Pattern Classification

PATTERN RECOGNITION

PATTERN RECOGNITION $\Rightarrow$ Pattern + Recognition

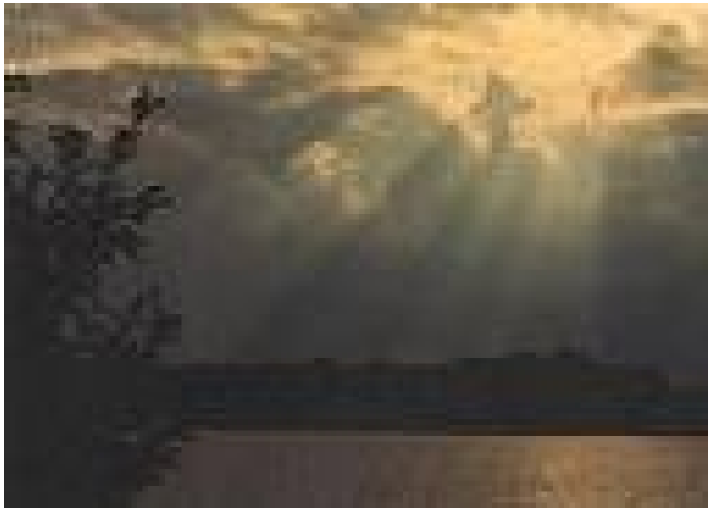
PATTERN : Pattern is a set of objects or phenomena or concepts where the elements of the set are similar to one another in certain ways/aspects. The Pattern are described by certain quantities, qualities, traits, notable features and so on.

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

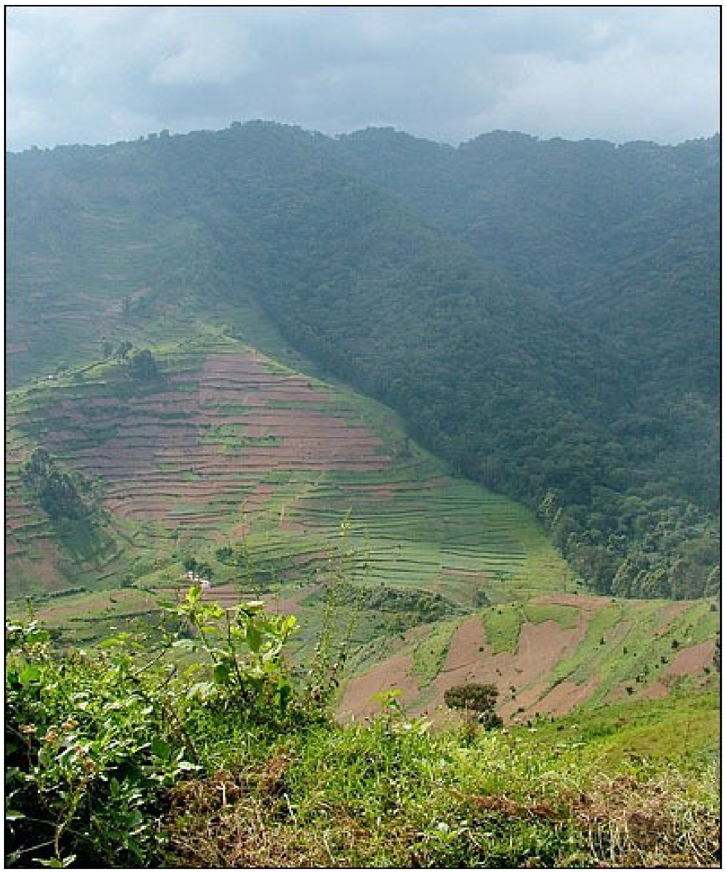
Humans have a pattern which is different from the pattern of animals. Each individuals has a pattern which is different from the patterns of others.

A RECOGNITION PROBLEM: MALE ... or ... FEMALE

Cloud Patterns

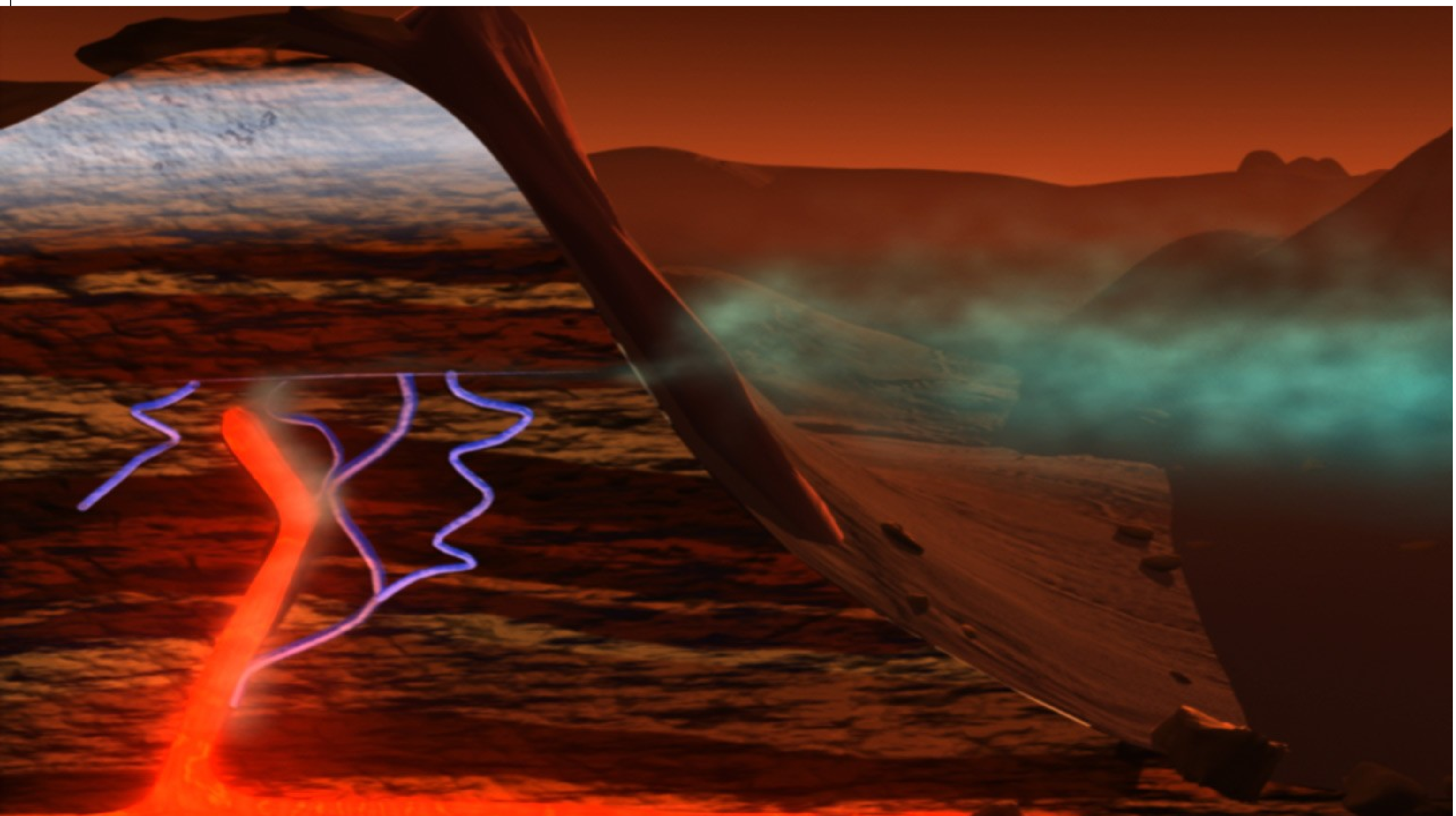


Forest and Cultivated Land



Coal Mine Detection

Natural Gas Detection



PATTERN RECOGNITION AREAS

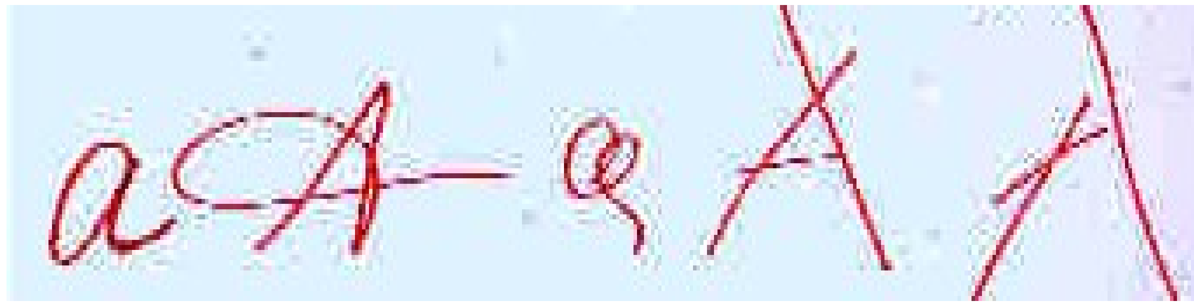
- **Optical Character Recognition (OCR)**
  - Sorting letters by postal code.
  - Reconstructing text from printed materials (such as reading machines for blind people).
- **Analysis and identification of human patterns**
  - Speech and voice recognition.
  - Finger prints and DNA mapping.
- **Banking and insurance applications**
  - Credit cards applicants classified by income, credit worthiness, mortgage amount, # of dependents, etc.
  - Car insurance (pattern including make of car, #of accidents, age, sex, driving habits, location, etc).
- **Diagnosis systems**
  - Medical diagnosis (disease vs. symptoms classification, X-Ray, EKG and tests analysis, etc).
  - Diagnosis of automotive malfunctioning
- **Prediction systems**
  - Weather forecasting (based on satellite data).
  - Analysis of seismic patterns

