Aston University

Some pages of this thesis may have been removed for copyright restrictions.

If you have discovered material in AURA which is unlawful e.g. breaches copyright, (either yours or that of a third party) or any other law, including but not limited to those relating to patent, trademark, confidentiality, data protection, obscenity, defamation, libel, then please read our Takedown Policy and contact the service immediately

ASTON UNIVERSITY

High Level Data Fusion

MARK D. BEDWORTH

Doctor of Philosophy

ASTON UNIVERSITY

April 1999

This copy of the thesis has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with its author and that no quotation from the thesis and no information derived from it may be published without proper acknowledgement.

ASTON UNIVERSITY

High Level Data Fusion

MARK D. BEDWORTH

Doctor of Philosophy, 1999

Thesis Summary

We address the question of how to obtain effective fusion of identification information such that it is robust to the quality of this information. As well as technical issues data fusion is encumbered with a collection of (potentially confusing) practical considerations. These considerations are described during the early chapters in which a framework for data fusion is developed. Following this process of diversification it becomes clear that the original question is not well posed and requires more precise specification. We use the framework to focus on some of the technical issues relevant to the question being addressed. We show that fusion of hard decisions through use of an adaptive version of the *maximum a posteriori* decision rule yields acceptable performance. Better performance is possible using probability level fusion as long as the probabilities are accurate. Of particular interest is the prevalence of overconfidence and the effect it has on fused performance. The production of accurate probabilities from poor quality data forms the latter part of the thesis. Two approaches are taken. Firstly the probabilities may be moderated at source (either analytically or numerically). Secondly, the probabilities may be transformed at the fusion centre. In each case an improvement in fused performance is demonstrated. We therefore conclude that in order to obtain robust fusion care should be taken to model the probabilities accurately; either at the source or centrally.

Keywords: Data Fusion, information quality, decision fusion, probability fusion, moderation.

To Ruth, Alice, Emily and Sam

Acknowledgements

The author would like to thank Professor David Lowe for having the faith to believe I would ever finish this work despite a continuing lack of written evidence. I would also like to thank him for his technical input in formulating the scope of this work and in the constructive comments that made the thesis what it is. I would further like to express thanks to Dr. West of DERA for funding the PhD work. I would also like to acknowledge the technical discussions I had with Anthony Heading during the first 18 months of the work on moderation, something that is central to the later chapters. During the last year, the useful discussions I had with Jane O'Brien helped greatly in structuring what appeared to be a *potpourri* of ideas.

The work presented in this thesis is my own. I would particularly like to thank several of my former colleagues who worked with me on certain elements of this work. Thanks to John Black, Keith Copsey and Phil Garner, all of who assisted in implementing software and providing some of the simulation results.

M Bedworth

Mark Bedworth, April 1999.

Contents

List of Figures

List of Tables

Preface

We begin by motivating the work and describing the organisation of the thesis. In data fusion systems multiple sources of data are combined (or fused) so that more information is available for the identification process than if a single sensor were employed. When the sensors are physically separated some communication is necessary and bandwidth constraints must be considered. This usually leads to a distributed or de-centralised architecture in which each sensor performs some processing before communicating its results to a fusion centre. The reliability of the information supplied to the fusion centre will often vary from one sensor to the next. Such variation in information quality has an effect on the way the sensors process their data, the way that the data is fused and the final performance of the fusion system itself. We therefore concentrate on the following question:

> *"How does one obtain effective fusion which is robust to the quality of the identification information being fused?"*

The issue has been pursued using a strategy of diversification followed by focussing.

In addition to the technical issues, data fusion is also encumbered with a collection of practical considerations. Despite its roots in the US military research of the 1980's there is no broad review of these considerations nor is there a standard data fusion framework. Part 1 (the diversification) provides a summary of this much-needed activity and establishes a framework for regarding data fusion systems. This part of the thesis provides an introduction to data fusion (Chapter 1) and brings together the main approaches (Chapter 2) and architectures (Chapter 3). This provides the context of the multi-layered, centralised, empirical approach adopted in the remainder of the thesis. Part 1 is a qualitative treatment of data fusion and comprises structured ideas rather than mathematics. It concludes with a reformulation of the original question as it applies to the fusion of hard and soft decisions.

In contrast, part 2 is devoted to the development of specific algorithms that deal with the quality of the identity information being fused (the focussing). In Chapter 4 we describe several decision-level fusion algorithms and analyse the effects of sensor reliability has on them (the so-called veto effect). The inability for decision fusion to adequately incorporate source reliability leads us to adopt an approach based on probabilities. The veto effect is re-addressed from the perspective of probability fusion. The development of methods to handle this problem forms the basis of the next few chapters. Chapter 5 motivates the use of probability fusion and assesses the effects of overconfidence. Chapter 6 and Chapter 7 deal with algorithms to moderate probabilities at source (first analytically and then heuristically). In Chapter 8 we develop methods for performing the moderation at the fusion centre.

We summarise the work with conclusions highlighting the main findings. A list of references is then provided. Appendix A contains the derivations of the key results presented in the main text and a description of the datasets used for evaluation is given in Appendix B.

Part 1:

A Data Fusion Framework

Chapter 1 Overview of Data Fusion

Data fusion is the technology for combining information from multiple sources that are separated spatially or temporally. It enables the user to make inferences about the data based on a richer variety of sources than would be possible using more conventional techniques applied to the individual sources. Data fusion is closely related to pattern processing and uses similar methods. The main points of difference between the two fields are outlined in Chapter 3 . The Joint Directors of Laboratories Data Fusion Group (JDL DFG) was formed in the US in the early 1980's to examine the then infant field of data fusion. Representatives from most of the US Government military research and development laboratories constructed a definition of data fusion thus [77]:

> *"Data fusion is a multi-level, multi-faceted process dealing with the automatic detection, association, correlation and combination of data and information from multiple sources".*

Modern intelligence, surveillance and reconnaissance (ISR) systems generally include large numbers of electronic sensor devices feeding into highly automated data processing and data communication networks. In addition collateral sources such as human intelligence (HUMINT), open source information (OSCINT) and encyclopaedic data may need to be incorporated (although we shall largely ignore the processing of this type of data in this thesis). The military requirement is to collate and combine this plethora of data in such a way that a condensed report may be presented to a commander so that he may make an informed and effective decision. The military intelligence process recognises many forms of information. The main source categories recognised in the UK intelligence cycle are shown in Table 1-1.

In order that the highest performance may be obtained from an automatic sensing system, the full range of signature-generating phenomena should be examined. Several sensors may be required to obtain adequate performance since the sensors that measure such phenomena are somewhat restricted in their perceptual coverage.

Category	Expansion	Description
IMINT	Image Intelligence	Visible or IR images
ACINT	Acoustic Intelligence	Sound or seismic waveforms
COMINT	Communications Intelligence	Communication emissions
SIGINT	Signals Intelligence	Non-communication emissions
ELINT	Electronic Intelligence	SIGINT + COMINT
RADINT	Radar Intelligence	Radar emissions
LASINT	Laser Intelligence	Laser emissions
OSCINT	Open Source Intelligence	News feeds or media broadcasts
HUMINT	Human Intelligence	Verbal or written reports from people

Table 1-1: The UK categories of intelligence sources.

The principal phenomena and sensor types are:

- Electromagnetic energy
  - ❑ Magnetic
  - ❑ Radar
  - ❑ Infrared
  - ❑ Visible
  - ❑ Ultraviolet
- Mechanical energy
  - ❑ Seismographic
  - ❑ Acoustic
  - ❑ Ultrasonic
- Chemical particles
- Nuclear particles

Further subdivisions may be made according to whether the sensor operates in passive or active mode, its frequency, bandwidth, polarisation and resolution. The absolute and relative locations and motions of the sensor platform(s) and the target(s) also affect the collection of the sensed signature, as do environmental variables such as weather, clutter and countermeasures [98].

Data fusion is widely applicable to modern systems with sensors, sources or databases of information. Over the last few years a data fusion community has developed with an increasing amount of international collaboration [25]. For political reasons the early funding of data fusion came from government sources and many of the initial applications were for military tasks. Such tasks include non co-operative target recognition (NCTR), multi-sensor multi-target tracking, situation assessment, battlefield surveillance, reconnaissance analysis and asset tasking [77]. Applications to non-defence remote sensing tasks and drug interdiction were next to be analysed using data fusion techniques. More recently, the developments in data fusion technology are beginning to find application in the commercial world, most notably in the aerospace, manufacturing and condition monitoring fields. Use of data fusion in such application areas may lead to increased robustness to jamming or source failure and extended spatial or temporal coverage. It may also increase confidence in decisions (resulting from less ambiguous or confirmed information), increased the detection probability and reduce the operator workload. We shall use the term fusion cell to denote a system sub-component where data is combined and data fusion occurs.

Chapter 2 Data Fusion Approaches

2.1 *Chapter Introduction*

The information about how to process and fuse data can come either from expertise or from data. We shall separate the use of experts into three approaches that relate to scientific knowledge of physical processes, expert opinion or the use the human expert directly to manually process the data during operation. There are therefore four possible approaches to designing fusion cells:

1. **Physical** modelling of the input processes
2. **Empirical** modelling of the input processes
3. **Expert** encapsulations of human knowledge of the interdependencies in the input information
4. **Human**-in-the-loop (HITL) analysis of the incoming data

In the following sections we shall examine examples of these four approaches to the design of fusion cells and select one of them for more detailed analysis in the remainder of the thesis.

2.2 *Physical Modelling*

Physical modelling is only possible when the processes governing the production of the separate streams of data are well (or at least well enough) understood. An example of physical modelling for data fusion is that of pixel-level fusion of infrared and visible light imagery as described in [130], [30], [131] and [31]. In this application a model of the heat flux at the surface of objects observed in the scene was developed and used to characterise the surface material of a scene. The resulting model used the visible light imagery to estimate the amount of energy incident on the patch of terrain corresponding to a single pixel in the image. This process itself may be regarded as a form of data fusion since collateral data (sensor orientation and position, local time of day and latitude of scene) are used to calculate the amount of sunlight expected to be available. The infrared image is used to estimate the temperature of the surface corresponding to the same patch of terrain. The two items of data together provide sufficient information to calculate a parameter, which largely depends on the material properties of the surface under scrutiny. An illustration of this technique applied to two synthetic images is shown in Figure 2-1. On the left is a synthetic downwards looking infrared image. The scene comprises an area of sloping terrain, a road network and four pent-roofed buildings. In the centre is the same field of view imaged in the visible spectrum. The third image (on the right) shows the output of the fusion system, which has assigned a material coding to each part of the scene.

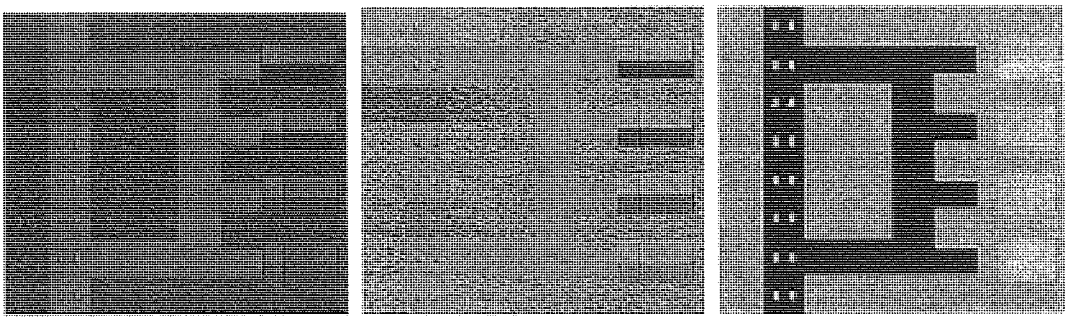

Figure 2-1: Physical modelling for image data fusion.

In more detail, the method uses infrared and visible-light images together with estimates of the wind speed, air temperature and the position of the sun to work out a parameter R. This parameter is defined as:

$$R = \frac{W_{bal}}{W_{abs}}$$

where the quantity W_{abs} is the absorbed solar heat flux, and W_{bal} is the heat flux lost from the surface by conduction, photosynthesis or transpiration. The value of R is found to be indicative of the localised material type of the surface under scrutiny. In the model it is assumed that the surface is composed of facets which correspond to the individual pixels in the infrared and visible-light images. Furthermore, it is assumed that each facet is thermally isolated from its neighbours. The model for the net heat flow at the surface of the facet is then given by:

$$W_{abs} = W_{cv} + W_{rad} + W_{bal}$$

Where W_{abs} is the absorbed heat flux, W_{cv} is the heat flux lost by the surface by convection and W_{bal} is balance the heat flux lost from the surface. See Appendix A.1. This is shown schematically in Figure 2-2.

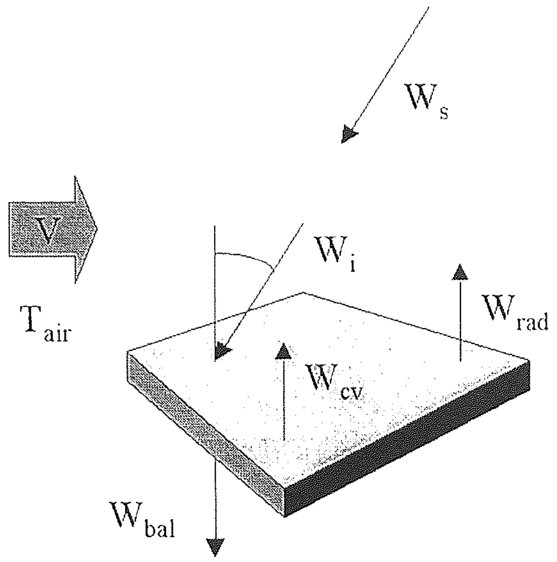

Figure 2-2: The heat flow model used for the infrared and visible light image fusion.

Two types of convection were modelled; free convection for zero wind speed and forced convection for non-zero wind speed. Each regime had several expressions for h dependent on the circumstances [90].

