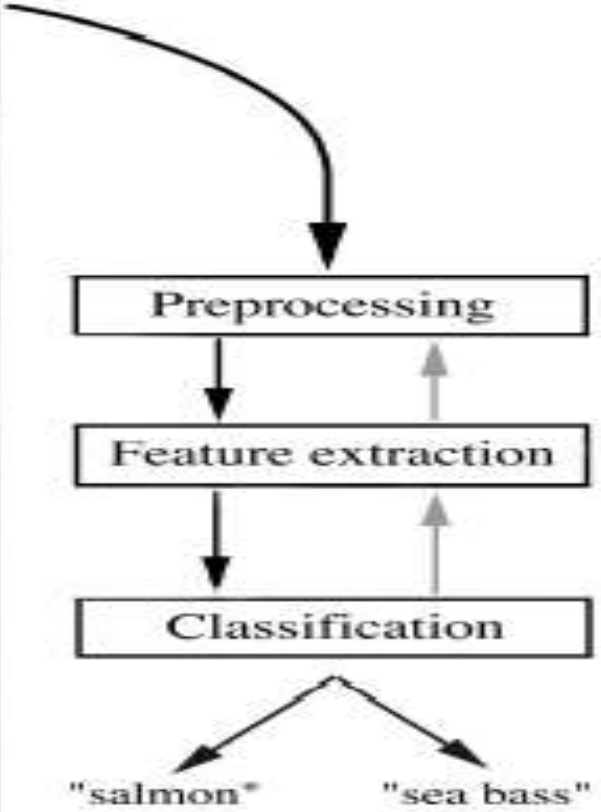
It is natural that we should seek to design and build machines that can recognize patterns. From automated speech recognition, fingerprint identification, optical character recognition, DNA sequence identification, and much more.

Moreover, in solving the myriad problems required to build such systems, we gain deeper understanding and appreciation for pattern recognition systems in the natural world- most particularly in humans. For some problems, such as speech and visual recognition, our design efforts may in fact be influenced by knowledge of how these are solved in nature, both in the algorithms we employ and in the design of special-purpose hardware.

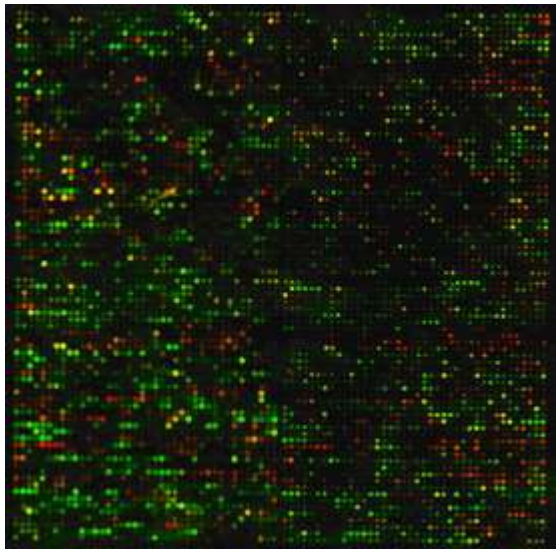
Model

Given that there are differences between the population of sea bass and that of salmon, we view them as having different models-different descriptions, which are typically mathematical in form. The overarching goal and approach in pattern classification is to hypothesize the class of these models, process the sensed data to eliminate noise (not due the models), and for any sensed pattern chose the conceptual toolbox of the designer of pattern recognition systems.

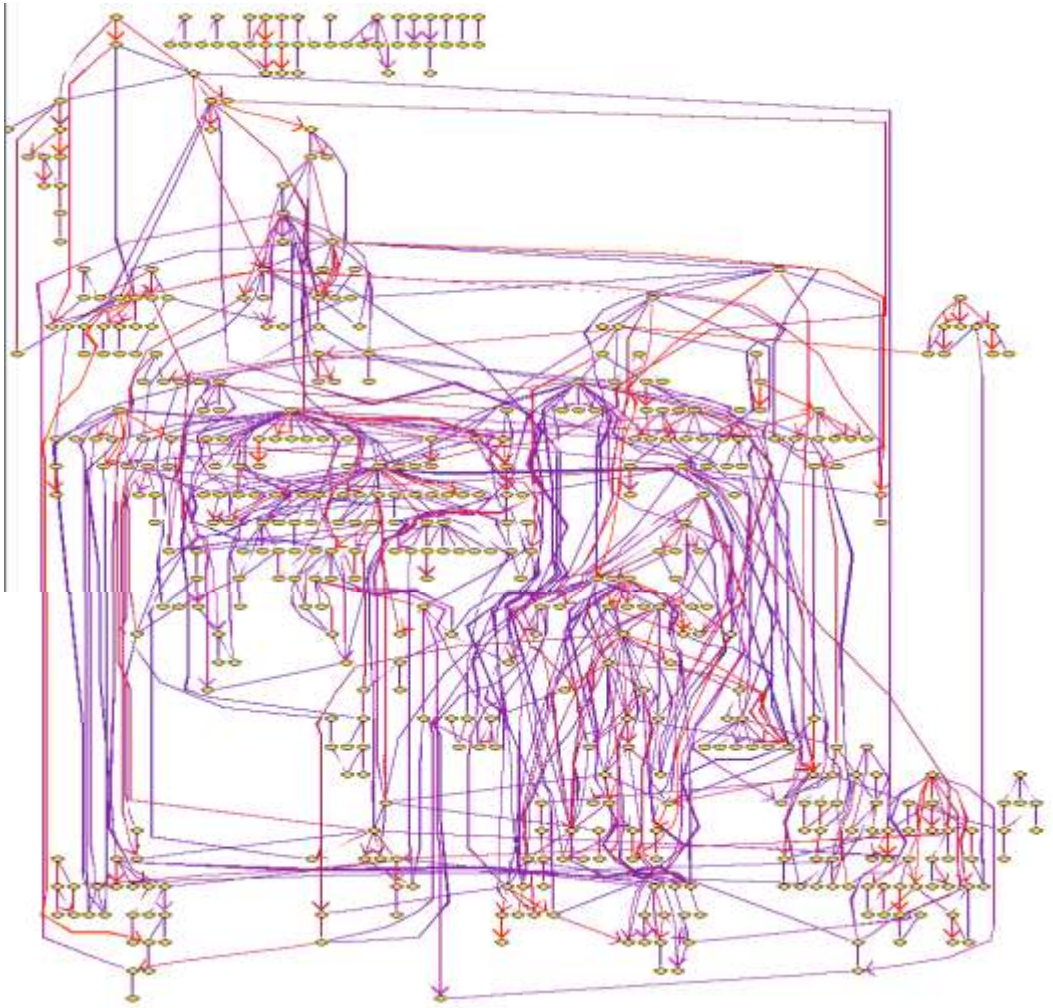
AN EXAMPLE


Preprocessing
Feature extraction
Classification
"salmon"
"sea bass"

Discovering networks



?



From data visualization to causal discovery

Classification Process

A Simple system to perform classification might have the following form as shown in the figure above. First the camera captures an image of the fish. Next, the camera's signals are preprocessed to simplify subsequent operations without losing relevant information. In particular, we might use a segmentation operation in which the images of different fish are somehow isolated from one another and from the background. The information from a single fish is then sent to a feature extractor, whose purpose is to reduce the data by measuring certain "features" of "properties.“

These features (or, more precisely, the values of these features) are then passed to a classifier that evaluates the evidence presented and makes a final decision as to the species.

Training Samples

The preprocessor might automatically adjust for average light level or threshold the image to remove the background of the conveyor belt, and so forth. For the moment let us pass over how the images of the fish might be segmented and consider how the feature extractor and classifier might be designed. Suppose somebody at the fish plant tells us that a sea bass is generally longer than a salmon.

These, then, give us our tentative models for the fish: Sea bass have some typical length, and this is greater than that for salmon. Then length becomes an obvious feature, and we might attempt to classify the fish merely by seeing whether or not the length l of a fish exceeds some critical value l*. To choose l* we could obtain some design or training samples of the different types of fish, make length measurements, and inspect the results.

Training Samples

- Suppose that we do this and obtain the histograms shown in Fig. 1.4. These disappointing histograms bear out the statement that sea bass are somewhat longer than salmon, on average, but it is clear that his single criterion is quite poor; no matter how we choose l*, we cannot reliably separate sea bass from salmon by length alone.

- Discouraged, but undeterred by these unpromising results, we try another feature, namely the average lightness of the fish scales. Now we are very careful to eliminate variations in illumination, because they can only obscure the models and corrupt our new classifier. The resulting histograms and critical value x*, shown in Fig. 1.5, are much more satisfactory: The classes are much better separated.