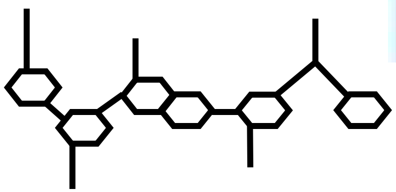
More Pattern Recognition Applications

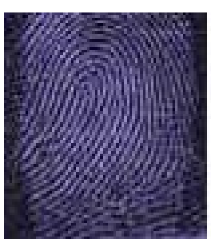
- *SENSORY*
- Vision
  - Face/Handwriting/Hand
- Speech
  - Speaker/Speech

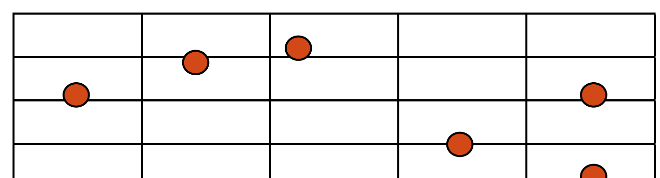
- *DATA*
- Text Categorization
- Information Retrieval
- Data Mining
- Matching

"A pattern is the opposite of a chaos; it is an entity unclearly defined, that could be given a name."









Characters

A v t u I h D U w K

Ç ş ğ İ ü Ü Ö Ğ

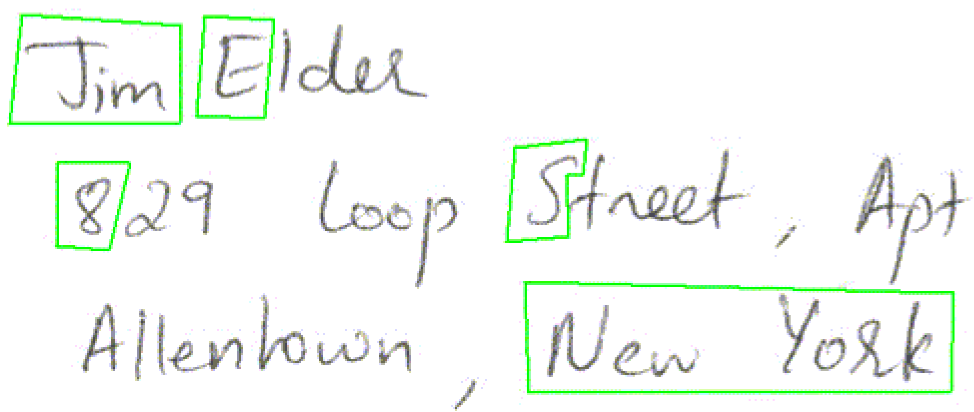
ع٧٤چك

К Ц Д

ζ ω Ψ Ω ξ θ

א ם ש ת ד נ

Handwriting



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.
- When we see a human being, we perceive a member of the same class of pattern. New class of patterns emerge when perhaps Martians or Extra-terrestrial beings come to earth.
- 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.

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.

MOTIVATION FOR THE STUDY OF PATTERN RECOGNITION

It is threefold.

- It is an essential part of the broader field of Artificial Intelligence, which is concerned with techniques, that enable computers to do things, that seem intelligent when done by people.

- It is an important aspect of applying computers to solve problems in science and engineering, since many of them involve analysis and classification of measurements, taken from physical processes.

- Pattern Recognition techniques provide a unified frame work to study a variety of techniques, in mathematics and computer science, that are individually useful in many different applications.

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.

Examples: Astrology/Palm history:

According to this methodology, it consists of the following

1. We observe the different planets/lines on hand.
2. We study the relationship between the planets/lines.
3. We study the relations between the position of planets/lines and situations in life and arrive at events.
4. We study the events and understand the law behind the events.
5. Using the law we can predict the future of a person.

TYPE OF PATTERNS

1.SPATIAL PATTERNS- These patterns are located in space.

Eg:- characters in character recognition

* images of ground covers in remote sensing
* images of medical diagnosis.

2.TEMPORAL PATTERN-These are distributed in time.

Eg:- Radar signal, speech recognition, sonar signal etc.

3.ABSTRACT PATTERNS-Here the patterns are distributed neither in space nor time.

Eg:- classification of people based on psychological tests.

* Medical diagnosis based on medical history and other medical tests.
* Classification of people based on language they speak.