For given infrared and visible-light images these equations could be used in reverse to estimate the heat flux balance and therefore the ratio R of the surface material. In experiments described in [130] it was shown that materials have characteristic ratios. Metallic objects have a low value of R (of varying values up to 0.25), building materials such as concrete, brick and asphalt have intermediate values of R (typically in the range 0.49→0.87) and vegetation has very high values of R (0.86 and higher, owing to the photosynthesis process).

The primary advantage of the technique is that the properties of surface materials are difficult to change and therefore the technique will be robust in a wide range of domains. The main disadvantages of this approach are the need for pixel-registered imagery and the high bandwidth of communication between sensors (both of which make this approach suitable only for single platform systems). Furthermore, it was shown in [31] that the amount of noise that can be tolerated is relatively small (although well within the state-of-the-art for such imagers). This may be a deciding factor if selecting this method for particular applications.

2.3 *Empirical Modelling*

The advances in pattern recognition may be readily transferred to data fusion. Several standard empirical techniques have been used for processing of multi-sensor data including linear discriminators [50], Gaussian classifiers and their variations [118], Kernel-based density estimation methods [137] and the nearest neighbour based pattern classification rules [51] and [81]. Several researchers have used so-called neural network architectures for data fusion. Methods such as the radial basis function network [40] and the multi-layer perceptron [148] are popular ([7] and [143] for example). Figure 2-3 shows the configuration for a multi-layer perceptron (MLP) fusion centre. In this model, the processing for each of the sensors is performed by a MLP, initially trained separately using standard methods. Two approaches may then be used for training of the fusion centre – either the fusion MLP is trained in isolation using the outputs of the sensor MLPs as input or the entire network is treated as a single MLP and optimised accordingly.

These techniques have also been used to fuse multi-sensor data at both the probability and the feature levels using slightly different MLP architectures (see Figure 2-3 and Figure 2-4). The author showed in [22] that by introducing an optimisation criterion which combined the sum squared error at the fused output and the separate outputs, it was possible to produce a fusion network which compromised between globally optimal and locally optimal features. Such a network was shown to be more robust to communication failure since the separate sensors could then operate autonomously.

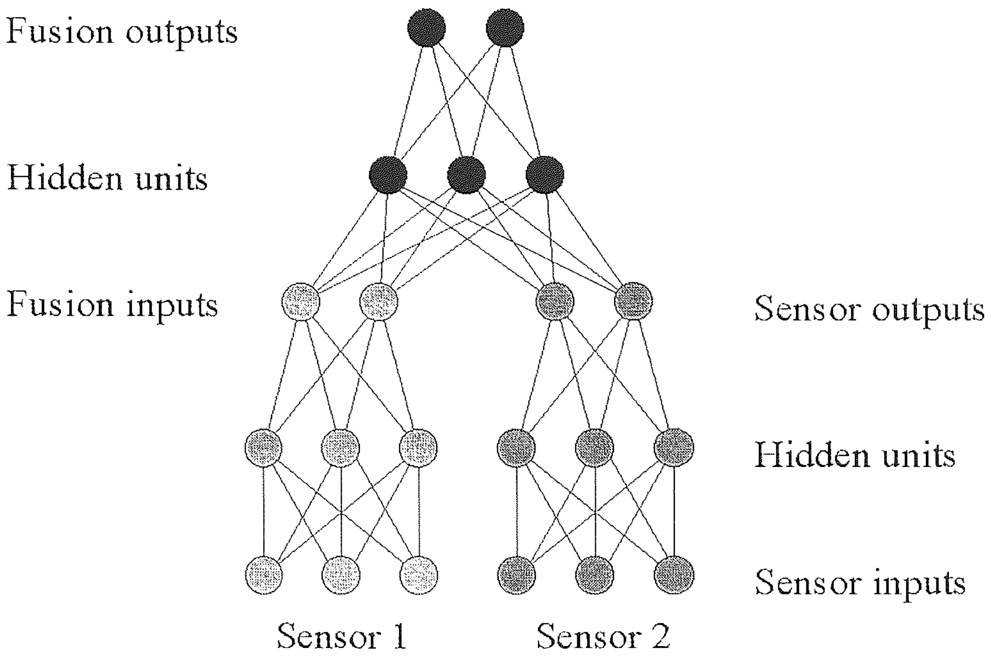

Figure 2-3: The probability level fusion configuration for the multi-layer perceptron.

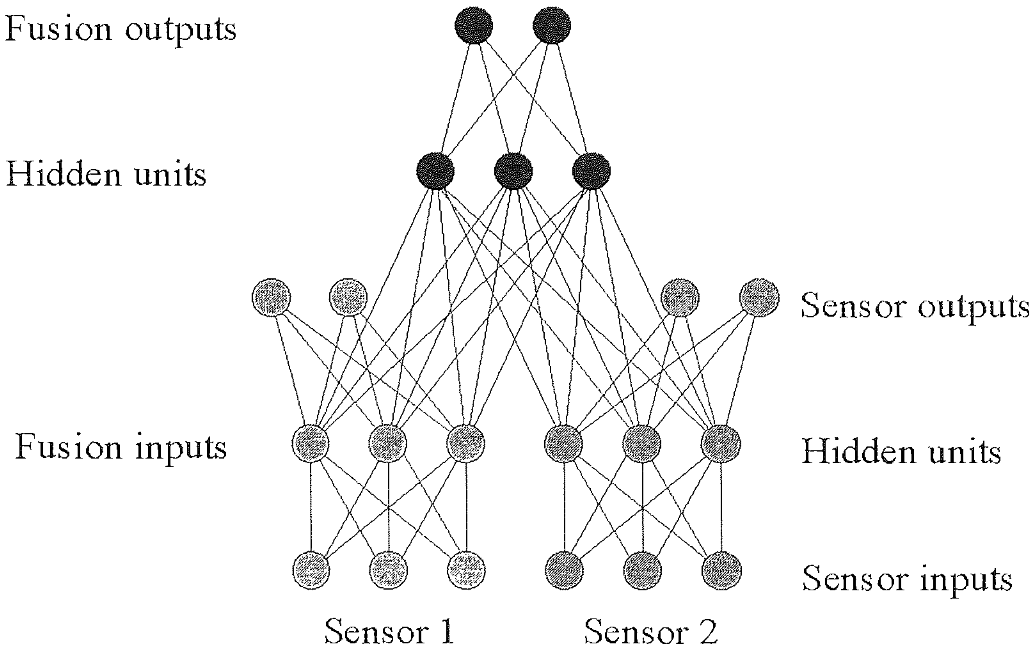

Figure 2-4: The feature level fusion configuration for the multi-layer perceptron.

2.4 *Expert Systems*

In the early development of automated data fusion systems, the intelligent knowledge-based system (IKBS) was seen as an appropriate technique. The approach was well established in the artificial intelligence (AI) community and had demonstrated potential on a number of related tasks. One of the first uses of IKBS technology for large-scale data fusion modelling was undertaken in work introduced in [105] and [106]. A similar, blackboard, architecture has subsequently been employed in several other, military data fusion systems [151], [152], [64] and [28]. The principal difficulties, which these researchers have identified with the IKBS or AI approach, are that there is a requirement for existing manual solutions and the availability of well-qualified and eloquent experts. In addition the way in which uncertainty can be expressed in the system (including uncertainty in the received information and in the rule base itself) is somewhat restricted and the system is sensitive to the introduction of new rules.

It does, however, provide an intuitive development environment which operators are generally able to identify with. The symbolic, rule-based, approach also gives the opportunity to give reasons for the decisions reached using simple *if-then* explanations.

More recent data fusion studies that call upon the knowledge of experts to design the fusion process have concentrated on the Bayesian belief network (BBN). A BBN is a network of interconnected nodes. Each node encapsulates the state of a particular facet of the system as a probability distribution. The links between the nodes represent the conditional probabilities of pairs of variables within the model. States of the system whose nodes are not connected are assumed to be independent [138]. For an introduction to Bayesian belief networks the reader is referred to [92]. The BBN model has been used for data fusion at the object level [155] and the situation assessment and threat assessment levels [66]. An example of the development of a BBN for information fusion for situation assessment is shown in Figure 2-5. The network encapsulates the fusion necessary in a simplified air defence scenario [18]. The pale grey entities to the left of the network concern properties of individual aircraft. The dark grey properties in the centre concern packages of aircraft and the mid-grey entities to the right concern the target.

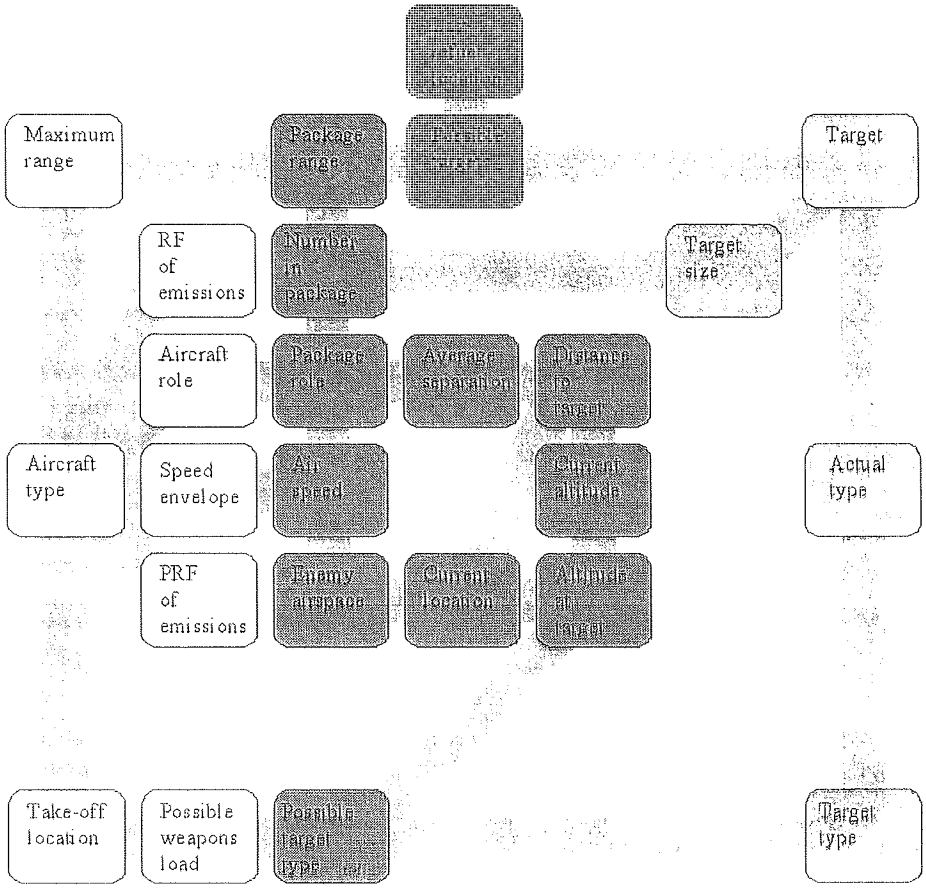

Figure 2-5: The air defence Bayesian belief network developed by the author.

In the BBN developed for this simplified application 25 nodes were defined and 28 conditional probability distributions between the entities these nodes represented were specified. The development process for this type of network is well structured and is loosely based on the CommonKADS KBS lifecycle [150]. It proceeds through the following steps:

- Identify a small number of abstract entities, or meta-nodes (in this example the entities aircraft, package and target were identified)
- Identify the principal qualities in each entity which depend on those in another entity (in this case the range, role, speed and weapons characteristics of the aircraft were determined to be linked to the associated quantities for the package. The target type, location and size were shared characteristics of the package and the target entities)
- Refine the abstract entities into a sub-network of nodes which represent more specific quantities which may be measured or otherwise ascertained
- Define the conditional dependence between individual nodes

The network as described was used to simulate known air missions flown during the 1991 Gulf conflict. The network was used in two ways. Firstly as an air defence analyst for determining likely targets given uncertain information on incoming aircraft. Secondly as a mission planning aid for targeting specific targets. It was noted that the network gave a plausible representation of the scenario and was robust to missing data. The approach is being further developed for data fusion by other researchers to admit the use of fuzzy rules [135] and automatic structuring [126].

2.5 *Human-in-the-Loop*

The integration of human operators into automated systems, which perform complex data fusion tasks, has been somewhat overlooked of late. Command and control (C^2) is a notable application where human-in-the-loop (HITL) processing is likely to be required for the foreseeable future. In this application the spectrum of socio-political, economic, military and interpersonal facets of a situation must be brought to bear on the decision-making process. Although automatic techniques can greatly accelerate the processing of sensor data and encyclopaedic knowledge, the commander is (currently) still required to make the final decision. This decision is often made under the dual pressures of time and resources. Adequate integration of human operators into such a process is little understood and, with a few exceptions such as [89], [97] and [43], has not been addressed in the data fusion community.

2.6 *Chapter Discussion*

In the preceding sections we have seen that the data fusion process may be approached in several different ways. The main advantages of physical modelling are that the theories are usually rigorous and need little maintenance, that the theories can be applied fairly directly and are widely applicable. The key limitations of physical modelling are that they may not be robust to missing or inaccurate input data and that they are inflexible. The advantages of empirical modelling are that there are many existing methods that can be tailored to specific applications and that they may be able to handle inaccuracies or missing data. The principal limitation of empirical modelling is that adequate training data is required. The advantages of expert models are that it exploits human insight and that it often offers compact solutions which are understood and accepted by human operators. The main limitation is in the availability of an expert from which the knowledge can be elicited. In some circumstances the knowledge capture itself is challenging [62] and the representation of the knowledge may prove awkward. Human-in-the-loop processing is most appropriate when the process involves a lot of common sense processing or where other techniques are not possible.

We shall select the empirical approach for the remaining chapters since it offers the flexibility to handle many data fusion problems and is amenable to quantitative analysis.

Chapter 3 Data Fusion Architectures

3.1 *Chapter Introduction*

In this chapter we introduce the concept of a system level approach to data fusion. Data fusion as a discipline only exists because of practical constraints, such as communications bandwidth or processing power, which affect systems. In the absence of such constraints all of the data would be available centrally and could be modelled jointly. In the following sections we describe a number of ways of organising the data fusion processing. This discussion of functional architectures will establish the context for the analyses that follow. We define two terms to describe the way in which data fusion algorithms may be embedded in a larger system. Firstly, we use the term *process model* to describe a sequence of processes, which must be undertaken before the system can be regarded as fully operational. Secondly, we define the term *functional topology* to mean the way in which these processes are distributed, the connectivity and data flows between their component parts. We shall expand on the options for each of these terms in the next two sections.

