UNIT 3

MULTIBIOMETRICS

TOPICS

- ***Introduction to multibiometrics***
- ***Sources of multiple evidence***
- ***Acquisition sequence***
- ***Processing sequence***
- ***Fusion level***
- ***Sensor level fusion***
- ***Feature level fusion***
- ***Score level fusion***
- ***Rank level fusion***
- ***Decision level fusion***

TOPICS

- ***Features Matching and Decision Making***
- ***Feature matching: null and alternative hypothesis h0, h1, Error type I/II, Matching score distribution, FM/FNM, ROC curve, DET curve, FAR/FRR curve.***
- ***Introduction to Various matching methods***
- ***LDA***
- ***PCA, Eigen vectors and values, 2D-PCA,***
- ***generalization to p-dim, covariance and correlation, algebra of PCA, projection of data***
- ***Introduction to decision theory and their examples***
- *Explanation – examples*

Multibiometrics

- Biometric systems can also be designed to recognize a person based on information acquired from **multiple biometric sources**.
- Such systems, known as *multibiometric systems*, are expected to be more accurate compared to unibiometric systems that rely on a single piece of biometric evidence

ADVANTAGES

- Availability of multiple biometric sources provides **redundancy** and f**ault-tolerance** in the sense that the recognition system continues to operate even when certain biometric acquisition modules fail.
- Alleviate the non-universality problem and reduce the failure to enroll errors. (For example, if a person cannot be enrolled in a fingerprint system due to worn-out ridge details or missing fingers he can still be identified using his other traits like face or iris.)
- Increase the resistance to **spoof attacks**. This is because it becomes increasingly difficult to circumvent(avoid) multiple biometric sources simultaneously

ADVANTAGES

- Provide a **degree of flexibility** in user authentication. (Suppose a user enrolls into the system using several different traits. Later, at the time of authentication, only a subset of these traits may be acquired based on the nature of the application under consideration and the convenience of the user.)
- Enable the search of a large biometric database in a **computationally efficient** manner.

DISADV

- It is **more expensive** than unibiometric systems due to the need for additional hardware (computational or storage resources) and larger enrollment and recognition times.
- Multibiometric systems are being increasingly deployed in many large-scale identification systems involving millions of users (e.g., border control or national identity systems) because of their ability to achieve **high recognition accuracy** based on existing technologies, which far outweighs the additional cost in such applications

FOUR DESIGN ISSUES

1. Information sources: What are the various sources of biometric information that should be used in a multibiometric system?
2. Mode of operation: Should the data corresponding to multiple biometric sources be acquired simultaneously in a parallel mode or in a sequence? Similarly, should the information acquired be processed sequentially or simultaneously?
3. Level of fusion: What type of information (i.e., raw data, features, match scores, or decisions) is to be fused?
4. Fusion approach: What fusion scheme should be employed to combine the information presented by multiple biometric sources?

Sources of Multiple Evidence

- Based on the sources of evidence, multibiometric systems can be classified into
  - multi-sensor,
  - multi-algorithm,
  - multi-instance,
  - multi-sample, and
  - multimodal systems.

- First four scenarios, multiple pieces of evidences are derived from a single biometric trait (e.g., fingerprint *or* iris), while in the fifth scenario (also called multimodal biometric system) multiple biometrictraits (e.g., fingerprint *and* iris) are used.
- UID(Unique Identification Authority) system in India uses all ten fingers and two irides, it is both multi-instance and multimodal. Such systems are called *hybrid* multibiometric systems.

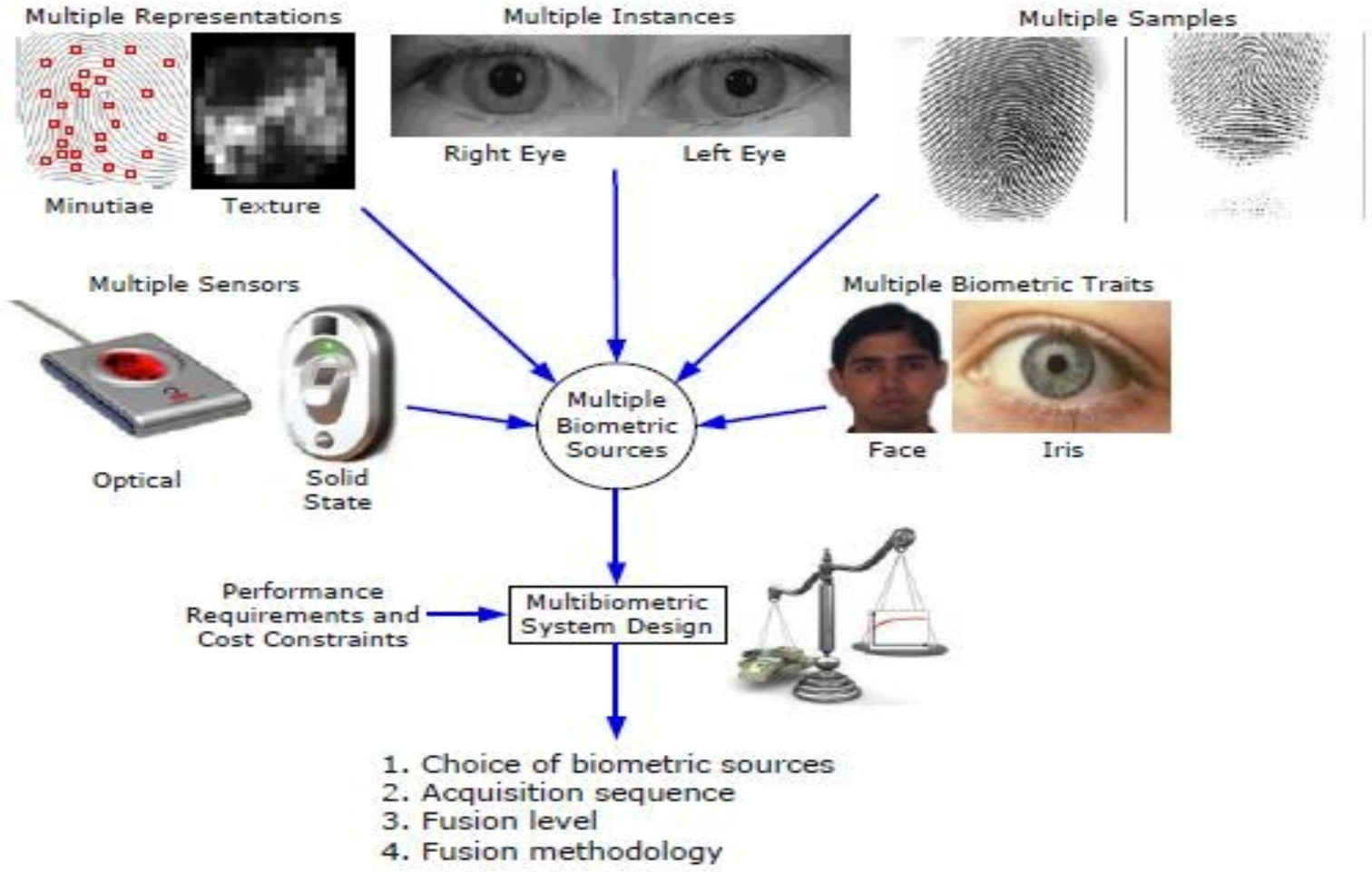
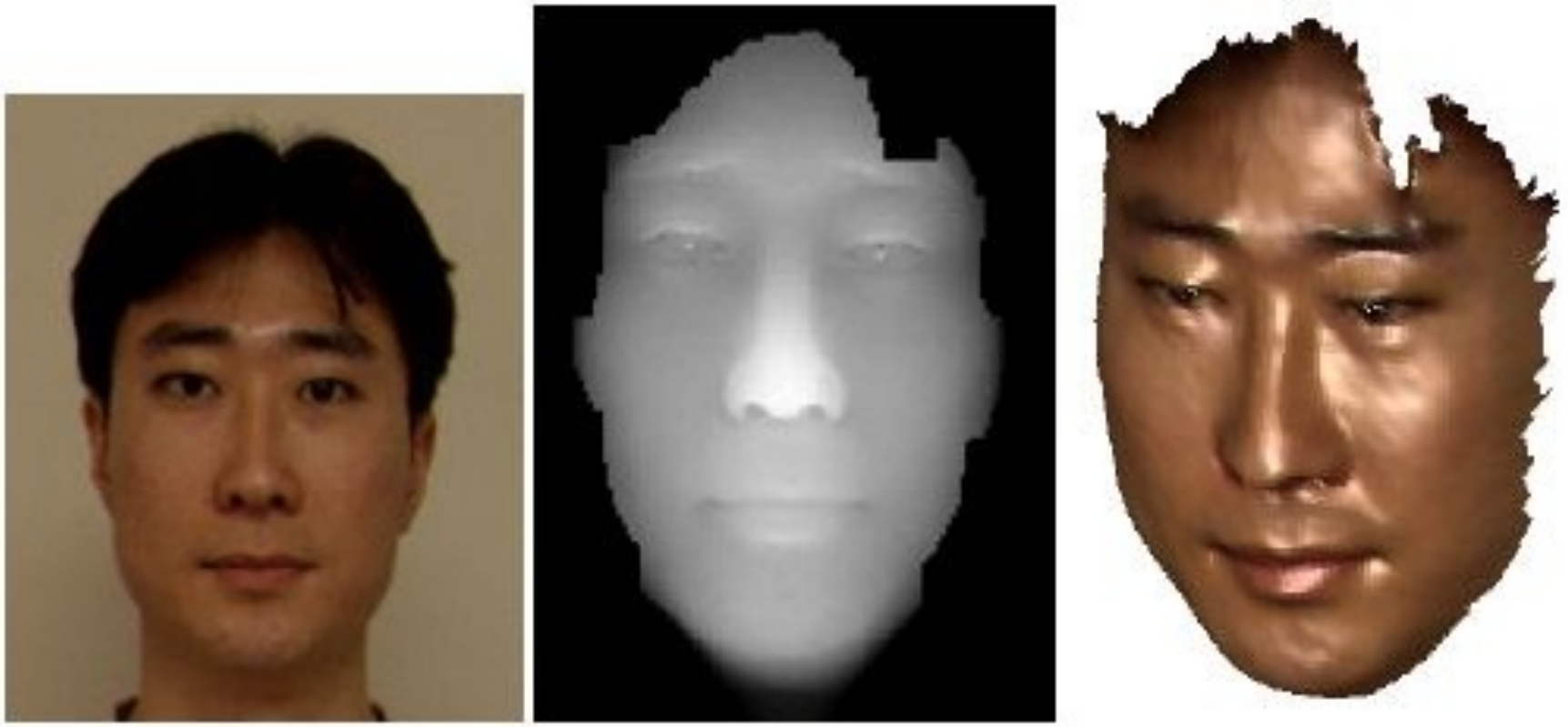


Fig. 6.2 Multibiometric systems utilize information from multiple biometric sources to establish an identity. Based on the information sources used, multibiometric systems can be classified into multi-sensor, multi-algorithm, multi-instance, multi-sample, and multimodal systems. In the first four scenarios, a single biometric trait provides multiple sources of evidence. In the fifth scenario, different biometric traits are used as sources of evidence. While in principle, a large number of sources can be combined to improve the identification accuracy, practical factors such as cost of deployment, small training sample size, accuracy requirements, throughput time, and user acceptance will limit the number of sources used in a particular application.

Multi-sensor systems

- In these systems, a single biometric trait is imaged or captured using **multiple sensors** in order to extract diverse information.
- For example, a system may record the two-dimensional texture content of a person's face using a CCD camera and the three-dimensional surface shape (also called the depth or range image) of the face using a range sensor in order to perform authentication

MULTI SENSOR



(a) (b) (c)

Fig. 6.3 Constructing a 3D face texture by combining the evidence presented by a 2D texture image and a 3D range image. (a) The 2D face texture of a person, (b) the corresponding 3D range (depth) image (here, blue represents farther distance from the camera while red represents a closer distance), and (c) the 3D surface after mapping the 2D texture information from (a).

MULTI SENSOR

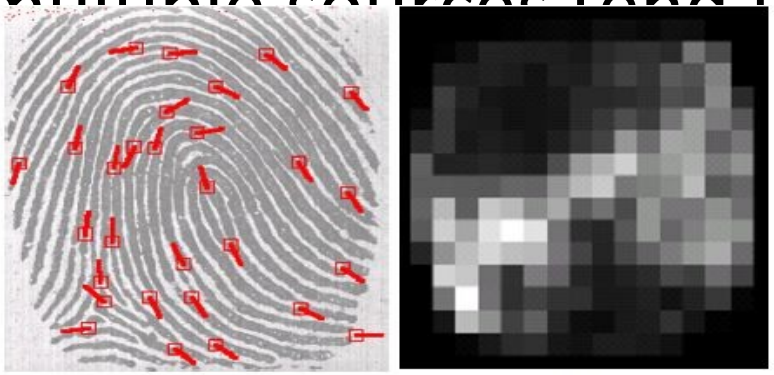
- If an optical and a capacitive fingerprint sensor are used to capture fingerprints, the two fingerprint images can be processed independently without spatially registering them.
- However, in this scenario, the user is required to interact with the sensors one at a time, leading to larger enrollment and verification times.

MULTI ALGORITHM

- In these systems, the same biometric data is processed using multiple algorithms. For example, a texture-based algorithm and a minutiae-based algorithm can operate on the same fingerprint image in order to extract diverse feature sets that can improve the performance of the system.
- This kind of system does not require the use of new sensors and, hence, is cost-effective.

MULTI ALGORITHM

- The main limitation of multi-algorithm systems is that since the different sources of evidence are obtained from the same raw data (e.g., right index fingerprint image), the multiple sources tend to be correlated



(a) (b)

Fig. 6.4 Extracting different sets of features from the same fingerprint image. (a) Minutia features extracted from a fingerprint image, (b) texture features extracted from the fingerprint image in (a).

- multi-algorithm system can use multiple feature sets (i.e., multiple representations) extracted from the same biometric data or multiple matching schemes operating on a single feature set.
- Another example of a system using multiple feature sets is a face recognition system that employs different feature extraction schemes like Principal Component Analysis (PCA), Independent Component
- Analysis (ICA), and Linear Discriminant Analysis (LDA) to encode (i.e., represent) a single face image.

Multi-instance systems

- These systems use multiple instances of the same body trait and are also sometimes referred to as **multi-unit systems**
- For example, the left and right index fingers, or the left and right irides of an individual may be used to verify an individual's identity.
- Automated Fingerprint Identification Systems (AFIS), that obtain tenprint information from a subject, can benefit from sensors that are able to rapidly acquire impressions of all ten fingers in three stages as shown in Figure 6.6.

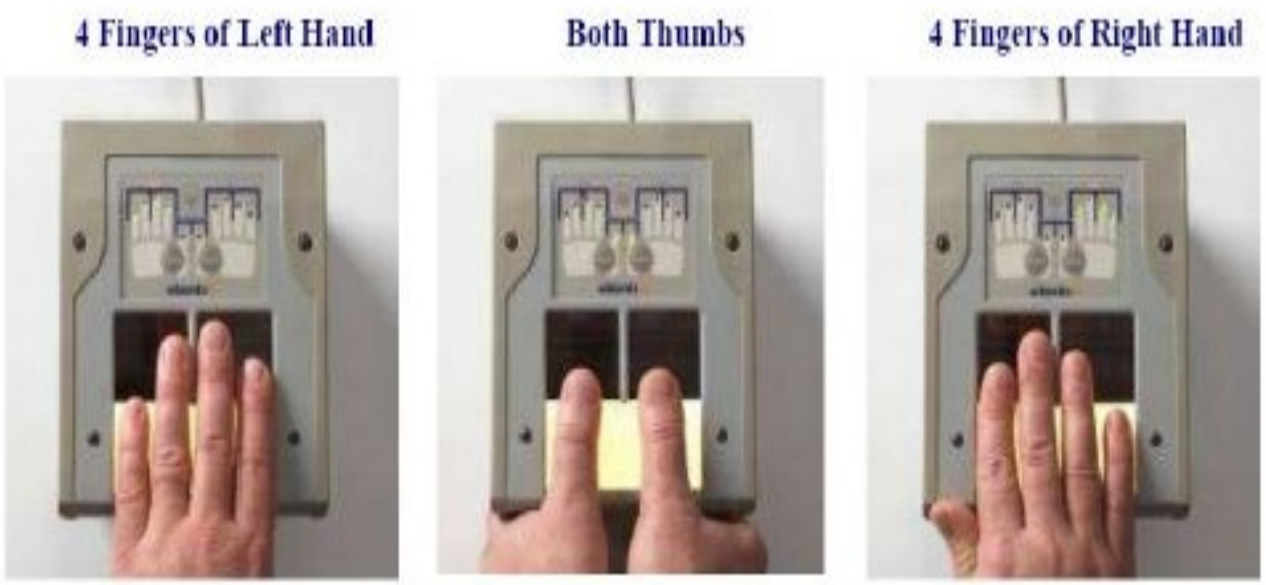


Fig. 6.6 A fingerprint sensor developed by Identix that allows rapid acquisition of all ten fingers in three steps. (Source: Nationwide Solutions)

- Multi-instance systems are often necessary in applications where the size of the system database (i.e., the number of enrolled individuals) is very large (the FBI's IAFIS database currently has more than 60 million ten-print images) and all ten fingers provide additional discriminatory information that is required for high search accuracy.

Multi-sample systems

- A single sensor may be used to acquire multiple samples of the same biometric trait in order to account for the variations that can occur in the trait, or to obtain a more complete representation of the underlying trait.
- A face system, for example, may capture (and store) the left and right profile images along with the frontal image of a person's face in order to account for variations in the facial pose

- One of the key issues in a multi-sample system is determining the *number* of samples that need to be acquired from a biometric trait.
- It is important that the procured samples represent the ***variability*** as well as the ***typicality*** of the individual's biometric data.