APPROACH TO PATTERN RECOGNITION

1. Statistical or decision theoretic or discriminant method.
2. Syntactic or Grammatical or structural approach.

STATISTICAL APPROACH

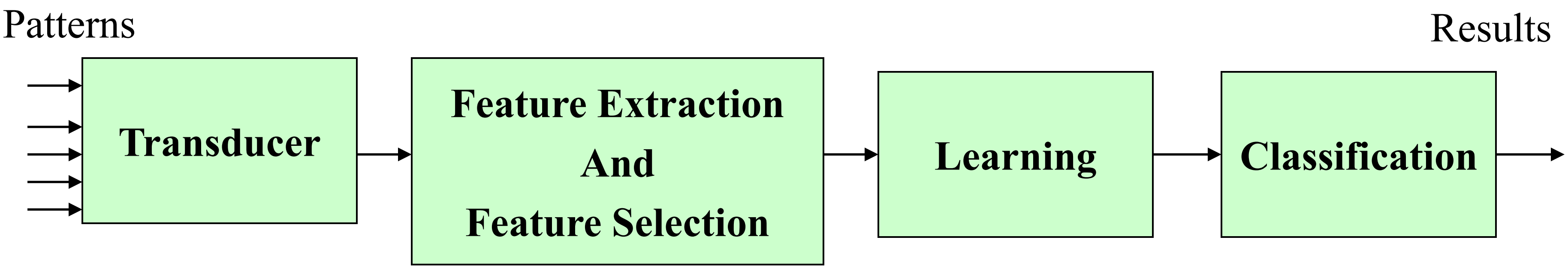


Fig1.1: Block diagram representation of statistical approach

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

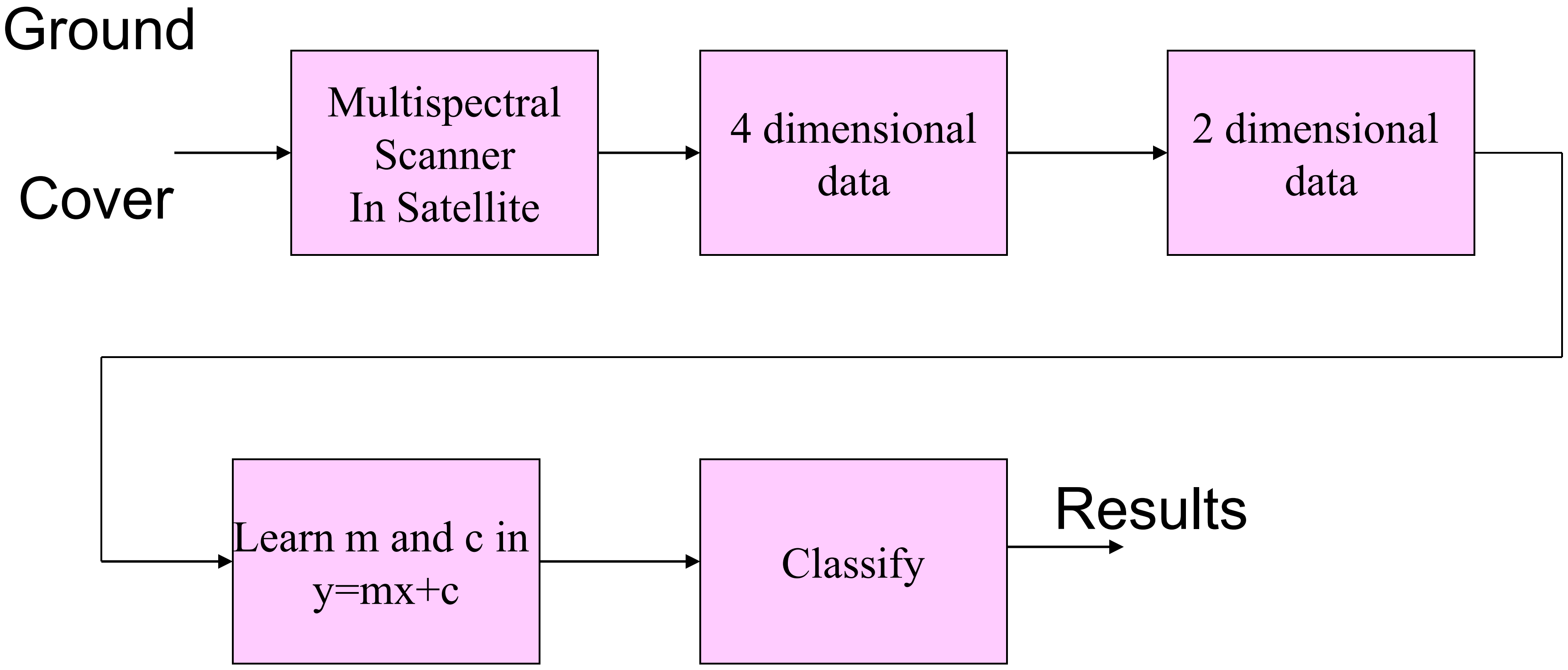


Fig 1.2: A Classification Method

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,

it is clear that reliable, accurate pattern recognition by machine would be immensely useful.

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

AN EXAMPLE

To illustrate the complexity of some of the types of problems involved,

let us consider the following imaginary and somewhat fanciful example.

Suppose that it fish packing plant wants to automate the process of sorting incoming fish on a conveyor belt according to species.

As a pilot project it is decided to try to separate sea bass from salmon using optical sensing.

We set up a camera, take some sample images, and begin to note some physical differences between the two types of fish-length, lightness, width, number and shape of fins, position of the mouth, and so on-and these suggest features to explore for use in our classifier. We also notice noise or variations in the images-variations in lighting, position of the fish on the conveyor, even "static" due to the electronics of the camera itself.

Model

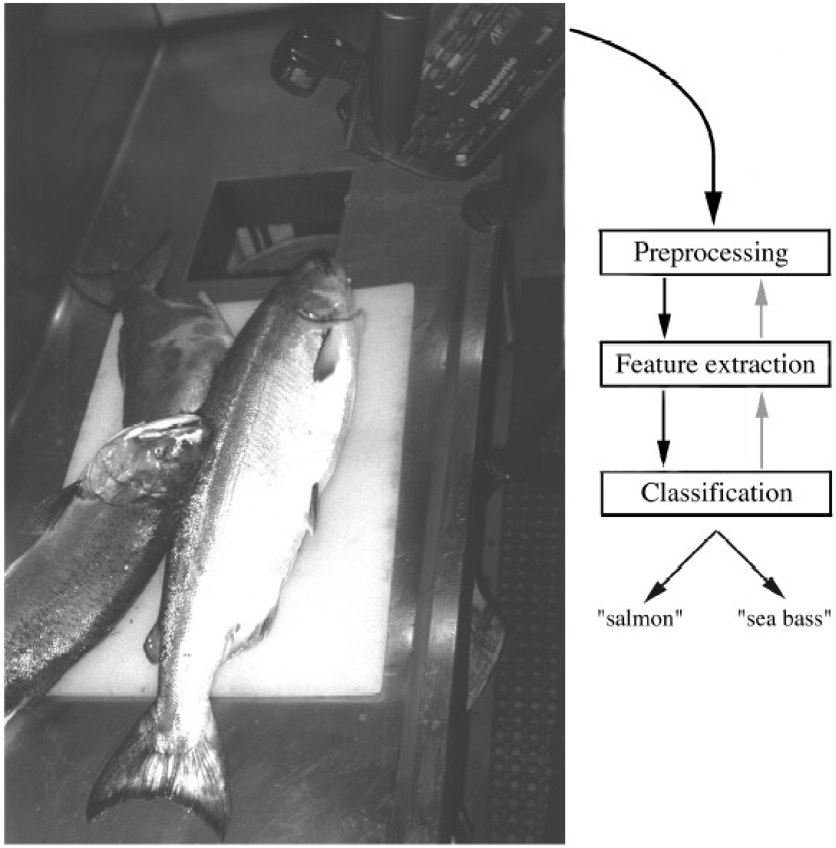
Given that there are differences between the population of sea bass and that of salmon,

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.

A Fishy Example I

- "Sorting incoming Fish on a conveyor according to species using optical sensing"
- Salmon or Sea Bass?



- Problem Analysis
  - Set up a camera and take some sample images to extract features
    - Length
    - Lightness
    - Width
    - Number and shape of fins
    - Position of the mouth, etc…

This is the set of all suggested features to explore for use in our classifier!

Solution by Stages

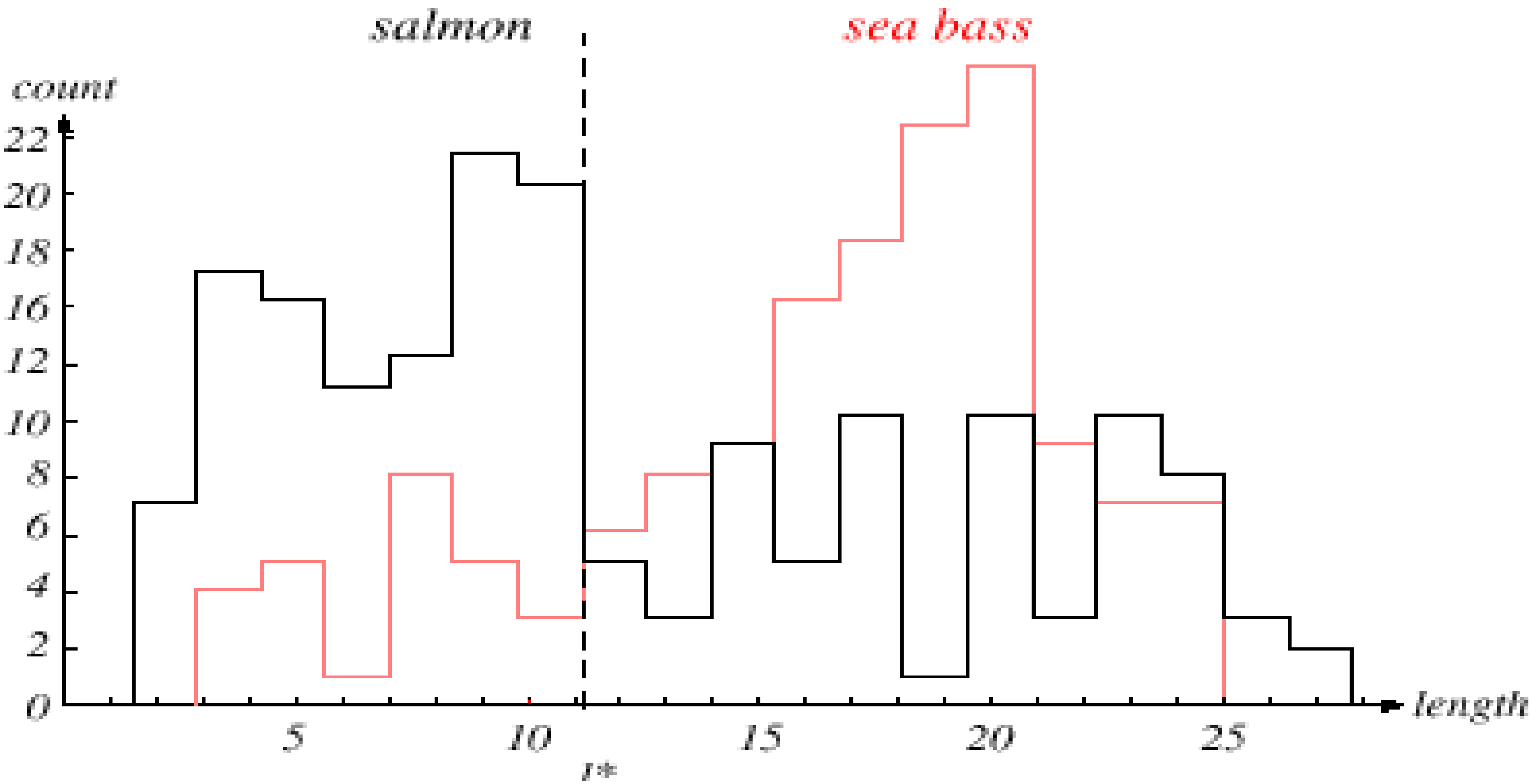
- Preprocess raw data from camera
- Segment isolated fish
- Extract features from each fish (length,width, brightness, etc.)
- Classify each fish