3.2 *Process Models*

The information processing requirements of modern automated systems require a plethora of data processing, data reduction and data combination capabilities. The specific requirements will, of course, vary between domains. However, a common strategy for implementing such systems is to modularise the processing into several components and to process the data either sequentially or hierarchically using these elements in turn.

3.2.1 Pattern Recognition Hierarchy

The UK Technology Foresight working group on Data Fusion and Data Processing [5], agreed the pattern recognition hierarchy model in 1996. Pattern recognition is an important task, which is encountered in many domains including NCTR, target acquisition, medical diagnosis, fault analysis, direct voice input and autonomous navigation. The recognition and decision support tasks are typically modularised into a set of four components (sensor and signal processing, feature extraction, pattern processing and decision making) as shown in Figure 3-1. In this model the flow of information is from raw data (at the lower levels) to abstracted information (at the higher levels). The diagram also illustrates the reduction in bandwidth, which occurs during the process.

These sub-problems have historically been viewed as essentially separate disciplines with only loose, interdisciplinary constraints imposed and interactions driven primarily through the personalities and contacts of the major research groups. As a result distinct frameworks, paradigms and expert knowledge have emerged in each.

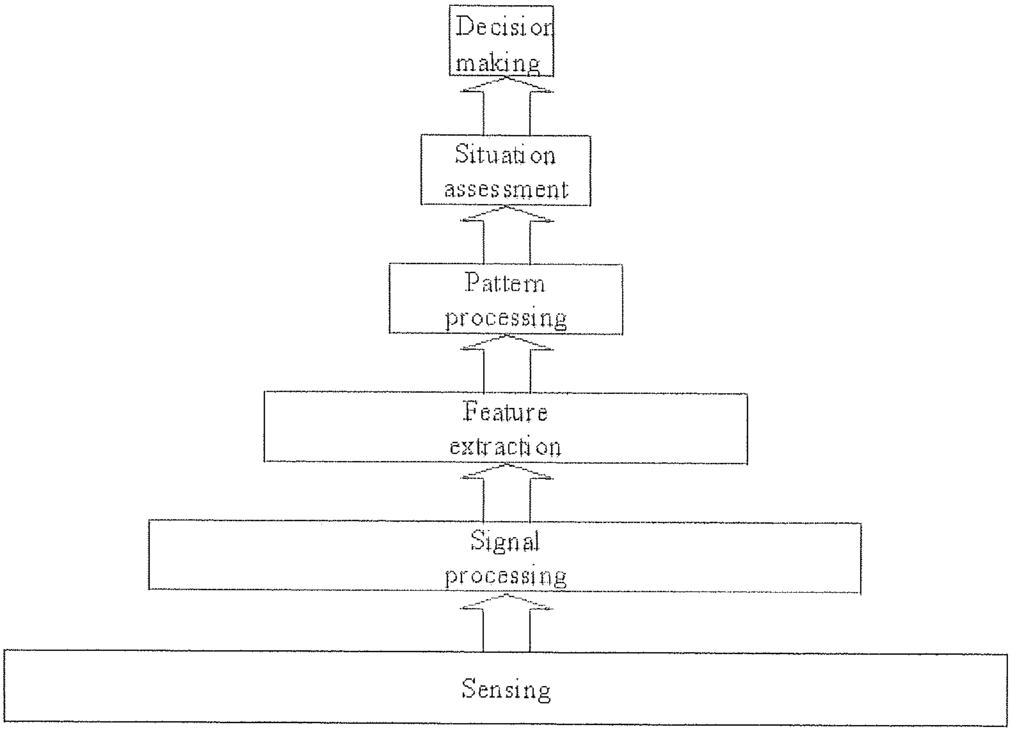

Figure 3-1: A schematic modularization of the pattern recognition process.

3.2.2 JDL Data Fusion Model

The Joint Directors of Laboratories group (JDL) Data Fusion Group (DFG) produced the JDL data fusion model shown in Figure 3-2. Shown in this model are the three levels of data fusion and one additional level of process refinement. Some preliminary filtering is assumed to take place since many data fusion systems will be overwhelmed by the volume of data produced by modern electronic sensor systems. Such filtering could be carried out on the grounds of the time of the event, the location of the event, the type of the event or the signature associated with the event. Once this pre-filtering has taken place the sensor data is passed on to the data fusion system itself.

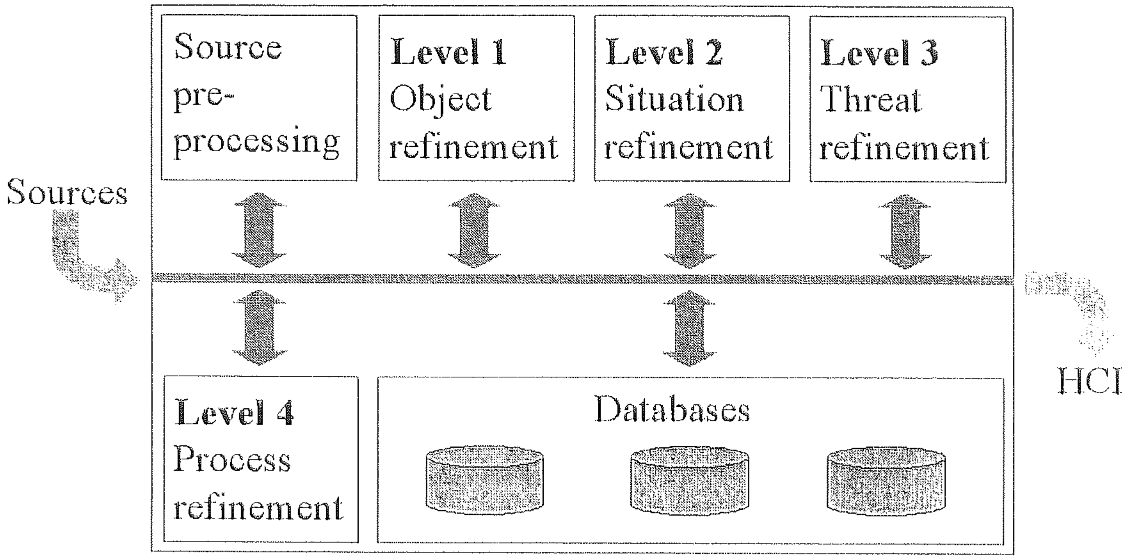
Source pre-processing
Level 1 Object refinement
Level 2 Situation refinement
Level 3 Threat refinement
Sources
HCI
Level 4 Process refinement
Databases

Figure 3-2: The JDL data fusion model as originally defined.

This model is based on the proposal in [77] and defines four levels of data fusion [79]:

1. **Object refinement** – the processing of identity or location / kinematic information pertaining to individual objects (for example single aircraft) within the area of interest. Information being fused at this stage is generally sensor-derived data. This type of data may be amenable to physical or empirical modelling. Much of the early work in data fusion was performed at JDL level 1. Typical functions undertaken at level 1 include:

   - ❑ Data alignment (spatial or temporal)
   - ❑ Correlation (gating, association or assignment)
   - ❑ Positional or kinematic estimation (using system or empirical models)
   - ❑ Identity estimation (using physical, feature-based or cognitive approaches)

2. **Situation refinement** – once information about individual objects is made available it can be fused at JDL level 2 to provide context. At this level sets of objects are combined into meaningful groups (for example by grouping a number of aircraft into a formation). It is at this stage of the fusion process that a useful picture of the outside world is first formed. At this level both sensor derived data and operator derived knowledge needs to be brought to bear on the task. Entities under consideration at level 1 are associated with:

- ❑ Other entities
- ❑ Environmental data
- ❑ Doctrinal information
- ❑ Performance data

3. **Threat refinement** – the impact that the evolving situation might have on the users well-being is assessed at JDL level 3. At this level of fusion both the input information and the collateral knowledge are generally quite abstract concepts, often involving human-in-the-loop (HITL) processing. Factors under consideration include:

   - ❑ Expected courses of action and intent estimation
   - ❑ Lethality and countermeasure assessment
   - ❑ Composition and deployment
   - ❑ Environmental effects

4. **Process refinement** – the management of the flow of information within the model and the tasking of collection assets are both implied at JDL level 4.

Additionally the JDL model identifies sources, databases and a human-computer-interface. The primary inputs, or sources, may include sensors, human sources and open sources. Internal databases of geographical or encyclopaedic data, for example, contain the necessary collateral information. The human-computer interface (HCI) which represents the primary output to the operator may include text, graphics and multimedia. Several workers interpret the JDL model as a sequential process with information flowing from sources and then through levels 1→4 in sequence before feeding back into source tasking.

The original JDL model was updated in late 1997 [154] to introduce a specific JDL level 0. Level 0 corresponds to the fusion of information at the pre-detection stage. Examples of such fusion are pixel level fusion or image registration. The new model also makes sensor management (asset availability, sensor tasking and task prioritisation) an explicit part of level 4 fusion, which was not clear in the original formulation. The revised JDL model is illustrated graphically in Figure 3-3. This model also incorporates level 0, sub-object refinement. As will be seen subsequently, the process levels may be broadly equated with the layers in the pattern recognition hierarchy.

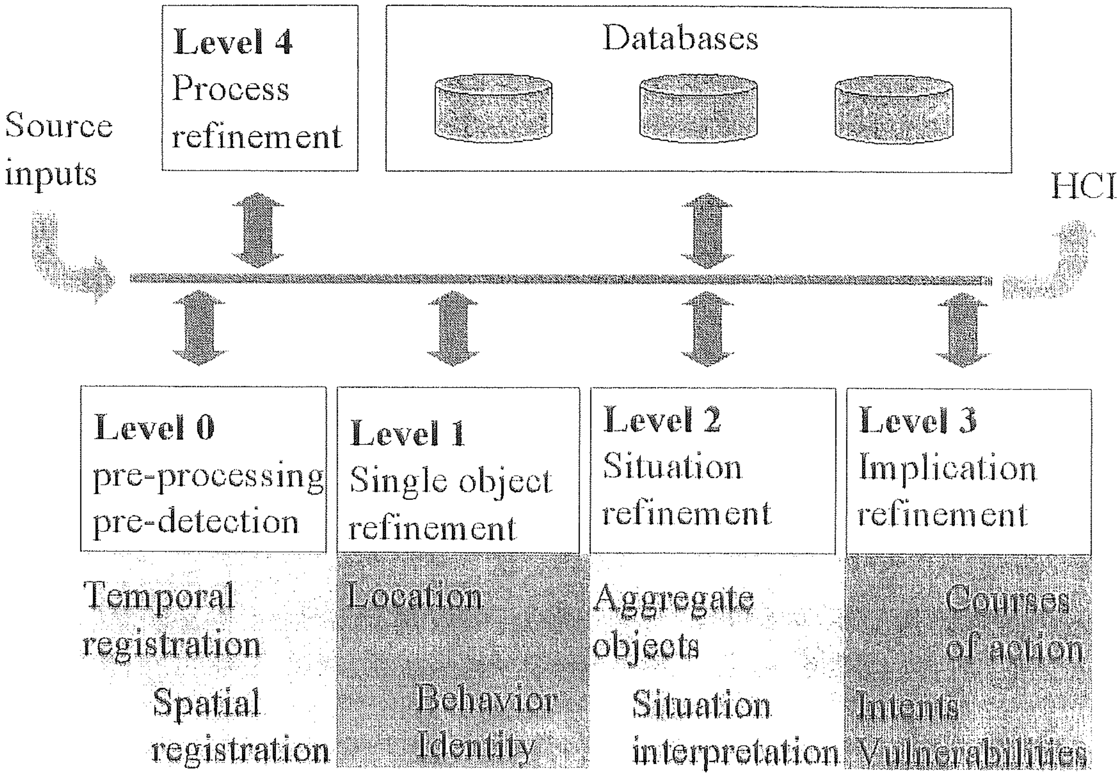

Figure 3-3: The revised JDL data fusion model.

3.2.3 OODA loop

The Observe-Orient-Decide-Act control loop was first described by Boyd in [36] and is shown graphically in Figure 3-4. The OODA loop is widely used in command and control analyses, particularly in military domains. The OODA loop identifies the principal stages at which a commander makes use of information available to him. In the first stage (*observe*) sensors and sources are used to collect raw data which may be collated and pre-processed into a form which can readily be assimilated. In the next (*orient*) stage, this data is analysed and a picture of the state of the system and the commander's role in it is developed. In the third (*decide*) phase, the command decision making is undertaken and in the final (*act*) phase the commands are put into action.

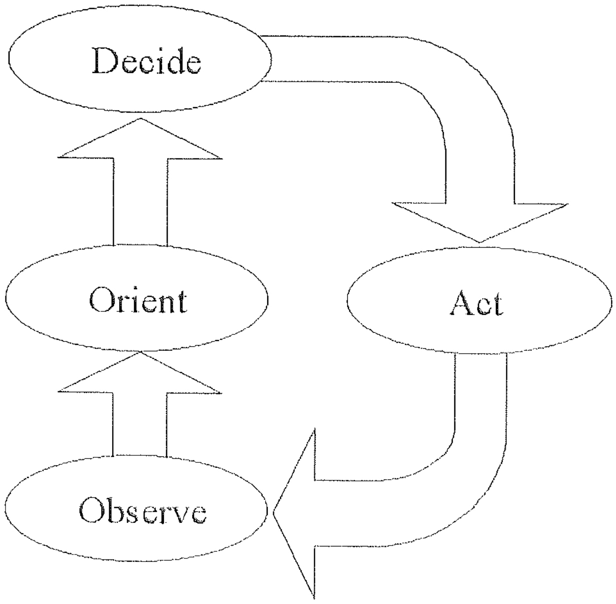

Figure 3-4: The Observe-Orient-Decide-Act (OODA) control loop.