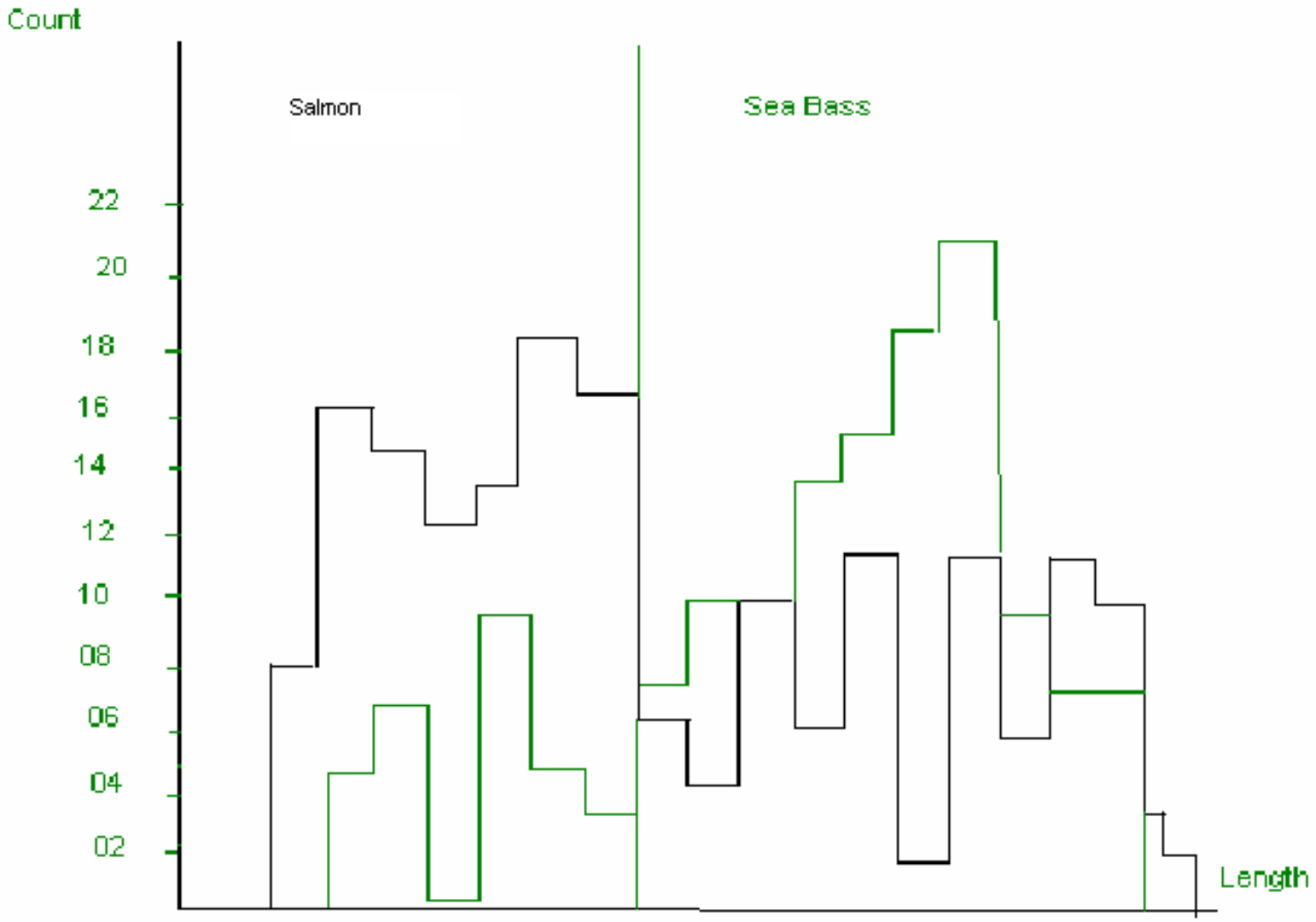


Fig 1.4: Histograms for the length feature for Salmon and Sea Bass Fish

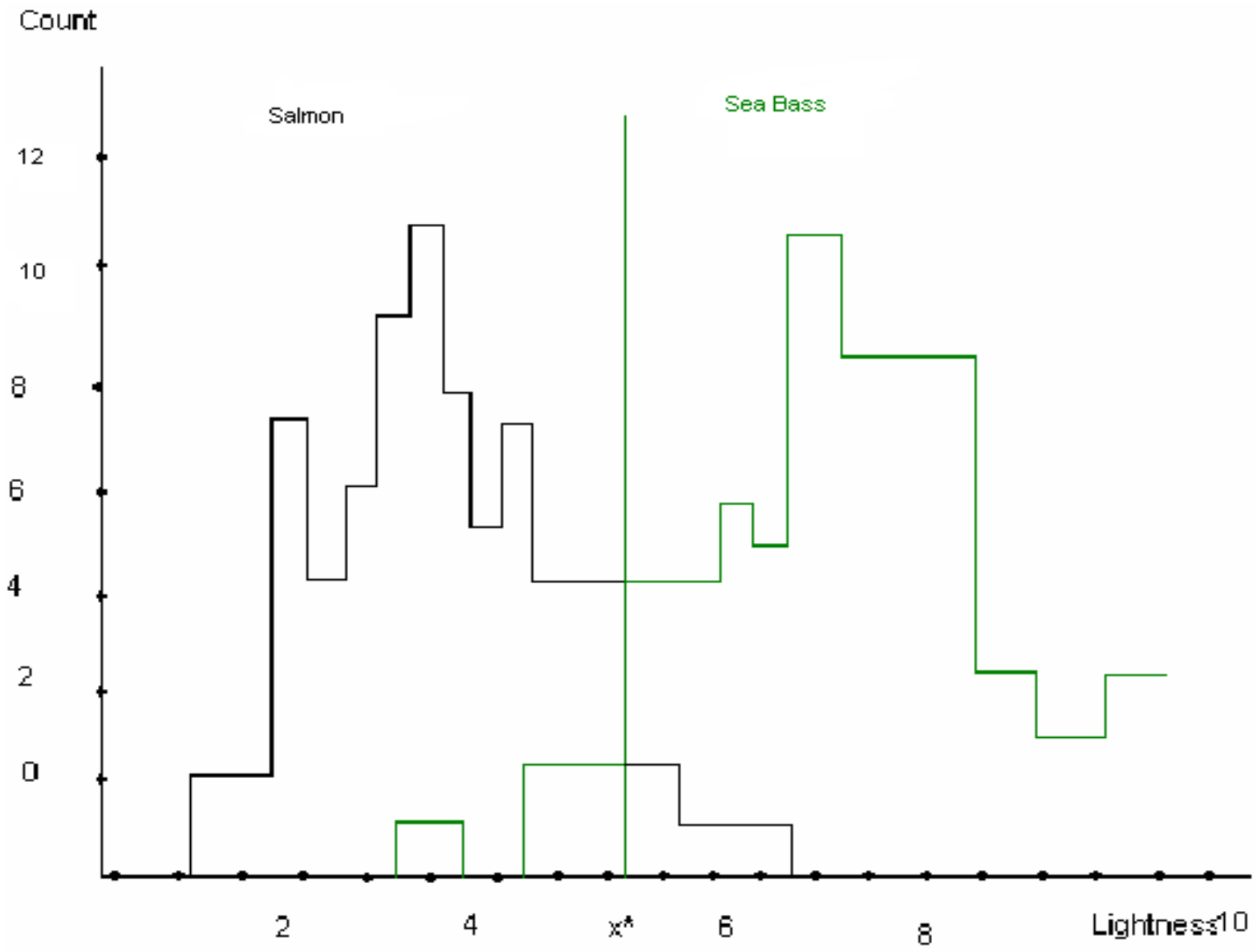


Fig 1.5 Histograms for the lightness feature of Salmon and Sea bass fish

Cost

So fair we have tacitly assumed that the consequences of our actions are equally costly: Deciding the fish was a sea bass when in fact it was a salmon was just as undesirable as the converse, Such a symmetry in the cost is often, but not invariably, the case. For instance, as a fish-packing company we may know that our customers easily accept occasional pieces of tasty salmon in their cans labeled " sea bass," but they object vigorously if a piece of sea bass appears in their cans labeled "salmon." If we want to stay in business, we should adjust our decisions to avoid antagonizing our customers, even if it means that more salmon makes its way into the cans of sea bass. In this case, then, we should move our decision boundary to smaller values of lightness, thereby reducing the number of sea bass that are classified as salmon (Fig.1.5). The more our customers object to getting sea bass with their salmon (i.e., the more costly this type of error) the lower we should set the decision threshold x* in Fig1.5.

Decision Theory, Decision Boundary

Based on the above discussion we can say that there is an overall single cost associated with our decision, and our true task is to make a decision rule (I,e., set of decision boundary) so as to minimize such a cost. This is the central task of decision theory of which pattern classification is perhaps the most important subfield.

Even if we know the costs associated with our decisions and choose the optimal critical value x*, we may be dissatisfied with the resulting performance.

It is observed that sea bass is wider than salmon which can be used as another feature. Now we have two features for classifying fish, the lightness X1 and the width X2. The feature extractor has thus reduced the image of each fish to a point or feature vector X in a two dimensional feature space, where,

$$X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$$

Decision Theory, Decision Boundary

Our problem now is to partition the feature space into two regions, where for all points in one region we will call the fish a sea bass, and for all points in the other we call it a salmon. Suppose that we measure the feature vectors for our samples and obtain the scattering of points shown in fig 1.6 .

This plot suggests the following decision rule for separating the fish: classify the unknown fish as the sea bass if its feature vector falls to the right of the decision boundary, else classify them as salmon.

This rule appears to do a good job of separating our samples and suggests that perhaps incorporating few more features would be more desirable. Besides the lightness and width of the fish we might include some shape parameter, such as the vertex angle of the dorsal fin, or the placement of the eyes (as expressed as a proportion of the mouth-to-tail distance), and so on.

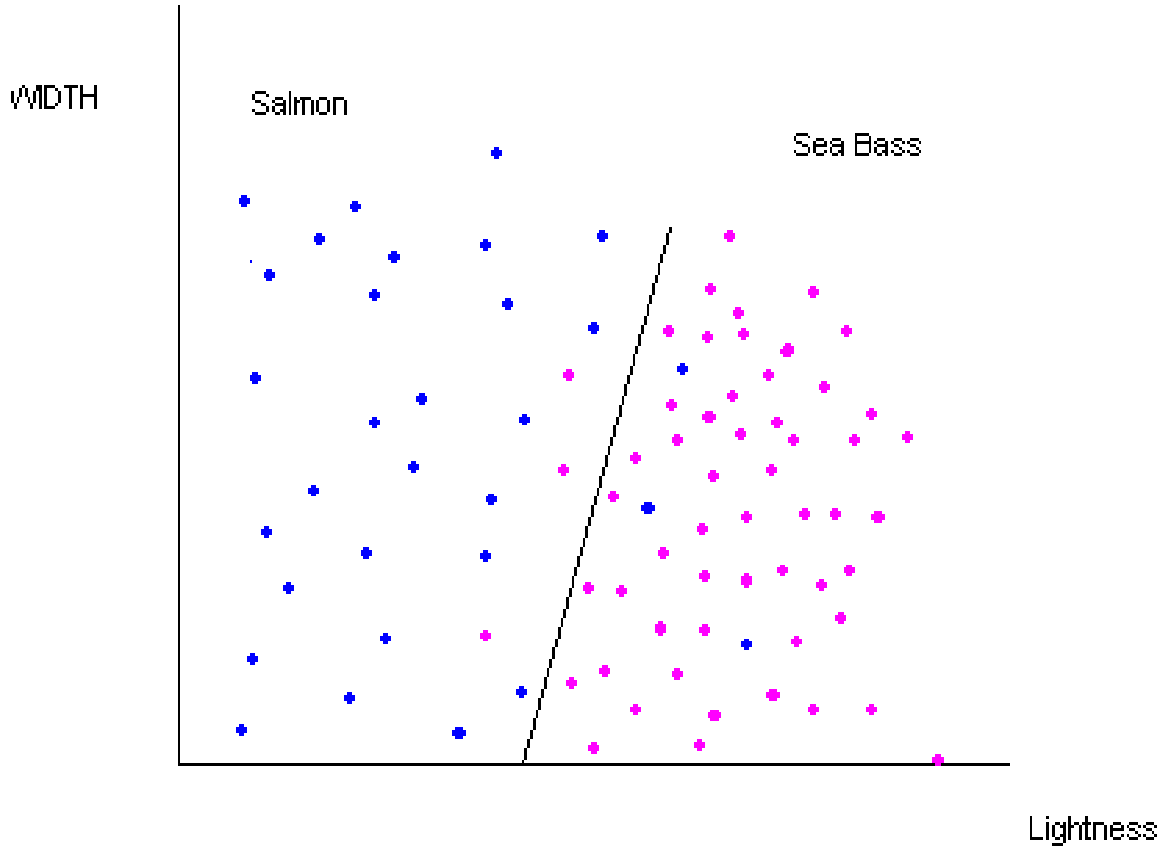


Fig 1.6: Linear Decision Boundary

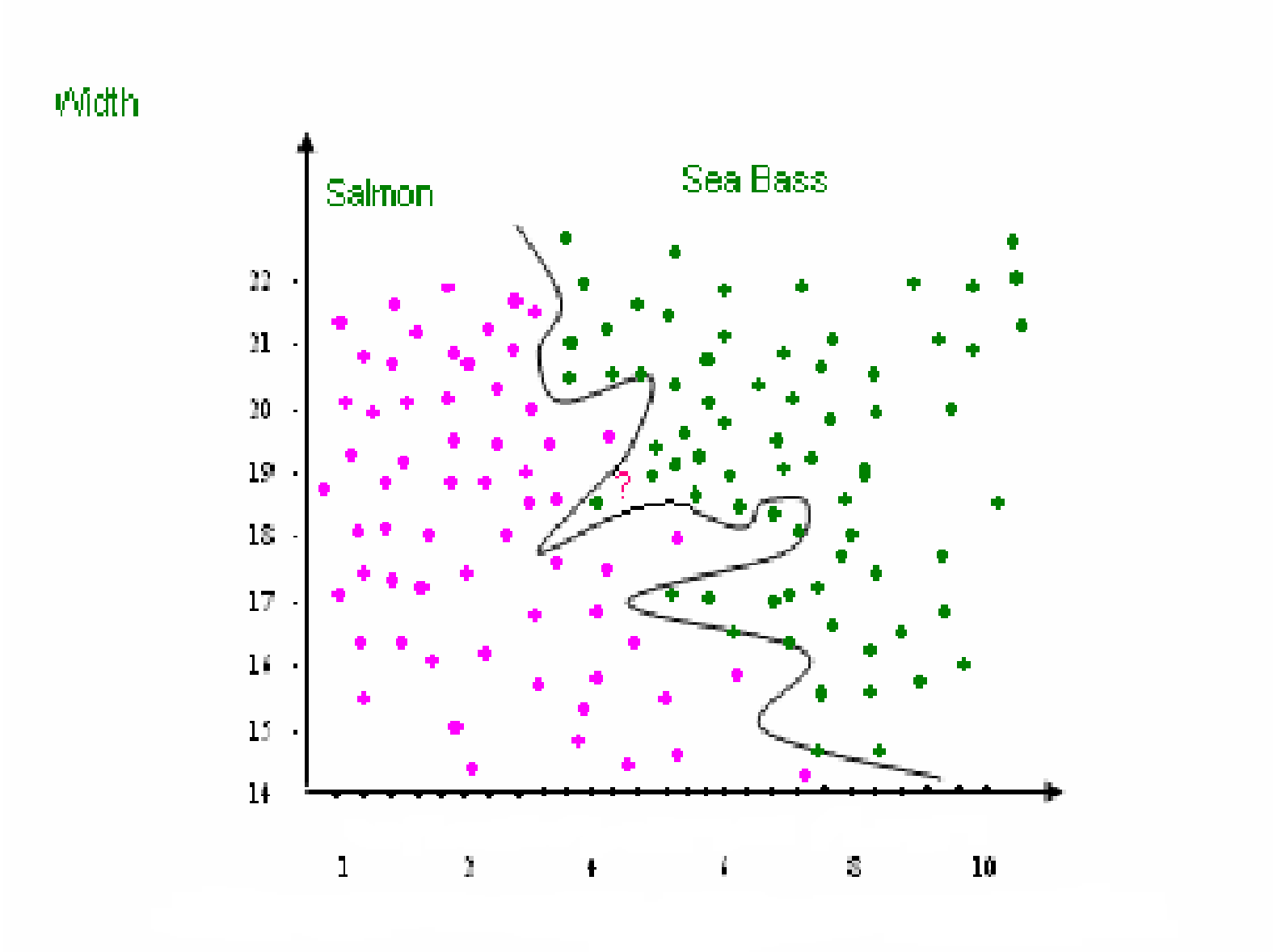
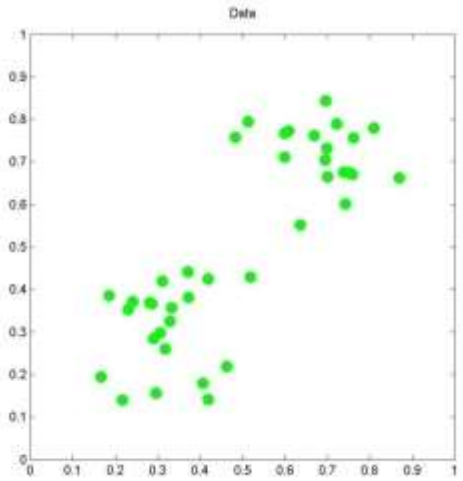