MULTIMODAL SYSTEMS

- Multimodal systems combine the evidence presented by different body traits for establishing identity. Some of the earliest multimodal biometric systems utilized face and voice features to establish the identity of an individual.
- The cost of deploying multimodal biometric systems is substantially more due to the requirement of multiple sensors and, consequently, the development of appropriate user interfaces

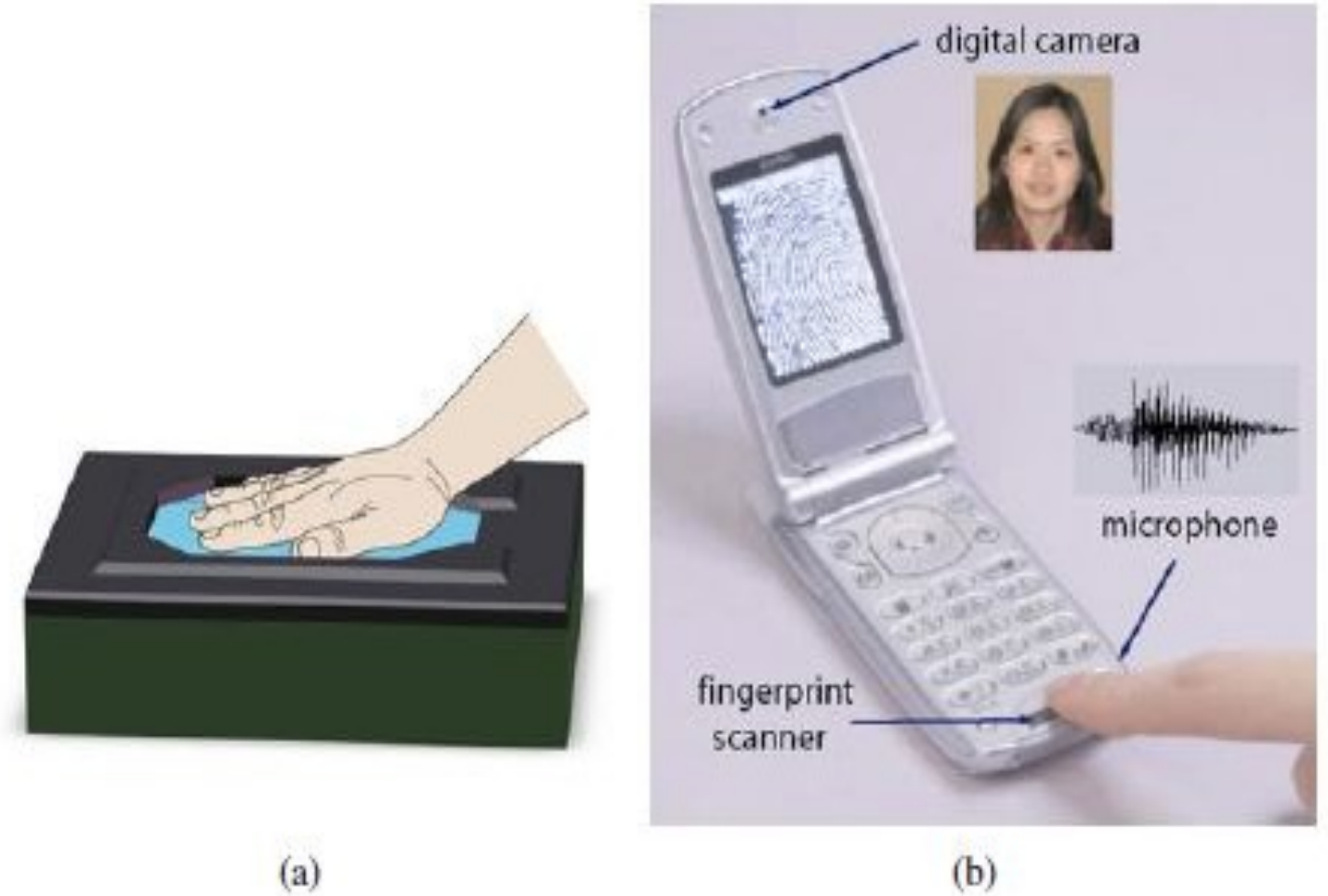


Fig. 6.8 Examples of interfaces that can record multibiometric data. (a) Concept diagram of a whole-hand scanner that can simultaneously acquire palmprint, fingerprints from all five fingers of a hand, and hand-shape (Source: Lumidigm Inc.), (b) a mobile phone that can acquire multiple modalities like fingerprint, face, and voice. Preliminary efforts have also been made to modify the camera on the phone to capture iris image as well.

Acquisition and Processing Architecture

- The order or sequence of biometric data acquisition has a bearing on the convenience imparted to the user.
- The sequence in which the procured biometric data is processed can significantly impact the throughput time in large-scale identification systems

Acquisition sequence

- The **acquisition sequence** in a multibiometric system refers to the order in which the various sources of evidence are acquired from an individual (in the case of multi-algorithm systems, only a single biometric sample is required and, therefore, the acquisition methodology is not an issue).

SERIAL AND PARALLEL ACQUISITION

- Typically, multi-biometric systems employ the serial acquisition approach, where the evidence is gathered sequentially, i.e., each source is independently obtained with a short time interval between successive acquisitions.
- In some cases, the evidence may be acquired simultaneously (in parallel fashion). For example the face and iris information of a user may be obtained simultaneously by utilizing two cameras housed in the same unit.
- Similarly, the face, voice, and lip movements of a user may be acquired simultaneously by capturing a video, and multiple fingerprints can be captured in parallel using a multi-finger slap scanner

Processing sequence

- The **processing sequence** adopted by a multi-biometric system refers to the order in which the acquired information is processed to render a decision which could be independent of the order in which the information is acquired.
- Thus, information may be *acquired sequentially* but *processed simultaneously* and vice

Serial mode

- In the **serial or cascade mode**, the processing of information takes place sequentially. In Figure, the fingerprint information of the user is first processed; if the fingerprint sub-system is unable to determine the user's identity, then the data corresponding to the face biometric is processed.

Serial mode of operation

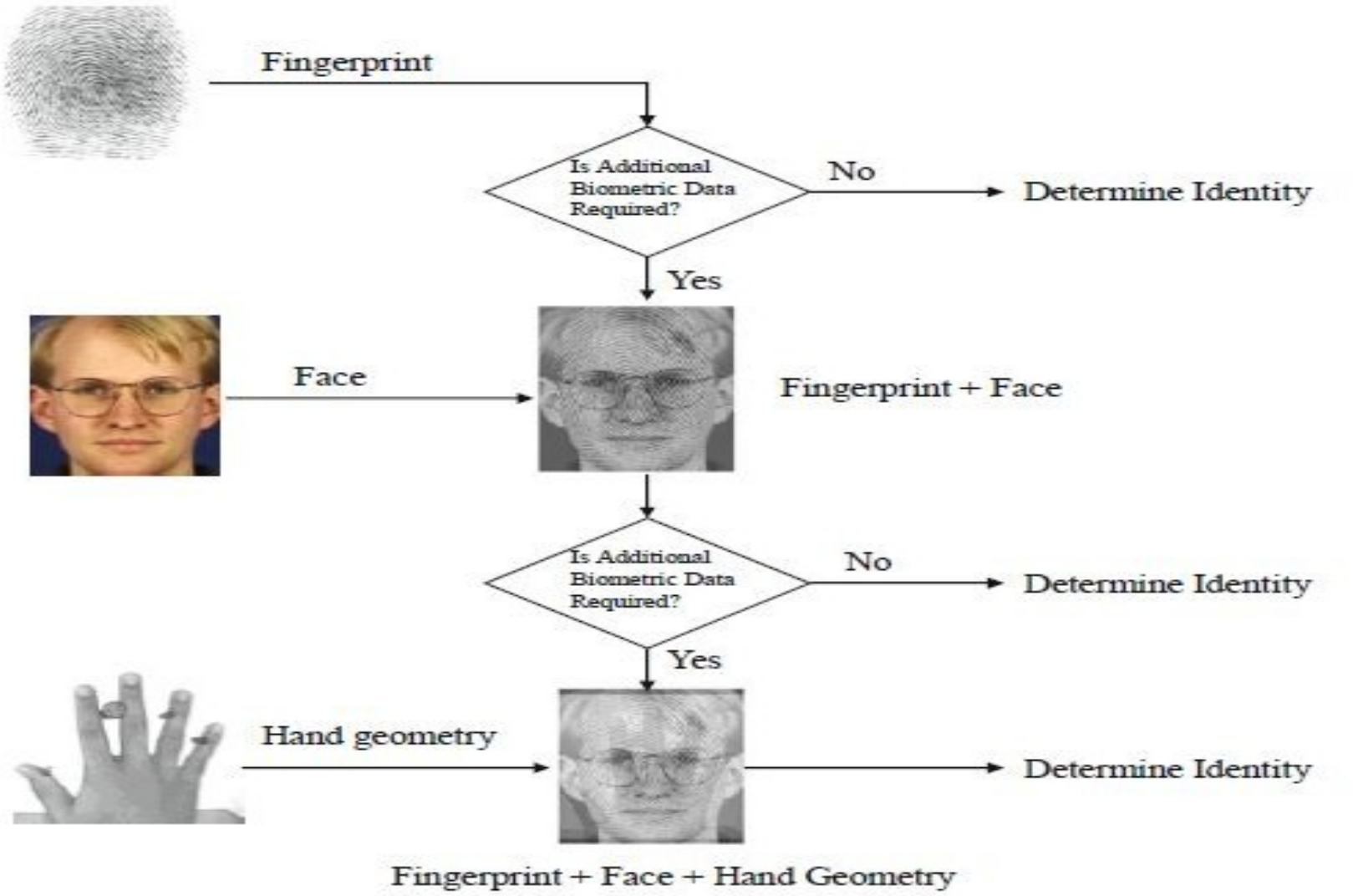


Fig. 6.11 In the cascade (or serial) mode of operation, evidence is incrementally processed in order to establish the user's identity. This scheme is also known as sequential pattern recognition. It enhances user convenience while reducing the average processing time since a decision can be made without having to acquire all the biometric traits.

Parallel

- In the parallel mode, each uni-biometric system processes its information independently at the same time and the processed information is combined using an appropriate fusion scheme
- A multi-biometric system designed to operate in the parallel mode generally has a higher accuracy because it utilizes more evidence about the user for recognition.
- Most practical multi-biometric systems have a parallel architecture because the primary goal of multi-biometric system designers has been to reduce the error rates of biometric systems and not necessarily the throughput and/or processing time.

PARALLEL MODE OF OPERATION

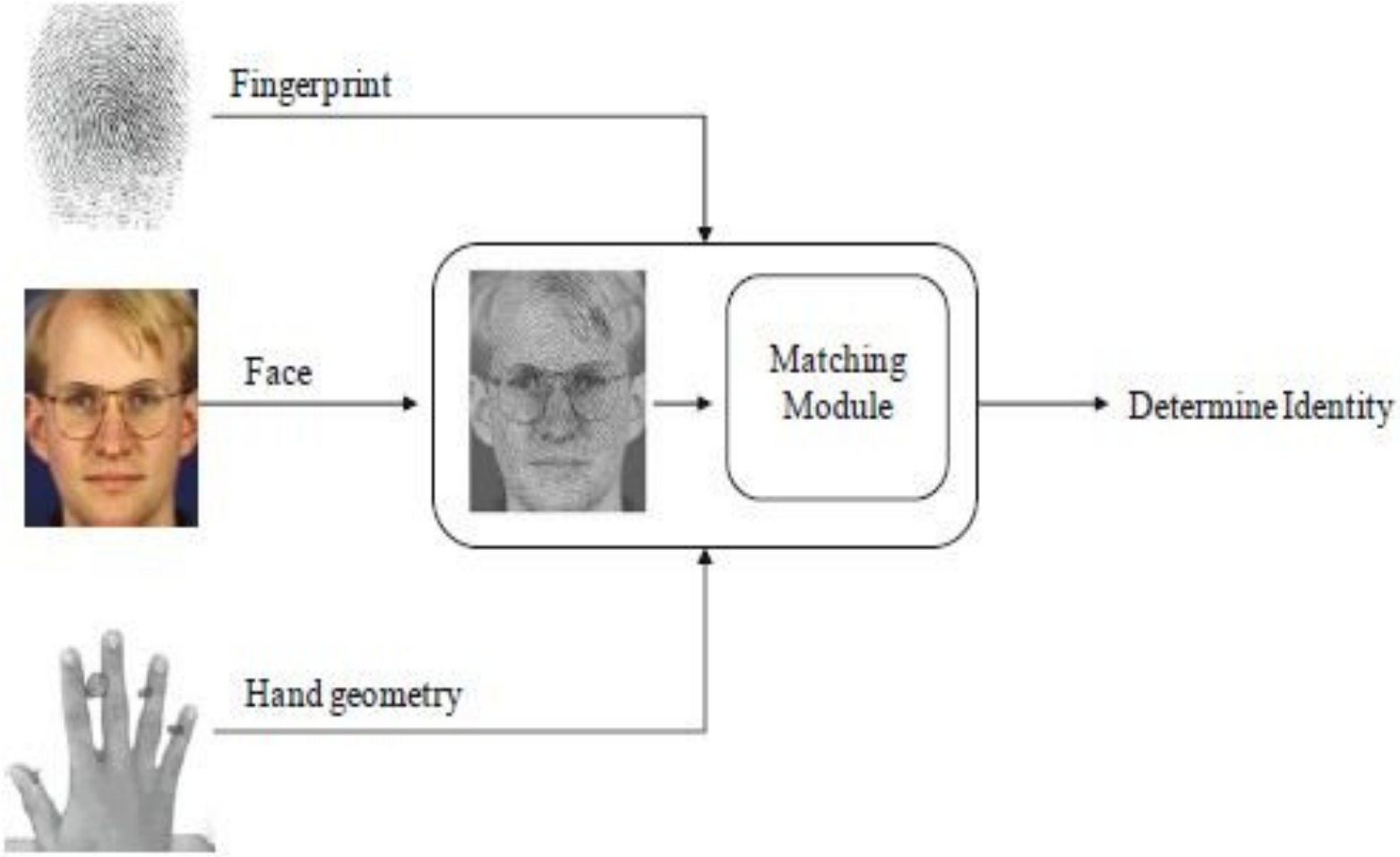
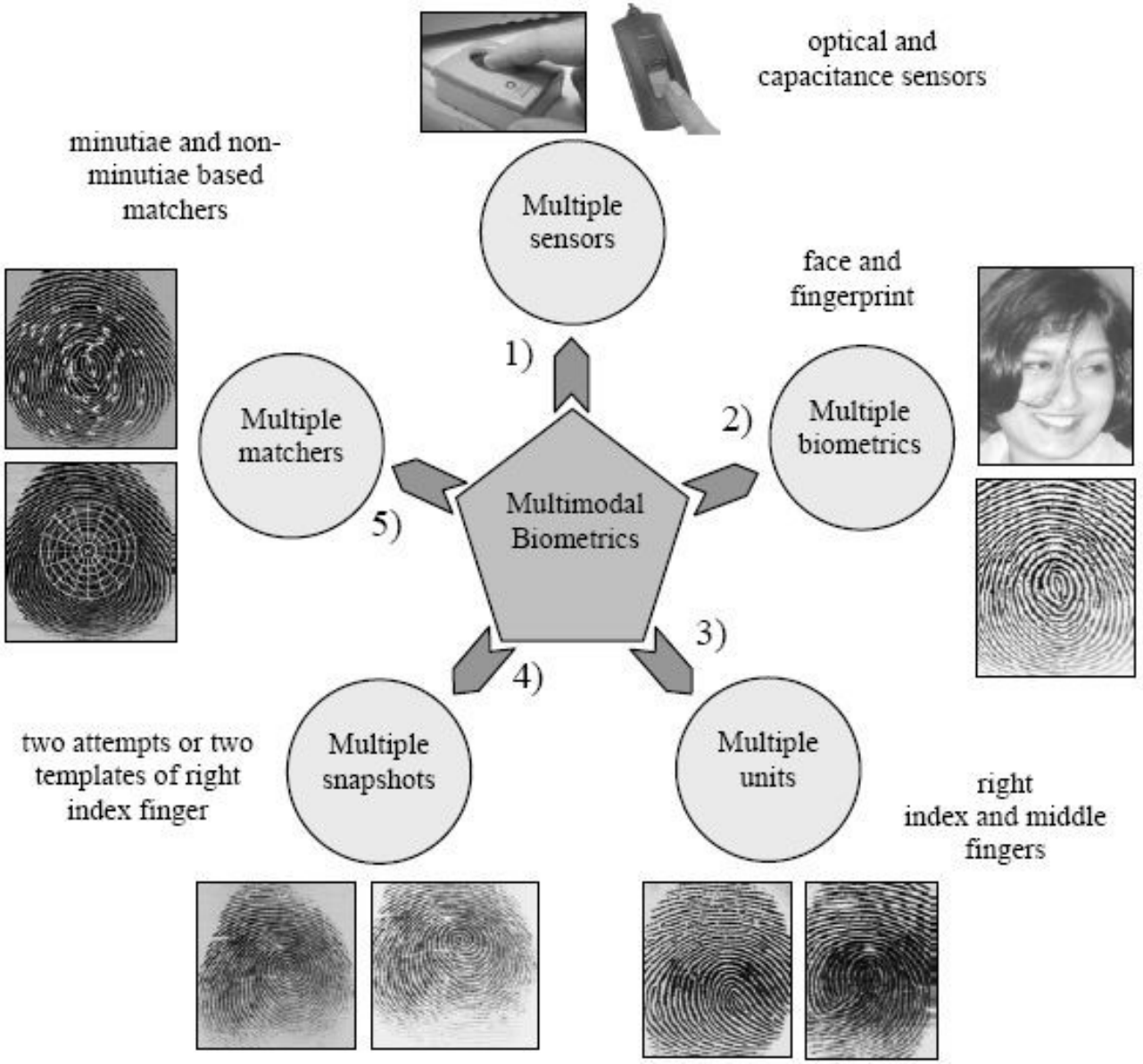


Fig. 6.12 In the parallel mode of operation, the evidence acquired from multiple sources is simultaneously processed in order to establish user's identity. Note that the evidence pertaining to the multiple sources may be acquired in a sequential fashion.

HYBRID

- It is also possible to have a hierarchical (tree-like) architecture to combine the advantages of both cascade and parallel architectures.
- In such a scheme, a subset of the acquired modalities may be combined in parallel, while the remaining modalities may be combined in a serial fashion.
- Such an architecture can be dynamically determined based on the quality of the individual biometric samples as well as when encountering missing biometric data.


optical and
capacitance sensors
minutiae and non-
minutiae based
matchers
Multiple
sensors
face and
fingerprint
1)
Multiple
matchers
2)
Multiple
biometrics
Multimodal
Biometrics
5)
3)
4)
two attempts or two
templates of right
index finger
Multiple
snapshots
Multiple
units
right
index and middle
fingers

FUSION LEVELS

- A fundamental issue in the design of a multi-biometric system is to determine the *type* of information that should be consolidated by the fusion module.
- In a typical biometric system, the amount of information available to the system gets compressed as we proceed from the sensor module to the decision module (see Figure).