3.2.4 Intelligence Cycle

Intelligence processing involves both information processing and information fusion. Although the information is often at a high level, the processes for handling intelligence products are broadly applicable to data fusion in general. There are a number of principles of intelligence:

- Central control (this avoids the possibility of duplication)
- Timeliness (this ensures that the intelligence is available fast enough to be useful)
- Systematic exploitation (makes sure that the outputs of the system are used appropriately)
- Objectivity (of the sources and the manner in which their information is processed)
- Accessibility (of the information)
- Responsiveness (to changing intelligence requirements)
- Source protection (to guarantee a source of information with increased longevity)
- Continuous review (of the process and the collection strategy)

Perhaps the most relevant factor to data fusion is that of objectivity. We will demonstrate in later chapters that intermediate decisions that are lacking objectivity, may propagate through data fusion systems and reduce overall performance. The issue of central control will be addressed in the next section on functional architectures. It will be seen that avoidance of duplication may be achieved in several ways. The UK intelligence cycle also lends itself to modelling the data fusion process. The cycle itself is depicted in Figure 3-5. Unlike the American model the British model does not include a specific planning and direction phase. The cycle comprises four phases (as in the OODA loop):

- **Collection** – collection assets such as electronic sensors or human derived sources are deployed to obtain raw intelligence data. In the world of intelligence the information is often presented in the form of an intelligence report – either free form text or in a predefined report format.

- **Collation** – associated intelligence reports are correlated and brought together. Some combination or compression may occur at this stage. Collated reports, however, may simply be packaged together ready for fusion at the next phase.

- **Evaluation** – the collated intelligence reports are fused and analysed. Historically, highly skilled human intelligence analysts have undertaken this process. The analysis may identify significant gaps in the intelligence collection. In this case, the analyst may be able to task a collection asset directly. More usual, however, is the inclusion of this requirement in the disseminated information.

- **Dissemination** – the fused intelligence is distributed to the users (usually commanders) who use the information to make decisions on their own actions and the required deployment of further collection assets.

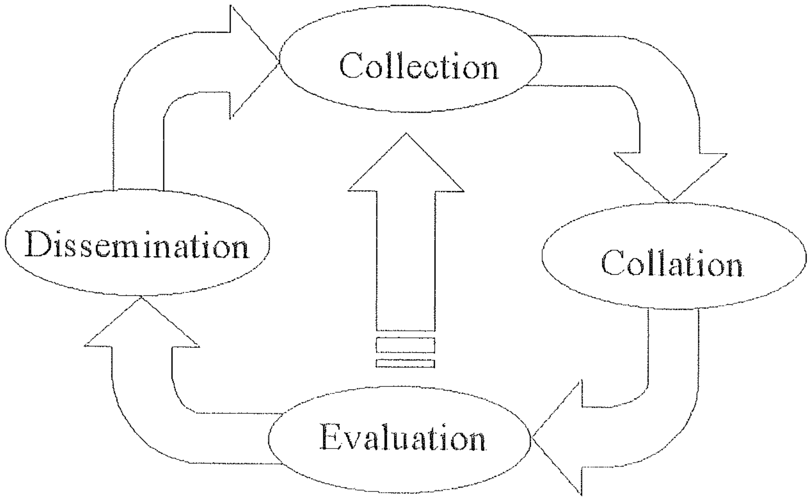

Figure 3-5: The UK intelligence cycle.

Each intelligence source is tagged with a confidence measure or *information grade*. The information grades used in the UK are numeric codes in the range 1→6 as shown inTable 3-1. Codes 1→5 can be regarded as representing approximate belief values. Code 6 indicates complete ignorance regarding the confidence that may be placed in the information. This is important since it provides our first example of the grading of information quality and raises the question of how such grading may be incorporated in the subsequent fusion. We shall return to this issue shortly.

1	Confirmed by other sources
2	Probably true
3	Possibly true
4	Doubtful
5	Improbable
6	Unknown confidence

Table 3-1: The information grade codes used by the UK military.

The different sources of information vary in accuracy, objectivity and reliability. In the UK, a system of reliability categories are also used, each category being assigned a letter code. The reliability of a source depends on the type of source and the historical reliability of that particular asset and is assessed by the intelligence analyst. The reliability codes are shown in Table 3-2.

A	Completely reliable
B	Usually reliable
C	Fairly reliable
D	Not usually reliable
E	Completely unreliable
F	Unknown reliability

Table 3-2: The UK intelligence source reliability coding scheme.

Intelligence analysts use these reliability codes to mediate the relative weightings given to different sources of information. The reliability codes are combined with the information grades in a somewhat ad hoc manner within the analyst cell. A rigorous means of assessing source reliability and automatically combining it with information grade is a desirable feature of any automatic data fusion system. The aim of combining accuracy and reliability will form one of the main thrusts of the research described in later chapters.

3.2.5 Comparison of Data Fusion Models

The four models described in the preceding sections can be compared and equivalencies identified where appropriate.

Table 3-3 shows a comparison between the process models described thus far. In some cases the equivalence is approximate. Greyed out boxes are not addressed by the specific model identified at the top of the column. It can be seen that there is some overlap in the way that the different models sub-divide the information flow from sensors to actions. The main differences correspond to the amount of detail with which particular processes are represented. This stems from the different uses of the various models and the emphasis they place on certain aspects of the information processing and fusion chain. As can be seen from Table 3-3, the information processing hierarchy contains the finest distinction between the lower levels of abstraction, the JDL model at the medium level and the OODA loop at the higher levels. The intelligence cycle covers all levels but in somewhat compressed detail. In [26] a universal architecture is proposed which encompasses the main approaches defined here.

Activity being undertaken	**Information Processing Hierarchy**	**JDL Data Fusion Model**	**OODA Loop**	**Intelligence Cycle**
Command execution	▨	▨	Act	Disseminate
Decision making process	Decision making	Level 4	Decide	
Threat assessment	▨	Level 3	Orient	Evaluate
Situation assessment	Situation assessment	Level 2		
Information processing	Pattern processing	Level 1		Collate
	Feature extraction			
Signal processing	Signal processing	Level 0	Observe	
Source / sensor acquisition	Sensing	▨		Collect

Table 3-3: A comparison of the four data fusion process models identified.

In the unified model the cyclic nature of the data fusion process is made explicit by retaining the general structure of the OODA loop. The fidelity of representation expressed by the pattern processing hierarchy is then easily incorporated into each of the four main process tasks. The points in the process where fusion may take place are explicitly located, see Figure 3-6.

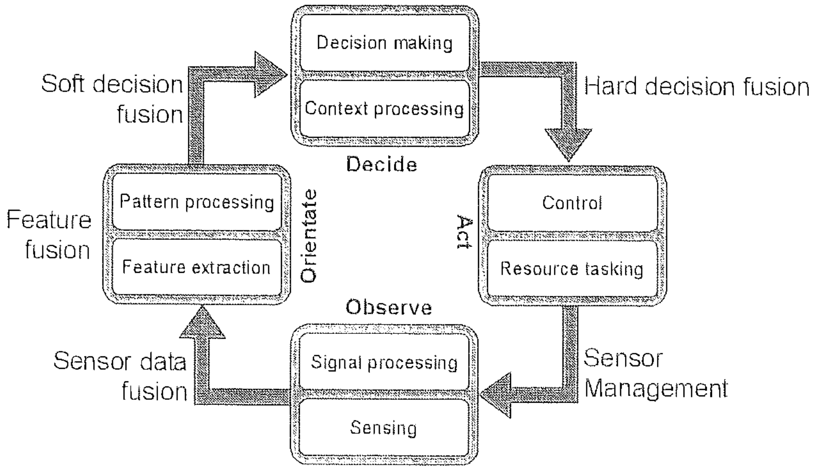

Figure 3-6: The unified data fusion process model.

3.3 *Functional Topologies*

In the preceding section we described the manner in which the data fusion process is organised. Since we have already established that the practical constraints of data fusion systems are also of importance it is necessary to examine the physical organisation of these processes in a data fusion topology.

3.3.1 Centralised

The centralised or, all-source, topology is perhaps the simplest arrangement for handling multi-sensor data. In this arrangement the raw sensor data from each of the sources is communicated to a single, central, fusion cell where the information is combined and passed on to the consumer. The arrangement is illustrated schematically in Figure 3-7. Each source is labelled "S", the single fusion cell is labelled "F" and the consumer of the information is labelled "C".

The principal advantage of the centralised topology is its ability to use all of the information collected by the sensors. The performance accuracy of the centralised topology is therefore potentially optimal. The main disadvantages of the centralised topology are twofold. Firstly the communications requirements, since the bandwidth necessary for communicating the raw sensor data to the fusion cell may be prohibitive (for example when image data needs to be transmitted over radio networks or satellite links). Secondly, the ability to produce a model that can effectively deal with all of the data sources.

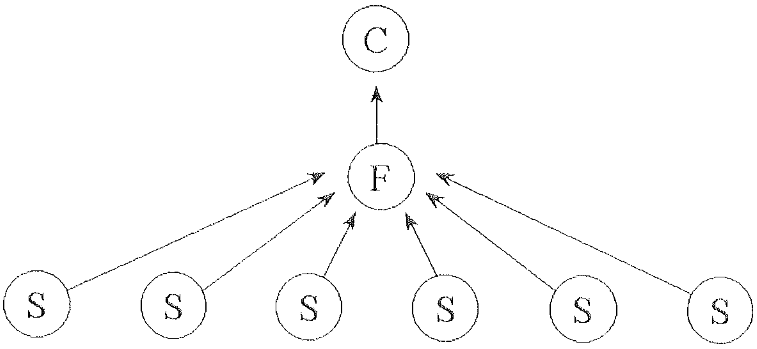

Figure 3-7: A centralised or all-source data fusion topology.

A reduced-bandwidth, centralised topology may be employed in some situations. In this case each source performs some local processing of the data to reduce communications loading. When other constraints admit the centralised topology it should be used in preference to the other topologies detailed in the following sections.

3.3.2 Hierarchical

The hierarchical topology, as illustrated in Figure 3-8, processes and fuses the information in stages. Nomenclature is identical to the previous diagram. In this case, however, several fusion cells combine the information before finally passing on the outcome to the consumer. Data fusion that is carried out on raw sensor data is termed low-level fusion and that which combines previously fused data is termed high-level fusion [108]. The principal advantages of the hierarchical data fusion topology are:

- Lighter processing load
- Distribution of collateral databases
- Reduced communications loading
- Faster user access resulting from reduced communications delays
- Prevention of information incest as a result of single data flow paths
- Easier modelling

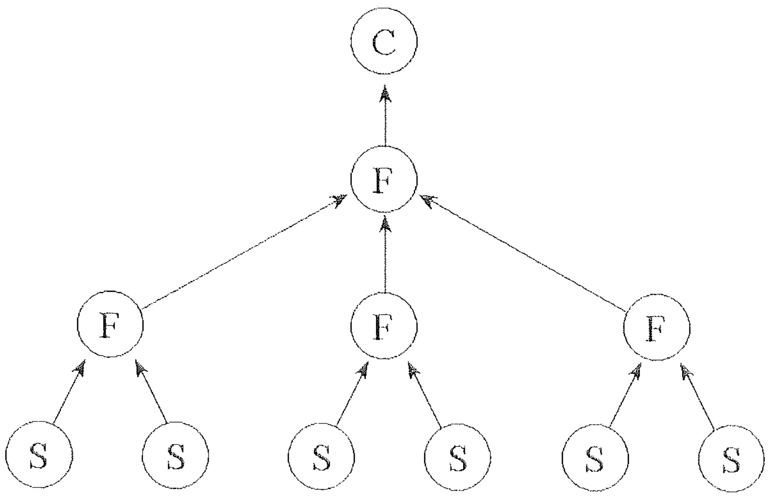

Figure 3-8: An example of a hierarchical data fusion topology.

In a hierarchical data fusion topology several levels of abstraction are possible. From high to low-level these are:

- **Decisions** – also known as symbolic fusion
- **Probabilities** – or belief value fusion
- **Features** – or intermediate-level fusion
- **Data** – or sensor level fusion

As was pointed in out by Dasarathy in [55], however, fusion may occur both at these levels and as a means of transforming between them. In the model proposed by Dasarathy (which omits probability-level fusion) there are five possible categories of fusion. Augmenting this model with the additional probability level we obtain a seven-layer model as depicted in Table 3-4. Note that the additional levels, which transform between representations, have direct analogues in classical pattern processing.

Input	**Output**	**Notation**	**Analogues**
Data	Data	DAI-DAO	*Data-level fusion*
Data	Features	DAI-FEO	Feature selection and feature extraction
Features	Features	FEI-FEO	*Feature-level fusion*
Features	Probabilities	FEI-PRO	Pattern recognition and pattern processing
Probabilities	Probabilities	PRI-PRO	*Probability-level fusion*
Probabilities	Decisions	PRI-DEO	Rational decision making
Decisions	Decisions	DEI-DEO	*Decision-level fusion*

Table 3-4: The seven possible levels of data fusion in the augmented Dasarathy model.

The advantages and disadvantages of conducting fusion at the various levels is depicted in Table 3-5.

Fusion level	**Bandwidth**	**Performance**	**Advantages**	**Limitations**
Decisions	Very low	Depends on system	Simplicity for large systems	Poor performance for small systems
Probabilities	Low	Often good	Bandwidth / performance trade-off	Sophisticated algorithms needed for correlated sources
Features	Moderate	Good→high	High performance	Difficult to select correct features
Data	High→very high	Potentially optimal	Possibility of using physical models	High bandwidth restricts use to single platform systems

Table 3-5: The advantages and disadvantages of the four levels of data fusion.

Even within this model there are a number of choices as to the specific format of the data being fused. For example in the three class problem comprising classes *X*, *Y* and *Z* the possible formats at the decision and probability levels are [168]:

- Hard declaration – most likely class is *Y*
- Hard declaration shortlist – best hypotheses are *{X,Y}*
- Ordered hard declaration shortlist – best hypotheses, in order are *{Y,X}*
- Ordered hypothesis list – ordered list, best choice first is *{Y,X,Z}*
- Soft declaration shortlist – best hypotheses are *{X,Y}* with $P_Y=0.6$ and $P_X=0.3$
- Soft declarations - $P_Y=0.6$, $P_X=0.3$ and $P_Z=0.1$

The first four formats are categorised as decision-level fusion and the remaining two formats as probability-level data fusion. In order to simplify our subsequent analyses we shall concentrate on the two extreme cases: decision level fusion in which only the most likely object type is reported and probability-level fusion in which a full set of soft declarations are supplied.