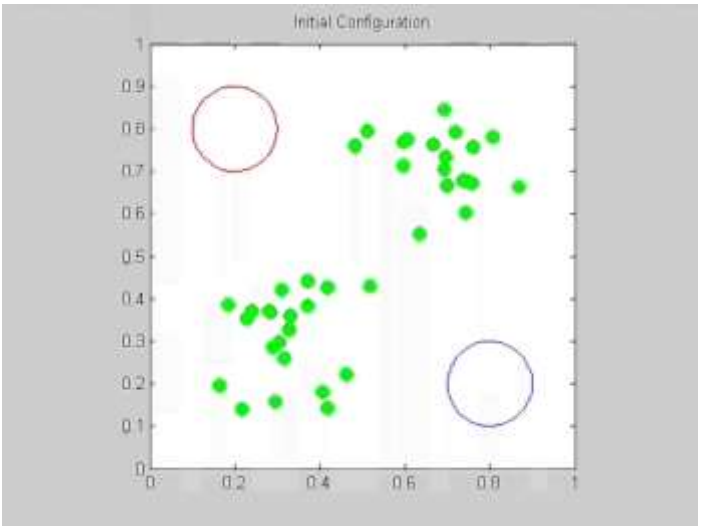
Fig 1.7: Non-Linear Decision Boundary for perfect Classification

K-means clustering

1. Guess number of clusters, K
2. Guess initial cluster centers, μ_1, μ_2
3. Assign data points x_i to nearest cluster center
4. Re-compute cluster centers based on assignments

Reiterate





Generalization

Suppose that features are too measure, or provides little improvement (or possibly even degrade the performance) in the approach described above, and that we are forced to make our decision based on the two features in fig., 1.6. If our models were extremely complicated, our classifier would have a decision boundary more complex than the simple straight line. In that case all the training patterns would be separated perfectly, as shown in Fig. 1.7, With such a "solution," though, our satisfaction would be premature because the central aim of designing a classifier is to suggest actions when presented with novel patterns, that is, fish not yet seen.

Generalization

- This is the issue of generalization. It is unlikely that the complex decision boundary in Fig.1.7 would provide good generalization. It is unlikely that the complex decision boundary in Fig. 1.7 would provide good generalization - it seems to be "turned" to the particular training samples, rather than some underlying characteristics or true model of all the sea bass and salmon that will have to be separated.
- Naturally, one approach would be to get more training samples for obtaining a better estimate of the true underlying characteristics, for instance the probability distributions of the categories. In some pattern recognition problems, however, the amount of such data we can obtain easily is often quite limited. Even with vast amount of training data in a continuous feature space though, if we followed the approach in Fig 1.7 our classifier would be unlikely to do well on novel patterns.

Generalization

- Rather, then, we might seek to "simplify" the recognizer, motivated by a belief that the underlying models will not require a decision boundary that is as complex as that in Fig 1.7. Indeed, we might be satisfied with the slightly poorer performance on the training samples if it means that our classifier will have better performance on novel patterns.
- But if designing a very complex recognizer is unlikely to give good generalization, precisely

  How should we quantify and favor simpler classifiers?

  How would our system automatically determine that simple curve in the figure 1.8 is preferable to the manifestly simpler straight line in fig1.6, or the compacted boundary in the fig1.7. Assuming that we somehow manage to optimize this tradeoff, can we then predict how well our system will generalize our new patterns?

  These are some of the central problems in statistical pattern organisation.

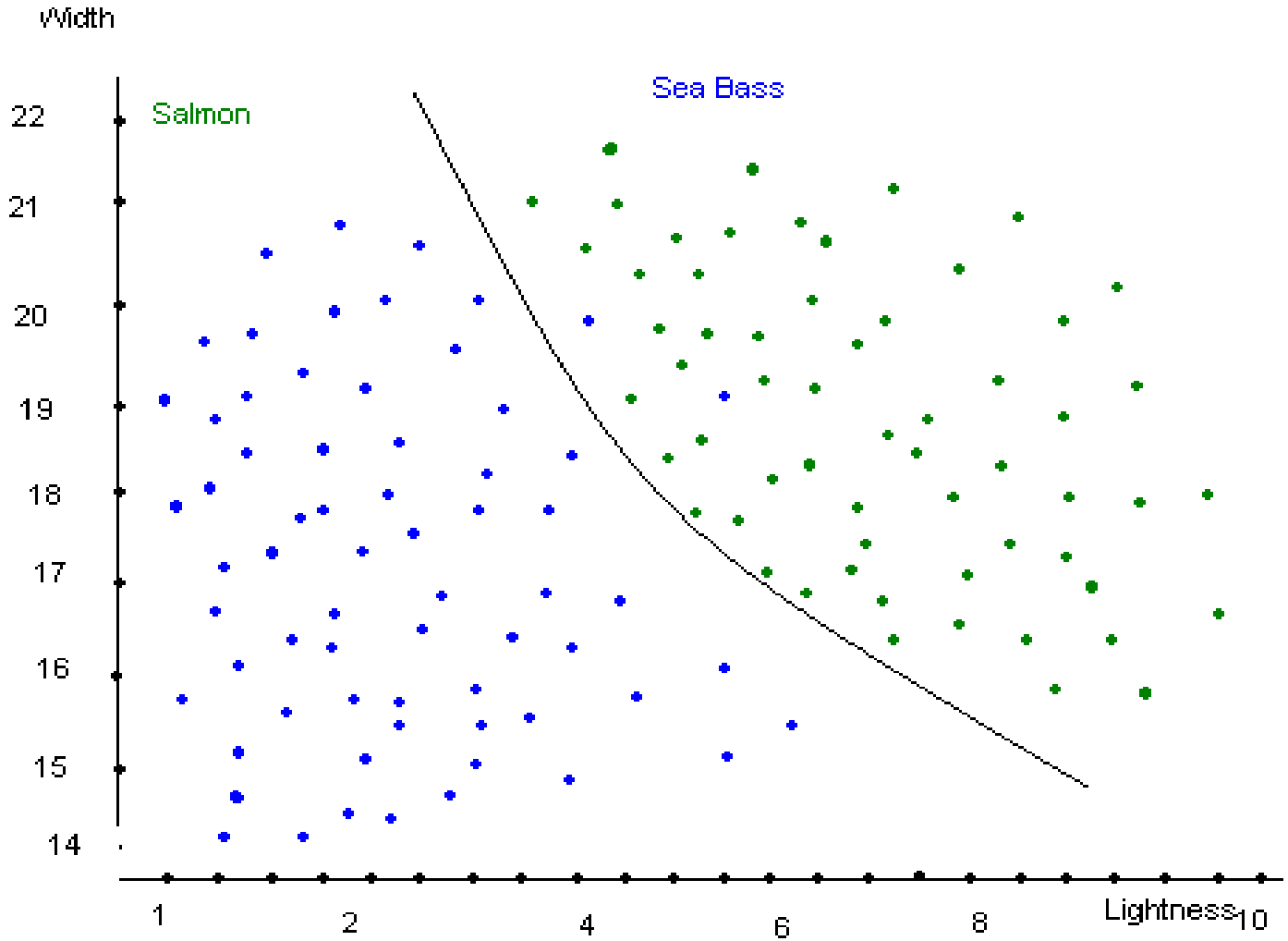


Fig 1.8: The decision boundary shown might represent the optimal tradeoff Between performance on the training set and simplicity of classifier

Generalization

- For the same incoming patterns, we might need to use a drastically different task or cost function, and this will lead to different actions altogether. We might, for instance, wish to separate the fish based on their sex – all females (of either species) from all males – if we wish to sell.
- The damaged fish (to prepare separately for food), and so on. Different decision tasks may require features and yield boundaries quite different from those useful for our original categorization problem.
- This makes it quite clear that our decisions are fundamentally task – or – cost specific, and that creating a single general purpose artificial pattern recognition device – that is one capable of acting accurately based on a wide variety of tasks – which is profoundly a difficult challenge.

PATTERN RECOGNITION SYSTEMS

In describing our hypothetical fish classification system, we distinguished between the three different operations of preprocessing, feature extraction and classification (see Fig. 1.3). Figure 1.9 shows a slightly more elaborate diagram of the components of a typical pattern recognition system. To understand the problem of designing such a system, we must understand the problems that each of these components must solve. Let us consider the operations of each component in term, and reflect on the kinds of problems that can arise.

PATTERN RECOGNITION SYSTEMS

- **Sensing**

  The input to a pattern recognition system is often some kind of a transducer, such as a camera or microphone array. The difficult of the problem may well depend on the characteristics and limitations of the transducer its bandwidth, resolution sensitivity, distortion, signal-to-noise ratio, latency etc. As important as it is in practice, the design of sensors for pattern recognition is beyond the scope of this book

- **Segmentation and Grouping**

  In our fish example, we tacitly assumed that each fish was isolated, separate from others on the conveyor belt, and could easily be distinguished from the conveyor belt.