SIZE OF THE DATA IS REDUCED

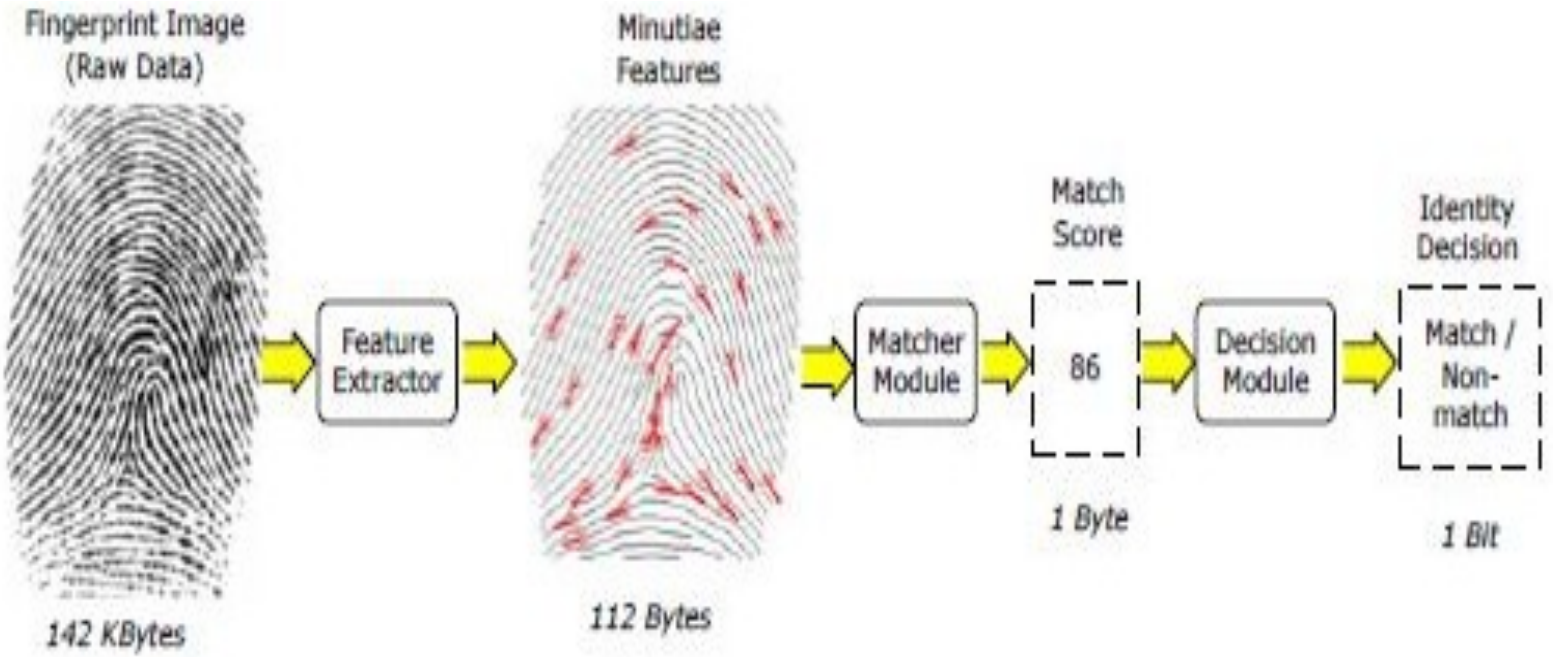


Fig. 6.13 The amount of information available for fusion gets reduced as one progresses along the various processing modules of a biometric system. The raw data represents the richest source of information, while the final decision (in a verification scenario) contains just a single bit of information. However, the raw data is corrupted by noise and may have large intra-class variability, which is typically reduced in the subsequent modules of the system.

BIOMETRIC FUSION

- Multi-biometric system, fusion can be accomplished by utilizing the information available in any of the four biometric modules (sensor, feature extractor, matcher, and decision modules).
- Figure shows the various levels of fusion possible in a multi-biometric system.
- Biometric fusion can be broadly classified into
- (a) fusion prior to matching, and
- (b) fusion after matching.

BIOMETRIC FUSION

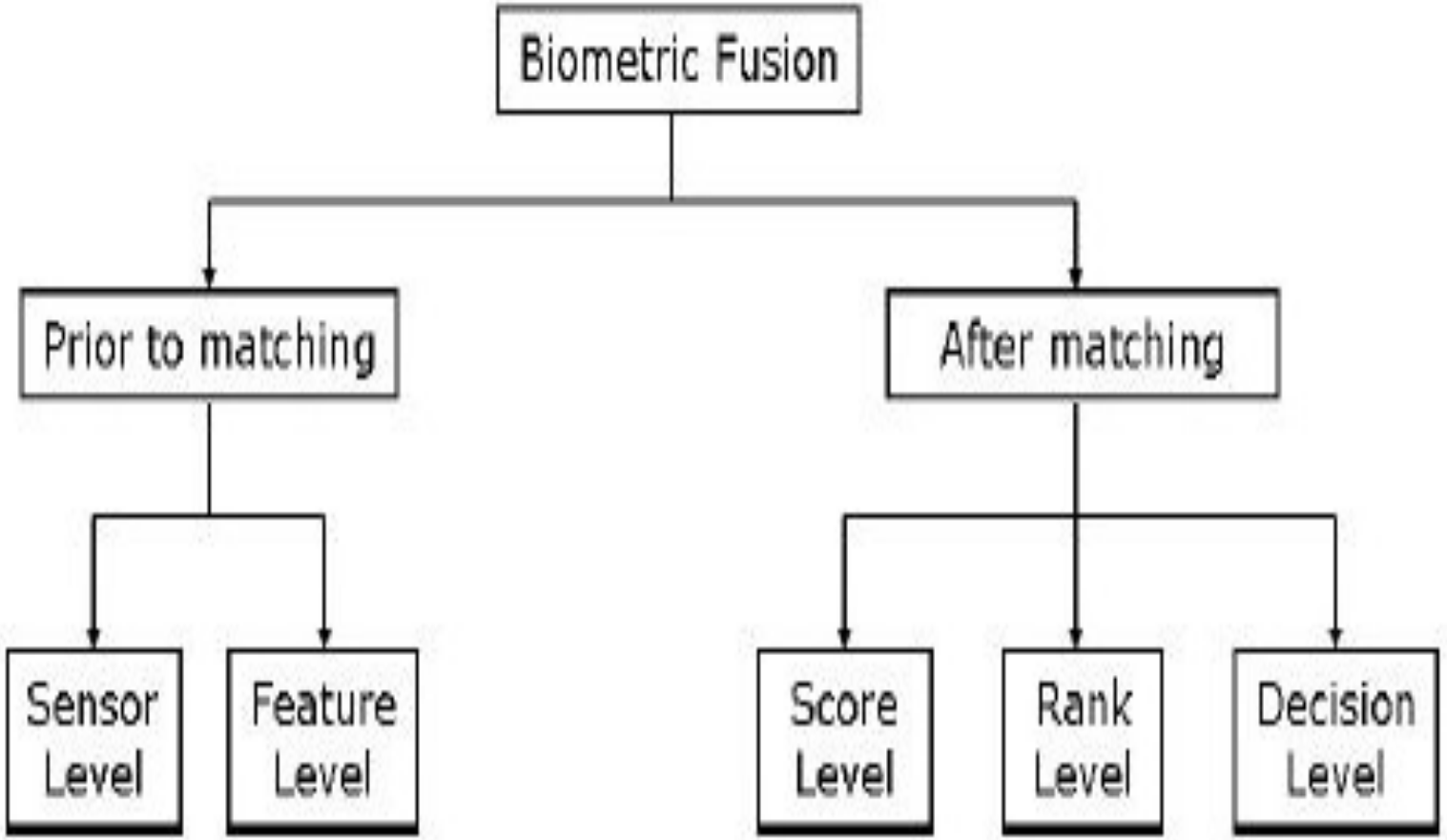


Fig. 6.14 Fusion can be accomplished at various levels in a biometric system. Most multibiometric systems fuse information at the score level or the decision level. Fusion at the rank level is applicable only to biometric systems operating in the identification mode.

PRIOR TO MATCHING &AFTER MATCHING

- Prior to matching, integration of information from multiple biometric sources can take place either at the sensor level or at the feature level.
- Schemes for integration of information after the classification/matcher stage can be further divided into three categories:
  - Fusion at the decision level,
  - Fusion at the rank level,
  - Fusion at the match score level.

COMPARISON

- Biometric systems that integrate information at an early stage of processing are believed to be more effective than those systems that perform integration at a later stage.
- Since the feature set contains richer information about the input biometric pattern than the match score or the decision label, integration at the feature level is expected to provide better recognition results than score or decision level fusion.

Different Fusion Strategies

- Fusion prior to matching
  - Sensor level fusion
  - Feature level fusion
- Fusion after matching
  - Match score fusion
  - Rank level fusion
  - Decision level fusion

Different Fusion Strategies (cont.)

- **Sensor level fusion**
  - Raw data from the sensor(s) are combined.
  - This is referred to **as image level or pixel level fusion**.
  - Sensor level fusion can benefit multi-sample systems which capture multiple snapshots of the same biometrics.
  - For example, 2D face images of an individual obtained from several cameras can be combined to form a 3D model of the face.
  - Another example of sensor level fusion is the mosaicing of multiple fingerprint impressions of a subject in order to construct a more elaborate fingerprint image.

Sensor level fusion

- Sensor level fusion can benefit **multi sample systems** which capture multiple snapshots of the same biometric.
- Small fingerprint sensor may capture two or more impressions of a person's fingerprint and create a composite fingerprint image that reveals more of underlying ridge structure. This process is known as mosaicing.
- Stiching algorithm is required to integrate various slices.
- Mosaicing has been attempted by researchers in face recognition where multiple 2D images representing different poses are stitched to generate a single image

Sensor level Fusion

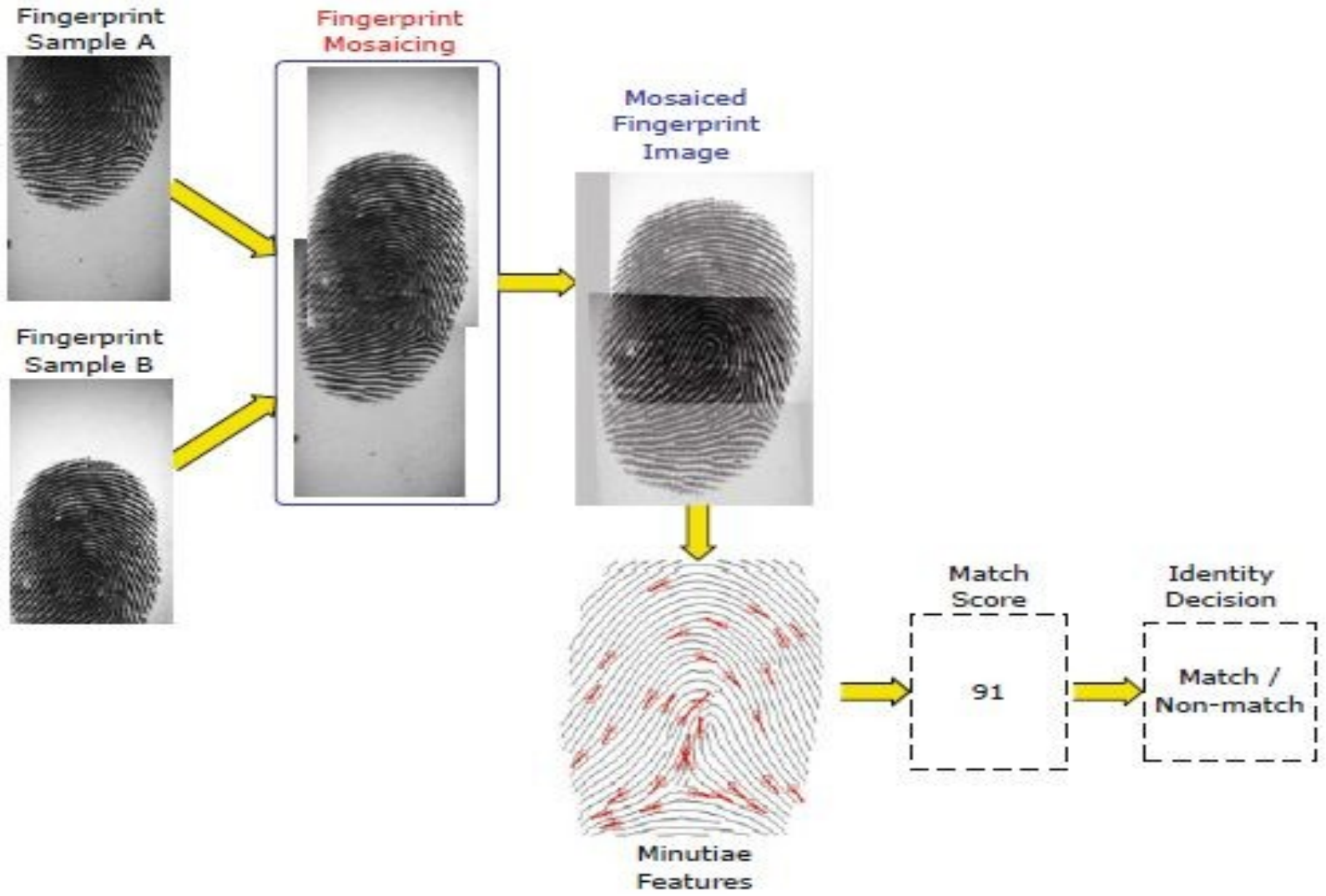


Fig. 6.16 Illustration of a sensor-level fusion scheme where multiple impressions of the same finger are stitched together using a process called mosaicing to generate a composite fingerprint image.

SENSOR level

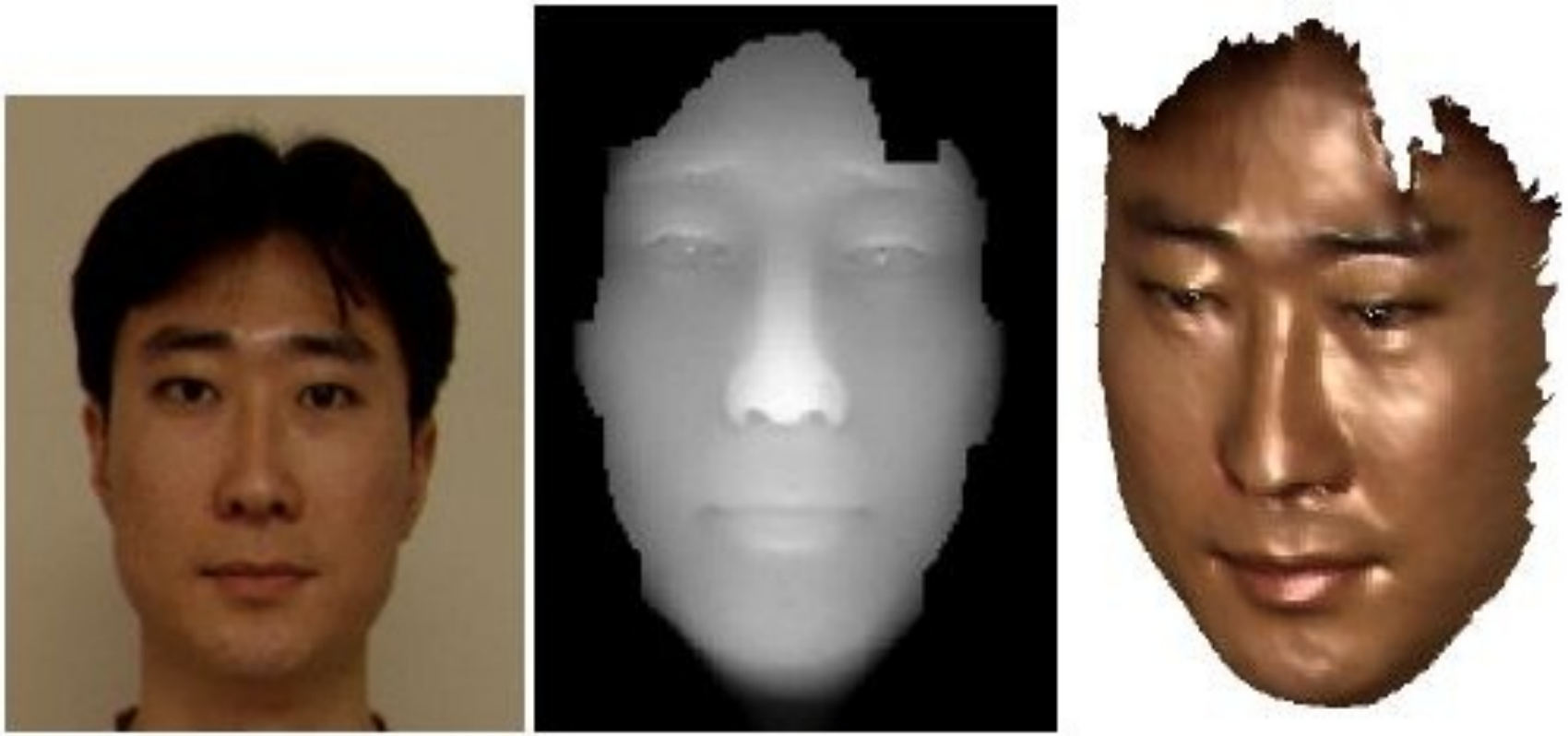


Fig. 6.3 Constructing a 3D face texture by combining the evidence presented by a 2D texture image and a 3D range image. (a) The 2D face texture of a person, (b) the corresponding 3D range (depth) image (here, blue represents farther distance from the camera while red represents a closer distance), and (c) the 3D surface after mapping the 2D texture information from (a).

Different Fusion Strategies (cont.)

- **Feature level fusion**
  - It refers to combine **different feature sets** extracted from multiple biometric sources.
  - When feature sets are homogeneous, a single resultant **feature vector** can be calculated as a **weighted average** of the individual feature vector
  - When the feature set are non-homogeneous , we can concatenate to form a single feature vector.
  - Sometime concatenation may not be possible (Fingerprint minutiae and Eigen-face coefficient)

Feature level sensor

- Feature-level fusion schemes can be categorized into two broad classes, namely, **homogeneous and heterogeneous**.
- A homogeneous feature fusion scheme is used when the feature sets to be combined are obtained by applying the **same feature extraction algorithm** to multiple samples of the same biometric trait (e.g., minutia sets from two impressions of the same finger). This approach is applicable to multi-sample and multi-sensor systems.
- Heterogeneous feature fusion techniques are required if the component feature sets originate from **different feature extraction algorithms** or from samples of different biometric traits (or different instances of the same trait).