3.3.3 Distributed

In a fully distributed data fusion system no single fusion cell is designated as the master fusion centre. This has both advantages and disadvantages. Since many existing (manual) data fusion systems are fully distributed, the distributed topology is often imposed rather than chosen. Distributed topologies may be driven by an information-push from the source or an information-pull from the consumer (or a mixture of the two). Since there is no single path along which data needs to flow to get from the sources to the consumer, the fully distributed topology is likely to be more robust to communications failures and processor downtime. This same facet of multiple interconnectivity also poses the main difficulty with the fully distributed topology. Because the same information may arrive at a fusion cell using more than one route it is necessary to ensure that it is not fused with itself. Such information incest would result in unjustified reinforcement. Some graphical information flow methods have been used to address the information incest problem [108]. If the information pedigree is tagged on to the data itself then the point at which self-reinforcement is likely to occur may be identified. Three methods for handling information incest were proposed:

- Restart the fusion using only the current information when incest is detected
- Sever those communications channels which give rise to the transmission of incestuous information
- Divide out the extra confidence which results from the incestuous information and proceed with the subsequent processing as normal

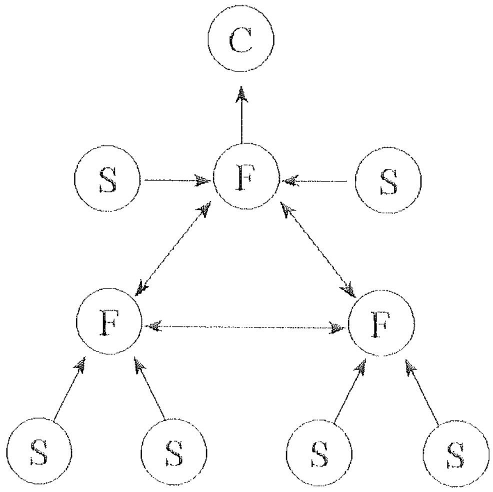

Figure 3-9: A fully distributed data fusion topology.

Figure 3-9 shows a fully distributed topology in which sources may supply information asynchronously to associated fusion cells, which exchange information appropriately. The consumer is provided with the outcome as it becomes updated.

3.4 *The Veto Effect*

In certain data fusion topologies the so-called veto effect may be observed. When most of the information sources provide erroneous data it is not unreasonable for the data fusion system to make an incorrect assessment (or more properly, to defer its decision). When the majority of the information sources are providing good quality data, however, it is far from desirable for the system as a whole to make an error. If the minority of erroneous sources is not given appropriate weight they can dominate the decisions made by the majority of good quality sensors. This is called the veto effect. Although the effect is possible in any of the architectures described above, it is most noticeable in the hierarchical topology. Figure 3-10 shows a pathological example of a hierarchical data fusion system drawing information from 27 sources. Although only 8 of the sources are providing erroneous information (indicated in black) the hierarchical nature of the processing (which in this case uses a majority rule) has allowed them to dominate.

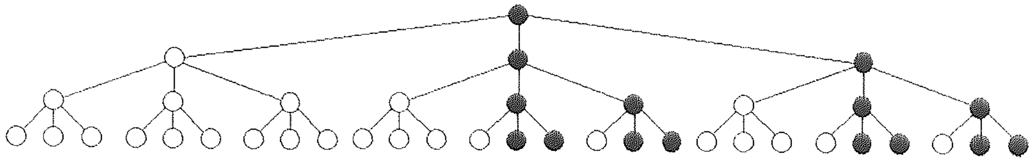

Figure 3-10: A pathological example of the veto effect in hierarchical data fusion.

The veto effect is also observed when the information is graded (as is the case in the intelligence process as described on page 32). In this case it is possible for the most certain sources to dominate the less certain sources. This is fine as long as those sources that claim certainty are actually correct. If the assessment is overconfident, then the veto effect can erroneously give them additional weight.

3.5 *Chapter Discussion*

We have seen that data fusion is generally a multi-layered activity. Raw data is converted into understanding via a number of levels of abstraction. During the data fusion process information is collated and combined as it progresses from one level of abstraction to the next. A fusion centre may exist at any level of abstraction. The input to such a fusion centre will comprise the output of several processes at lower levels. During this collation process the amount of co-located information increases. During the combination process, however, irrelevant information is discarded and the reduced quantity (but of more concentrated quality) of information is passed on to the next level of abstraction. Figure 3-11 shows the knowledge pyramid, which illustrates that as the level of abstraction increases so the information bandwidth decreases.

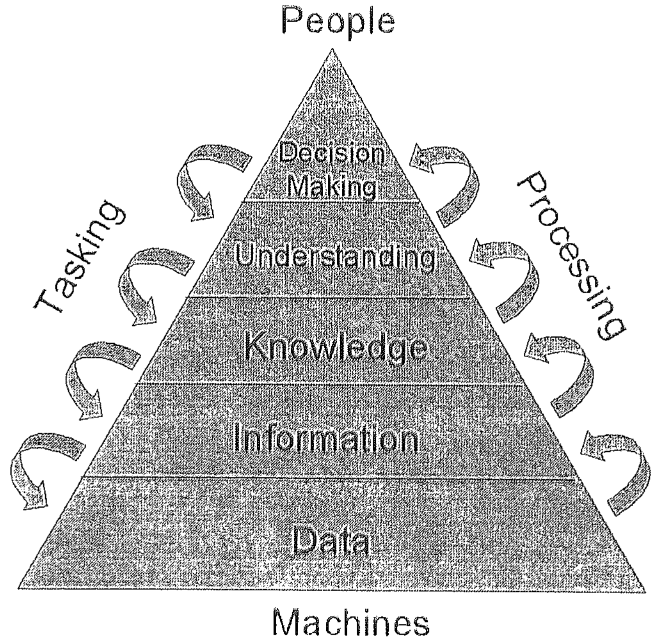

Figure 3-11: The knowledge pyramid of data fusion.

We therefore find the prospect of fusion at the decision level highly attractive since it requires minimal data communications bandwidth and therefore makes a centralised fusion architecture feasible for even large systems. We also suspect that the robustness of such a decision level centralised architecture would show robustness in the presence of erroneous information from a minority of sources. In part 2 we will address the key question posed in the preface with respect to multi-layered data fusion at high levels of abstraction. In the next chapter we will address decision level fusion. We shall quantify this robustness of various decision level fusion architectures and expand on the *maximum a posteriori* approaches to decision level fusion.

We introduced process models and functional topologies for the data fusion process. It was made clear that most process models use a layered approach to data fusion in which fusion may occur at increasing levels of abstraction. This was further examined in the context of functional topologies. It was suggested that fusion at either the probability or decision levels offer an appropriate low-bandwidth solution for data fusion across separate sensor platforms. Furthermore, it was noted that hierarchical fusion architectures allow easier modelling and a faster, more robust data fusion system that is not susceptible to information incest. It was noted, however, that such hierarchical systems are somewhat sensitive to faulty sensor information (the veto effect). Techniques for overcoming the veto effect for these two approaches will be studied in more detail in the following chapters.

Part 2:

Data Fusion Methods which Deal with Information Quality

Chapter 4 Decision Fusion

4.1 *Chapter Introduction*

Fusion of decision-level information comprised the initial forays into the new field of data fusion during the early 1980's. Decision fusion represents the extreme situation of fusing data from multiple, distributed sensors using minimal communications bandwidth between the sensors and the fusion centre. Decision-level data fusion is most appropriate when there are large numbers of sensors, as described in the preceding sections. It has been the subject of significant research over the last two decades, primarily because the problem is amenable to theoretical analysis. In this chapter we shall concentrate on decision level fusion because of these reasons. We shall begin the study of decision-level fusion with an examination of the veto effect since we saw in Section 3.4 that this can reduce the benefits of data fusion.

4.2 *Decision Fusion Experiments with the Veto Effect*

We shall examine the veto effect in which some data sources provide erroneous information to the fusion process. We shall analyse the robustness of various decision-level topologies to increasing proportions of misleading information. In the following experiments a decision level data fusion system comprising 16 sources was simulated. At this preliminary stage the fusion rule was simple majority voting. Three fusion topologies were used:

- Centralised topology in which all 16 sources were connected directly to the fusion cell
- Hierarchical topology in which 5 fusion cells were used, each with a fan-in of 4 sources
- Hierarchical topology in which 15 fusion cells were used, each with a fan-in of 2 sources

These topologies are illustrated in Figure 4-1, Figure 4-2 and Figure 4-3 and clearly illustrate that the fusion is carried out in one, two or four stages respectively.

For each topology a series of experiments was performed. In each case four of the sources (just 25% of the total) were made to report the incorrect class. The reports were assigned randomly to the sources. Furthermore, a certain proportion of the sources was set as deferring their decision. The proportion of deferring sources varied from none to all sixteen. Again, these were allocated randomly. For each allocation the fused decision was calculated using a simple majority rule method.

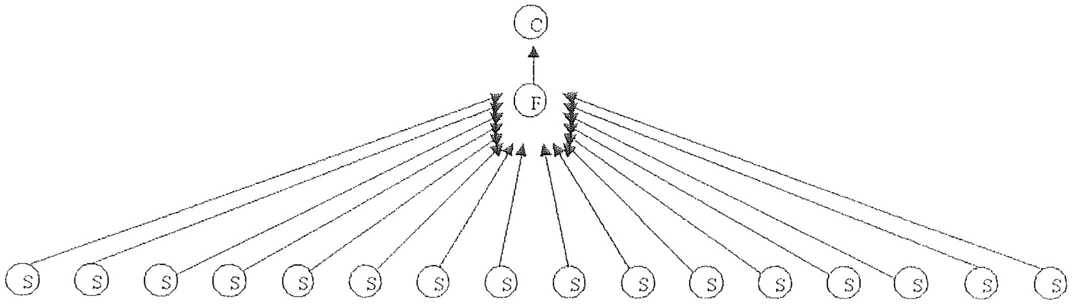

Figure 4-1: The centralised topology used in the veto experiments.

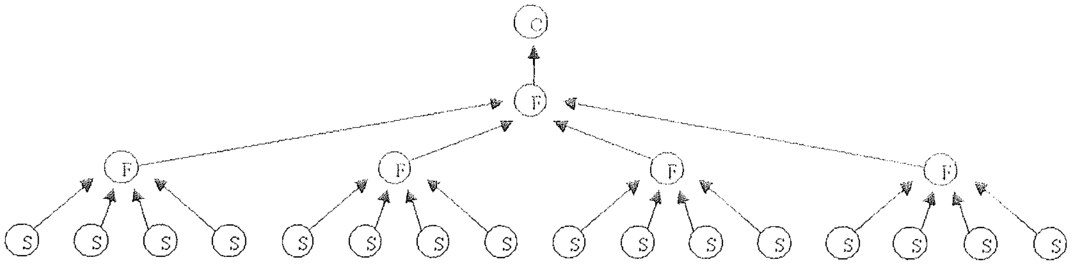

Figure 4-2: The shallow hierarchical fusion topology used in the veto experiments.

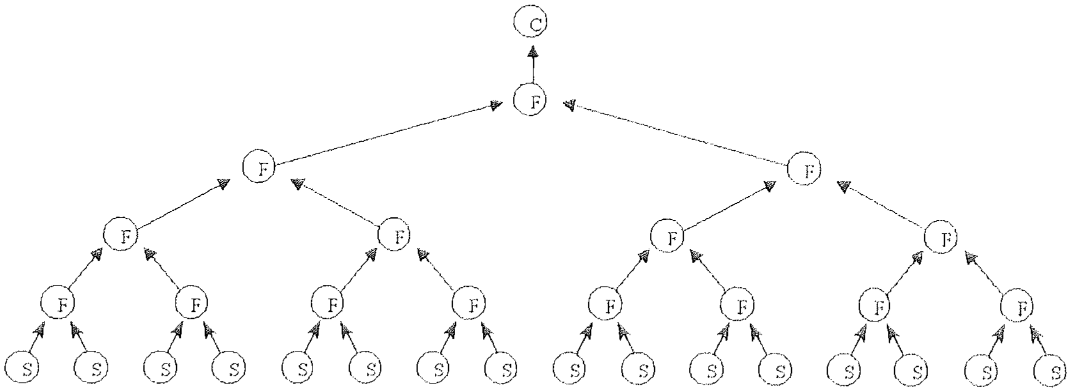

Figure 4-3: The deep hierarchical topology used in the veto effect experiments.

The percentage of 10,000 such experiments that gave rise to the correct fused decision being output was recorded and is shown in Figure 4-4. The solid line shows the performance of a centralised topology while the other lines show hierarchical topologies with a fan-in of four (dotted) and two (dashed). In each case 25% of sources was providing an incorrect decision. This graph shows that a centralised topology shows the most resilience to the veto effect with little or no degradation in fused performance with over half of the sources deferring their decision. Both graphs for the hierarchical topologies show less robustness with significant drops in performance with fewer than half of the sources deferring. In this case the hierarchical topology having the larger fan-in of four sources performed slightly better than the deeper topology associated with the two-fold fan-in.