PATTERN RECOGNITION SYSTEMS

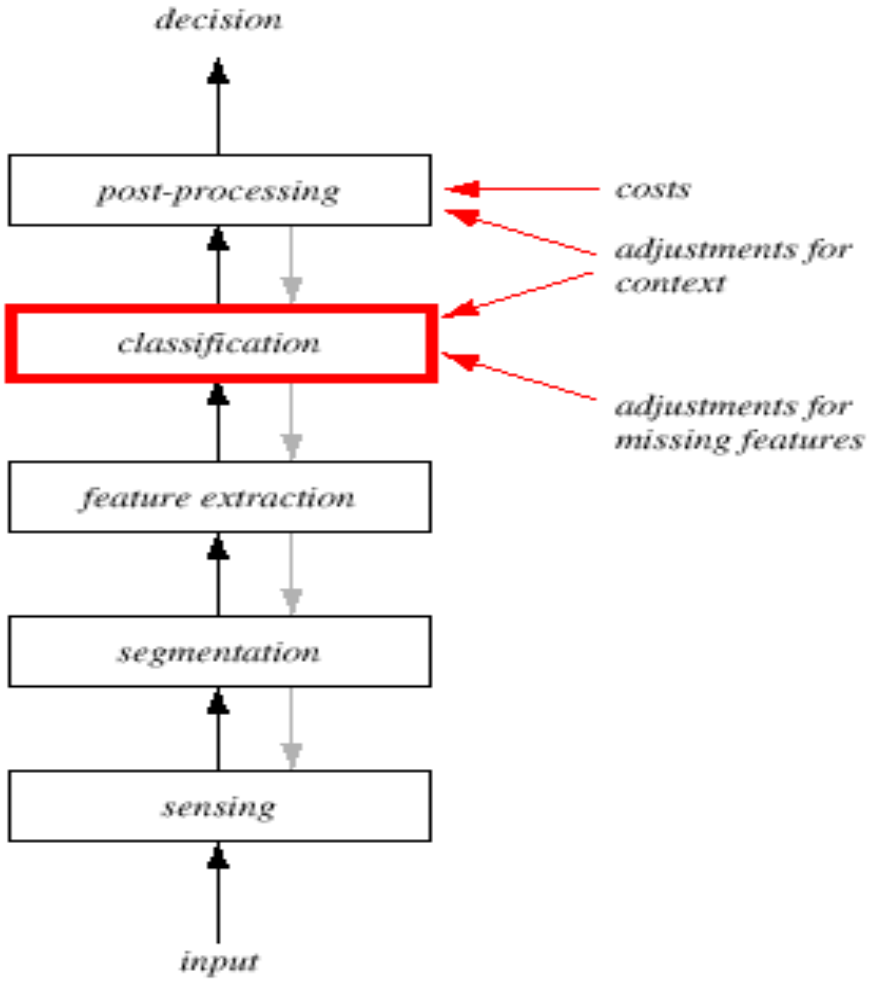


Fig 1.9 Components of a typical Pattern Recognition System

PATTERN RECOGNITION SYSTEMS

In practice, the fish would often be abutting or overlapping, and our system would have to determine where one fish ends and the next begins-the individual patterns have to be segmented. If we have already recognized the fish then it would be easier to segment their images. But how can we segment the images before they have been categorized, or categorize them before they have been segmented? It seems we need a way to know when we have switched from one model to another, or to know when we just have background or "no category." How can this be done?

Segmentation is one of the deepest problems in pattern recognition.

PATTERN RECOGNITION SYSTEMS

- **Feature Extraction**

The conceptual boundary between feature extraction and classification proper is somewhat arbitrary: An ideal feature extractor would yield a representation that makes the job of the classifier trivial, conversely, an omnipotent classifier would not need the help of a of a sophisticated feature extractor. The distinction is forced upon us for practical, rather than theoretical reasons.

PATTERN RECOGNITION SYSTEMS – Feature Extraction

Invariant Features : The traditional goal of the feature extractor is to characterize an object to be recognized by measurements whose values are very similar for objects in the same category, and very different for objects in different categories. This leads to the idea of seeking distinguishing features that are invariant to irrelevant transformations of the input. In our fish example, the absolute location of a fish on the conveyor belt is irrelevant to the category, and thus our representation should also be insensitive to the absolute location of the fish. Ideally, in this case we want the features to be invariant to translation, whether horizontal or vertical. Because rotation is also irrelevant for classification, we would also like the features to be invariant to rotation. Finally, the size of the fish may not be important-a young, small salmon is still a salmon. Thus, we may also want the features to be invariant to scale, In general, features that describe properties such as shape, color and many kinds of texture are invariant to translation, rotation and scale.

PATTERN RECOGNITION SYSTEMS – Feature Extraction

- Occlusion and Projective Distortion

The problem of finding rotation invariant features from an overhead image of a fish on a conveyor belt is simplified by the fact that the fish is likely to be lying flat, and the axis of rotation is always parallel to the camera's line of sight. A more general invariance would be for rotations about an arbitrary line in three dimensions. The image of even such a "simple" object as a coffee cup undergoes radical variation as the cup is rotated to an arbitrary angle:

The handle may become occluded-that is, hidden by another part. The bottom of the inside volume come into view, the circular lip appear oval or a straight line or event obscured, and so forth. Furthermore, if the distance between the cup and the camera can change, the image is subject to projective distortion. How might we ensure that the features are invariant to such complex transformations? Or should we define different subcategories for the image of a cup and achieve the rotation invariance at a higher level of processing?

PATTERN RECOGNITION SYSTEMS – Feature Extraction

In speech recognition, we want features that are invariant to translations in time and to changes in the overall amplitude. We may also want features that are insensitive to the duration of the word, i.e., invariant to the rate at which the pattern evolves. Rate variation is a serious problem in speech recognition. Not only do different people talk at different rates, but also even a single talker may vary in rate, causing the speech signal to change in complex ways.

Likewise, cursive handwriting varies in complex ways as the writer speeds up the placement of dots on the I's and cross bars on the t's and f's, are the first casualties of rate of rate increase, while the appearance of I's and e's are relatively inviolate. How can we make a recognizer that changes its representations for some categories differently from that for others under such rate variation?

PATTERN RECOGNITION SYSTEMS – Feature Extraction

Deformation

A large number of highly complex transformations arise in pattern recognition, and many are domain specific .We might wish to make our handwritten optical character recognizer insensitive to the overall thickness of the pen line, for instance .Far more sever are transformations such as no rigid deformations that arise in three dimensional object recognition, such as the radical variation in the image of your hand as you grasp as object or snap your fingers. Similarly ,variations in illumination or the complex effects of cast shadows may need to be taken into account.

PATTERN RECOGNITION SYSTEMS – Feature Extraction

Feature Selection

As with segmentation, the task of feature extraction is much more problem-and domain-dependent than is classification proper, and thus requires knowledge of the domain. A good feature extractor for sorting fish would probably be of little use for identifying fingerprints, or classifying photomicrographs of blood cells. However, some of the principles of pattern classification can be used in the design of the feature extractor. Although the pattern classification techniques presented in this book cannot substitute for domain knowledge, (they can be helpful in making the feature values less sensitive to noise.) In some cases, they can also be used to select the most valuable features from a larger set of candidate features.

PATTERN RECOGNITION SYSTEMS – Classification

The task of the classifier component proper of a full system is to use (the feature vector provided by the feature extractor to assign the object to a category) Most of this book is concerned with the design of the classifier. Because perfect classification performance is often impossible, a more general task is to determine the probability for each of the possible categories. The abstraction provided by the feature-vector representation of the input data enables the development of a largely domain-independent theory of classification.

PATTERN RECOGNITION SYSTEMS – Classification

- Noise

The degree of difficulty of the classification problem depends on the variability in the feature values for objects in the same category relative to the difference between feature values for objects in different categories. The variability of feature values for objects in the same category may be due to complexity, and may be due to noise.

We define noise in very general terms: any property of the sensed pattern, which is not due to the true underlying model but instead to randomness in the word or the sensors. All nontrivial decision and pattern recognition problems involve noise in some form. What is the best way to design a classifier to cope with this variability? What is the best performance that is possible?

PATTERN RECOGNITION SYSTEMS – Classification

One problem that arises in practice is that it may not always be possible to determine the values of all of the features for a particular input. In our hypothetical system for fish classification, for example, it may not be possible to determine the width of the fish because of occlusion by another fish. How should the categorize compensate?

Since our two-feature recognizer never had a single-variable criterion value (x* determined in anticipation of the possible absence of a feature) (cf. Fig. 1.3), how shall it make the best decision using only the feature present? The naïve method. Of merely assuming that the value of the missing feature is zero or the average of the values for the patterns already seen, is provably no optimal. Likewise, how should we train a classifier or use one when some features are missing?

PATTERN RECOGNITION SYSTEMS – Post Processing

A classifier rarely exists in a vacuum, Instead, it is generally to be used to recommend actions (put this fish in this bucket, put that fish in that bucket), each action having an associated cost. The post-processor uses the output of the classifier to decide on the recommended action.

PATTERN RECOGNITION SYSTEMS – Post Processing

Error rate Risk Conceptually, the simplest measure of classifier performance is the classification error rate-the percentage of new patterns that are assigned to the wrong category. Thus, it is common to seek minimum-error-rate classification. However, it may be much better to recommend actions that will minimize the total expected cost, which is called the risk. How do we incorporate knowledge about costs and how will they affect our classification decision? Can we captivate the total risk and thus tell whether our classifier is acceptable even before we field it? Can we estimate the lowest possible risk of any classifier, to see how close ours meets this ideal, or whether the problem is simply too hard overall?

PATTERN RECOGNITION SYSTEMS – Post Processing

Context The post processed might also be able to exploit contest input dependent information other than from the target pattern itself-to improve system performance. Suppose in an optical character recognition system we encounter a acquiesce that looks like T/-\E C/-\T. Even though the system may be unable to classify each /-\ as an isolated character, in the context of English it is clear that the first instance should be an H and the second an A. Context can be highly complex and abstract. The utterance "jeetyet?" may seem nonsensical, unless you hear it spoken by a friend in the context of the cafeteria at lunchtime- "did you eat yet?" How can such a visual and temporal context influence your recognition of speech?

PATTERN RECOGNITION SYSTEMS – Post Processing

Multiple Classifiers:

In our fish example we saw how using multiple features could lead to improved recognition. We might imagine that we could also do better if we used multiple classifiers, each classifier operating on different aspects of the input. For example, we might combine the results of acoustic recognition and lip reading to improve the performance of a speech recognizer.

If all of the classifiers agree on a particular pattern, there is no difficulty. But suppose they disagree. How should a "super" classifier pool the evidence from the component recognizers to achieve the best decision? Imagine calling in ten experts for determining whether or not a particular fish is diseased. While nine agree that the fish is healthy, one expert does not. Who is the Crazy Man right? It may be that the lone dissenter is the only one familiar with the particular very rare symptoms in the fish, and is in fact correct, How would the "super" categorizer know when to base a decision on a minority opinion, even from an expert in one small domain who is not well-qualified to judge throughout a broad range of problems?

PATTERN RECOGNITION SYSTEMS – Post Processing

Our purpose was to emphasize the complexity of pattern recognition problems and to dispel naïve hope that any single approach has the power to solve all pattern recognition problems. The methods presented in this book are primarily useful for the classification step. However, performance on difficult pattern recognition problems generally requires exploiting domain-specific knowledge.

LEARNING AND ADAPTATION

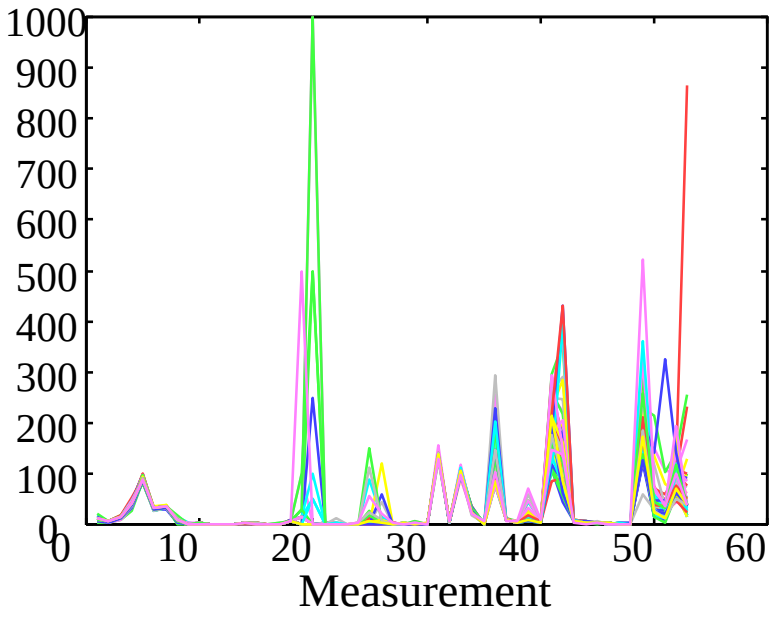
In the broadest sense, any method that incorporates information from the training samples in the design of a classifier employs learning. Because nearly all practical or interesting pattern recognition problems are so hard that we cannot guess the best classification decision ahead of time, we shall spend the great majority of our time here considering learning. Creating classifiers then involves positing some general form of model, or form of the classifier, and using training patterns to learn or estimate the unknown parameters of the model. Learning refers to some form of algorithm for reducing the error on a set of training data. A range of gradient descent algorithms that alter a classifier's parameters in order to reduce an error measure now permeate the field of statistical pattern recognition , and these will demand a great deal of our attention. Learning comes in several general forms.