Homogeneous feature fusion

- Homogeneous feature fusion can be used for **template update or template improvement**.
- **Template update:** A template in the database can be updated based on the evidence presented by the current feature set in order to reflect (possibly) permanent changes in a person's biometric.
- A simple scheme would be to take average of two feature vectors corresponding to two instances of biometric signal and use the average feature vector as the new template

- **Template improvement**: In the case of fingerprints, the minutiae information available in two impressions can be combined by appropriately aligning the two prints and then removing duplicate minutia, thereby generating a larger minutia set (see Figure).
- This process, known as template improvement, can also be used to remove spurious minutiae points that may be present in a feature set.
- While template update is used to accommodate temporal changes in a person's biometric, the purpose of template improvement is to increase the number of features (*and* decrease the number of spurious features) whilst retaining its integrity

Homogenous feature fusion

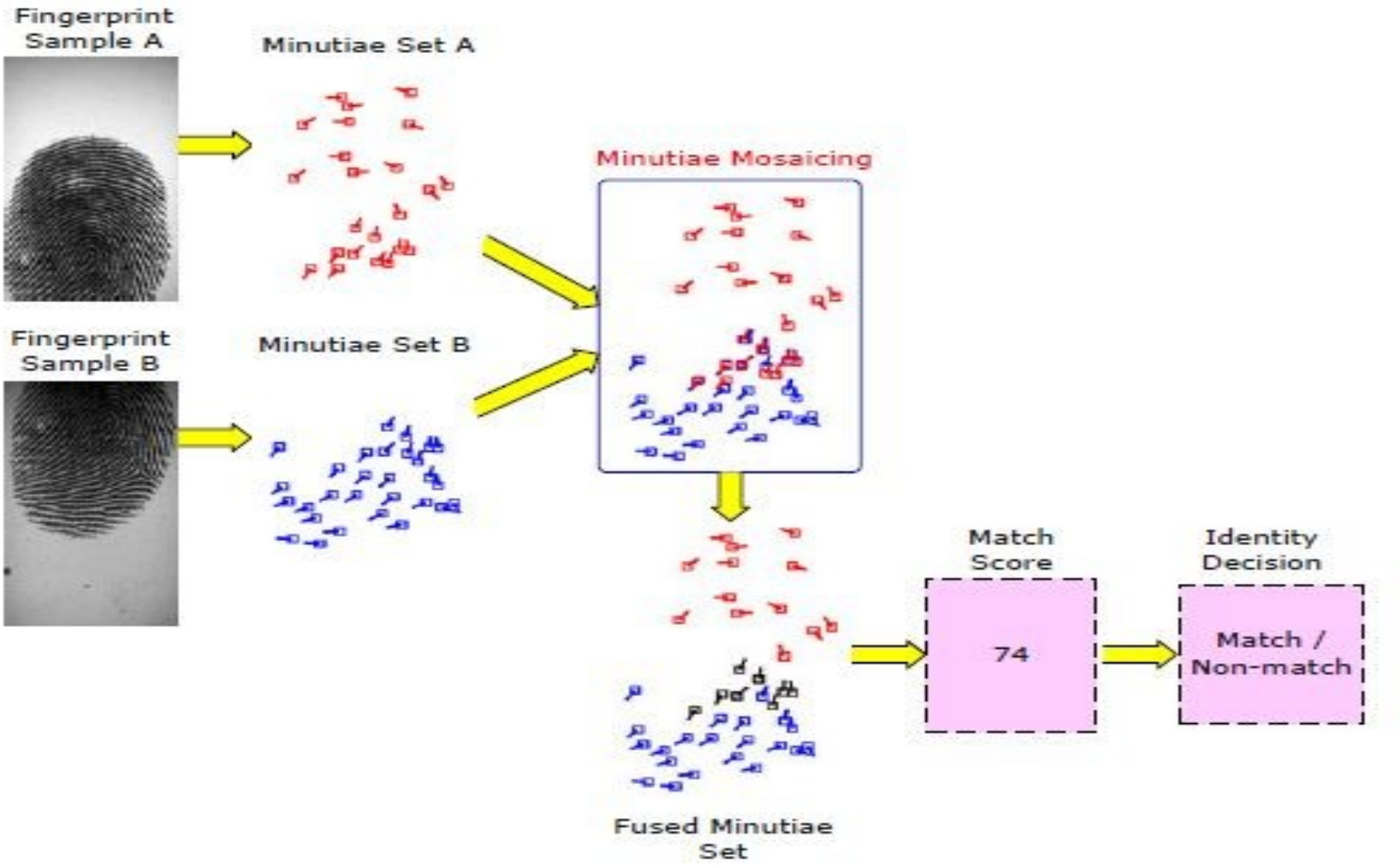


Fig. 6.18 Illustration of a homogeneous feature fusion (template improvement) scheme where minutia sets extracted from multiple impressions of the same finger are reconciled to generate a larger minutia set.

Heterogeneous feature fusion

- How does one consolidate feature sets originating from different algorithms and different biometric modalities?
- Feature level fusion is difficult to achieve in such cases because of the following reasons:
- The relationship between the feature spaces of different biometric systems may not be known.
- The feature sets of multiple modalities may be incompatible. For example, the minutiae set of fingerprints and the eigen-coefficients of face have different representation schemes.
- Concatenating two feature vectors might lead to the curse-of-dimensionality problem, where increasing the number of features might actually degrade the system performance especially in the presence of small number of training samples. The curse of dimensionality basically means that the error increases with the increase in the number of features
- Most commercial biometric systems do not provide access to the

Heterogenous

To avoid the above said issue follow the

- **Feature Normalization**
- **Feature Selection or Transformation**
- The main purpose of feature normalization is to modify the **location** and **scale** parameters of individual feature values to transform the value into a common domain.

Min max normalization

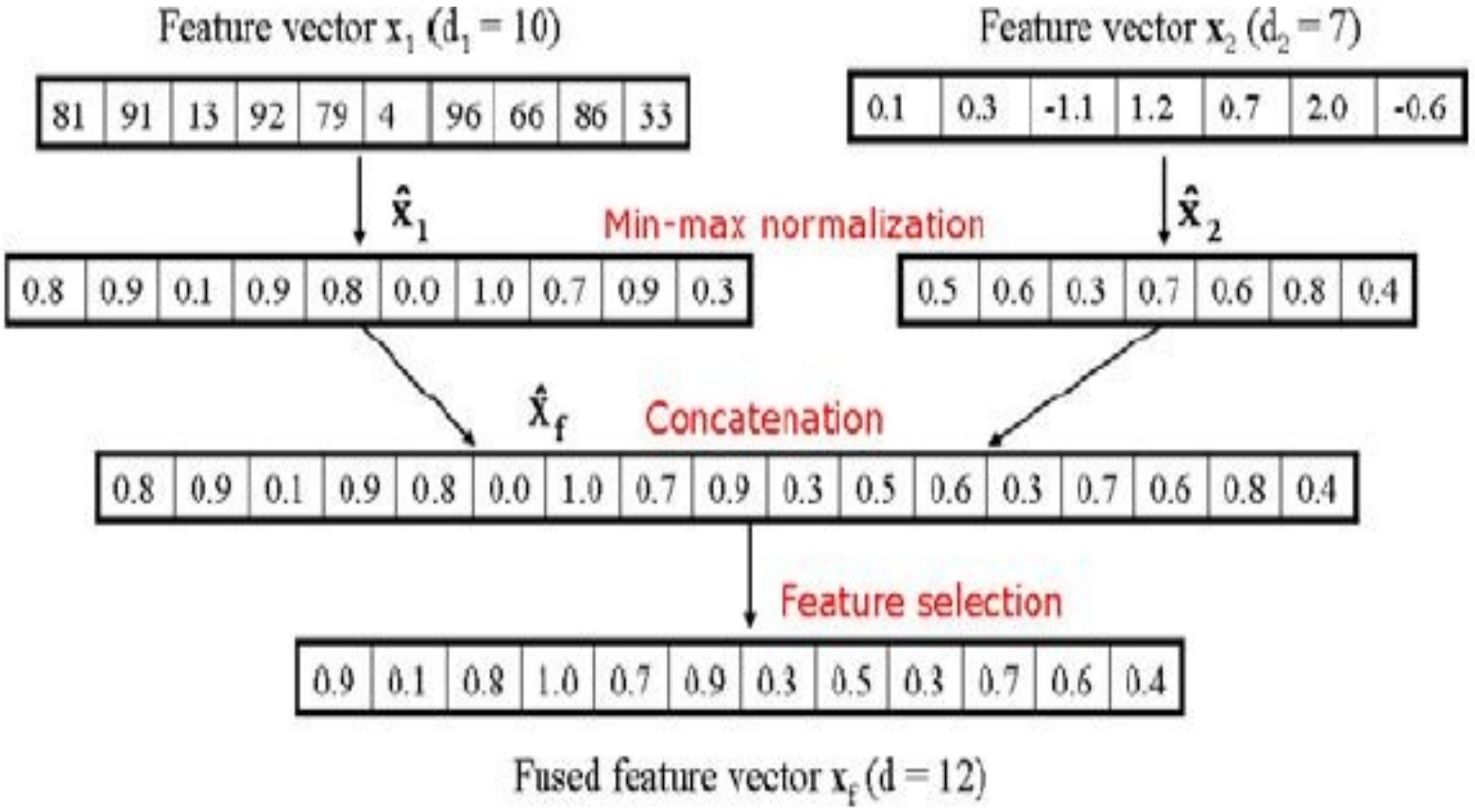


Fig. 6.19 A simple scheme for the fusion of two heterogeneous feature vectors whose lengths are fixed across all users. In this example, min-max normalization is performed based on the assumption that the ranges of feature values are $[0, 100]$ and $[-3, 3]$ for the first and second feature vectors, respectively.

Min max

A simple normalization scheme that is often used in practice is the min-max normalization scheme, which transforms the features values such that they fall in the range $[0, 1]$, irrespective of their original values. Let x and $\hat{x}$ denote a feature value before and after normalization, respectively. The min-max technique computes $\hat{x}$ as

$$\hat{x} = \frac{x - \min(h_x)}{\max(h_x) - \min(h_x)}, \tag{6.4}$$

where h_x is the function that generates x, and $\min(h_x)$ and $\max(h_x)$ represent the minimum and maximum of all possible x values that will be observed, respectively. The min-max technique is effective when the minimum and the maximum values of the component feature values are known beforehand. In cases where such infor-

Feature normalization

- The goal of feature normalization is to modify the location (mean) and scale (variance) of the features values via a transformation function in order to map them into a common domain.
- Adopting an appropriate normalization scheme also helps address the problem of outliers in feature values.

Feature selection scheme

- Feature selection is a dimensionality reduction scheme that entails choosing a minimal feature set of size d, $d < (d1+d2)$, such that a criterion (objective) function applied to the training set of feature vectors is optimized.

Different Fusion Strategies (cont.)

- Features are normalized
- Feature selection schemes are employed to reduce the dimension size of the feature vector
  - Principal Component Analysis (PCA)
  - Linear Discriminant Analysis (LDA)

Different Fusion Strategies (cont.)

- **Score level fusion**
  - Scores generated from different matching modules are combined to produce a single score.
  - Final decision is taken by considering the fused score.
  - Normalization and Similarity/ Dissimilarity Score
  - There are various approaches possible for combining the individual scores.
    - Product rule
    - Sum rule
    - Weighed sum rule
    - Max rule and median rule

Score level

- When match scores output by different biometric matchers are consolidated in order to arrive at a final recognition decision, fusion is said to be done at the score level.
- This is also known as fusion at the measurement level or confidence level.
- After the raw data and feature vectors representations, the next level of fusion is based on match scores.
- It is relatively easy to access and combine the scores generated by different biometric matchers.
- Consequently, score-level fusion is the most commonly used approach in multibiometric systems

- Score-level fusion methodologies will vary depending on whether the multibiometric system operates in the verification or identification mode.
- Score fusion in a multibiometric verification system can be considered as a two-class pattern classification problem, where the goal is to determine whether the query corresponds to a “genuine” user or an “impostor”.

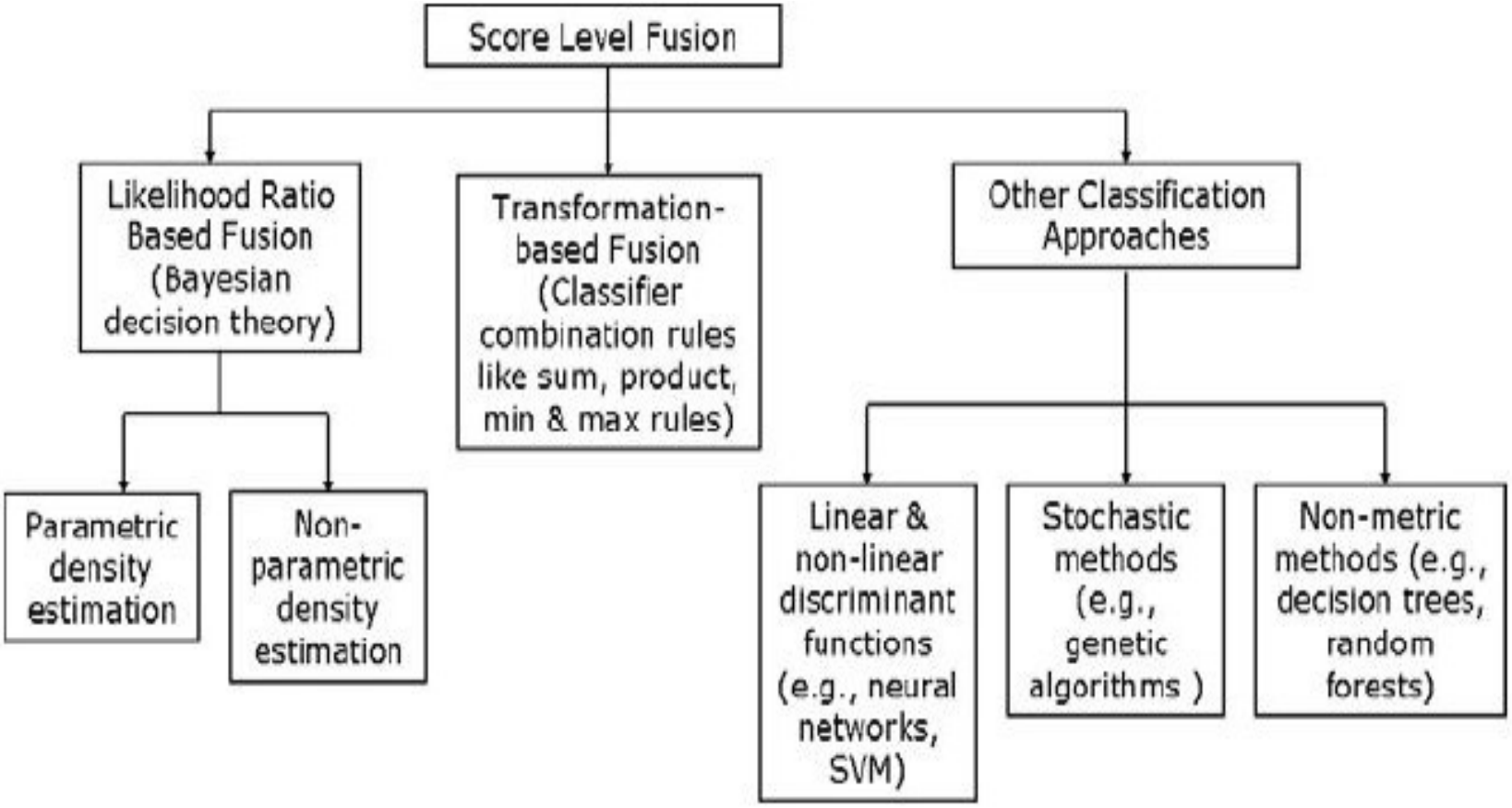


Fig. 6.22 Taxonomy of classification approaches that can be used for score level fusion in a multi-biometric verification system. Note that the above categorization is not strict and some score fusion techniques may be categorized under multiple approaches or may involve more than one basic approach.

Different Fusion Strategies (cont.)

- **Rank level fusion**
  - For identification, output is the ranks of enrolled identities.
  - This fusion scheme is to consolidate the ranks of individual biometric systems to derive a fused rank for each identity.
  - It reveals less information than match scores. However, unlike match scores, the ranking output by multiple biometric systems are comparable.
  - No normalization is needed and this makes the rank level fusion schemes simpler to implement compared to the score level fusion techniques.
    - Highest rank method
    - Logistic regression method

Rank method

- When a biometric system operates in the identification mode, the output of the system can be viewed as a ranking of the enrolled identities.
- The goal of rank-level fusion schemes is to consolidate all the ranks output by the individual biometric subsystems in order to derive a consensus rank for each identity.
- Ranks provide more insight into the decision-making process of the matcher compared to just the identity of the best match, but they reveal less information than match score

Rank method

Let $\mathbf{R} = [r_{n,m}]$ be the rank matrix in a multibiometric system, where $r_{n,m}$ is the rank assigned to identity I_n by the m^{th} matcher, $m = 1, \cdots, M$ and $n = 1, \cdots, N$. Let $\hat{r}_n$ be a statistic computed for user I_n such that the user with the lowest value of $\hat{r}$ is assigned the highest consensus (or reordered) rank. The following three well-known methods can be used to compute the statistic $\hat{r}$.

Highest rank

Highest Rank Method: In the highest rank method, each user is assigned the highest rank (minimum r value) as computed by different matchers, i.e., the statistic for user I_n is

$$\hat{r}_n = \min_{m=1}^{M} r_{n,m}. \tag{6.27}$$

Borda count

Borda Count Method: The Borda count method uses the sum of the ranks assigned by the individual matchers to calculate the value of $\hat{r}$, i.e., the statistic for user I_n is

$$\hat{r}_n = \sum_{m=1}^{M} r_{n,m}. \tag{6.28}$$

The magnitude of the Borda count for each user is a measure of the degree of agreement among the different matchers on whether the input belongs to that user. The Borda count method assumes that the ranks assigned to the users by the matchers are statistically independent and all the matchers perform equally well.

Logistic regression

Logistic Regression Method: The logistic regression method is a generalization of the Borda count method where a weighted sum of the individual ranks is calculated, i.e., the statistic for user I_n is

$$\hat{r}_k = \sum_{m=1}^{M} w_m r_{n,m}. \tag{6.29}$$

The weight, w_m, to be assigned to the m^{th} matcher, $m = 1,\ldots,M$, is determined by logistic regression. The logistic regression method is useful when the different biometric matchers have significant differences in their accuracies. One limitation of this method is that it requires a training phase to determine the weights.