- Preprocessing
  - Use a segmentation operation to isolate fishes from one another and from the background
- Information from a single fish is sent to a feature extractor whose purpose is to reduce the data by measuring certain features
- The features are passed to a classifier


Preprocessing
Feature extraction
Classification
"salmon"
"sea bass"

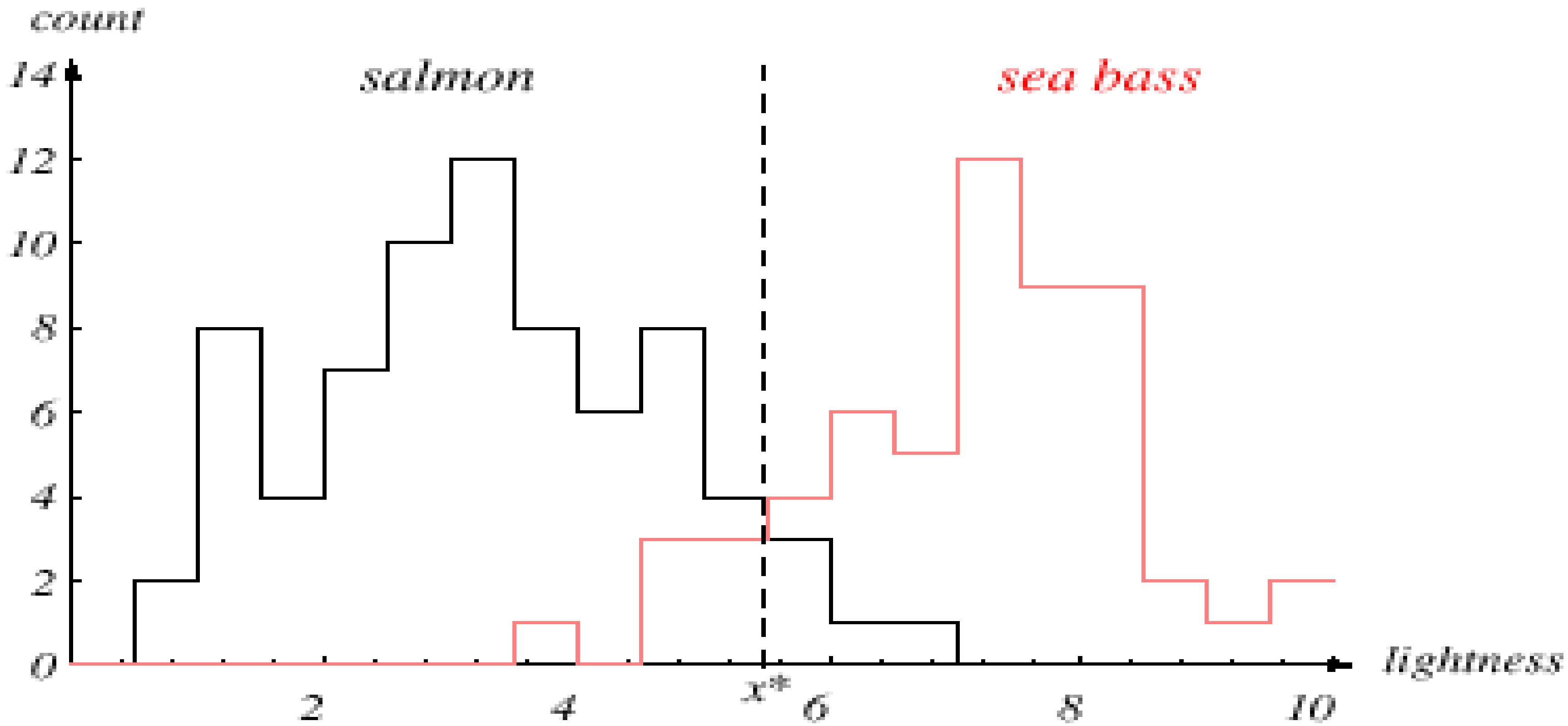
- Classification

Select the length of the fish as a possible feature for discrimination


salmon
sea bass
count
22
20
18
16
12
10
8
6
4
2
0
5
10
15
20
25
l*
length

The length is a poor feature alone!

Select the lightness as a possible feature.



count
salmon
sea bass
14
12
10
8
6
4
2
0
2
4
x*
6
8
10
lightness

“Customers do not want sea bass in their cans of salmon”

- Threshold decision boundary and cost relationship
- Move our decision boundary toward smaller values of lightness in order to minimize the cost (reduce the number of sea bass that are classified salmon!)