Data Presentation

- Example: 53 Blood and urine measurements (wet chemistry) from 65 people (33 alcoholics, 32 non-alcoholics).
- Matrix Format

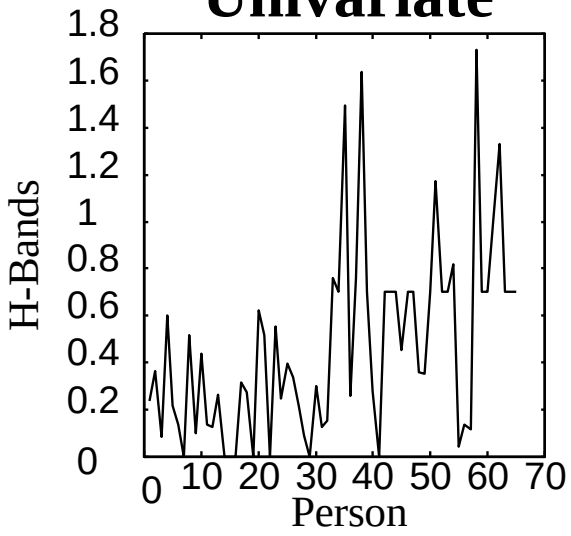
	H-WBC	H-RBC	H-Hgb	H-Hct	H-MCV	H-MCH	H-MCHC
A1	8.0000	4.8200	14.1000	41.0000	85.0000	29.0000	34.0000
A2	7.3000	5.0200	14.7000	43.0000	86.0000	29.0000	34.0000
A3	4.3000	4.4800	14.1000	41.0000	91.0000	32.0000	35.0000
A4	7.5000	4.4700	14.9000	45.0000	101.0000	33.0000	33.0000
A5	7.3000	5.5200	15.4000	46.0000	84.0000	28.0000	33.0000
A6	6.9000	4.8600	16.0000	47.0000	97.0000	33.0000	34.0000
A7	7.8000	4.6800	14.7000	43.0000	92.0000	31.0000	34.0000
A8	8.6000	4.8200	15.8000	42.0000	88.0000	33.0000	37.0000
A9	5.1000	4.7100	14.0000	43.0000	92.0000	30.0000	32.0000

- Spectral Format

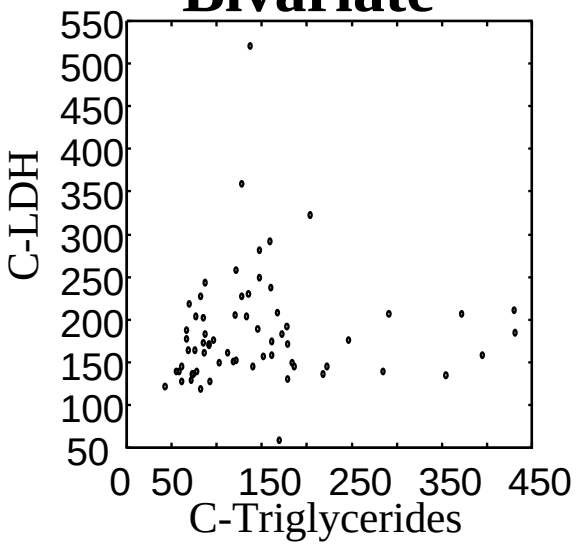


Data Presentation

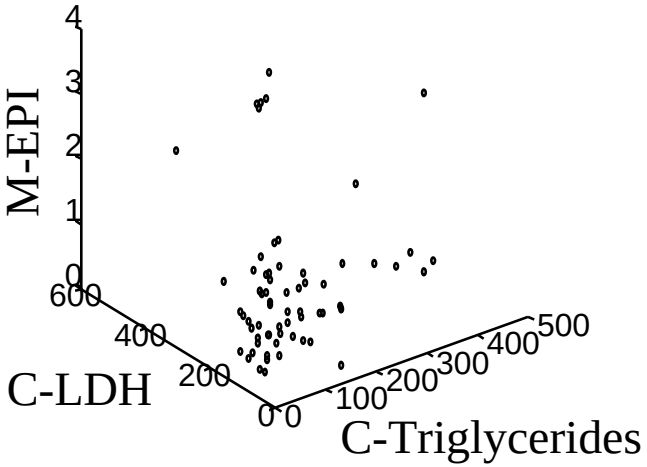
Univariate



Bivariate



Trivariate

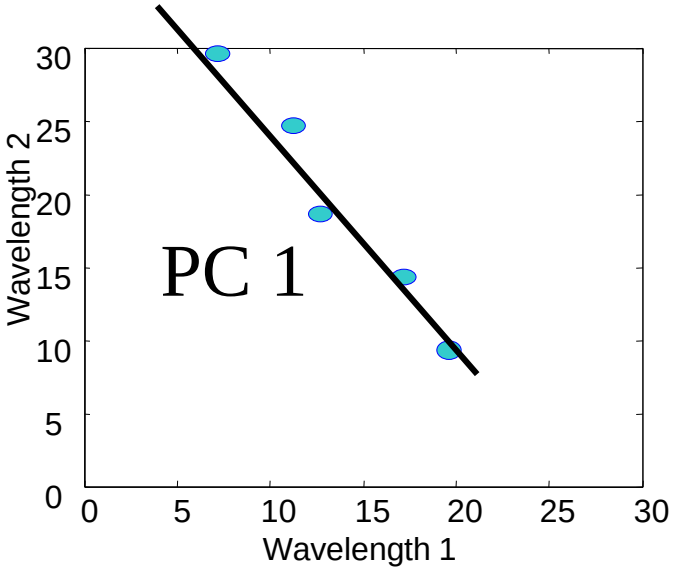


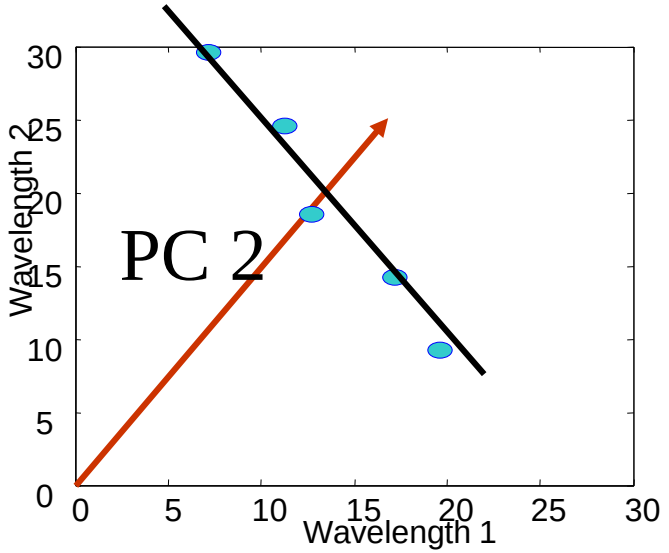
Data Presentation

- Better presentation than ordinate axes?
- Do we need a 53 dimension space to view data?
- How to find the ‘best’ low dimension space that conveys maximum useful information?
- One answer: Find “Principal Components”

Principal Components

- All principal components (PCs) start at the origin of the ordinate axes.
- First PC is direction of maximum variance from origin
- Subsequent PCs are orthogonal to 1st PC and describe maximum residual variance





The Goal

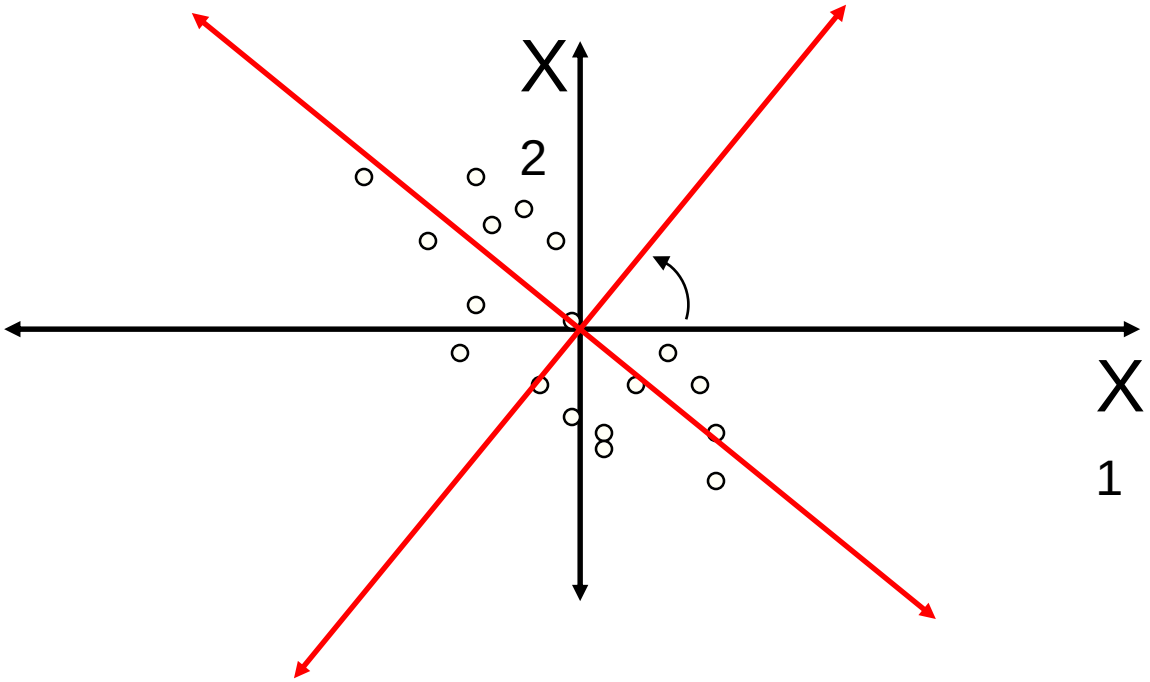
We wish to explain/summarize the underlying variance-covariance structure of a large set of variables through a few linear combinations of these variables.

Applications

- Uses:
  - Data Visualization
  - Data Reduction
  - Data Classification
  - Trend Analysis
  - Factor Analysis
  - Noise Reduction
- Examples:
  - How many unique “sub-sets” are in the sample?
  - How are they similar / different?
  - What are the underlying factors that influence the samples?
  - Which time / temporal trends are (anti)correlated?
  - Which measurements are needed to differentiate?
  - How to best present what is “interesting”?
  - Which “sub-set” does this new sample rightfully belong?

Trick: Rotate Coordinate Axes

Suppose we have a population measured on p random variables $X_1,\ldots,X_p$. Note that these random variables represent the p-axes of the Cartesian coordinate system in which the population resides. Our goal is to develop a new set of p axes (linear combinations of the original p axes) in the directions of greatest variability:



This is accomplished by rotating the axes.

Principal component analysis

It is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called **principal components**. The number of principal components is less than or equal to the number of original variables.

PCA: *General*

From k original variables: $x_1,x_2,\ldots,x_k$:

Produce k new variables: $y_1,y_2,\ldots,y_k$:

$y_1 = a_{11}x_1 + a_{12}x_2 + \ldots + a_{1k}x_k$

$y_2 = a_{21}x_1 + a_{22}x_2 + \ldots + a_{2k}x_k$

...