Rank all three methods

Face Matcher 1

User Identity	Rank
Alice	3
Bob	1
Charlie	4
Dick	2

Face Matcher 2

User Identity	Rank
Alice	1
Bob	2
Charlie	3
Dick	4

Rank Fusion Module

	Highest Rank		Borda Count		Logistic Regression ($w_1 = 0.8$, $w_2 = 0.2$)	
User Identity	Fused Score	Reordered Rank	Fused Score	Reordered Rank	Fused Score	Reordered Rank
Alice	1	1	4	2	2.6	3
Bob	1	2	3	1	1.2	1
Charlie	3	4	7	4	3.8	4
Dick	2	3	6	3	2.4	2

Fig. 6.27 An illustration of rank-level fusion as performed by the highest rank method, Borda count, and logistic regression. In this example, the three fusion schemes assign different consensus ranks to the individual identities.

Different Fusion Strategies (cont.)

- **Decision level fusion**
  - Decision level fusion is the highest level fusion of biometric evidences.
  - Fusion is carried out at the abstract or decision level when only the decisions output by the individual biometric matchers are available.
  - It logically combines accept/reject matching decisions of different matchers.
    - "AND" and "OR" rule
    - Majority voting
    - Weighted majority voting
    - Bayesian decision fusion

DECISION LEVEL

- Many commercial off-the-shelf (COTS) biometric matchers provide access only to the final recognition decision.
- When such COTS matchers are used to build a multi-biometric system, only decision-level fusion is feasible.
- Decision-level fusion include “AND” and “OR” rules, majority voting, weighted majority voting, Bayesian decision fusion, the Dempster-Shafer theory of evidence, and behavior knowledge space.

" AND" and "OR" Rules

- In a multibiometric verification system, the simplest method of combining decisions output by the different matchers is to use the AND and OR rules.
- The output of the AND rule is a "match" only when all the *M* biometric matchers agree that the input sample matches with the template.
- On the contrary, the OR rule outputs a "match" decision as long as at least one of the *M* matchers decides that the input sample matches with the template.
- The limitation of these two rules is their tendency to result in extreme operating points.
- When the AND rule is applied, the False Accept Rate (FAR) of the multibiometric system is extremely low (lower than the FAR of the individual matchers) while the False Reject Rate (FRR) is high (greater than the FRR of the individual matchers).
- The OR rule leads to higher FAR and lower FRR than the individual matchers.
- When one biometric matcher has a substantially higher equal error rate compared to the other matcher, the combination of the two matchers

MAJORITY VOTING

Majority Voting: The most common approach for decision-level fusion is majority voting where the input biometric sample is assigned to that class ("genuine" or "impostor" for verification systems and identity I_k for identification systems) on which a majority of the matchers agree. If there are M biometric matchers, the input sample is assigned to a class if at least $\hat{m}$ of the matchers agree on that class, where

$$\hat{m} = \begin{cases} \frac{M}{2} + 1 \text{ if } M \text{ is even,} \\ \frac{M+1}{2} \quad \text{otherwise.} \end{cases} \tag{6.30}$$

When none of the classes is supported by $\hat{m}$ matchers, a "reject" decision is output by the system. Majority voting assumes that all the matchers perform equally well. The advantages of majority voting are: (a) no apriori knowledge about the matchers is needed, and (b) no training is required to come up with the final decision.

MAJORITY VOTING

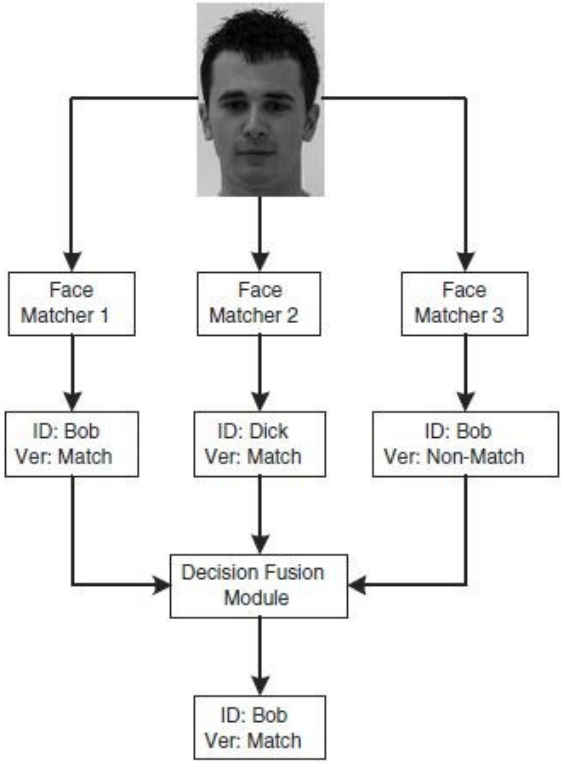


Fig. 6.28 Flow of information when decisions provided by multiple biometric matchers are combined using the majority vote fusion scheme. Here "ID" and "Ver" represent the identification and verification modes of recognition operation, respectively. For the verification mode, the claimed identity is Bob.

Weighted Majority Voting

Weighted Majority Voting: When the matchers used in a multibiometric system do not have similar recognition accuracy, it is reasonable to assign higher weights to the decisions made by the more accurate matchers. In order to facilitate this weighting, the labels output by the individual matchers are converted into degrees of support as follows.

$$\tilde{s}_{n,m} = \begin{cases} 1, \text{ if output of the } m^{th} \text{ matcher is class } n, \\ 0, \text{ otherwise,} \end{cases} \tag{6.31}$$

weighted

where $m = 1, \ldots, M$ and $n = 0, 1$ for verification or $n = 1, 2, \cdots, N$ for identification. The decision rule based on weighted voting can be stated as

$$\text{Decide in favor of class } k \text{ if } \sum_{m=1}^{M} w_m \tilde{s}_{k,m} > \sum_{m=1}^{M} w_m \tilde{s}_{n,m}, \ \forall\, n,\ k \neq n, \qquad (6.32)$$

where w_m is the weight assigned to the m^{th} matcher.

Bayesian decision

- Bayesian decision fusion scheme relies on transforming the discrete decision labels output by the individual matchers into continuous probability values.

- ***Feature Matching and Decision Making***
- ***Feature matching: null and alternative hypothesis h0, h1, Error type I/II, Matching score distribution, FM/FNM, ROC curve, DET curve, FAR/FRR curve.***
- for FAR,FRR,FM,FNM ROC refer unit 1 slides

Errors

Failure to Acquire (FTA)

1. Common reasons of failure to acquire
   - biometric characteristic could not be presented (due to illness, non-conformant presentation, etc.)
   - sensor's failure
   - data processing failure (e.g. segmentation failed)
   - low quality of data
2. Estimator of the failure to acquire probability (FTA) calculated for attempts:

$$\mathrm{FTA} = \frac{\text{number of failed attempts}}{\text{number of all attempts}}$$

Failure to Enroll (FTE)

1. Common reasons of failure to enroll
   - as for FTA and:
   - biometric features could not be calculated
   - test verification failed
2. Estimator of the failure to enroll probability (FTE):

$$\mathrm{FTE} = \frac{\text{number of failed enrollments}}{\text{number of all enrollments}}$$

3. NOTE: restrictive enrollment procedures increase FTE, but decrease FMR/FNMR

Types of biometric samples

1. Genuine: originating from the same class (the same eye, finger, signature, etc.)
2. Impostor or random forgeries or zero-effort attempts: originating from different classes in a form as they were acquired (different eye, finger, signature, etc.)
3. Skilled forgeries: prepared (with some effort) to imitate the sample of a given class (iris printout, gummy finger, skilled forgery in signature recognition, etc.)

Matching and decision making

1. Matching
   - calculation of matching score or similarity/dissimilarity score

     similarity = 0 → samples are totally unalike
     dissimilarity = 0 → samples are identical
   - NOTE: 'similarity' and 'dissimilarity' are not always complementary
2. Possible decisions
   - match/non-match based on comparison the similarity/dissimilarity score with decision threshold

Kinds of errors

- Given two biometric samples we can have two possible hypothesis
  - **Null hypothesis** H0 => two samples are matching
  - **Alternative hypothesis** H1 => two samples are not matching
- Definition of hypothesis depends on biometric applications.
- different application can have different definition of errors.
- But matching module decides whether H0 is true or H1 is true.

1. Hypotheses
   - null hypothesis (H_0)
     - the hypothesis that is tested (assumed to be true)
     - we try to find reasons to reject H_0
     - acceptance of H_0 does not mean that H_0 is true; it only means that we did not find reasons to reject it
   - alternative hypothesis (H_1)
     - we are leaning towards accepting H_1 if H_0 is rejected
2. We consider the simplest case
   - H_0 and H_1 are simple hypotheses, namely:
     - completely specify the population distribution
     - we assume that we know this distribution

Null and simple hypothesis H_0

- iris X is the iris of Adam Czajka

Alternative and simple hypothesis H_1

- iris X is the iris of John Doe
- iris X is one of the N known irides

Alternative and composite hypothesis H_1

- iris X is not the iris of Adam Czajka

A Type I error means rejecting the null hypothesis when it's actually true, while a Type II error means failing to reject the null hypothesis when it's actually false.

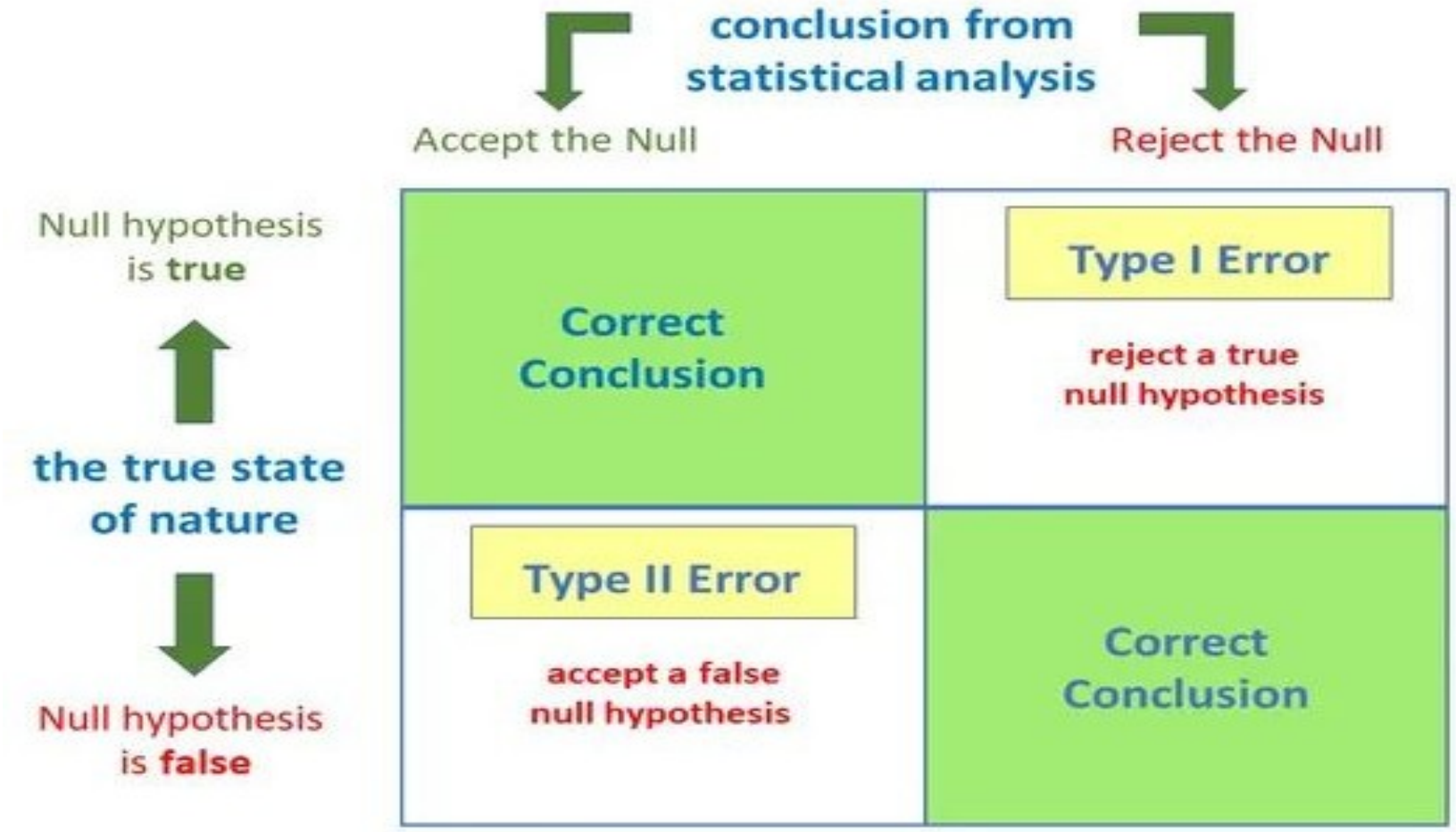
Because we are making a decision based on a finite sample, there is a possibility that we will make mistakes.

The possible outcomes are:

	H_0 is true	H_1 is true
Do not reject H_0	Correct decision	Type II error
Reject H_0	Type I error	Correct decision

Type 1 False Positive

Type 2 False negative

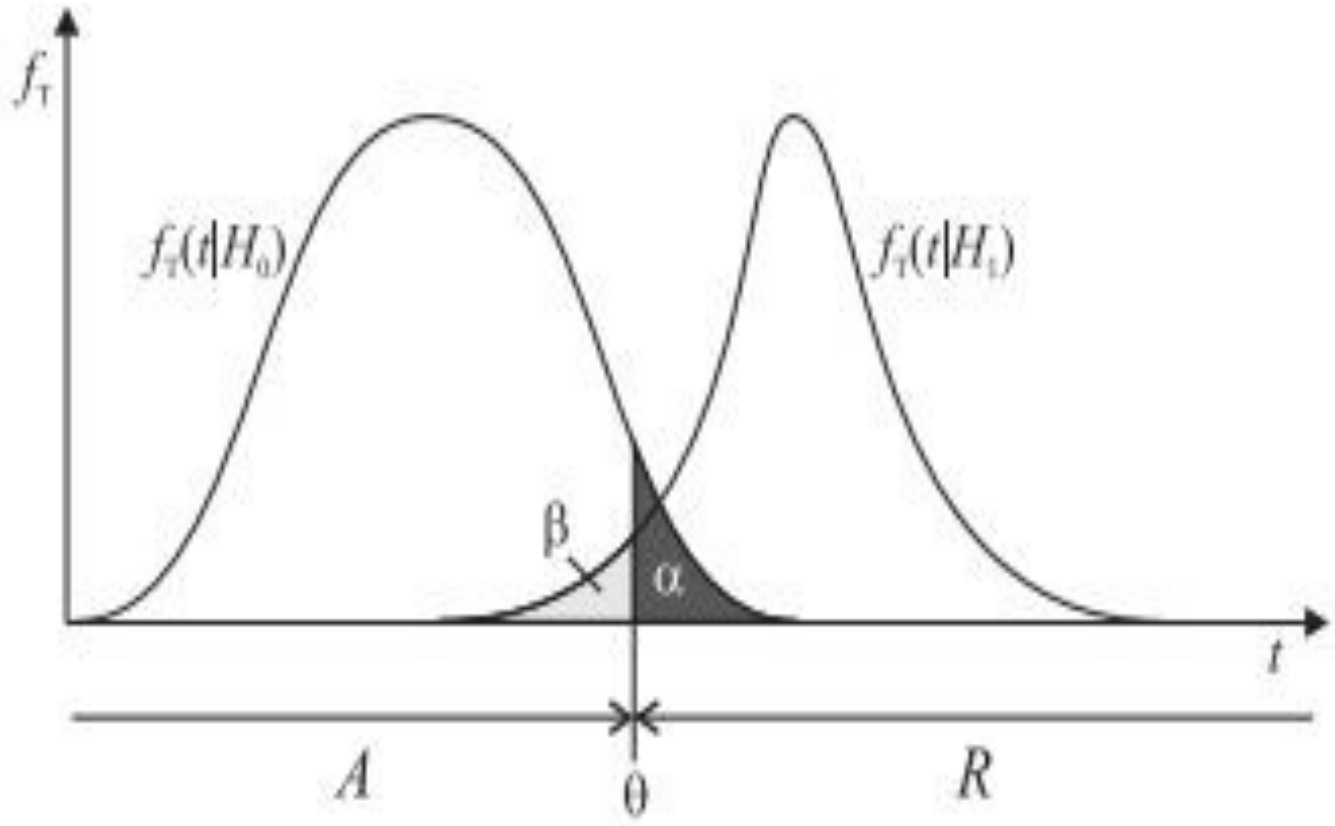


Type 1

- The acceptance of H1 when H0 is true is called a Type I error.
- The probability of committing a type I error is called the **level of significance** and is denoted by alpha.

Type 2

- Failure to reject H0 when H1 is true is called a Type II error.
- The probability of committing a type II error is denoted by beta.