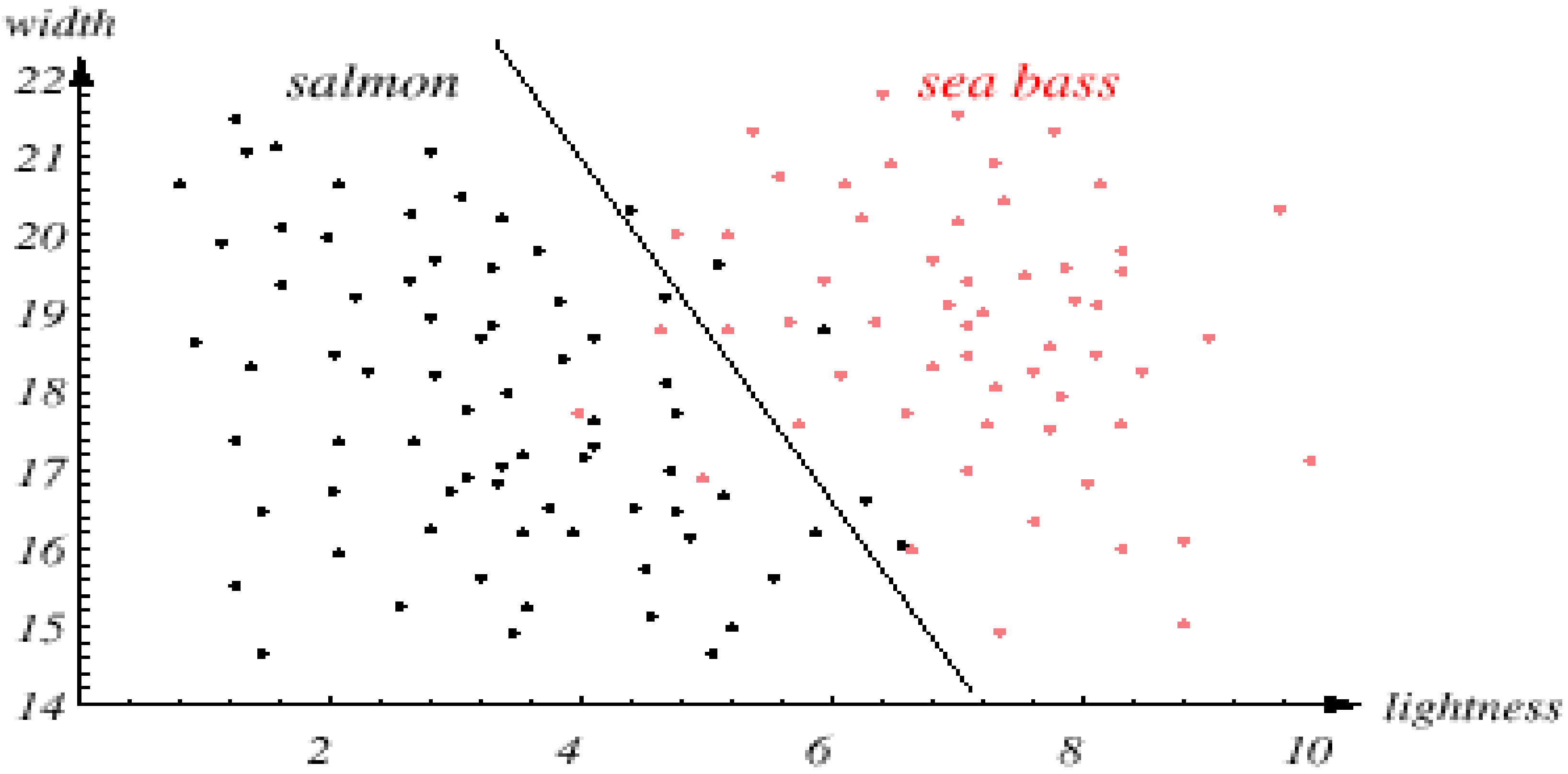
Task of decision theory

- Adopt the lightness and add the width of the fish

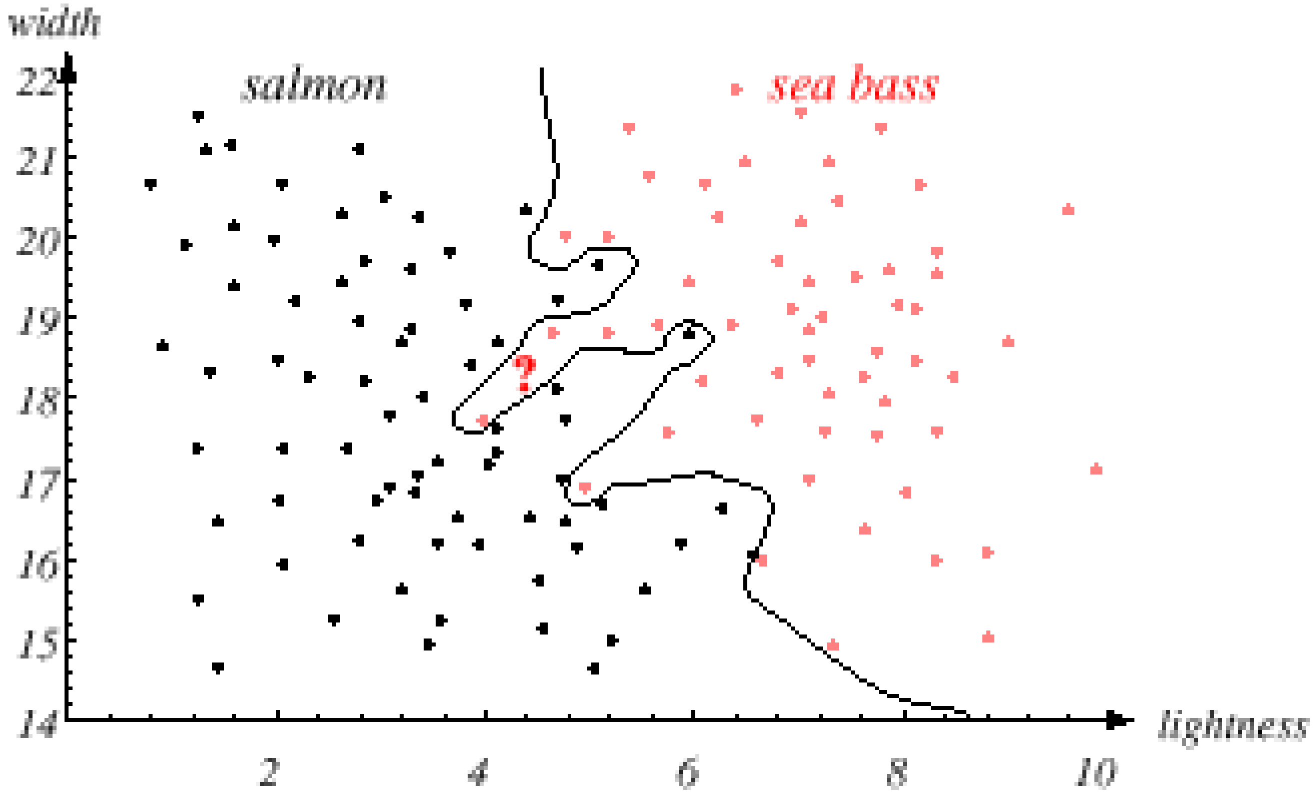
Fish $\longrightarrow$ $x = [x_1, x_2]$

Lightness

Width

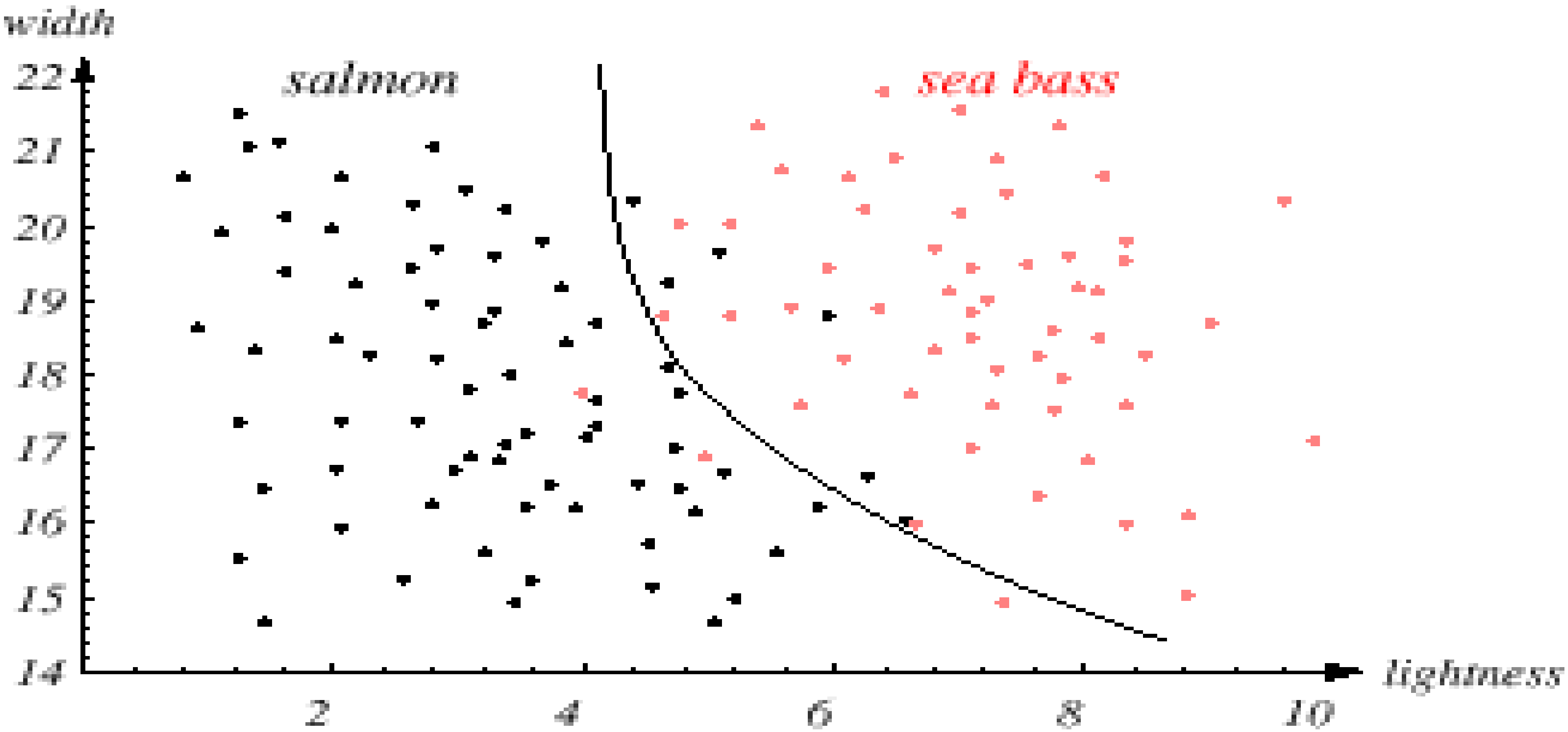

width
salmon
sea bass
22
21
20
19
18
17
16
15
14
2
4
6
8
10
lightness

- We might add other features that are not correlated with the ones we already have. A precaution should be taken not to reduce the performance by adding such “noisy features”
- Ideally, the best decision boundary should be the one which provides an optimal performance such as in the following figure:


width
salmon
sea bass
?
22
21
20
19
18
17
16
15
14
2
4
6
8
10
lightness

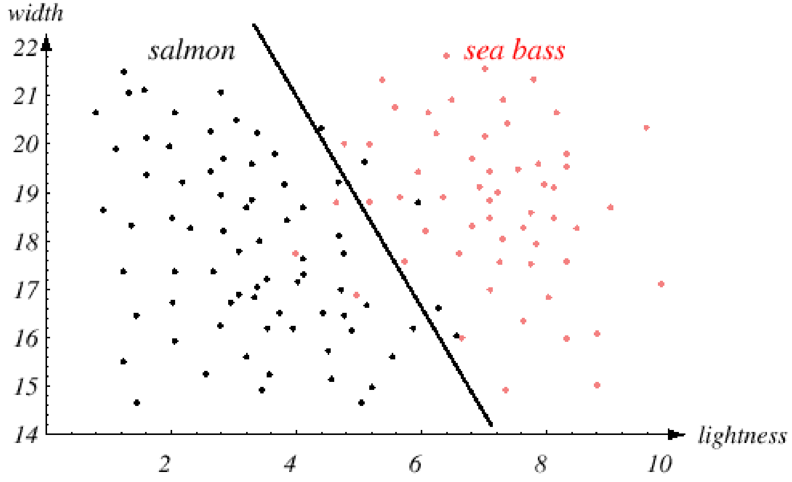
- However, our satisfaction is premature because the central aim of designing a classifier is to correctly classify novel input