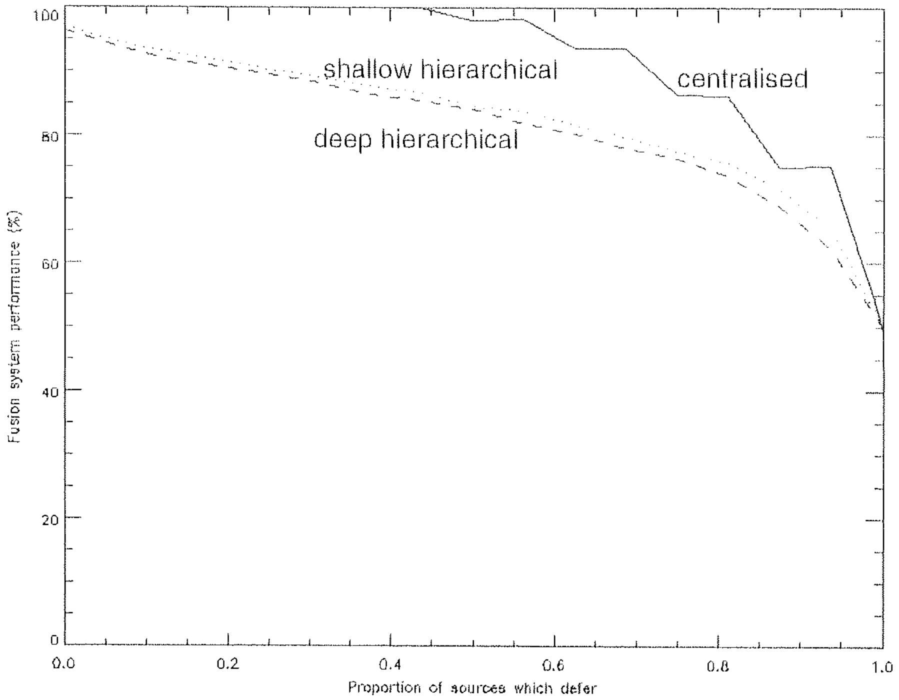

Figure 4-4: The veto effect for a data fusion system comprising 16 sources.

Further experiments were performed to analyse the performance of the same system as a function of the proportion of the sources that reported an incorrect decision. In Figure 4-5 the performance for a system with no deferring sources is shown. Figure 4-6 and Figure 4-7 show the same information for a deferral rate of 25% and 50% respectively. As in the previous experiments three topologies were assessed (centralised and hierarchical with a fan-in of either 2 or 4). It can be seen from the graph that the centralised architecture again yields greatest robustness to a minority of faulty sensor data with the hierarchical topology having a fan-in of two giving marginally the worst results. The effect is reversed for scenarios in which the majority of sensors are faulty. This is, however, a somewhat artificial situation that is not likely to be encountered in a well-designed data fusion system.

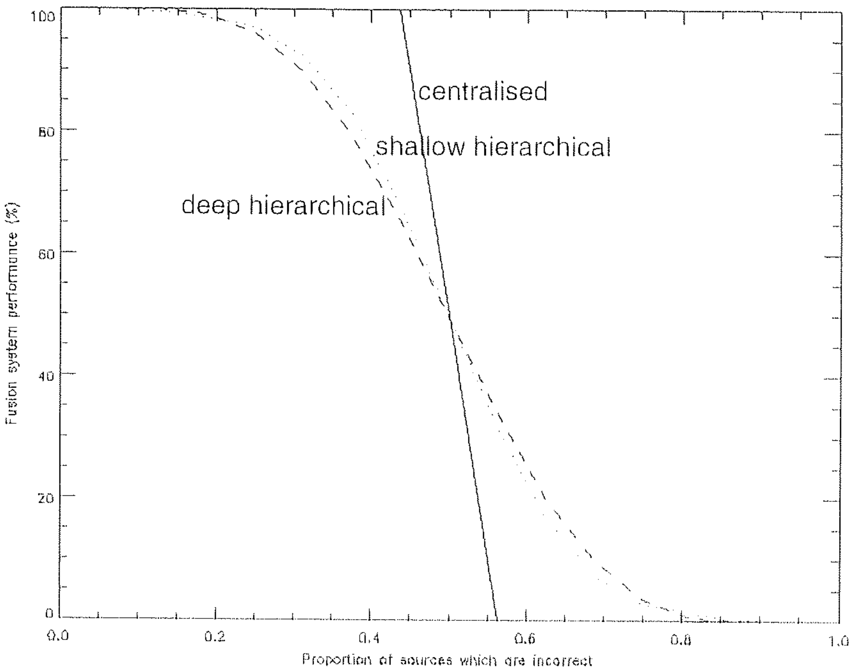

Figure 4-5: The robustness of fusion for centralised and hierarchical topologies.

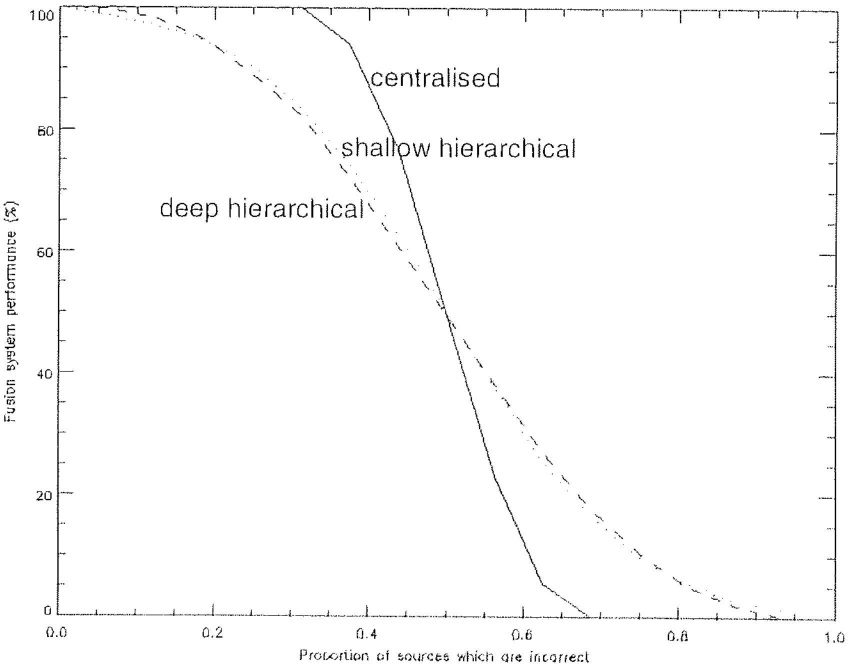

Figure 4-6: The fusion robustness at a deferral rate of 25%.

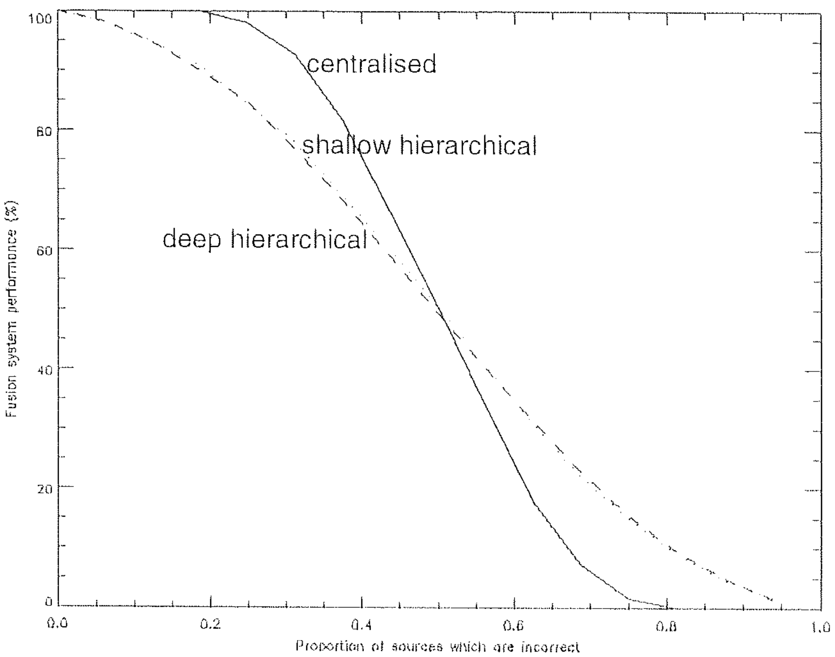

Figure 4-7: The fusion robustness for a deferral rate of 50%.

4.3 *Decision Fusion Formulation*

We shall now describe a formulation of decision fusion as applied to the identification problem. Consider an object recognition task with a set of N target classes, $C \in \{c_1, c_2, \cdots c_N\}$, which has a prior distribution of $P(C) = \{P(c_1), P(c_2), \cdots P(c_N)\}$. A particular sensor, S_i, makes local decisions, $\hat{c}_i \in \{\hat{c}_{i?}, \hat{c}_{i1}, \hat{c}_{i2}, \cdots \hat{c}_{iN}\}$, where $\hat{c}_{i?}$ indicates that sensor i defers its decision. We may later abbreviate $\hat{c}_{ij}$ to $\hat{c}_i$ in the interests of brevity if the particular value of j is unimportant. We characterise the i^{th} sensor by specifying a sensor transition matrix, T_i. The matrix T_i is defined such that the element $t_i(j,k)$ gives the probability that the i^{th} sensor will cause an output of k when the j^{th} object type is actually present. For example:

$$T_1 = \begin{pmatrix} 0.6 & 0.3 & 0.1 \\ 0.2 & 0.7 & 0.1 \end{pmatrix}$$

shows a typical sensor characteristic matrix for sensor 1. The meaning of this description is shown in the table below.

	Sensor output decision		
Actual object class	Object class 1 $\hat{c}_1$	Object class 2 $\hat{c}_2$	Decision deferred $\hat{c}_?$
Class 1 (c_1)	0.6	0.3	0.1
Class 2 (c_2)	0.2	0.7	0.1

Table 4-1: A typical sensor characterisation matrix

The probability that the sensor will correctly identify an unknown object may be obtained by forming a weighted average of the values on the matching diagonal since:

$$P(\hat{c}_i = C) = \sum_{j=1}^{N} P(c_i = \hat{c}_{ij} \mid c_j) P(c_j)$$

which we shall refer to as the expected classification performance. Taking equal priors for the two classes above we obtain an expected classification performance of 65% for T_1. A similar method may be used to evaluate the expected error rate, which in this case is 25%. The remaining 10% of the time sensor T_1 defers its decision.

4.3.1 Maximum A Posteriori Decision Fusion

We shall now develop a rational rule for decision fusion based on these sensor characteristics. In this case the posterior conditional probabilities for each class are computed for each possible set of sensor outputs and the maximum is selected as the fused decision [129]. The strategy minimises the probability of making an incorrect classification given the information provided (the classification labels from the separate sources). The maximum a posteriori (MAP) fused decision is relatively straightforward to calculate given certain knowledge about the behaviour of the sensors. A simple worked example will serve to illustrate the principle. Suppose we have three sensors with transition matrices:

$$T_1 = \begin{pmatrix} 0.6 & 0.2 & 0.2 \\ 0.1 & 0.7 & 0.2 \end{pmatrix} \quad T_2 = \begin{pmatrix} 0.6 & 0.3 & 0.1 \\ 0.4 & 0.5 & 0.1 \end{pmatrix} \quad T_3 = \begin{pmatrix} 0.8 & 0.1 & 0.1 \\ 0.1 & 0.8 & 0.1 \end{pmatrix}$$

Qualitatively, we observe that sensor 1 is often uncertain but otherwise quite well tuned, sensor 2 is less frequently uncertain but also less accurate and sensor 3 is both less uncertain and well tuned. Let us choose a particular set of sensor outputs using the sensor characteristics described above. Assume that sensors 1 and 2 both decide on target number 1 and sensor 3 decides on target number 2. If the sensor outputs are conditionally independent (a strong assumption to which we shall return later) then the probability of this event is given by:

$$P(\hat{c}_{11}, \hat{c}_{21}, \hat{c}_{32} \mid c_1) = P(\hat{c}_{11} \mid c_1) \times P(\hat{c}_{21} \mid c_1) \times P(\hat{c}_{32} \mid c_1) = 0.6 \times 0.6 \times 0.1 = 0.036$$

if the target were of type c_1 and

$$P(\hat{c}_{11}, \hat{c}_{21}, \hat{c}_{32} \mid c_2) = P(\hat{c}_{11} \mid c_2) \times P(\hat{c}_{21} \mid c_2) \times P(\hat{c}_{32} \mid c_2) = 0.1 \times 0.4 \times 0.8 = 0.032$$

if the target were of type c_2.

$P(\hat{c}_{1?}, \hat{c}_{2?}, \hat{c}_{3?} \mid c_1) = 0.002$	$P(\hat{c}_{1?}, \hat{c}_{2?}, \hat{c}_{31} \mid c_1) = 0.016$	$P(\hat{c}_{1?}, \hat{c}_{2?}, \hat{c}_{32} \mid c_1) = 0.002$
$P(\hat{c}_{1?}, \hat{c}_{21}, \hat{c}_{3?} \mid c_1) = 0.012$	$P(\hat{c}_{1?}, \hat{c}_{21}, \hat{c}_{31} \mid c_1) = 0.096$	$P(\hat{c}_{1?}, \hat{c}_{21}, \hat{c}_{32} \mid c_1) = 0.012$
$P(\hat{c}_{1?}, \hat{c}_{22}, \hat{c}_{3?} \mid c_1) = 0.006$	$P(\hat{c}_{1?}, \hat{c}_{22}, \hat{c}_{31} \mid c_1) = 0.048$	$P(\hat{c}_{1?}, \hat{c}_{22}, \hat{c}_{32} \mid c_1) = 0.006$
$P(\hat{c}_{11}, \hat{c}_{2?}, \hat{c}_{3?} \mid c_1) = 0.006$	$P(\hat{c}_{11}, \hat{c}_{2?}, \hat{c}_{31} \mid c_1) = 0.048$	$P(\hat{c}_{11}, \hat{c}_{2?}, \hat{c}_{32} \mid c_1) = 0.006$
$P(\hat{c}_{11}, \hat{c}_{21}, \hat{c}_{3?} \mid c_1) = 0.036$	$P(\hat{c}_{11}, \hat{c}_{21}, \hat{c}_{31} \mid c_1) = 0.288$	$P(\hat{c}_{11}, \hat{c}_{21}, \hat{c}_{32} \mid c_1) = 0.036$
$P(\hat{c}_{11}, \hat{c}_{22}, \hat{c}_{3?} \mid c_1) = 0.018$	$P(\hat{c}_{11}, \hat{c}_{22}, \hat{c}_{31} \mid c_1) = 0.144$	$P(\hat{c}_{11}, \hat{c}_{22}, \hat{c}_{32} \mid c_1) = 0.018$
$P(\hat{c}_{12}, \hat{c}_{2?}, \hat{c}_{3?} \mid c_1) = 0.002$	$P(\hat{c}_{12}, \hat{c}_{2?}, \hat{c}_{31} \mid c_1) = 0.016$	$P(\hat{c}_{12}, \hat{c}_{2?}, \hat{c}_{32} \mid c_1) = 0.002$
$P(\hat{c}_{12}, \hat{c}_{21}, \hat{c}_{3?} \mid c_1) = 0.012$	$P(\hat{c}_{12}, \hat{c}_{21}, \hat{c}_{31} \mid c_1) = 0.096$	$P(\hat{c}_{12}, \hat{c}_{21}, \hat{c}_{32} \mid c_1) = 0.012$
$P(\hat{c}_{12}, \hat{c}_{22}, \hat{c}_{3?} \mid c_1) = 0.006$	$P(\hat{c}_{12}, \hat{c}_{22}, \hat{c}_{31} \mid c_1) = 0.048$	$P(\hat{c}_{12}, \hat{c}_{22}, \hat{c}_{32} \mid c_1) = 0.006$

Table 4-2: The probabilities of the 27 possible sensor output triplets.