H_0: sample comes from a distribution "0" $\rightarrow$ T statistic has a distribution $f_T(t|H_0)$

H_1: sample comes from a distribution "1" $\rightarrow$ T statistic has a distribution $f_T(t|H_1)$

		decision	
		no reason to reject	reject
hypothesis	true	OK $(1-\alpha)$	type I error (α)
	false	type II error (β)	OK $(1-\beta)$

where α: significance level, $1-\beta$: power of the test

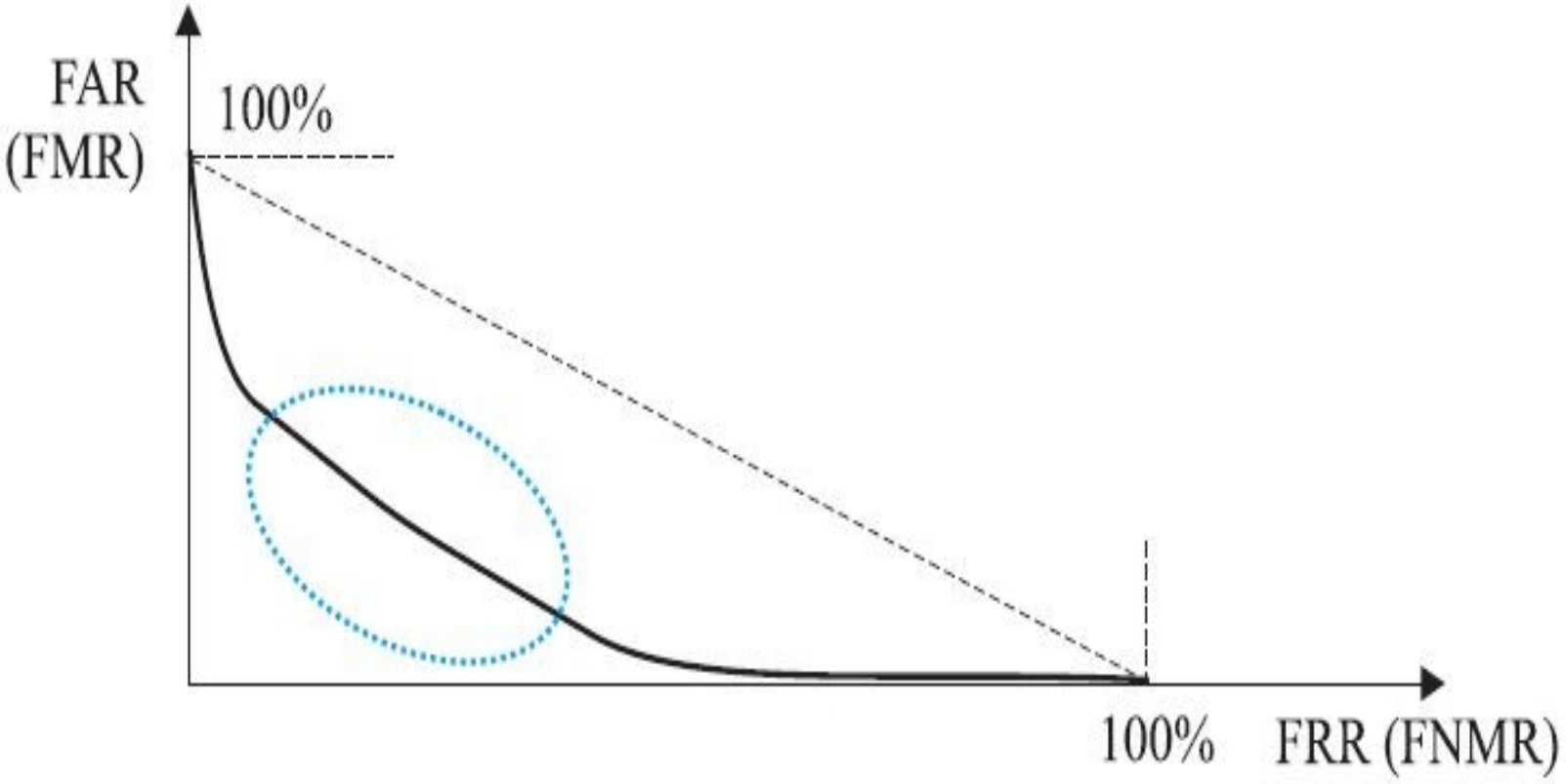
- DET -- A **detection error tradeoff** (DET) graph is **a graphical plot of error rates for binary classification systems**, plotting the false rejection rate vs. false acceptance rate.

  DET curves give the user direct feedback of the detection error tradeoff to aid in operating point analysis.

  The user can deduct directly from the DET-curve plot at which rate false-negative error rate will improve when willing to accept an increase in false-positive error rate (or vice-versa).

Detection Error Tradeoff (DET)

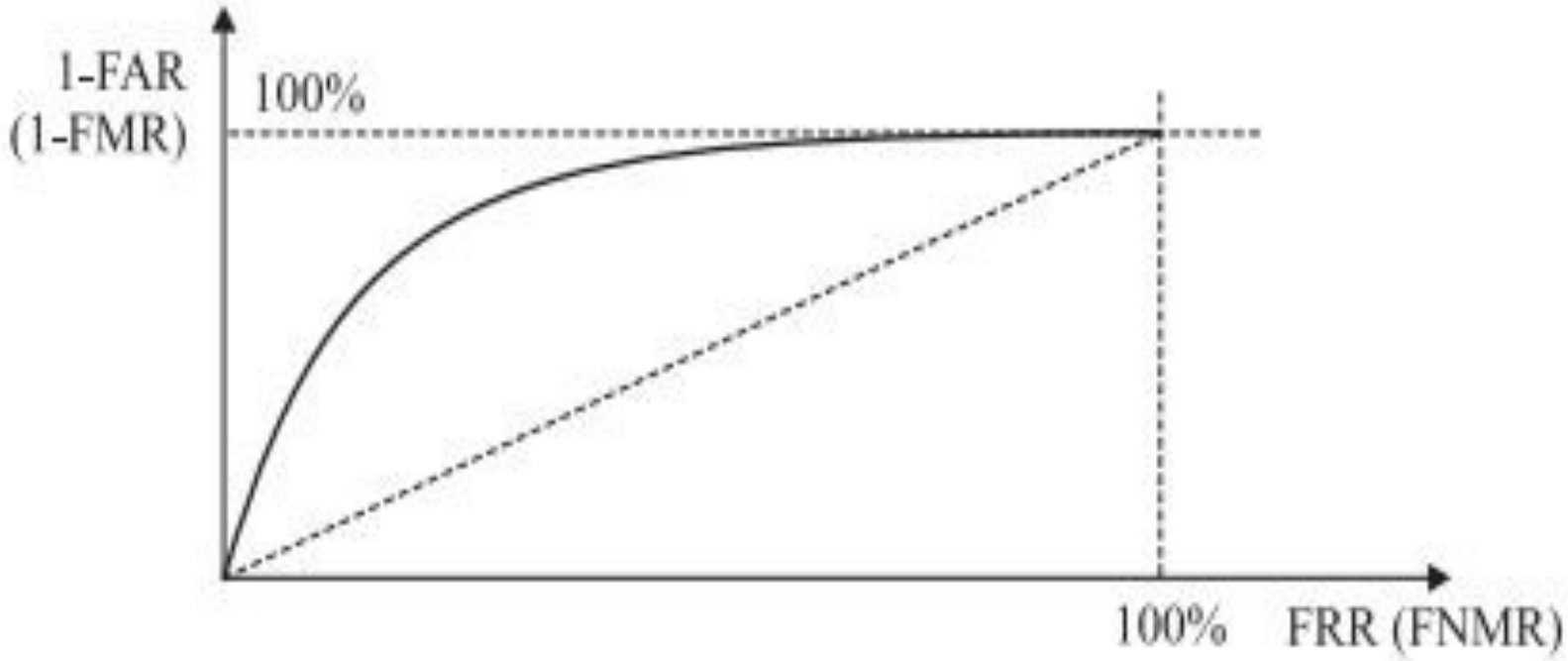
Use of the inverse normal CDF (cumulative probability distribution) for expressing the estimator values (makes the curve in the analysis area to be roughly a straight line)



ROC

Receiver Operating Characteristic (ROC)

Parametric curve joining FAR and FRR, or FMR and FNMR.
Acceptance threshold is the parameter.



Feature Matching and Decision Making

Matching

- Matcher is a system that takes two samples of biometric data and returns a score that indicates their similarity and dissimilarity
- Similarity and dissimilarity measures highly depends on
  -data acquisition device
  -precision of representation of biometric samples
- degree of uniqueness of the samples overtime etc

Kinds of errors

- Given two biometric samples we can have two possible hypothesis
  -Null hypothesis H0 => two samples are matching
- Alternative hypothesis H1 => two samples are not matching
- Definition of hypothesis depends on biometric applications.
- different application can have different definition of errors.
- But matching engine decides whether H0 is true or H1 is true.

Terminology

- A biometric verification system makes two types of errors:
  - (i) mistaking biometric measurements from two different persons to be from the same person (called *false match*),
  - (ii) mistaking two biometric measurements from the same person to be from two different persons (called *false non-match*).
- These two types of errors are often termed as *false accept* and *false reject*, respectively.

Terminology

- The frequency with which this False Match occurs is called False match rate **(FMR)** or False Acceptance rate **(FAR) or Type – I error.**
- The frequency with which this False Non Match occurs is called False non match rate **(FNMR)** or False rejection rate **(FAR) or Type – II error.**

Terminology

- The system performance at all the operating points (thresholds, *t*) can be depicted in the form of a ***Receiver Operating Characteristic*** **(ROC)** curve.
- A ROC curve is a plot of FMR against (1-FNMR) or FNMR for various threshold values, *t*

- Mathematically the errors in a verification system can be formulated as follows. If the stored biometric template of the user I is represented by XI and the acquired input for recognition is represented by XQ, then the null and alternate hypotheses are:

- *H*0: input *XQ* does not come from the same person as the template *XI*;
- *H*1: input *XQ* comes from the same person as the template *XI*.
- The associated decisions are as follows:
  - *D*0: person is not who she claims to be;
  - *D*1: person is who she claims to be.
- The decision rule is as follows: if the matching score $S(XQ, XI)$ is less than the system threshold t, then decide $D0$, else decide $D1$.

- **Type I:** false match ($D1$ is decided when $H0$ is true);
- **Type II:** false non-match ($D0$ is decided when $H1$ is true).
- FMR is the probability of type I error (also called significance level in hypothesis testing) and FNMR is the probability of type II error:

$$\text{FMR} = P(D_1 | H_0);$$
$$\text{FNMR} = P(D_0 | H_1).$$

- The expression (1-FNMR) is also called the power of the hypothesis test.

Match Score distribution

- If S is the similarity measure between two biometric samples, then we decide H0 is True if S > t (threshold), otherwise H1 is true.
- Decision becomes sometimes very hard as there is no scope to tell “do not know” or “too close”.
  - This is known as decision making without exception handling
- Reliability of the score is influenced by many factors :
  - Variations in sensors
  - Variations in data acquisition
  - Variations in data representation

Observations

- Even though two samples are from identical biometric, the similarity score is rarely 1, unless two samples are copies of each other. But the score is usually high (i.e) Match score or genuine score for identical individual tend to be high.
- Even though two samples are from different individuals, the score is not ZERO. But the score is usually low . (i.e) Non match score or imposter scores for two different individuals tend to be low.

Error rates

- To evaluate the accuracy of a fingerprint biometric system, one must collect scores generated from multiple images of the same finger (the distribution $p(S(XQ, XI)|H1)$), and scores generated from a number of images from different fingers (the distribution $p(S(XQ, XI)|H0)$)

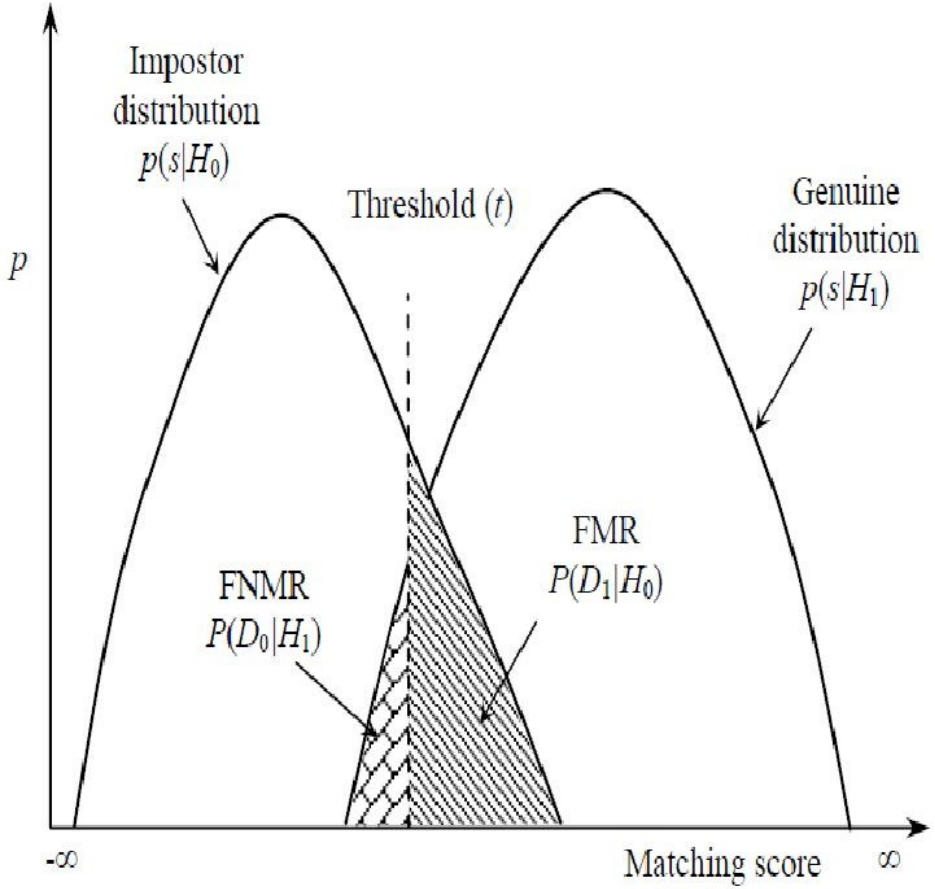
Error rates

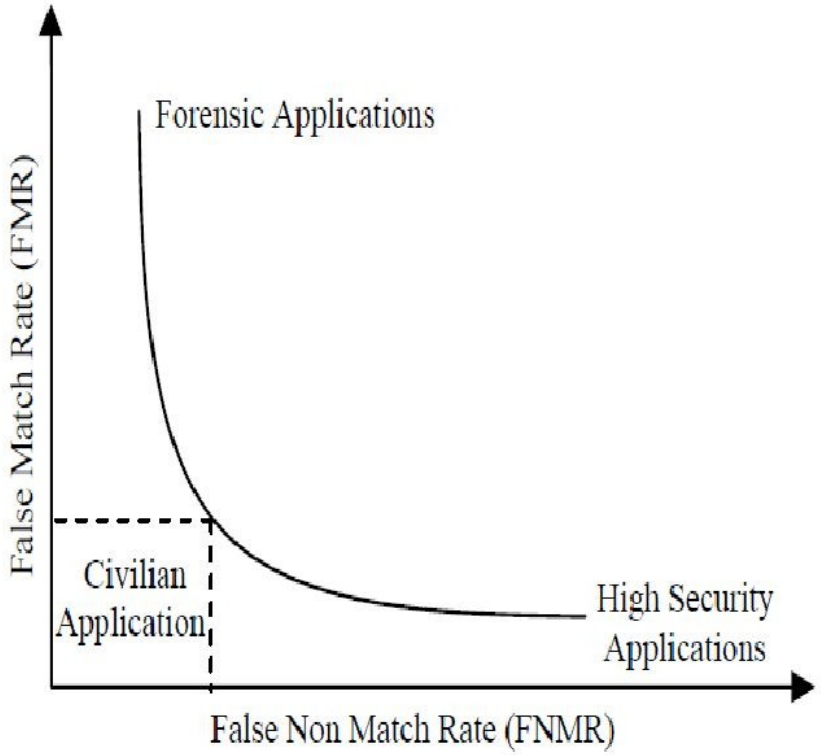
- For biometric application, Match score distribution and non match score distribution always overlap.
- It is not possible to select a t (threshold) such that FMR = 0 and FNMR = 0.
- it should be selected in a such a way that system operates in optimal fashion.
- FMR and FNMR are inversely related.
- There is a trade-off between false match **rate (FMR)** and **false non-match rate (FNMR)** in every biometric system.
- In fact, both FMR and FNMR are functions of the system threshold *t*; if *t* is decreased to make the system more tolerant to input variations and noise, then FMR increases.
- On the other hand, if *t* is raised to make the system more secure, then FNMR increases accordingly.

Receiver Operating Curve

- Suppose the integrals can be computed for any t.
- Then, FMR(t) and FNMR(t) give error rates when the match decision is made at some t
- These errors can be plotted against each other as a two dimensional curve
  - ROC(t) = (FMR(t),FNMR(t))
- That is, FMR and FNMR behaviour is expressed in terms of ROC curve

ROC curve and FAR/FRR curve



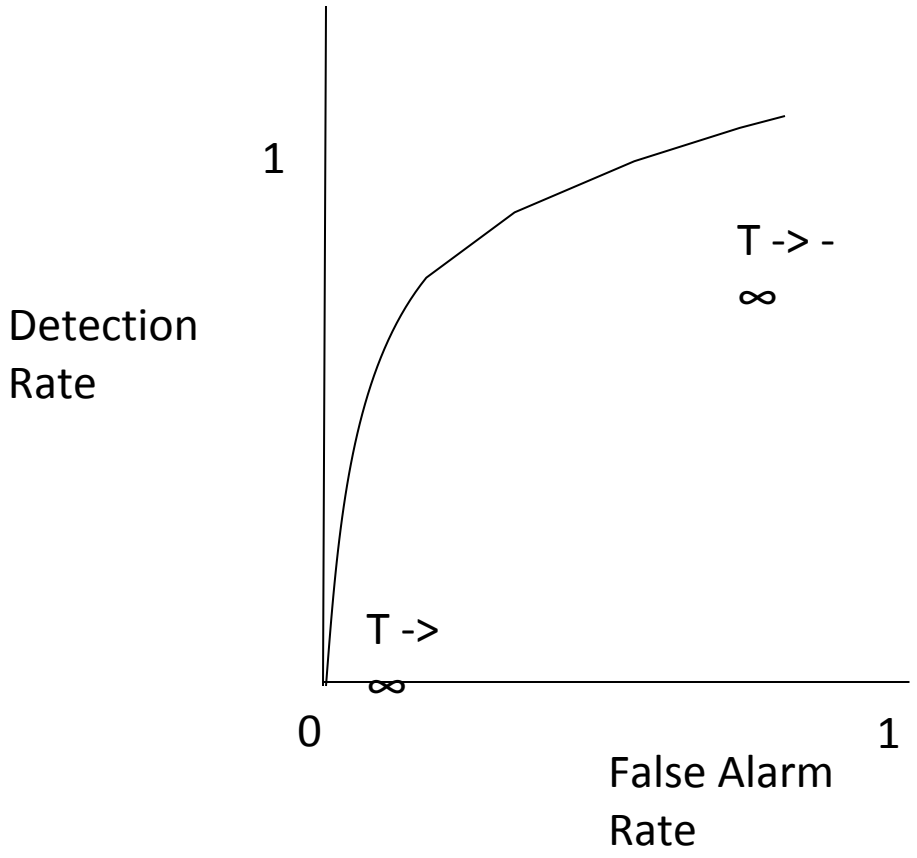


ROC

- When t is set low, FMR is high and FNMR is low; conversely when t is high FNMR is high and FMR is low
- Thus the matcher can be operated using any threshold t which defines a point on ROC. This is known as Operating Point of the matcher.
- The operating point can be specified by choosing any one of t, FMR and FNMR.

Variations of ROC

- There are number of variations of ROCs that are used expressing the information
- One way is already shown.
- Others can be obtained by plotting one or both the probabilities on a logarithmic scale.
- Sometimes one plots Correct Match Rate [i.e 1-FNMR] against FMR. This is called **Detection Error Trade-off (DET) curve.**
- Along y-axis we have [1-FNMR] which is correct detection rate and along x-axis we put FMR [False Alarm Rate]
- Detection rate goes to 1 when false alarm rate goes to 1..