$y_k = a_{k1}x_1 + a_{k2}x_2 + \ldots + a_{kk}x_k$

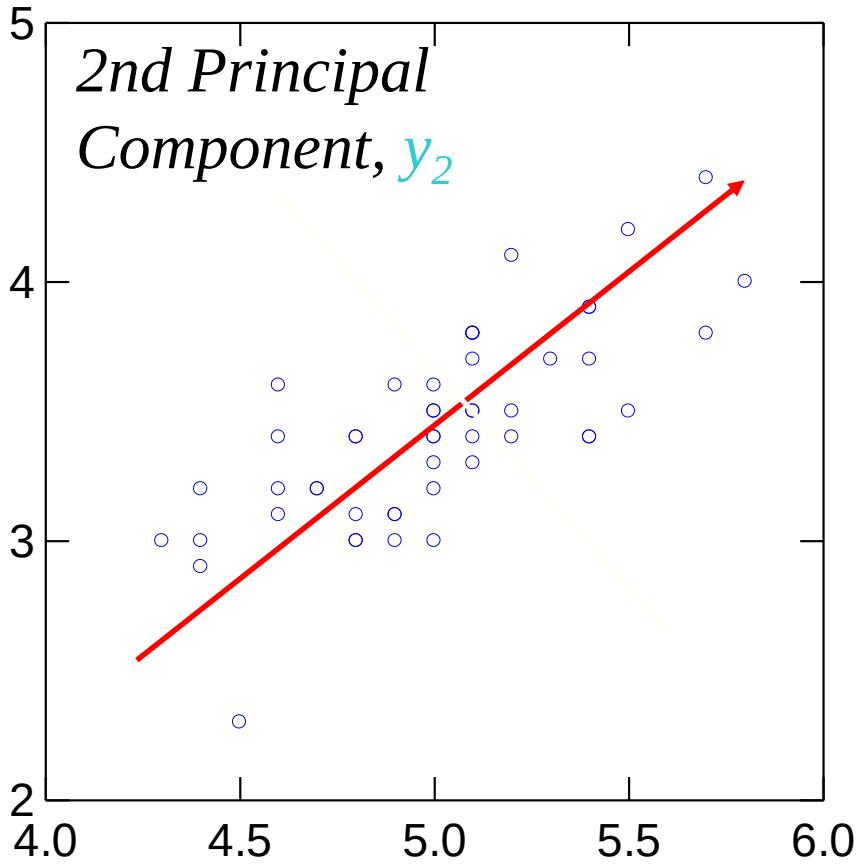
such that:

y_k's are uncorrelated (orthogonal)

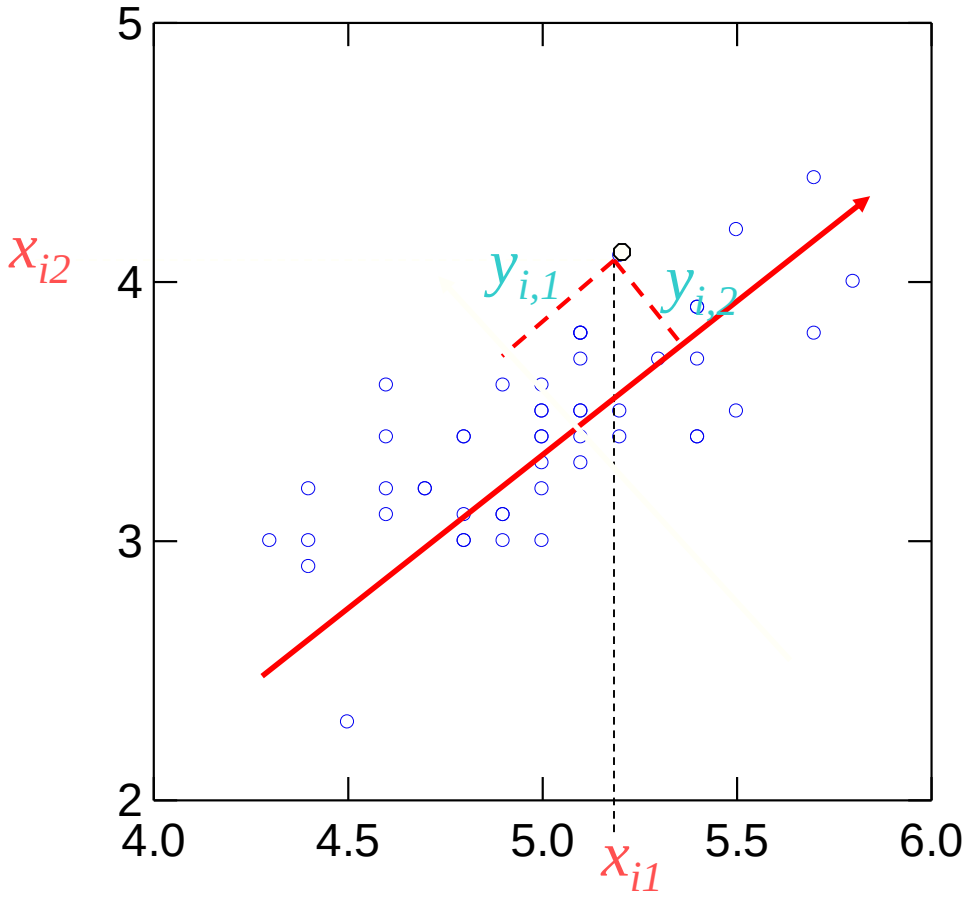
y_1 explains as much as possible of original variance in data set

y_2 explains as much as possible of remaining variance

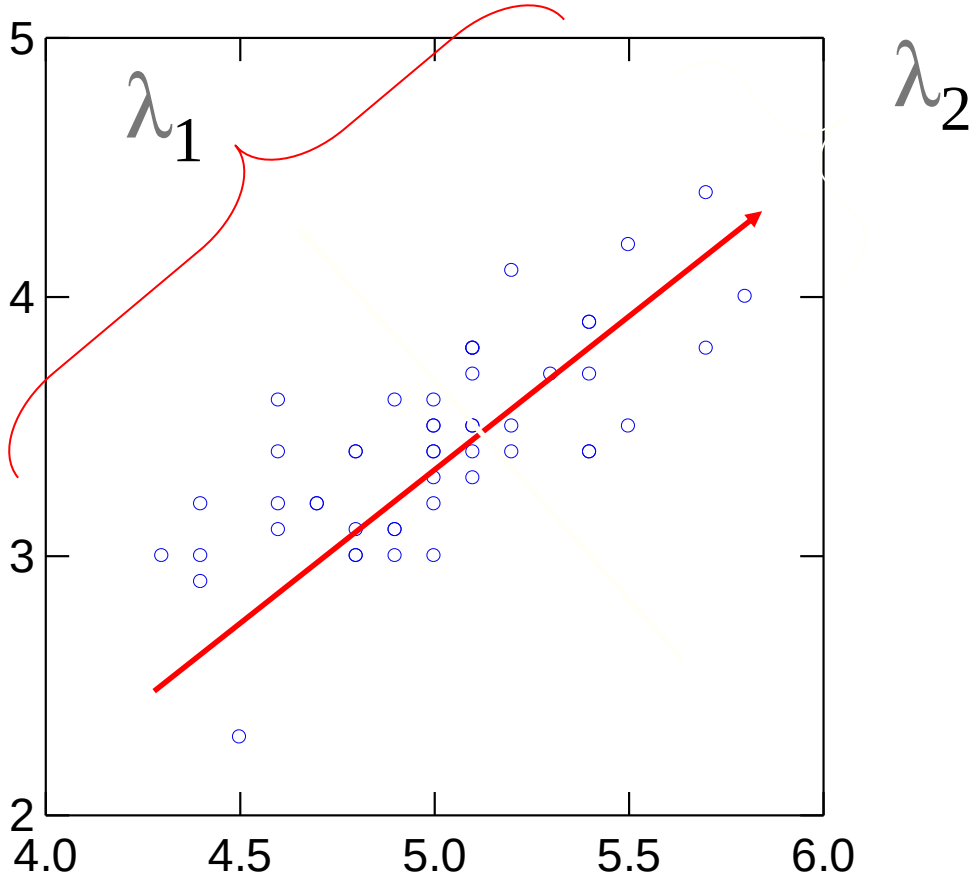
etc.


2nd Principal
Component, y_2
1st Principal
Component, y_1
5
4
3
2
4.0
4.5
5.0
5.5
6.0

PCA Scores



PCA Eigenvalues



PCA

This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it isorthogonal to (i.e.,

Pattern Recognition

Pattern recognition is:

1. The name of the journal of the Pattern Recognition Society.

2. A research area in which patterns in data are found, recognized, discovered, …whatever.

3. It includes
   - classification
   - clustering
   - data mining
   - ….

Two Schools of Thought

1. Statistical Pattern Recognition

   The data is reduced to vectors of numbers and statistical techniques are used for the tasks to be performed.

2. Structural Pattern Recognition

   The data is converted to a discrete structure (such as a grammar or a graph) and the techniques are related to computer science subjects (such as parsing and graph matching).

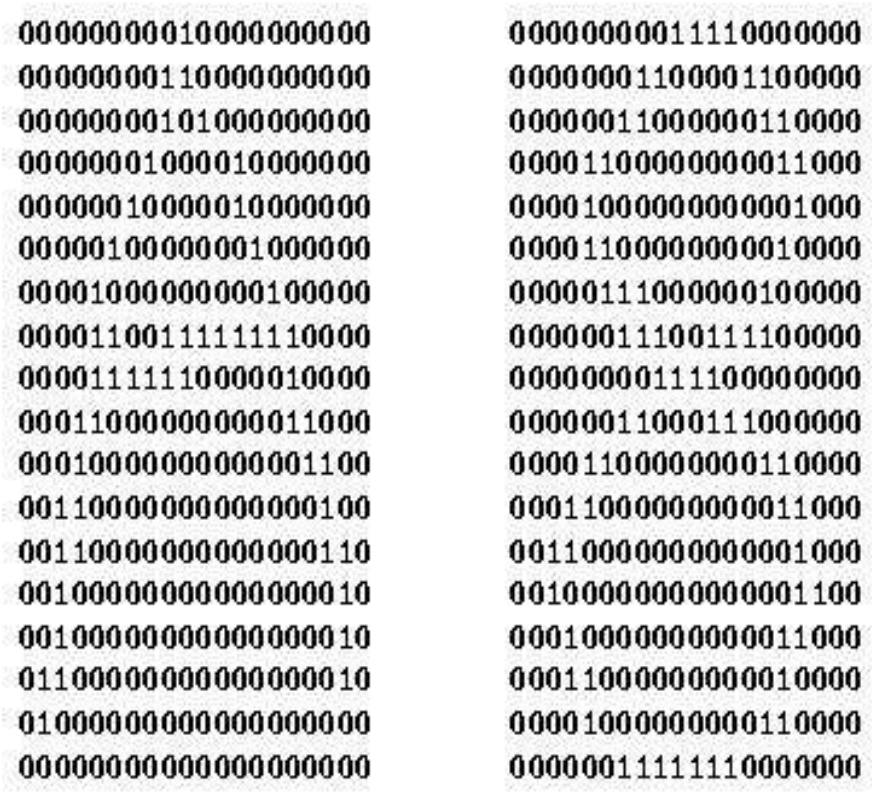
Classification in Statistical PR

- A class is a set of objects having some important properties in common
- A feature extractor is a program that inputs the data (image) and extracts features that can be used in classification.
- A classifier is a program that inputs the feature vector and assigns it to one of a set of designated classes or to the “reject” class.

With what kinds of classes do you work?