In a similar way we may calculate the probability of each of the possible output triplets given one target or the other. The full set of $3^3 = 27$ sensor outputs for target type c_1 is shown in Table 4-2.

Now we require the probability of a particular target given that we observe a specific output triplet. This is easily calculated using Bayes' rule, for example:

$$P(c_1 \mid \hat{c}_1, \hat{c}_2, \hat{c}_3) = \frac{P(\hat{c}_1, \hat{c}_2, \hat{c}_3 \mid c_1)P(c_1)}{P(\hat{c}_1, \hat{c}_2, \hat{c}_3)} = \frac{P(\hat{c}_1, \hat{c}_2, \hat{c}_3 \mid c_1)P(c_1)}{\sum_{i=1}^{N} P(\hat{c}_1, \hat{c}_2, \hat{c}_3 \mid c_i)P(c_i)}$$

and the posterior estimate of the target class c_1 given this declaration is

$$P(c_1 \mid \hat{c}_{11}, \hat{c}_{21}, \hat{c}_{32}) = \frac{0.036 \times P(c_1)}{0.036 \times P(c_1) + 0.032 \times P(c_2)}$$

and similarly for c_2

$$P(c_2 \mid \hat{c}_{11}, \hat{c}_{21}, \hat{c}_{32}) = \frac{0.032 \times P(c_2)}{0.036 \times P(c_1) + 0.032 \times P(c_2)}$$

If the prior probabilities of classes c_1 and c_2 are, say, 0.4 and 0.6 respectively we obtain the MAP decision probabilities of

$$P(c_1 \mid \hat{c}_{11}, \hat{c}_{21}, \hat{c}_{32}) \approx 0.429$$

and

$$P(c_2 \mid \hat{c}_{11}, \hat{c}_{21}, \hat{c}_{32}) \approx 0.571$$

and in this case we declare as class c_2 being the MAP fused decision (despite the case that two out of the three sensors declared the class as being c_1). The MAP fused transition matrix, T^F, can be found by summing the values for all triples which yield a particular fused decision, conditioned on a specific target. For example element T_{11}^F is calculated by summing all values for which the fused decision was target type c_1, when the actual target type was also type c_1. Summing these values (indicated by shading in the table above) results in the probability 0.824. For the above sensor characteristics the full MAP fused transition matrix, T^F, is given by:

$$T^{F_{MAP}} \approx \begin{pmatrix} 0.824 & 0.176 & 0.000 \\ 0.083 & 0.917 & 0.000 \end{pmatrix}$$

Note that the fused decision may not be deferred using this simplest scheme.

The expected error rate of this fused decision-maker may be calculated as approximately 12%. It may be observed that this is better than the first two sensors (at 14% and 36%) but worse than the third sensor, S_3, which has an error rate of just 10%. However, sensor S_3 additionally defers its decision 10% of the time whereas the fused system is forced to make a decision every time.

4.3.2 Maximum Likelihood Decision Fusion

The maximum likelihood estimate of the fused class probabilities may be obtained by ignoring the priors on the target class in equations given for the MAP fusion method [129]. For the same example illustrated in the previous section we have:

$$P(c_1 \mid \hat{c}_{11}, \hat{c}_{21}, \hat{c}_{32}) \approx \frac{0.036}{0.036 + 0.032} \approx 0.529$$

and similarly for c_2

$$P(c_2 \mid \hat{c}_{11}, \hat{c}_{21}, \hat{c}_{32}) \approx \frac{0.032}{0.036 + 0.032} \approx 0.471$$

and so the ML fused decision in this case is in favour of class c_1 (the opposite of the MAP decision). It is readily seen that the ML decision fusion rule and the MAP decision fusion rule are identical when the *a priori* target probabilities for the classes under consideration are equal (or alternatively when we have no evidence to support the hypothesis that they are different). The ML fused transition matrix is calculated in exactly the same manner as detailed in the previous section. In this case the ML fused transition matrix is found to be:

$$T^{F_{ML}} \approx \begin{pmatrix} 0.909 & 0.091 & 0.000 \\ 0.151 & 0.849 & 0.000 \end{pmatrix}$$

and the expected error rate is approximately 13% - slightly worse than that of the MAP decision fusion method but nevertheless comparing favourably with the performance of the individual sensors.

4.3.3 Deferral with MAP and ML Decision Fusion

When characterising the performance of individual sensors we included the possibility of a sensor reporting a state of uncertainty in the class type (or deferring the decision). The same option of deferral may be introduced to either of the decision fusion schemes described in the preceding sections. One simple, yet effective, rule is to require that the probability (either MAP or ML) associated with the preferred class exceeds some pre-set threshold. The higher the threshold is set, the more confident the fusion process is required to be before outputting a fused decision. For example, say the decision deferral threshold is 0.6. This means that only decisions whose probability exceeds 0.6 will be reported. All others will be deferred. Using the sensor characteristics defined in the previous sections we obtain the new fused transition matrices for the MAP estimate:

$$T^{F_{MAP}} \approx \begin{pmatrix} 0.812 & 0.090 & 0.098 \\ 0.075 & 0.848 & 0.077 \end{pmatrix}$$

with an expected error rate of approximately 8% and about 8% of decision being deferred. The corresponding matrix for the ML decision fusion method is:

$$T^{F_{ML}} \approx \begin{pmatrix} 0.812 & 0.084 & 0.104 \\ 0.075 & 0.840 & 0.085 \end{pmatrix}$$

with an expected error rate of just under 8% but over 9% deferrals. Each of these methods now performs better than the individual sensors (both in terms of higher decision accuracy and a lower deferral rate). In reality the value of the threshold would be carefully chosen with due consideration for the detrimental effects of a fast, but incorrect decision versus waiting for a slower (and possibly too slow) correct decision. A rigorous approach to this problem would use utility theory (described later) to set the threshold to that value which minimised the expected cost of the decision (deferral or otherwise). This approach is not expanded on here.

4.3.4 Fair Ballot Decision Fusion

Perhaps the simplest approach to decision fusion is the use of a simple ballot in which each sensor casts a single vote and the majority decision is adopted. This is essentially the rule that was employed during the veto experiments at the start of the chapter. Deferring sensors are assumed to abstain, tied ballots result in a deferral by fusion process.

There are $3^3 = 27$ possible outputs from these three sensors. The list below details each possible triple and the fair ballot fused decision associated with them.

$P(\hat{c}_{1?},\hat{c}_{2?},\hat{c}_{3?} \mid c_1) \rightarrow F_?$	$P(\hat{c}_{1?},\hat{c}_{2?},\hat{c}_{31} \mid c_1) \rightarrow F_1$	$P(\hat{c}_{1?},\hat{c}_{2?},\hat{c}_{32} \mid c_1) \rightarrow F_2$
$P(\hat{c}_{1?},\hat{c}_{21},\hat{c}_{3?} \mid c_1) \rightarrow F_1$	$P(\hat{c}_{1?},\hat{c}_{21},\hat{c}_{31} \mid c_1) \rightarrow F_1$	$P(\hat{c}_{1?},\hat{c}_{21},\hat{c}_{32} \mid c_1) \rightarrow F_?$
$P(\hat{c}_{1?},\hat{c}_{22},\hat{c}_{3?} \mid c_1) \rightarrow F_2$	$P(\hat{c}_{1?},\hat{c}_{22},\hat{c}_{31} \mid c_1) \rightarrow F_?$	$P(\hat{c}_{1?},\hat{c}_{22},\hat{c}_{32} \mid c_1) \rightarrow F_2$
$P(\hat{c}_{11},\hat{c}_{2?},\hat{c}_{3?} \mid c_1) \rightarrow F_1$	$P(\hat{c}_{11},\hat{c}_{2?},\hat{c}_{31} \mid c_1) \rightarrow F_1$	$P(\hat{c}_{11},\hat{c}_{2?},\hat{c}_{32} \mid c_1) \rightarrow F_?$
$P(\hat{c}_{11},\hat{c}_{21},\hat{c}_{3?} \mid c_1) \rightarrow F_1$	$P(\hat{c}_{11},\hat{c}_{21},\hat{c}_{31} \mid c_1) \rightarrow F_1$	$P(\hat{c}_{11},\hat{c}_{21},\hat{c}_{32} \mid c_1) \rightarrow F_1$
$P(\hat{c}_{11},\hat{c}_{22},\hat{c}_{3?} \mid c_1) \rightarrow F_?$	$P(\hat{c}_{11},\hat{c}_{22},\hat{c}_{31} \mid c_1) \rightarrow F_1$	$P(\hat{c}_{11},\hat{c}_{22},\hat{c}_{32} \mid c_1) \rightarrow F_2$
$P(\hat{c}_{12},\hat{c}_{2?},\hat{c}_{3?} \mid c_1) \rightarrow F_2$	$P(\hat{c}_{12},\hat{c}_{2?},\hat{c}_{31} \mid c_1) \rightarrow F_?$	$P(\hat{c}_{12},\hat{c}_{2?},\hat{c}_{32} \mid c_1) \rightarrow F_2$
$P(\hat{c}_{12},\hat{c}_{21},\hat{c}_{3?} \mid c_1) \rightarrow F_?$	$P(\hat{c}_{12},\hat{c}_{21},\hat{c}_{31} \mid c_1) \rightarrow F_1$	$P(\hat{c}_{12},\hat{c}_{21},\hat{c}_{32} \mid c_1) \rightarrow F_2$
$P(\hat{c}_{12},\hat{c}_{22},\hat{c}_{3?} \mid c_1) \rightarrow F_2$	$P(\hat{c}_{12},\hat{c}_{22},\hat{c}_{31} \mid c_1) \rightarrow F_2$	$P(\hat{c}_{12},\hat{c}_{22},\hat{c}_{32} \mid c_1) \rightarrow F_2$

Table 4-3: The fair ballot decisions for the 27 possible sensor output triplets.

The fair ballot fused transition matrix is calculated in exactly the same manner as detailed in previous sections. In this case the matrix is found to be:

$$T^{F_{FB}} \approx \begin{pmatrix} 0.778 & 0.108 & 0.114 \\ 0.093 & 0.783 & 0.124 \end{pmatrix}$$

and the expected error rate is approximately 10% - slightly worse than either the MAP or ML decision fusion methods with deferral. Furthermore, the expected deferral rate is 12%. This is higher than either the MAP or the ML fusion methods.

4.4 *Estimating Sensor Characteristics*

It should have been noted from the above treatment of decision fusion methods that good estimates of the sensor characteristics are required to obtain optimum fused performance. Although much effort has been put into the development of decision fusion algorithms themselves, little thought has been given to the methodologies with which they could be applied in real systems. The sensor characteristics are often obtained by observing sensor performance under known conditions and estimating the proportion events for which the sensor output was correct, incorrect or was deferred. The collection of appropriate data is a major factor in determining the operational performance of data fusion systems.

4.4.1 Lack of Quality Data

In any pattern recognition task there will be an inherent degree of uncertainty, which it is not possible to eradicate. However, this uncertainty can be compounded by an inappropriate choice of database. It is important to understand the implications that a small sample has on the results of sensor characterisation experiments and to maintain a sense of what constitutes a sample of reasonable size.

There are many factors, both theoretical and practical, which determine the amount of data that should be used for this characterisation process. The amount of data that actually is collected will to a large extent be constrained by the data collection budget and the availability of examples. In practice it may be more efficient *to "collect an intermediate size sample of good quality than a large, but messy, dataset."* [45]. These comments notwithstanding the following list of cases would require a large amount of quality data to be collected and used for the individual sensor characterisation:

1. the highest possible performance is required
2. the classification method used is of high complexity
3. high confidence is needed to be placed in the test results
4. the cardinality of the feature set is large
5. the separation between the separate classes is small
6. if little knowledge is available about the problem.

In the following analyses the number of patterns from class i which are available to design the classifier is denoted by N_i. The number of classes is denoted by C and the number of features or measurements present in each pattern (the input dimensionality) is denoted by D.

The partitioning of data into a design set and a test set is a traditional technique for developing a classifier and estimating its performance on unseen data. The more accurate estimators of performance such as the leave-one-out method are often not feasible for complex pattern recognition tasks in which adaptive methods are used. It has been shown [84] that for large samples the optimum partitioning of the data allocates at least 50% of patterns to the test set. This partitioning strategy has been shown to be sub-optimal for small samples as pointed out in later literature [69]. The more training data which is available the better the estimates of the posterior distributions are likely to be (as long as the training data is representative). If the form of the underlying distributions is known, and the mathematics is tractable, then it may be possible to calculate the effect which a small sample size has on the performance and to make appropriate allowances to the output probability estimates. For example, if the data is isotropic-normally distributed then the correct Bayesian posterior distribution is a Student's *t*-distribution [16].

However, if the data distributions, or the models, which are used to fit the data, are more complex, then an analytical approach is not usually feasible. In this case an adaptive technique is sometimes used in which parameters of the model are adjusted on the training set in the hope that they will produce adequate performance on the test set. This assumption is fine if the performance on the training set (re-substitution performance) is similar to the performance on the test set (generalisation performance). It has been shown that the ratio between the number of training samples from each class and the dimensionality of each pattern vector, $N/_D$, is a suitable indicator of the adequacy of the design database. A more complicated scheme has been suggested ([116] for example), in which the model parameters are set using a two part criterion; the usual fit to the data is augmented by a complexity penalty term which moderates the output.