1
Detection Rate
T -> -∞
T -> ∞
0
1
False Alarm Rate

Principal Component Analysis

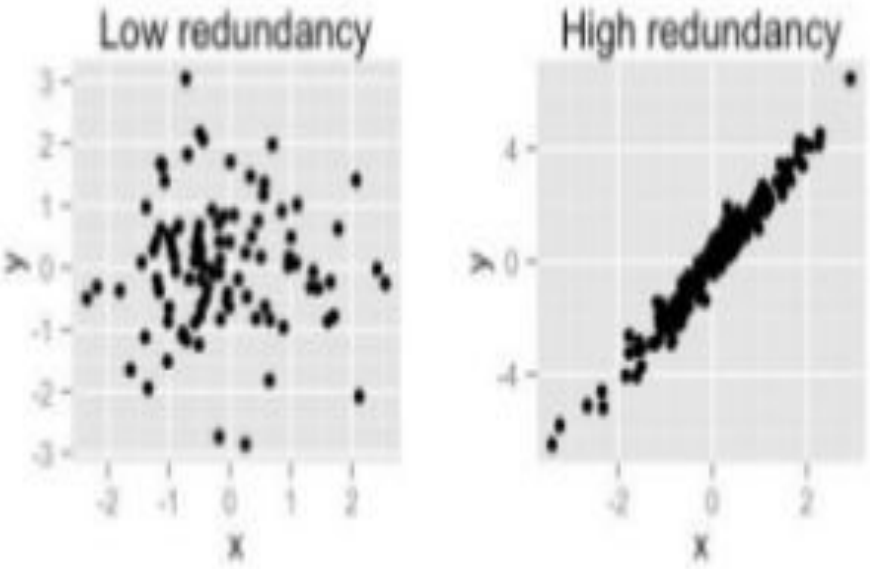
- A mathematical procedure with ***simple matrix operations*** from ***linear algebra and statistics*** to transform a number of *correlated variables into smaller number of uncorrelated variables* called principal components.
- Emphasize variation and bring out strong patterns in a dataset.
- To make data easy to ***explore and visualize***.
- Still contains most of the information in the large set.

Goals of PCA

- To *identify hidden pattern* in a data set
- To *reduce the dimensionality* of the data by removing the noise and redundancy in the data
- To identify *correlated variables*

Principal Component Analysis (contd...)

- PCA method is particularly useful when the variables within the data set are highly correlated.
- **Correlation** indicates that there is **redundancy** in the data.
- PCA can be used to reduce the original variables into a smaller number of new variables (= **principal components**) explaining most of the variance in the original variables.



Principal Component Analysis (contd...)

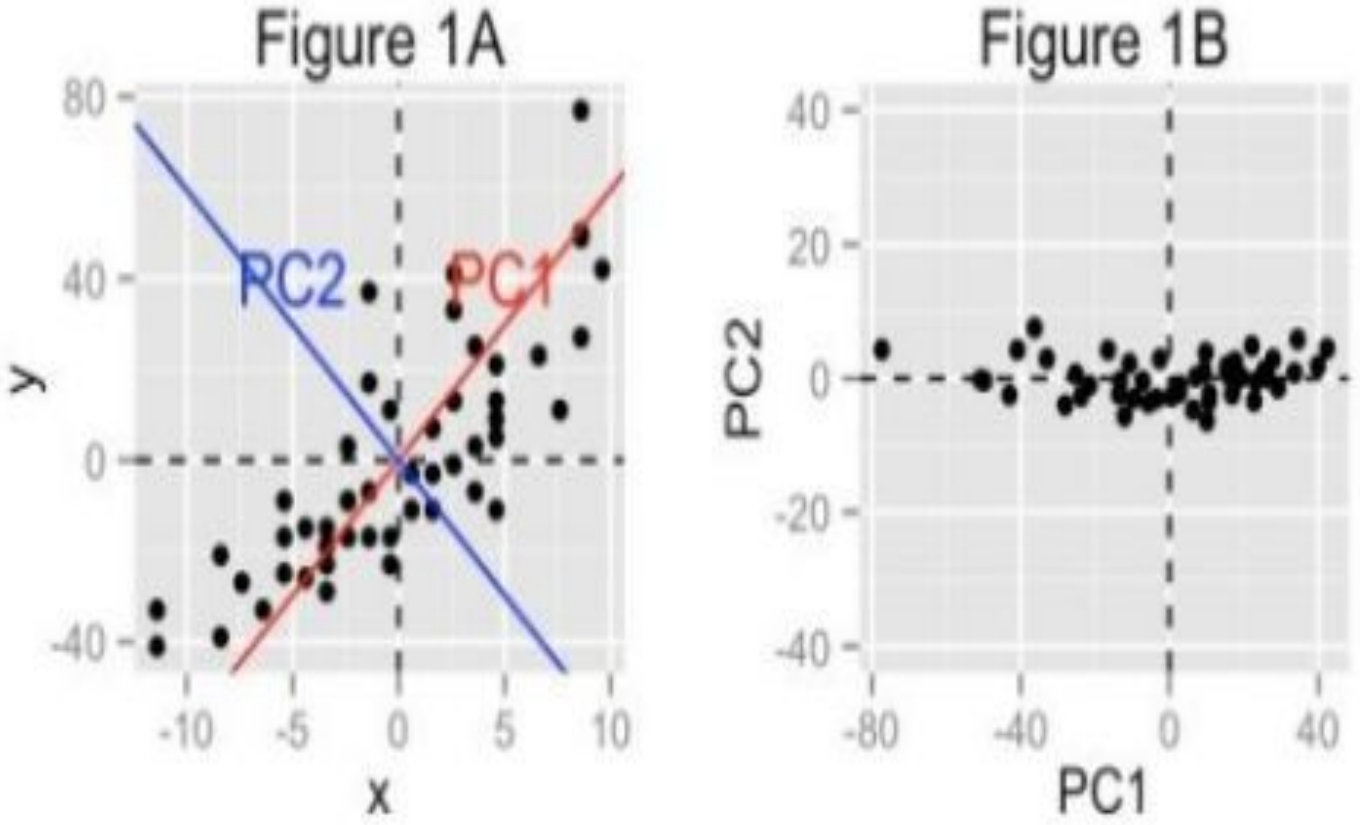


Figure 1A: The data are represented in the X-Y coordinate system

Figure 1B: The *PC1 axis* is the **first principal direction** along which the samples show the largest variation

- PCA finds a **new set of dimensions** (or a set of basis of views) such that all the dimensions are **orthogonal** (and hence linearly independent) and **ranked** according to the variance of data along them.
- It means **more important principle axis occurs first**.
- (more important = more variance/more spread out data)

How does PCA work

- Calculate the **covariance matrix X** of data points.
- Calculate **Eigen vectors** and corresponding **Eigen values.**
- **Sort** the Eigen vectors according to their Eigen values in **decreasing order**.
- Choose **first k Eigen vectors** and that will be the **new k dimensions**.
- **Transform** the original n dimensional data points into k dimensions.

The goal of PCA

- Find **linearly independent dimensions** (or basis of views) which can losslessly represent the data points.
- Those newly found dimensions should allow us to **predict/reconstruct the original dimensions**.
- The reconstruction/projection error should be minimized.

Steps in PCA

- **Standardize** the data.
- Compute the **covariance matrix** of the features from the dataset.
- Perform **Eigen Decompositon** on the covariance matrix.
- **Order** the eigenvectors in **decreasing** order based on the **magnitude** of their corresponding **eigenvalues**.
- **Determine k**, the **number of top principal components** to select.
- Construct the **projection matrix** from the chosen number of top principal components.
- Compute the **new k-dimensional feature space**.

1. Standardize the Dataset

- Assume we have the below dataset which has 4 features and a total of 5 training examples.
- Data standardization is about making sure that data is internally consistent; that is, each data type has the same content and format

f1	f2	f3	f4
1	2	3	4
5	5	6	7
1	4	2	3
5	3	2	1
8	1	2	2

- First, we need to standardize the dataset and for that, we need to calculate the mean and standard deviation for each feature.

$$x_{new} = \frac{x - \mu}{\sigma}$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

	f1	f2	f3	f4
μ =	4	3	3	3.4
σ =	3	1.58114	1.73205	2.30217

- After applying the formula for each feature in the dataset is transformed as below:

f1	f2	f3	f4
-1	-0.63246	0	0.26062
0.33333	1.26491	1.73205	1.56374
-1	0.63246	-0.57735	-0.17375
0.33333	0	-0.57735	-1.04249
1.33333	-1.26491	-0.57735	-0.60812

Calculate the covariance matrix for the whole dataset

For Population

$$Cov(x,y) = \frac{\sum (x_i - \bar{x}) * (y_i - \bar{y})}{N}$$

For Sample

$$Cov(x,y) = \frac{\sum (x_i - \bar{x}) * (y_i - \bar{y})}{(N-1)}$$

- the covariance matrix for the given dataset will be calculated as below

	f1	f2	f3	f4
f1	var(f1)	cov(f1,f2)	cov(f1,f3)	cov(f1,f4)
f2	cov(f2,f1)	var(f2)	cov(f2,f3)	cov(f2,f4)
f3	cov(f3,f1)	cov(f3,f2)	var(f3)	cov(f3,f4)
f4	cov(f4,f1)	cov(f4,f2)	cov(f4,f3)	var(f4)

- Since we have standardized the dataset, so the **mean for each feature is 0** and the standard deviation is 1.
- var(f1) = ((-1.0-0)² + (0.33-0)² + (-1.0-0)² +(0.33–0)² +(1.33–0)²)/5
  var (f1) = 0.8
- cov(f1,f2) =
  ((-1.0–0)*(-0.632456-0) +
  (0.33–0)*(1.264911-0) +
  (-1.0–0)* (0.632456-0)+
  (0.33–0)*(0.000000 -0)+
  (1.33–0)*(-1.264911–0))/5
  cov(f1,f2 = -0.25298

- In the similar way be can calculate the other covariances and which will result in the below covariance matrix

	f1	f2	f3	f4
f1	0.8	-0.25298	0.03849	-0.14479
f2	-0.25298	0.8	0.51121	0.4945
f3	0.03849	0.51121	0.8	0.75236
f4	-0.14479	0.4945	0.75236	0.8

Calculate eigenvalues and eigen vectors.

- An **eigenvector** is a nonzero vector that changes at most by a scalar factor when that linear transformation is applied to it.
- The corresponding **eigenvalue** is the factor by which the eigenvector is scaled.
- Let A be a square matrix (in our case the covariance matrix), v a vector and λ a scalar that satisfies $Av = \lambda v$, then λ is called eigenvalue associated with eigenvector v of A.
- Rearranging the above equation,

- $Av-\lambda v = 0$; $(A-\lambda I)v = 0$
- Since we have already know v is a non- zero vector, only way this equation can be equal to zero, if
- $\det(A-\lambda I) = 0$

	f1	f2	f3	f4
f1	0.8 - λ	-0.25298	0.03849	-0.14479
f2	-0.25298	0.8- λ	0.51121	0.4945
f3	0.03849	0.51121	0.8 - λ	0.75236
f4	-0.14479	0.4945	0.75236	0.8 - λ

- Solving the above equation = 0
- ***λ = 2.51579324 , 1.0652885 , 0.39388704 , 0.02503121***

Eigenvectors:

Solving the (A-λI)v = 0 equation for v vector with different λ values:

$$\begin{pmatrix} 0.800000-\lambda & -(0.252982) & 0.038490 & -(0.144791) \\ -(0.252982) & 0.800000-\lambda & 0.511208 & 0.494498 \\ 0.038490 & 0.511208 & 0.800000-\lambda & 0.752355 \\ -(0.144791) & 0.494498 & 0.752355 & 0.800000-\lambda \end{pmatrix} \times \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{pmatrix} = 0$$

Going by the same approach, we can calculate the eigen vectors for the other eigen values. We can from a matrix using the eigen vectors.

- For λ = *2.51579324, solving the above equation using Cramer's rule, the values for v vector are*

e1	e2	e3	e4
0.161960	-0.917059	-0.307071	0.196162
-0.524048	0.206922	-0.817319	0.120610
-0.585896	-0.320539	0.188250	-0.720099
-0.596547	-0.115935	0.449733	0.654547

- *v1 = 0.16195986*
  v2 = -0.52404813
  v3 = -0.58589647
  v4 = -0.59654663
- eigenvectors(4 * 4 matrix)

***4.* Sort eigenvalues and their corresponding eigenvectors.**

Since eigenvalues are already sorted in this case so no need to sort them again.

***5.* Pick k eigenvalues and form a matrix of eigenvectors**

f we choose the top 2 eigenvectors, the matrix will look like this:

Top 2 eigenvectors(4*2 matrix)

e1	e2
0.161960	-0.917059
-0.524048	0.206922
-0.585896	-0.320539
-0.596547	-0.115935

Feature matrix * top k eigenvectors = Transformed Data

- **6. Transform the original matrix.**
- Feature matrix * top k eigenvectors
  = Transformed Data

f1	f2	f3	f4
-1.000000	-0.632456	0.000000	0.260623
0.333333	1.264911	1.732051	1.563740
-1.000000	0.632456	-0.577350	-0.173749
0.333333	0.000000	-0.577350	-1.042493
1.333333	-1.264911	-0.577350	-0.608121

(5,4)

*

e1	e2
0.161960	-0.917059
-0.524048	0.206922
-0.585896	-0.320539
-0.596547	-0.115935

(4,2)

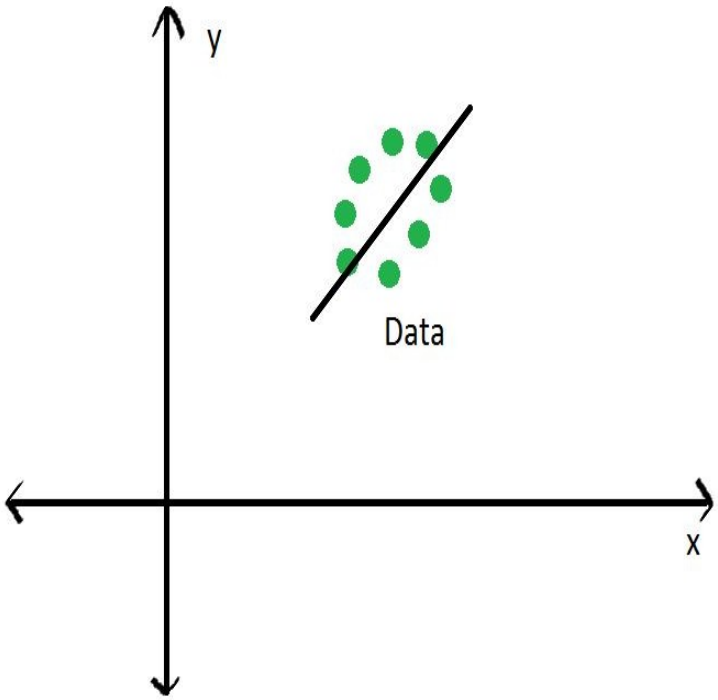
=

nf1	nf2
0.014003	0.755975
-2.556534	-0.780432
-0.051480	1.253135
1.014150	0.000239
1.579861	-1.228917

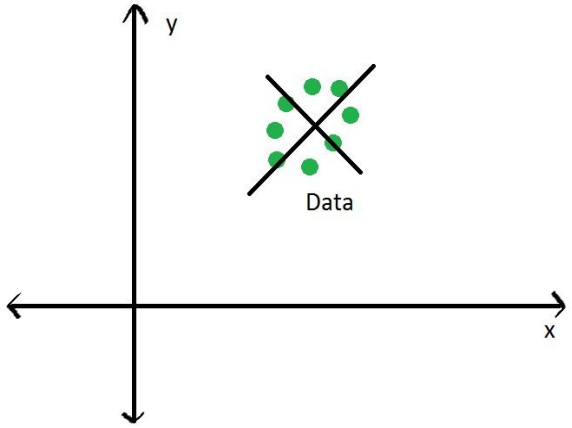
(5,2)

Eigenvectors and eigenvalues

- The **eigenvectors** and **eigenvalues** of a covariance (or correlation) matrix represent the "core" of a **PCA**: The **eigenvectors** (principal components) determine the directions of the new feature space, and the **eigenvalues** determine their magnitude.
- **Deduce the Eigen:**
- Suppose we have plotted a *scatter plot* of random variables, and a line of best fit is drawn between these points.
- This ***line of best fit***, shows the direction of maximum variance in the dataset.
- The Eigenvector is the direction of that line, while the eigenvalue is a number that tells us how the data set is spread out on the line which is an Eigenvector


y
Data
x

- The main principal component, depicted by the black line, is the first Eigenvector. The second Eigenvector will be p**erpendicular or orthogonal** to the first one. The reason the two Eigenvectors are orthogonal to each other is because the Eigenvectors should be able to span the whole x-y area. Naturally, a line perpendicular to the black line will be our new Y axis, the other principal component.



- We are going to *rotate* our data to fit these new axes. But what will the coordinates of the rotated data be?
- To convert the data into the new axes, we will multiply the original X, Y data by *Eigenvectors*, which indicate the direction of the new axes (principal components).
- But first, we need to deduce the Eigenvectors (there are two — one per axis). Each Eigenvector will correspond to an E*igenvalue*, whose magnitude indicates how much of the data's variability is explained by its Eigenvector.

•From the definition of Eigenvalue and Eigenvector:

[*Covariance matrix*].[*Eigenvector*] =
[*Eigenvalue*].[*Eigenvector*]

Steps to implement PCA in 2D dataset

Step 1: **Normalize** the data

Step 2: Calculate the **covariance matrix**

Step 3: Calculate the **eigenvalues and eigenvectors**

Step 4: **Choosing** principal components

Step 5: Forming a **feature vector**

Step 6: **Forming Principal Components**

Step 1: Normalization

This is done by subtracting the respective means from the numbers in the respective column. So if we have two dimensions X and Y, all X become x- and all Y become y-.

For all X; $x\text{-} = X - \mu_x$

For all Y; $y\text{-} = Y - \mu_y$

This produces a dataset whose mean is zero.

Step 2: Calculation of correlation

$$\text{Matrix (covariance)} = \begin{bmatrix} var(x) & var(x,y) \\ var(y,x) & var(y) \end{bmatrix}$$

Covariance Matrix for Iris Dataset

	Sepal.Length	*Sepal.Width*	*Petal.Length*	*Petal.Width*
Sepal.Length	0.69	-0.04	1.27	0.52
Sepal.Width	-0.04	0.19	-0.33	-0.12
Petal.Length	1.27	-0.33	3.12	1.30
Petal.Width	0.52	-0.12	1.30	0.58

Calculation of correlation(contd...)