Feature Vector Representation

- X=[x1, x2, … , xn], each xj a real number
- xj may be an object measurement
- xj may be count of object parts
- Example: object rep. [#holes, #strokes, moments, …]

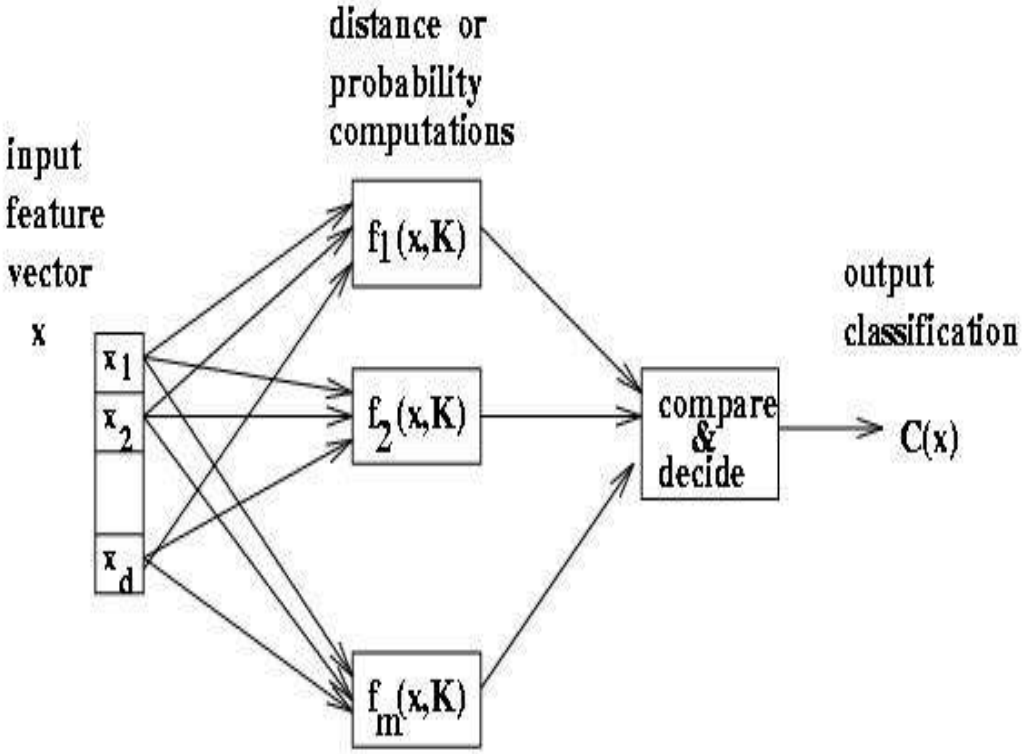


Some Terminology

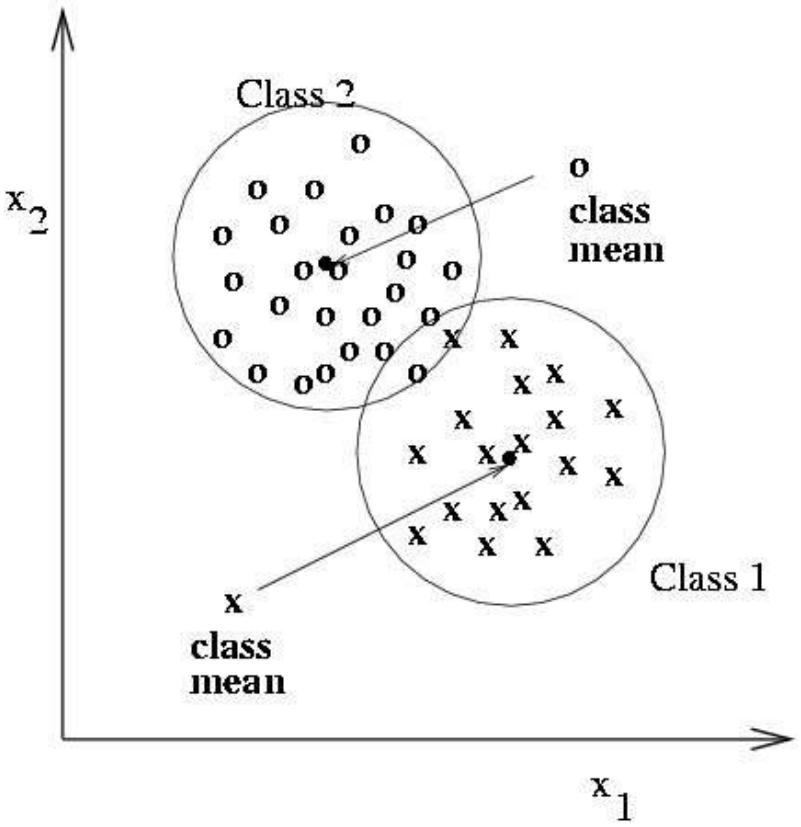
- Classes: set of m known categories of objects
  - (a) might have a known description for each
  - (b) might have a set of samples for each
- Reject Class:
  - a generic class for objects not in any of the designated known classes
- Classifier:
  - Assigns object to a class based on features

Discriminant functions

- Functions f(x, K) perform some computation on feature vector x
- Knowledge K from training or programming is used
- Final stage determines class



Classification using nearest class mean



- Compute the Euclidean distance between feature vector X and the mean of each class.
- Choose closest class, if close enough (reject otherwise)

Nearest Neighbor Classification

- Keep all the training samples in some efficient look-up structure.
- Find the nearest neighbor of the feature vector to be classified and assign the class of the neighbor.
- Can be extended to K nearest neighbors.

Bayesian decision-making

- classify into class ω_i that is most likely based on observations $\mathbf{X}$
- In order to compute the likelihoods given the measurement $\mathbf{X}$, the following distributions are needed.

$$\textbf{class conditional distribution}: \quad p(x|\omega_i) \; for \; each \; class \quad omega_i \tag{1}$$

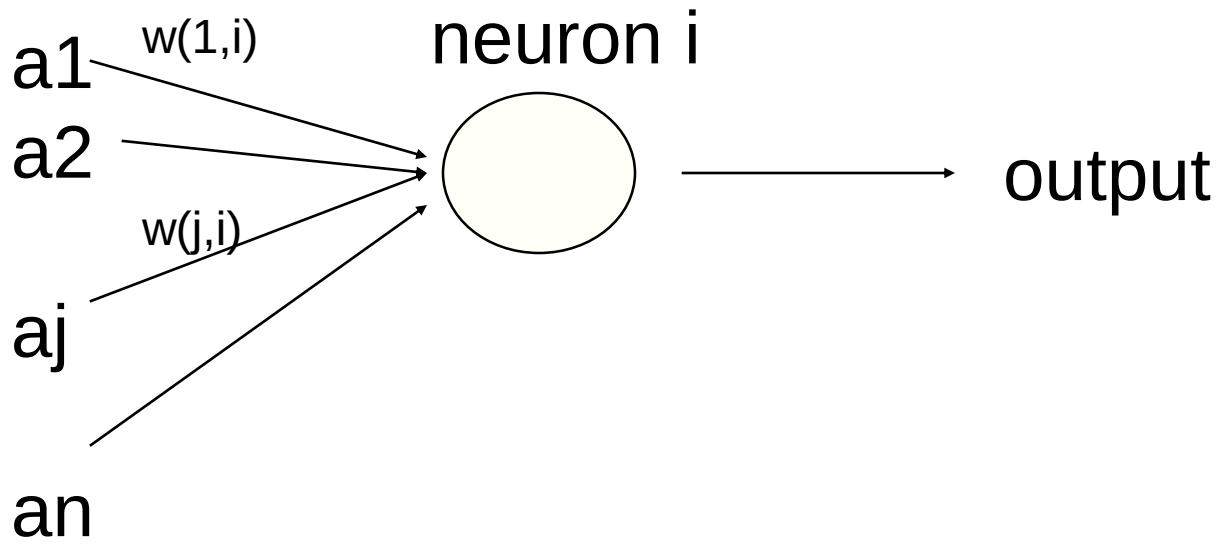
$$\textbf{a priori probability}: \quad P(\omega_i) \; for \; each \; class \; \omega_i \tag{2}$$

$$\textbf{unconditional distribution}: \quad p(x) \tag{3}$$

- use Bayes rule if all of the classes ω_i are disjoint

$$P(\omega_i|x) = \frac{p(x|\omega_i)P(\omega_i)}{p(x)} = \frac{p(x|\omega_i)P(\omega_i)}{\sum_{i=1,m} p(x|\omega_i)P(\omega_i)} \tag{4}$$

NN Functions



output = g ($\sum$ aj * w(j,i))

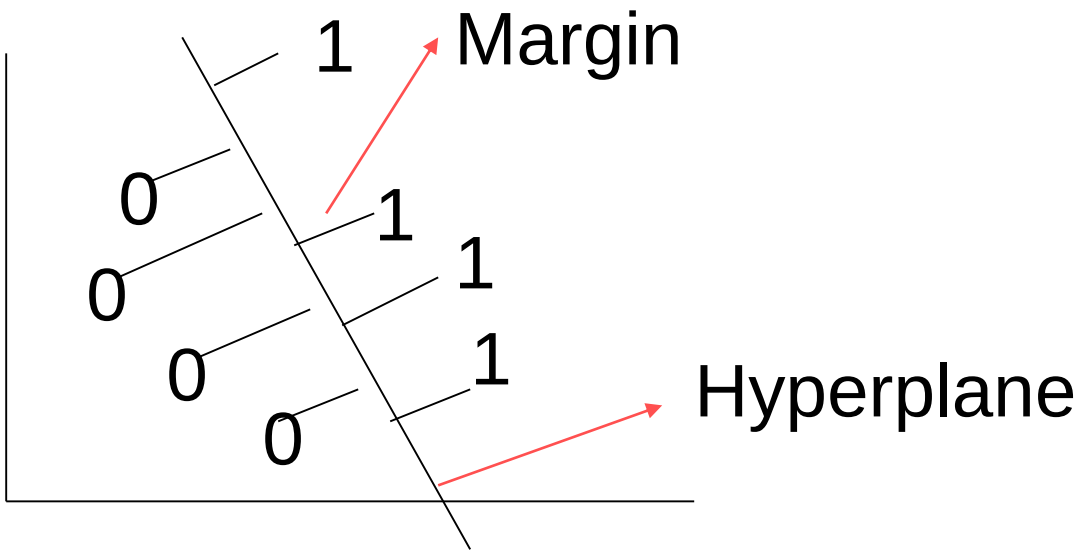
Function g is commonly a step function, sign function, or sigmoid function (see text).

Support Vector Machines (SVM)

Support vector machines are learning algorithms that try to find a hyperplane that separates the differently classified data the most.
They are based on two key ideas:

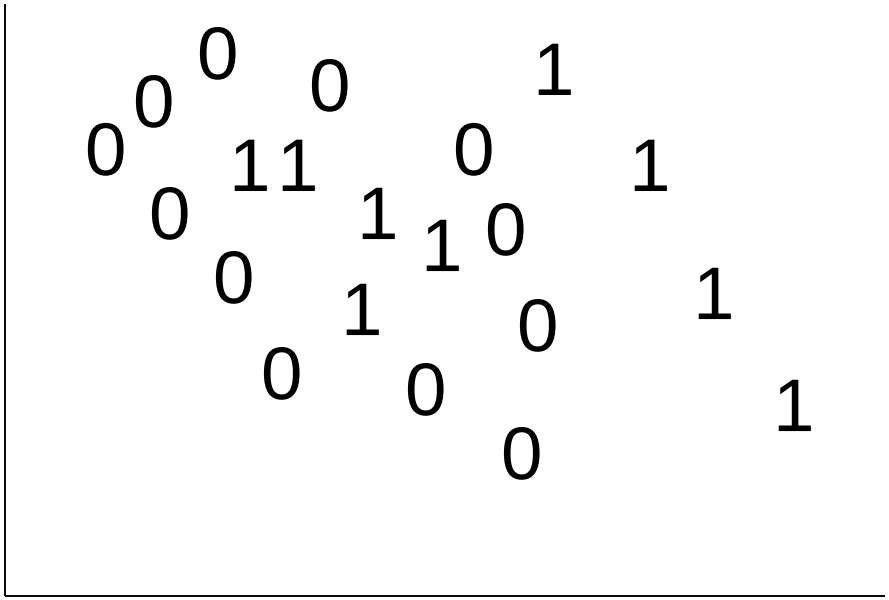
- Maximum margin hyperplanes
- A kernel ‘trick’.

Maximal Margin



Find the hyperplane with maximal margin for all the points. This originates an optimization problem Which has a unique solution (convex problem).