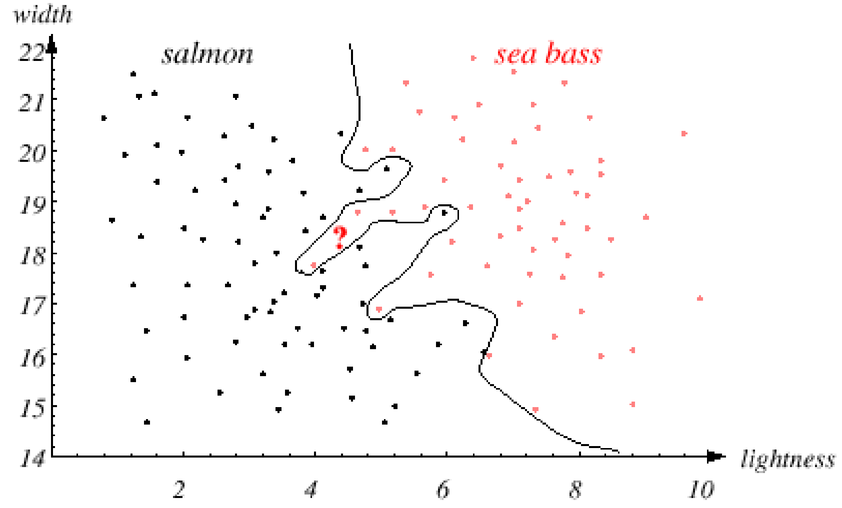
Issue of generalization!


width
22
21
20
19
18
17
16
15
14
salmon
sea bass
2
4
6
8
10
lightness

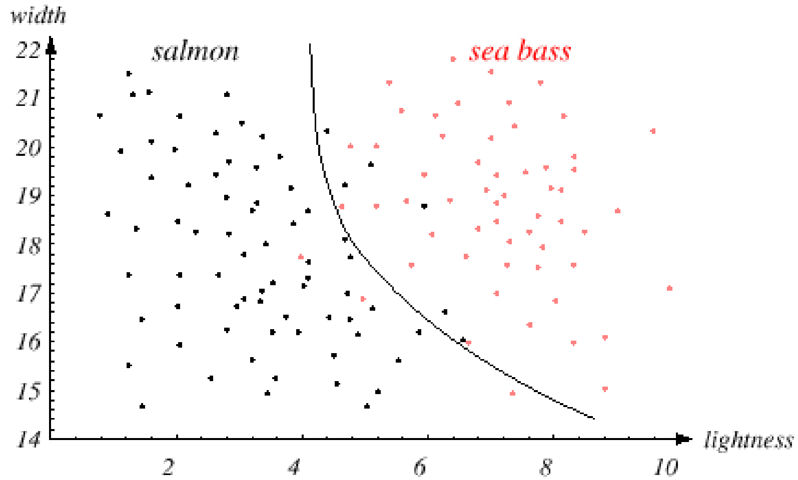
Decision Boundaries

Observe: Can do much better with two features





Caveat: overfitting!



Classification Process

A Simple system to perform classification might have the following form as shown in the figure 1.3.

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

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.