If **x** and **y** be two variables with length n,

The variance of **x** , variance of **y** and variance of **x & y** is given by following equations.

$$\sigma_{xx}^2 = \frac{\sum_i (x_i - m_x)(x_i - m_x)}{n-1}$$

$$\sigma_{yy}^2 = \frac{\sum_i (y_i - m_y)(y_i - m_y)}{n-1}$$

$$\sigma_{xy}^2 = \frac{\sum_i (x_i - m_x)(y_i - m_y)}{n-1}$$

m_x : mean of **x** variables

m_y : mean of **y** variables

Calculation of correlation (contd...)

Correlation is the index to measure how strongly two variable are related to each other. The value of the same ranges for ***-1*** to ***+1***. (i.e. $-1 < r < 1$)

If $r<0$, variables are ***negatively correlated*** (e.g. x increases when y increases)

If $r>0$, variables are ***positively correlated*** (e.g. x increases when y decreases)

If $r=0$, variables has ***no correlation***

Step 3: Calculation of Eigenvalue and Eigenvector

Calculate Eigenvalue and Eigenvector of the covariance matrix using power method:

$$|\lambda - A| = 0$$

Where, I is an identity matrix of same dimension as A
λ is eigenvalue.

For each value of λ, corresponding eigenvector 'v' is obtained by solving:

$$(\lambda - A)v = 0$$

Step 4: Choosing Component

- Eigenvalues from largest to smallest so that it gives us the components in order or significance.
- If we have a dataset with *n* variables, then we have the corresponding *n* eigenvalues and eigenvectors.
- Eigenvector (v) corresponding to largest eigenvalue (λ) is called first principal component.
- To reduce the dimensions, we choose the first *p* eigenvalues and ignore the rest.

Step 5: Forming Principal Components

$$\text{NewData} = \text{FeatureVector}^T \times \text{ScaledData}^T$$

NewData is the Matrix consisting of the principal components,

FeatureVector is the matrix we formed using the eigenvectors we chose to keep, and

ScaledData is the scaled version of original dataset

Linear Discriminant Analysis

Introduction

- Linear Discriminant Analysis or LDA is a dimensionality reduction technique.
- It is used as a pre-processing step in Machine Learning and applications of pattern classification.
- The goal of LDA is to project the features in higher dimensional space onto a lower-dimensional space in order to avoid the curse of dimensionality and also reduce resources and dimensional costs.

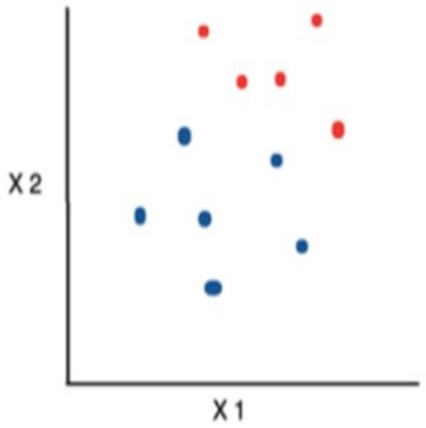
- Both LDA and PCA are linear transformation techniques:
- **LDA is a supervised whereas PCA is unsupervised** – PCA ignores class labels.
- In contrast to PCA, LDA attempts to find a feature subspace that maximizes class separability

Dimensionality Reduction

- The techniques of dimensionality reduction are important in applications of Machine Learning, Data Mining, Bioinformatics, and Information Retrieval.
- The main agenda is to remove the redundant and dependent features by changing the dataset onto a lower-dimensional space.
- In simple terms, they reduce the dimensions (i.e. variables) in a particular dataset while retaining most of the data.

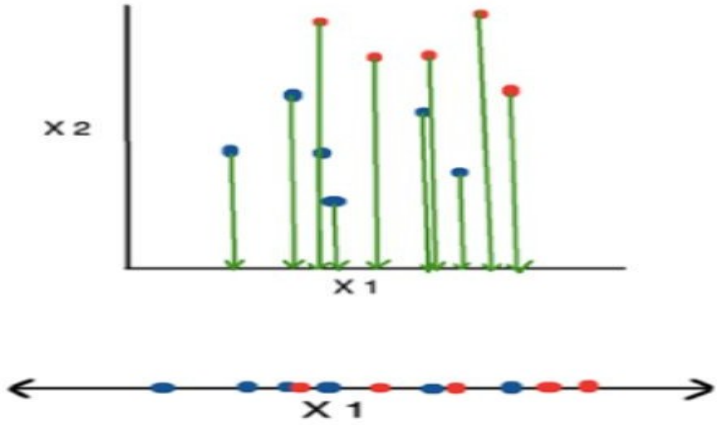
Practical approach to an LDA model

- Consider a situation where you have plotted the relationship between two variables where each color represents a different class. One is shown with a red color and the other with blue



Practical approach to an LDA model

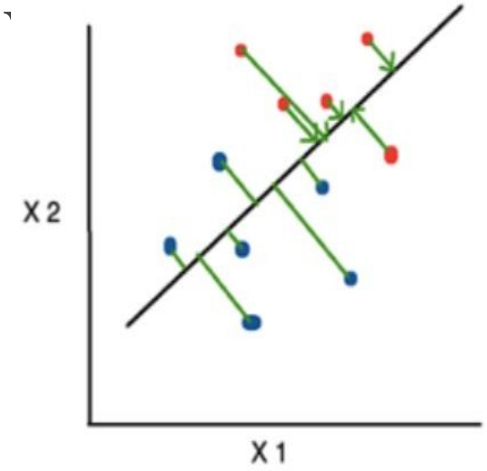
- If you are willing to reduce the number of dimensions to 1, you can just project everything to the x-axis as shown below:

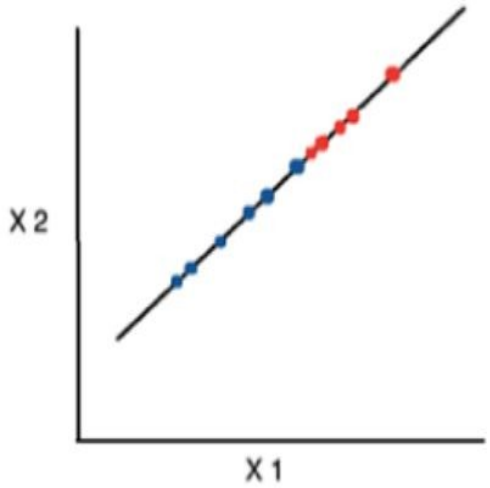


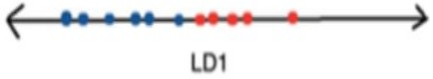
- This approach neglects any helpful information provided by the second feature. However, you can use LDA to plot it.

Practical approach to an LDA model

- The advantage of LDA is that it uses information from both the features to create a new axis which in turn minimizes the variance and maximizes the class distance of the two







How it works

- LDA focuses primarily on projecting the features in higher dimension space to lower dimensions. You can achieve this in three steps:
- Firstly, you need to calculate the separability between classes which is the distance between the mean of different classes. This is called the *between-class variance*.

$$S_b = \sum_{i=1}^{g} N_i (\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})^T$$

- Secondly, calculate the distance between the mean and sample of each class. It is also called the within-class variance.

$$S_w = \sum_{i=1}^{g} (N_i - 1) S_i = \sum_{i=1}^{g} \sum_{j=1}^{N_i} (x_{i,j} - \bar{x}_i)(x_{i,j} - \bar{x}_i)^T$$

How it works

- Finally, construct the lower-dimensional space which maximizes the between-class variance and minimizes the within-class variance. P is considered as the lower-space projection, also ca criterion.

$$P_{lda} = \arg\max_{P} \frac{\left| P^T S_b P \right|}{\left| P^T S_w P \right|}$$

-

How does an LDA model make predictions?

- LDA models use **Bayes' Theorem** to estimate probabilities.
- They make predictions based upon the probability that a new input dataset belongs to each class.
- The class which has the highest probability is considered the output class and then the LDA makes a prediction.
- The prediction is made simply by the use of Bayes' Theorem which estimates the probability of the output class given the input.

How does an LDA model make predictions?

They also make use of the probability of each class and the probability of the data belonging to each class:

P(Y=x|X=x) = **[**(Plk * fk(x))**]** / **[**(sum(PlI * fl(x))**]** Where x = input, k = output class.

Plk = Nk/n or base probability of each class observed in the training data.

It is also called prior probability in Bayes' Theorem. fk(x) = estimated probability of x belonging to class k

How does an LDA model make predictions?

- The f(x) is plotted using a Gaussian Distribution function and then it is plugged into the equation above and the result we get is the equation as follows:
- $Dk(x) = x*(mean/\Sigma^2) - (mean^2/(2*\Sigma^2)) + ln(PIk)$
- The Dk(x) is called the discriminant function for class k given input x, mean, Σ^2 and Plk are all estimated from the data and the class is calculated as having the largest value, will be considered in the output classification.

What is decision theory?

- Decision theory is theory about decisions.
- The subject is not a very unified one.
- To the contrary, there are many different ways to theorize about decisions, and therefore also many different research traditions.
- This text attempts to reflect some of the diversity of the subject. Its emphasis lies on the less (mathematically) technical aspects of decision theory.

- **Decision theory** (or the **theory of choice** not to be confused with choice theory) is the study of an agent's choices.
- Decision theory can be broken into two branches:
- normative
- descriptive

Two types

- normative decision theory, which analyzes the outcomes of decisions or determines the optimal decisions given constraints and assumptions
- descriptive decision theory, which analyzes *how* agents actually make the decisions they do.

PCA Problem 1

Q: 1 Given the following data, use PCA to reduce the dimension from 2 to 1

Feature	Example 1	Example 2	Example 3	Example 4
x	4	8	13	7
y	11	4	5	14

Step 1 : Dataset

Feature	Ex. 1	Ex. 2	Ex. 3	Ex. 4
x	4	8	13	7
y	11	4	5	14

No. of features, $n = 2$

No of samples, $N = 4$

Step 2: Computation of mean of variables.

$$\bar{x} = \frac{4+8+13+7}{4} = 8$$

$$\bar{y} = \frac{11+4+5+14}{4} = 8.5$$

(x,x), (x,y) (y,x), (y,y)

n

n^2

i) Covariance of all ordered pairs

$$Cov(x_i,x) = \frac{1}{N-1}\sum_{k=1}^{N}(x_{ik}-\bar{x}_i)(x_{jk}-\bar{x}_j)$$

$$cov(x,x) = \frac{1}{N-1}\sum_{k=1}^{N}(x_i-\bar{x})^2$$

$$= \frac{1}{4-1}\left[(4-8)^2+(8-8)^2+(13-8)^2+(7-8)^2\right]$$

$$= 14$$

$$Cov(x,y) = \frac{1}{4-1}\Big[(4-8)(11-8.5)+(8-8)(4-8.5)$$
$$+(12-8)(5-8.5)+$$
$$(7-8)(14-8.5)\Big]$$

$$= -11$$

$$Cov(y,x) = Cov(x,y) = -11$$

$$Cov(y,y) = \frac{1}{4-1}\Big[(11-8.5)^2+(4-8.5)^2+$$
$$(5-8.5)^2+(14-8.5)^2\Big]$$

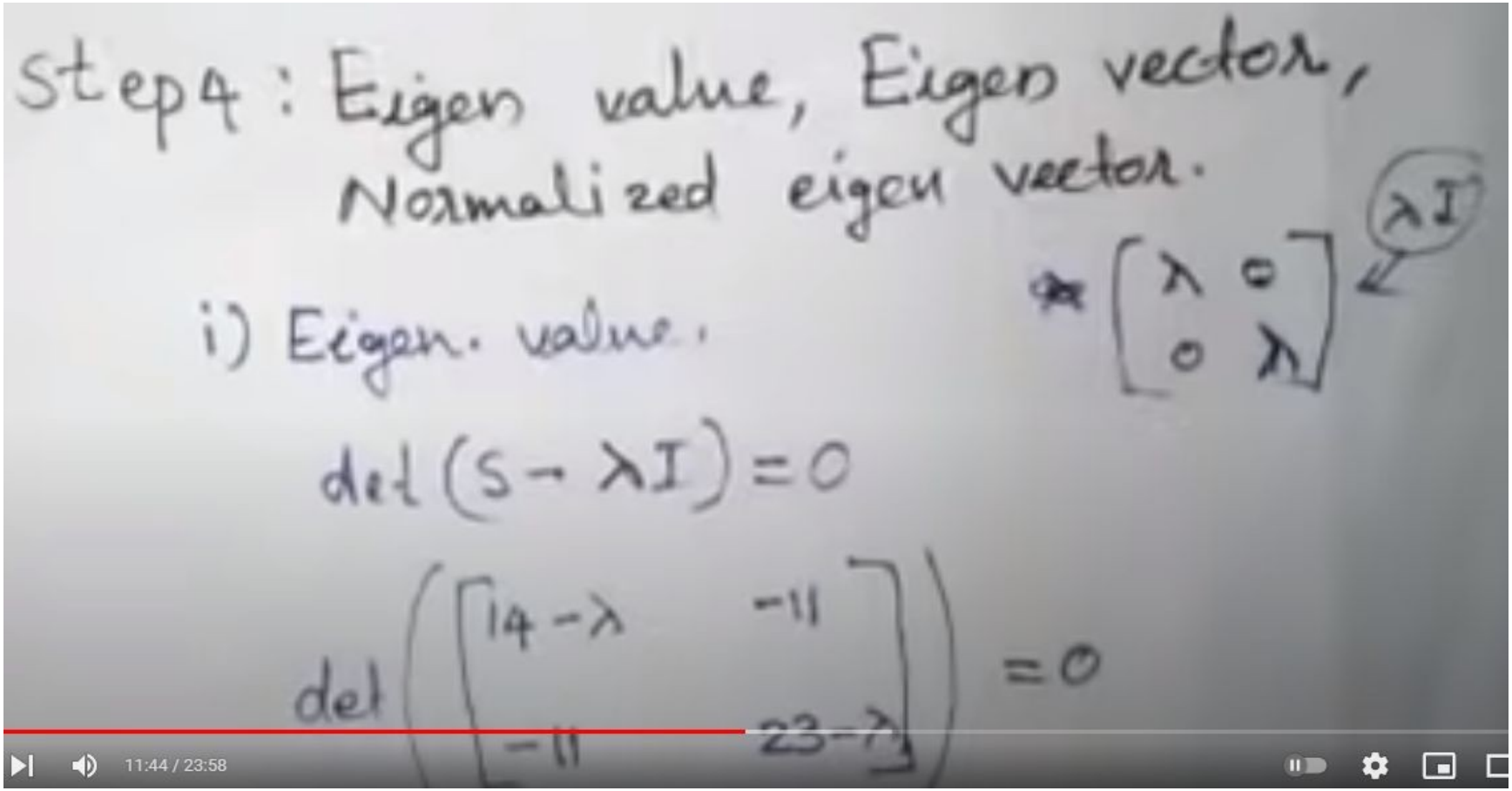
$$= 23$$

$n \times n$

2×2

$$S = \begin{bmatrix} \text{cov}(x,x) & \text{cov}(x,y) \\ \text{cov}(y,x) & \text{cov}(y,y) \end{bmatrix}$$

$$= \begin{bmatrix} 14 & -11 \\ -11 & 23 \end{bmatrix}$$


Step 4 : Eigen value, Eigen vector,
Normalized eigen vector.
i) Eigen. value.
$\lambda I = \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}$
$\det(S - \lambda I) = 0$
$\det\left(\begin{bmatrix} 14-\lambda & -11 \\ -11 & 23-\lambda \end{bmatrix}\right) = 0$
11:44 / 23:58

$$\det\left(\begin{bmatrix} 14-\lambda & -11 \\ -11 & 23-\lambda \end{bmatrix}\right) = 0$$

$$(14-\lambda)(23-\lambda) - (-11 \times -11) = 0$$

$$\lambda^2 - 37\lambda + 201 = 0$$

$$\lambda = 30.3849,\ 6.6151$$

$$\lambda^2 - 37\lambda + 201 = 0$$

$$\lambda = 30.3849,\ 6.6151$$

$$\lambda_1 > \lambda_2$$

$$\lambda_1 = 30.3849$$

$$\lambda_2 = 6.6151$$

2×1

$U_1 =$

ii) Eigen vector of λ_1

$$(S - \lambda_1 I) U_1 = 0$$

$$\begin{bmatrix} 14-\lambda_1 & -11 \\ -11 & 23-\lambda_1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = 0$$

$$\begin{bmatrix} (14-\lambda_1)u_1 - 11u_2 \\ -11u_1 + (23-\lambda_1)u_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$(14-\lambda_1)u_1 - 11u_2 = 0$ ✓

$-11u_1 + (23-\lambda)u_2 = 0$

$$\frac{u_1}{11} = \frac{u_2}{14-\lambda_1} = t$$

When $t = 1$

$u_1 = 11$

$u_2 = 14$

$(14-\lambda_1)u_1 - 11u_2 = 0$ ✓

$-11u_1 + (23-\lambda_1)u_2 = 0$

$$\frac{u_1}{11} = \frac{u_2}{14-\lambda_1} = t$$

When $t = 1$

$u_1 = 11$

$u_2 = 14$

λ_2 11 $14-\lambda_1$

When t=1

$u_1 = 11$

$u_2 = 14 - \lambda_1$

Eigen vector U_1 of $\lambda_1 = \begin{bmatrix} 11 \\ 14-\lambda_1 \end{bmatrix}$

$= \begin{bmatrix} 11 \\ 14 - 30.3841 \end{bmatrix} = \begin{bmatrix} 11 \\ -16.3841 \end{bmatrix}$

iii) Normalize the eigen vector U_1

$$e_1 = \begin{bmatrix} \frac{11}{\sqrt{11^2+(16.38)^2}} \\ \frac{-16.3849}{\sqrt{11^2+(-16.11)^2}} \end{bmatrix}$$

$$= \begin{bmatrix} 0.5574 \\ -0.8303 \end{bmatrix}$$

$$e_2 = \begin{bmatrix} 0.830 \\ 0.557 \end{bmatrix}$$

Step 5: Derive new dataset

Step 5: Derive new dataset

	Ex.1	Ex.2	Ex.3	Ex.4
First principal pc$_1$ component	P_{11}	P_{12}	P_{13}	P_{14}

$$P_{11} = e_1^T \begin{bmatrix} 4-8 \\ 11-8.5 \end{bmatrix}$$

$$P_{11} = e_1^T \begin{bmatrix} 4-8 \\ 11-8.5 \end{bmatrix}$$

$$= \begin{bmatrix} 0.5574 & -0.8303 \end{bmatrix} \begin{bmatrix} -4 \\ 2.5 \end{bmatrix}$$

$$= -4.3052$$

$$P_{13} = 5.6928$$

$$P_{14} = -5.1238$$

	Eigen 1	Eigen 2	Eigen 3	Eigen 4
PC1	-4.3052	3.7361	5.6928	-5.1238