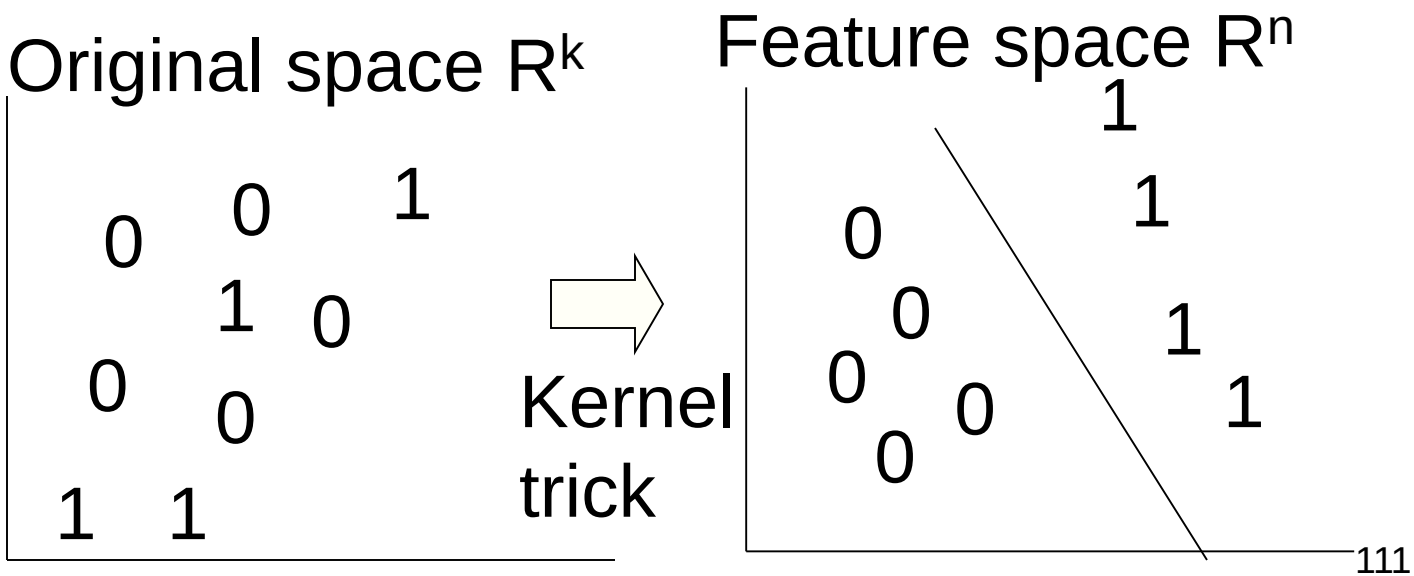
Non-separable data



What can be done if data cannot be separated with a hyperplane?

The kernel trick

The SVM algorithm implicitly maps the original data to a feature space of possibly infinite dimension in which data (which is not separable in the original space) becomes separable in the feature space.



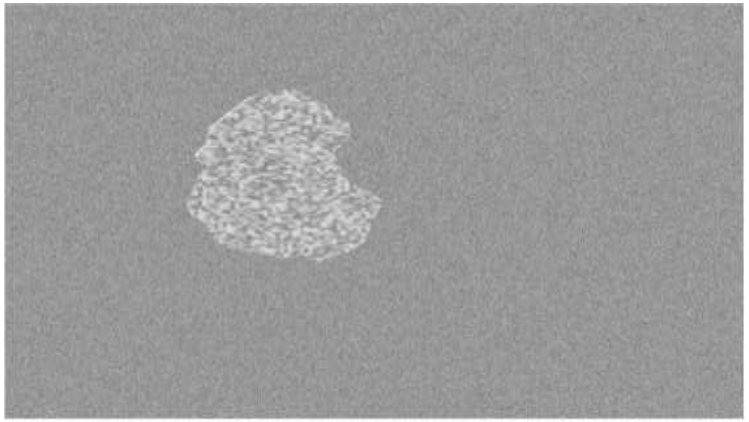
A spectrum of machine learning tasks

Statistics----------Artificial Intelligence

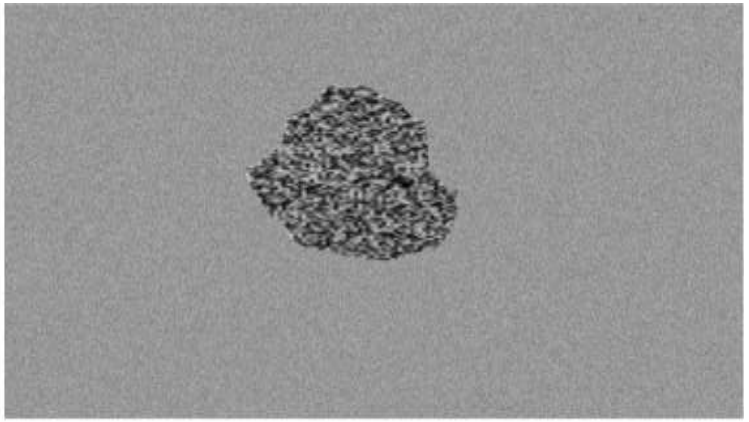
- Low-dimensional data (e.g. less than 100 dimensions)
- Lots of noise in the data
- There is not much structure in the data, and what structure there is, can be represented by a fairly simple model.
- The main problem is distinguishing true structure from noise.

- High-dimensional data (e.g. more than 100 dimensions)
- The noise is not sufficient to obscure the structure in the data if we process it right.
- There is a huge amount of structure in the data, but the structure is too complicated to be represented by a simple model.
- The main problem is figuring out a way to represent the complicated structure that allows it to be learned.

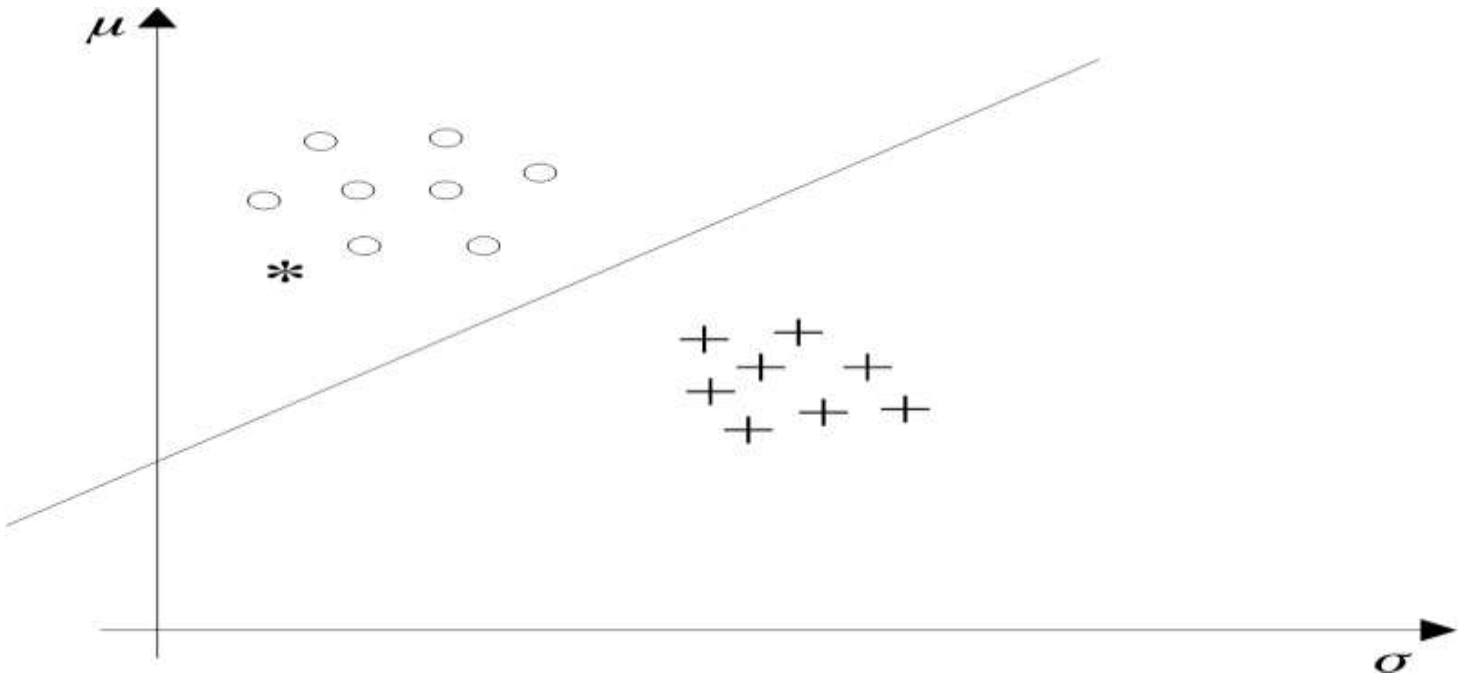
An example:



(a)

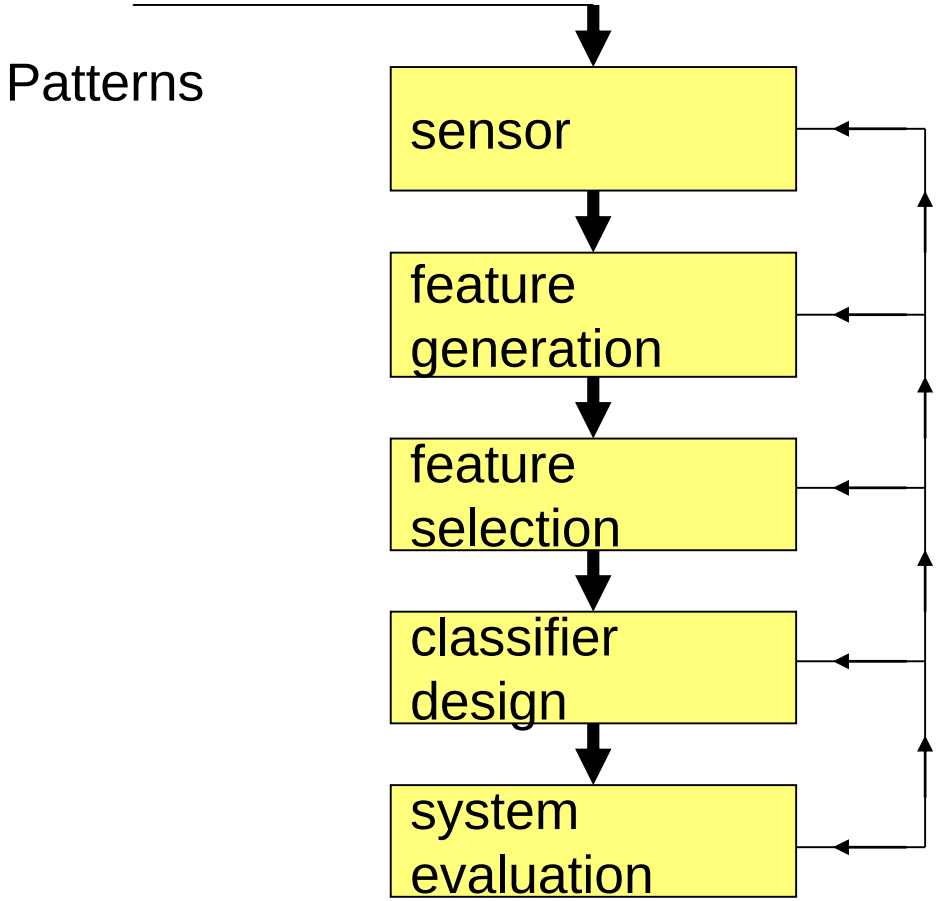


(b)



- The classifier consists of a set of functions, whose values, computed at $\underline{x}$, determine the class to which the corresponding pattern belongs

- Classification system overview



CLASSIFIERS BASED ON BAYES DECISION THEORY

- Statistical nature of feature vectors

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

- Assign the pattern represented by feature vector $\underline{x}$ to the most probable of the available classes

$$\omega_1, \omega_2, \ldots, \omega_M$$

That is $\underline{x} \rightarrow \omega_i : P(\omega_i | \underline{x})$ maximum

The Bayesian framework

- The Bayesian framework assumes that we always have a prior distribution for everything.
  - The prior may be very vague.
  - When we see some data, we combine our prior distribution with a likelihood term to get a posterior distribution.
  - The likelihood term takes into account how probable the observed data is given the parameters of the model.
    - It favors parameter settings that make the data likely.
    - It fights the prior
    - With enough data the likelihood terms always win.

Bayes Theorem

joint probability

conditional probability

$$p(D)\,p(W \mid D) = p(D,W) = p(W)\,p(D \mid W)$$

Prior probability of weight vector W

Probability of observed data given W

$$p(W \mid D) = \frac{p(W)\;p(D \mid W)}{p(D)}$$

Posterior probability of weight vector W given training data D

$$\int_W p(W)\,p(D \mid W)$$

THANK YOU