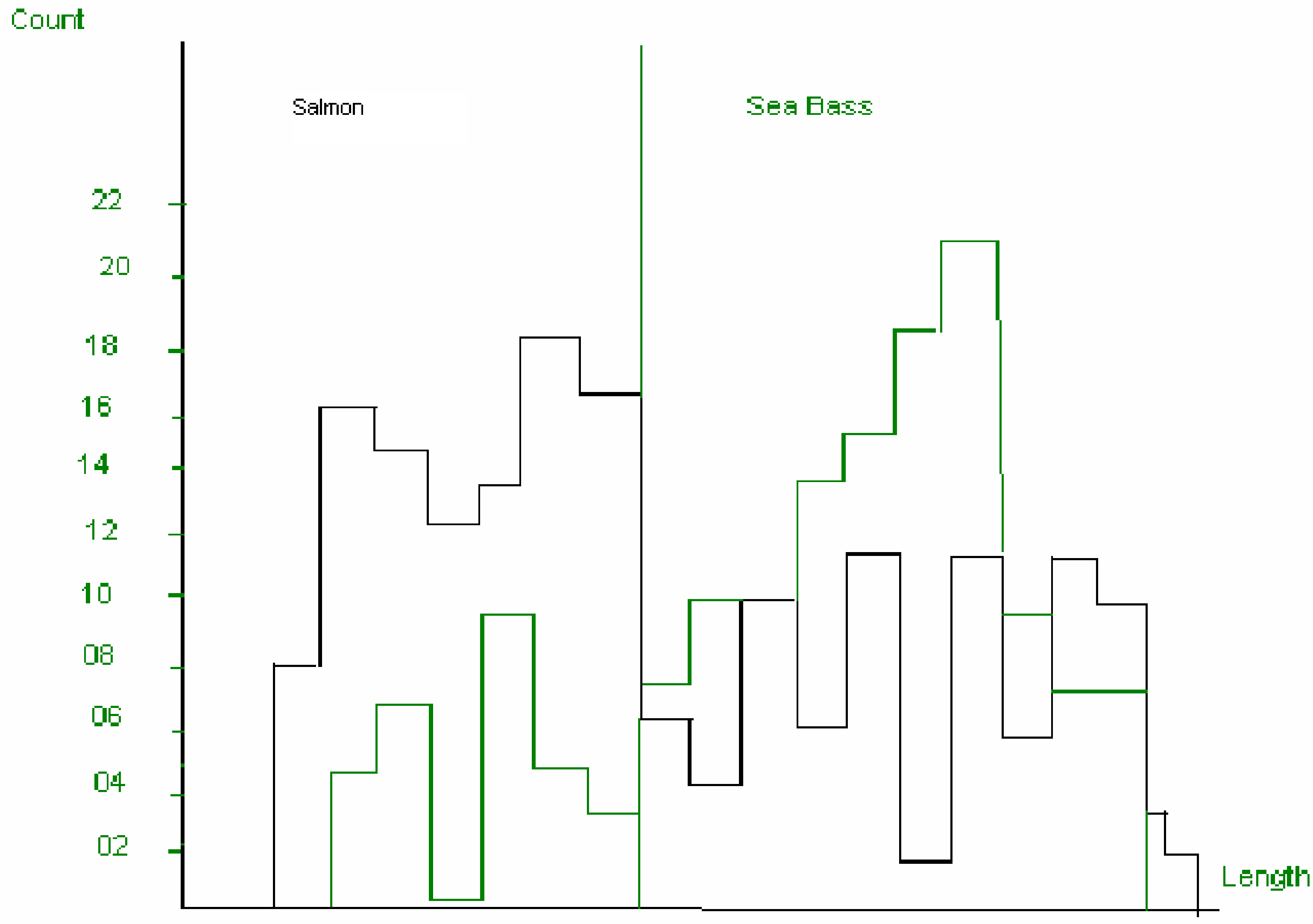


Fig 1.4: Histograms for the length feature for Salmon and Sea Bass Fish

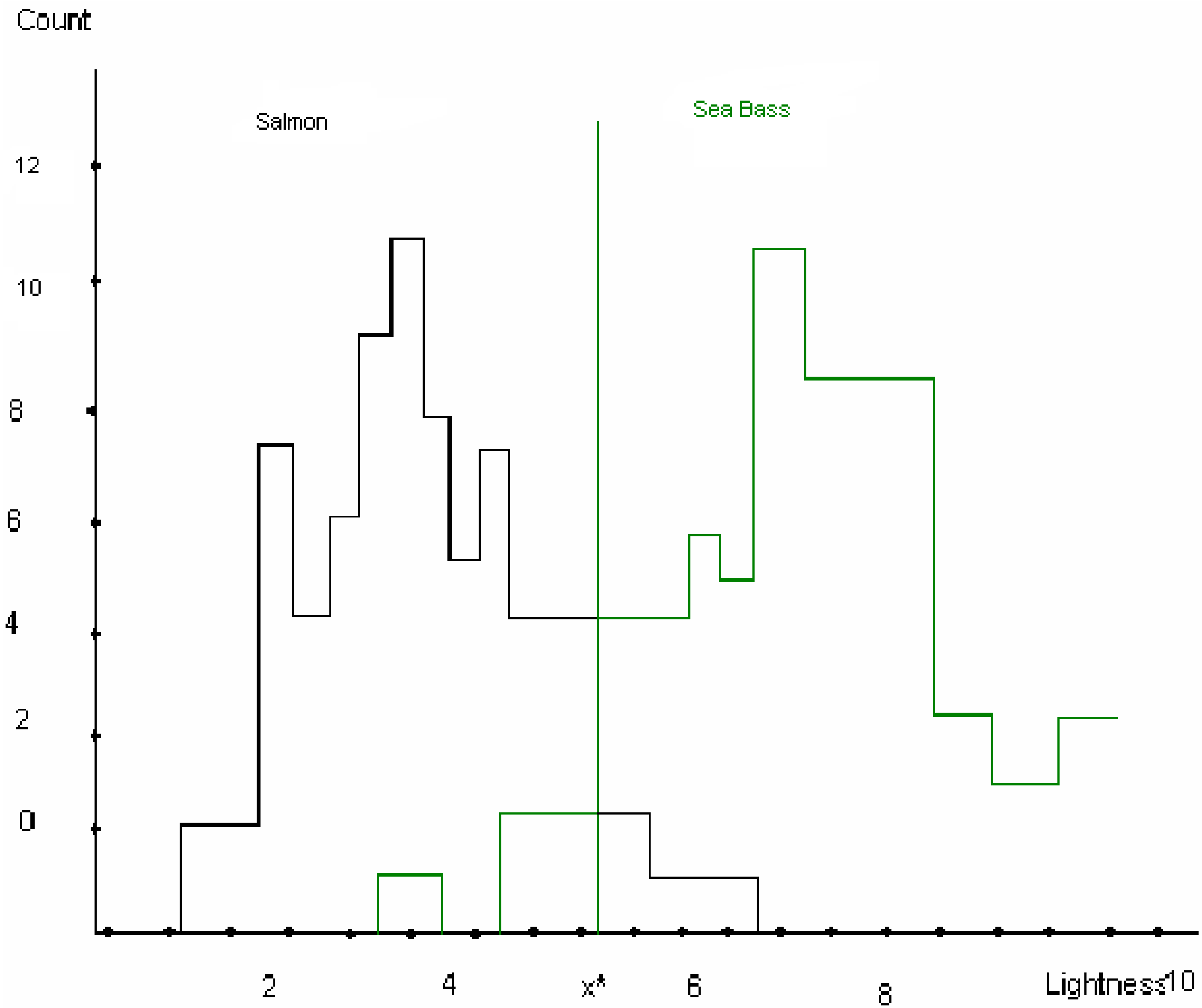


Fig 1.5 Histograms for the lightness feature of Salmon and Sea bass fish

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,

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

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.

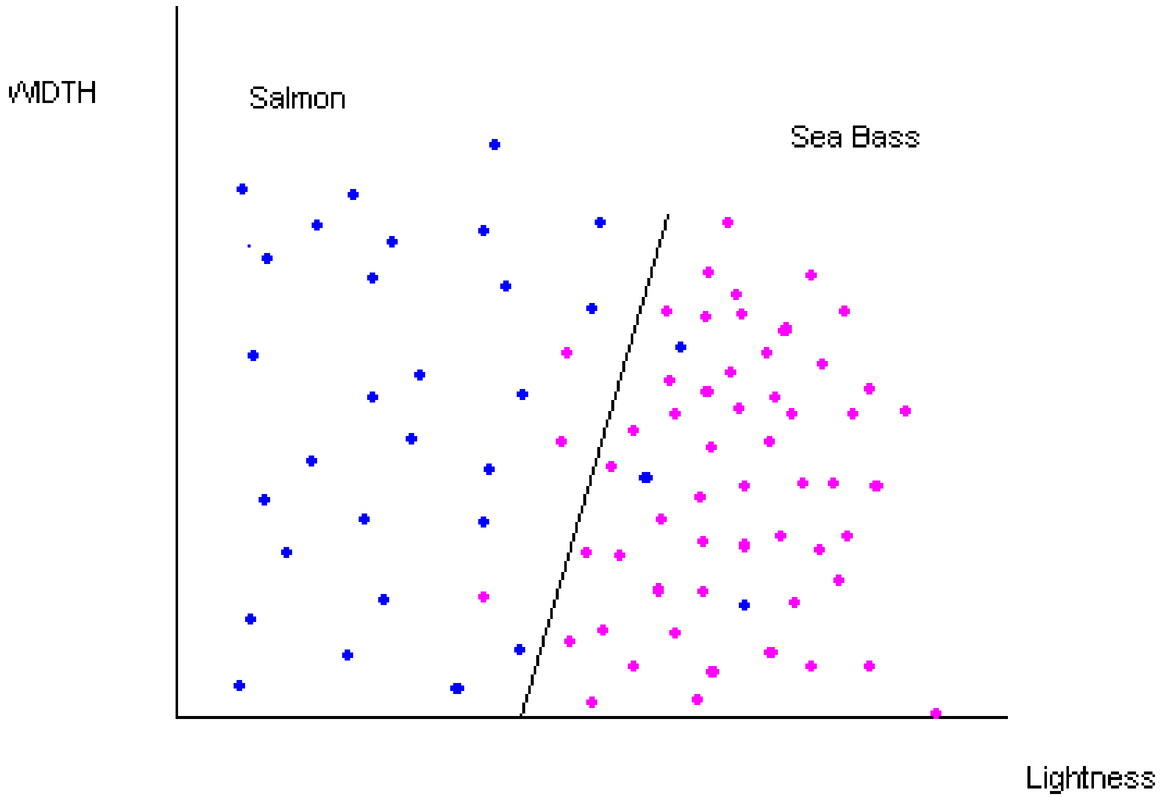


Fig 1.6: Linear Decision Boundary

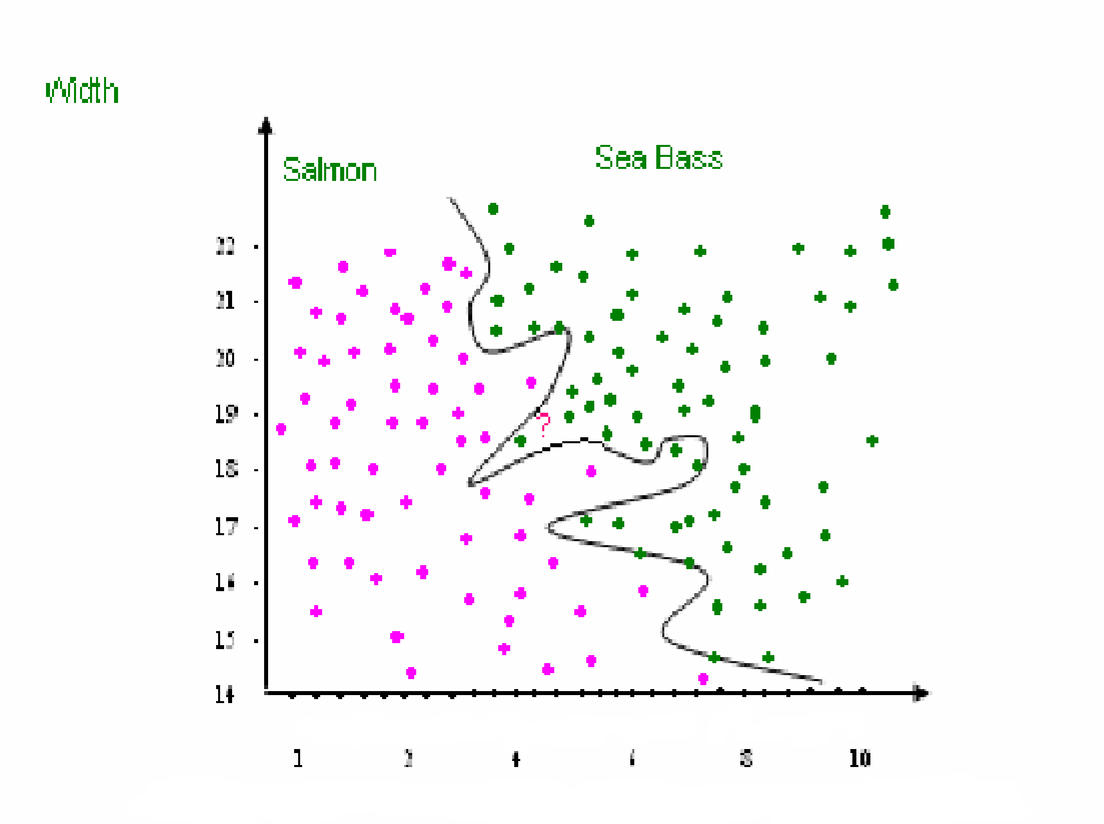


Fig 1.7: Non-Linear Decision Boundary for perfect Classification

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.

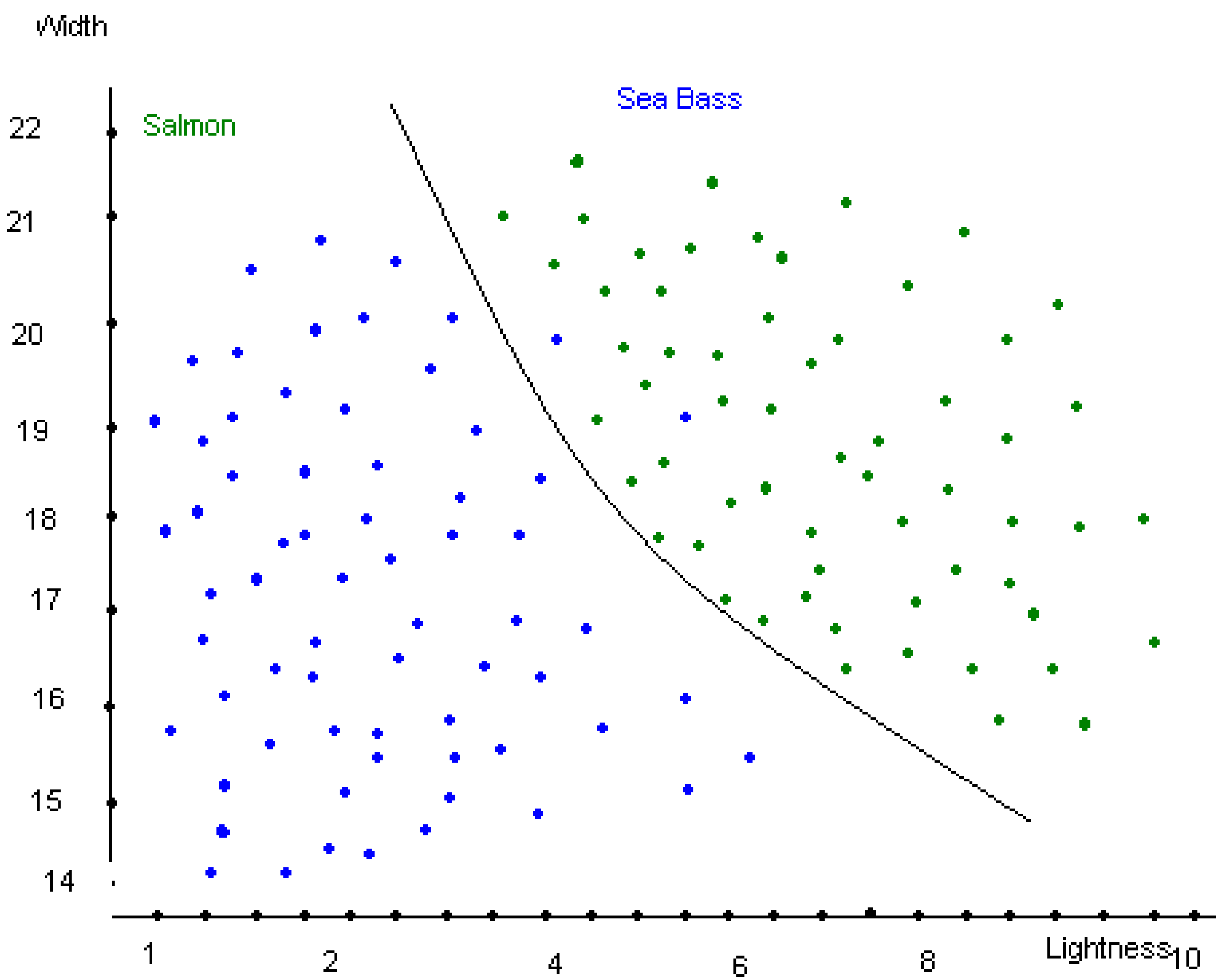


Fig 1.8: The decision boundary shown might represent the optimal tradeoff Between performance on the training set and simplicity of classifier

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

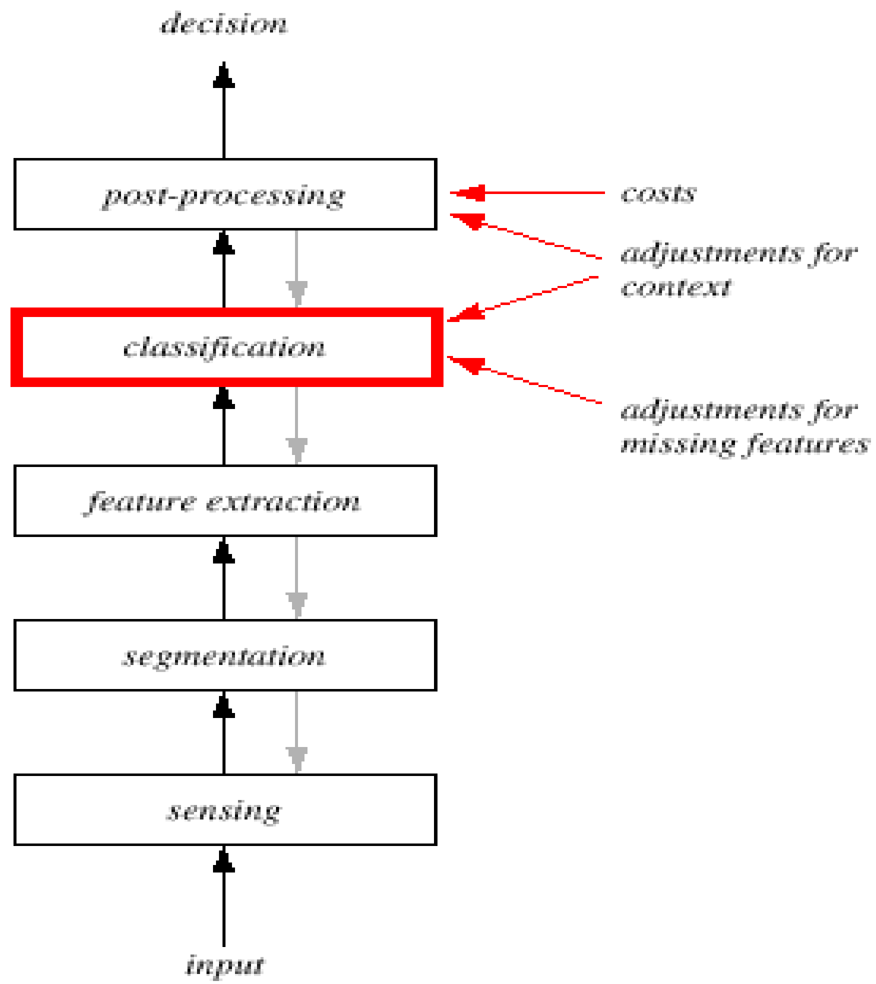


Fig 1.9 Components of a typical Pattern Recognition System

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

Rate

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

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

LEARNING AND ADAPTATION

Supervised learning

In supervised learning, a teacher provides a category label or cost for each pattern in a training set, and seeks to reduce the sum of the costs for these patterns. How can we be sure that a particular learning algorithm is powerful enough to learn the solution to a given problem and that it will be stable to parameter variations? How can we determine if it will converge in finite time or if it will scale reasonably with the number of training patterns, the number of input features or the number of categories? How can we ensure that the learning algorithm appropriately favors " simple" solutions (as in fig 1.8) rather than complicated one (as in fig 1.7)?

LEARNING AND ADAPTATION

Unsupervised learning

In unsupervised learning or clustering there is no explicit teacher, and the systems form clusters or "natural groupings" of the input patterns. "Natural" is always defined explicitly or implicitly in the clustering system itself; and given a particular set of patterns or cost function, different clustering algorithms lead to different clusters. Often the user will set the hypothesized number of different clusters ahead of time, but how should this be done? How do we avoid inappropriate representations?

LEARNING AND ADAPTATION

Reinforcement learning

The most typical way to train a classifier is to present an input, compute its tentative category label, and use the known target category label to improve the classifier. For instance, in optical character recognition, the input might be an image of a character, the actual output of the classifier the category label “R”, and the desired output a “B”. In reinforcement learning or learning with a critic, no desired category signal is given; instead, the only teaching feedback is that the tentative category is right or wrong . This is analogous to a critic who merely states that something is right or wrong, but does not say specifically how it is wrong. In pattern classification, it is most common that such reinforcement is binary—either the tentative decision is correct or it is not. How can the system learn from such non-specific feedback?

CONCLUSION

- Listener seems to be overwhelmed by the number, complexity and magnitude of the sub-problems of Pattern Recognition
- Many of these sub-problems can indeed be solved
- Many fascinating unsolved problems still remain