4.4.2 Bias in Re-substitution Performance

It has long since been known that the expected performance of a classifier tested on the data with which it was designed is higher than the performance expected on a test set of patterns drawn from the same distribution as the design set. Some analytic results and considerable experimental verifications are available. Others [68] have investigated the size of this effect.

A number of schemes have been suggested for providing the experimenter with a means of obtaining estimates of the true generalisation performance (the holdout technique or cross validation technique for example). The disadvantage of these and other schemes is that they can impose heavy computational demands. Modern pattern recognition methods such as adaptive networks cannot, in general, use such techniques. With such adaptive methods the efficacy of a classifier is evaluated and the parameters iteratively tuned in order to improve the performance. The usual approach is to divide the available data into a design set and a test set and to hope that the bias introduced by adapting only on the design set does not render the performance as evaluated on the test set useless. The size of design set for which the re-substitution performance does not differ too much from the expected generalisation performance is therefore of great practical importance.

The following computer experiment was performed: N samples were generated from each of two D-dimensional normal distributions with unit variance and separation S. The separation was varied between zero and five standard deviations giving Bayes' classification error rates of 50.0%, 30.8%, 15.9%, 6.7%, 2.3% and 0.6% respectively. D was varied between 1 and 5. The means of the distributions were estimated from the training database and the recognition performance both on the training data and on a large testing dataset consisting of 100,000 examples were measured. The re-substitution classification performance and generalisation classification accuracy was plotted against the indicator N/D. Each classification performance value for a given sample size, dimensionality and separation is the mean of 20 experiments. Figure 4-8 shows the average of the curves for the five values of D (each point is therefore the average of 100 experiments).

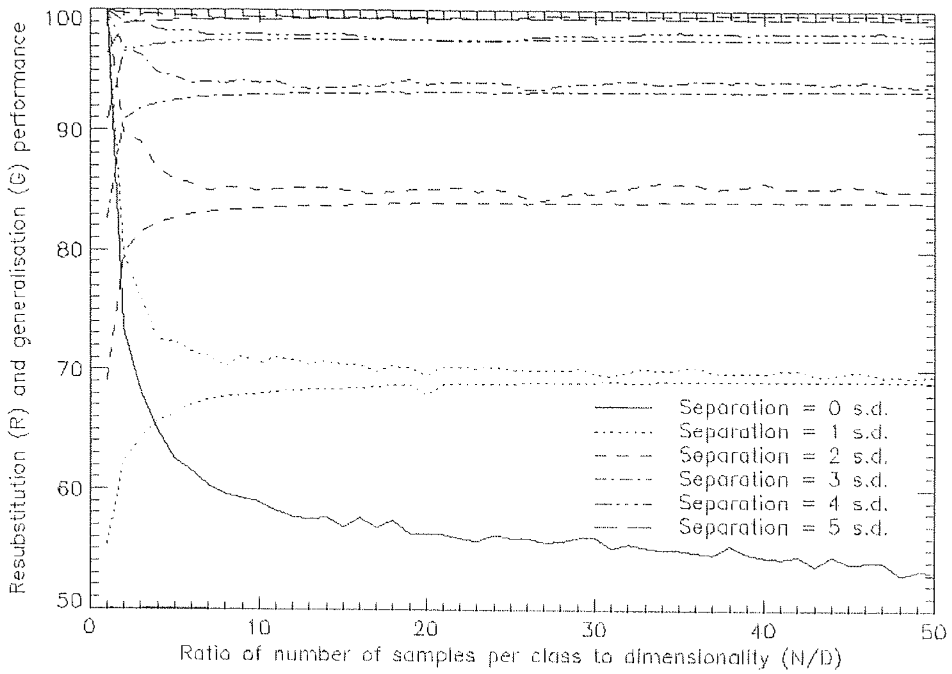

Figure 4-8: The re-substitution and generalisation performance for a Bayes' classifier.

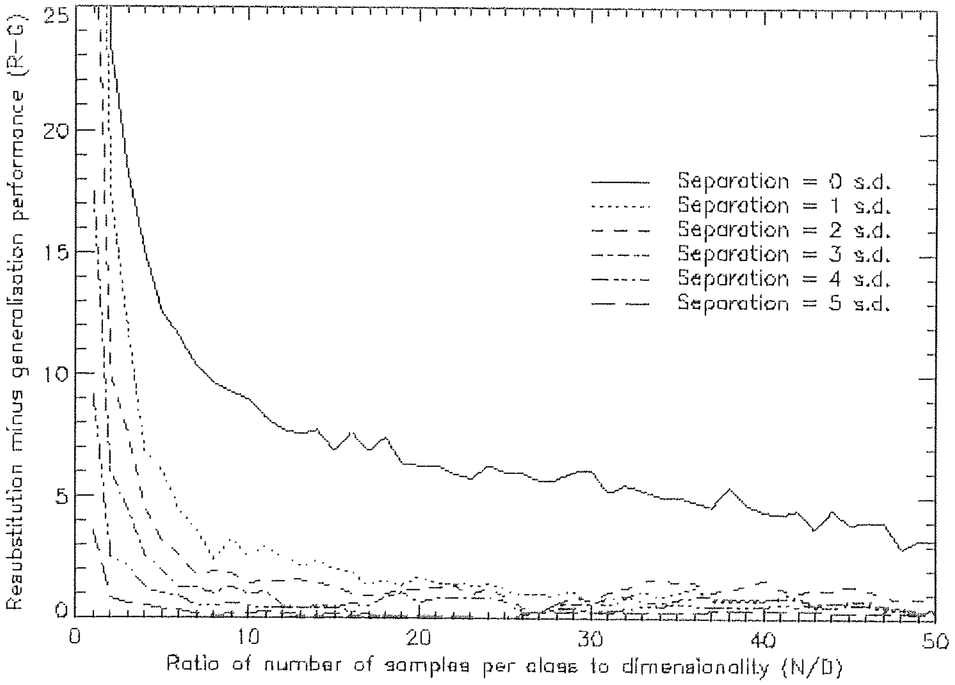

Figure 4-9: The difference between the re-substitution and generalisation performance.

The difference between the re-substitution misclassification error and the generalisation misclassification error is plotted in Figure 4-9. It can be noted that a shoulder is present in these curves at which a further decrease in difference requires considerably more data. The point at which a reasonable amount of data is available depends on the Bayes' error rate but appears to lie between 5 and 10 for the distributions analysed in the experiments described above. This heuristic rule is confirmed by other research [95]. They have analysed the dispersion in the estimate of covariance matrices for multivariate normal distributions. Their results suggested that the "*performance of the estimators [is] statistically reasonable [when the number of training samples per class is] about five times the number of features*". Other research workers [146] have compiled a thorough analysis of the data requirements of five classification techniques from the family of Gaussian classifiers. Their findings again point to the ratio of sample size and dimensionality as the appropriate indicator. Using a hypothetical problem with a Bayes error rate of 1% it was found that an increase in misclassification rate to 1.5% would occur when $N/_D$ was between 1.0 (for the nearest class mean classifier) and 10.0 (for the Gaussian classifier).

The extension of these results to distributions other than normal and to classification problems involving more than two classes is not easy. The results presented above are to be taken as a guide to the minimum amounts of data necessary. More complex distributions will necessarily require a larger number of design patterns.

4.4.3 Confidence in Test Set Performance

It is not sensible to place perfect deterministic confidence in a procedure that is derived from the analysis of a probabilistic event. When designing the system it is wise to determine the confidence that can be placed in it. The degree to which the in-service performance of the classifier can be approximated by the performance on a test set will vary according to: how representative the test set is, how many patterns there are in the test set and how correlated those patterns are. We consider next the case in which the test set patterns are representative and independent. In this case the observed error rate has a binomial distribution.

The confidence, which is to be placed in the performance of the classifier and the number of patterns used to assess that performance, are related. Tighter confidence intervals being associated with greater amounts of test data. The relationship can be used to plan in advance the amount of data that is necessary to provide a desired confidence level; or to determine the confidence interval for a given amount of test data.

Suppose a recogniser is specified with a maximum misclassification rate of 1%. Suppose also that it is required that the user should have 95% confidence that this specification has been met (*i.e.* that the probability that the underlying misclassification rate is <1% is 0.95). Furthermore, suppose that an independent test set is collected and the recogniser makes no errors in classifying patterns from it. How many patterns should the test set contain?

Assuming a flat prior on the misclassification rate we have

$$P(E_G \leq 0.01 \mid E_T, N_T) = \frac{\int_0^{0.01} dE\, P(E_T \mid E, N_T)}{\int_0^1 dE\, P(E_T \mid E, N_T)} = \frac{\int_0^{0.01} dE\,(1-E)^{N_T}}{\int_0^1 dE\,(1-E)^{N_T}} = 1 - 0.99^{N_T+1}$$

which we have specified must be greater than 0.95. This gives

$$N_T > \frac{\log(1-0.95)}{\log(0.99)} - 1 \approx 297$$

The builder of the recogniser would therefore need to collect about 300 patterns for testing the recogniser. If the required misclassification rate was less than 0.1% then the same confidence would necessitate the use of nearly 3,000 test samples, all of which would need to have been correctly classified by the recogniser.

This principle can be extended to provide the desired confidence limits for any measured performance rate. It has been shown that the confidence that should be placed in a classification error is independent of the number of possible classes [84]. We arbitrarily show the workings for the 95% confidence interval - any other interval can be calculated in a similar way. We define the 95% confidence interval to be bounded by a pair of values such that the underlying error rate E_g falls below the lower value with probability no greater than 0.025 and above the higher value, again, with the probability no greater than 0.025. If the patterns in the test data are independent then the probability of misclassification will have a binomial distribution. For large test samples, say greater than 100, this binomial distribution can be approximated by a normal distribution [84] and the calculation of the confidence limits is very straightforward. The same approximation continues to be widely used when the number of test samples is much smaller (see Figure 4-10 for example).

$$B(p, N) \approx N(\mu, \sigma^2)$$

where

$$\mu = Np \text{ and } \sigma^2 = Np(1-p)$$

The 95% confidence intervals can then be estimated by appropriate use of tables or approximations. Figure 4-10 shows these confidence intervals for samples sizes of 2, 5, 10, 15, 20, 30, 50, 100, 250 and 1000 samples.

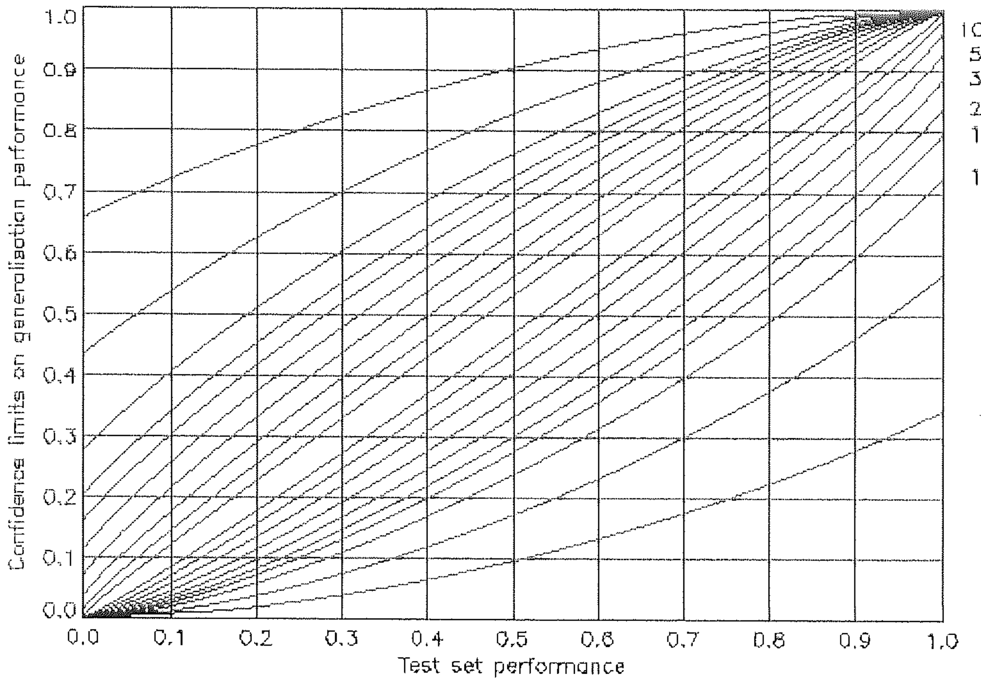

Figure 4-10: The 95% confidence using the Gaussian approximation to the binomial.

For smaller sample size the procedure is somewhat more laborious as outlined in [48][1] and [46]. In such cases we find an approximation to the inverse of the incomplete beta function based on that of Hastings [2] useful. We give only the lower confidence limit, L_l, the upper confidence limit follows from symmetry: $L_u(x) = 1 - L_l(x)$. In this approximation N denotes the total number of patterns in the test set and E_T the proportion of test samples misclassified. The value of y_p is chosen depending on the confidence level - the value of $y_p = 1.96$ used here corresponds to the 95% confidence limit. The approximation holds when at least one test pattern is either correctly or incorrectly classified (see Appendix A.2).

[1] The extensively reproduced confidence limits given in Clopper and Pearson (1934) are based on a linear interpolation of tables from the Medical Research Council's Reports. The resulting confidence intervals are tighter than are strictly justified.

This is shown in Figure 4-11 for *N* equal to 2, 5, 10, 15, 20, 30, 50, 100, 250 and 1000. The graph is read as follows: the measured test set error rate is located on the horizontal scale and the points at which a vertical from it crosses the upper and lower confidence limits for the respective sample size are noted. The values of the 95% confidence limits on the underlying error rate can then be read off the vertical scale. For instance, in a typical recognition experiment using 1300 test patterns with a measured error rate of 30% we obtain a 95% confidence interval of 27.5→32.5%. This method has been used for several applications studies [11], [12] and [14]. It has proved to be a useful technique for assessing significance of results and gaining some feel for the reliability of the predicted error rate.

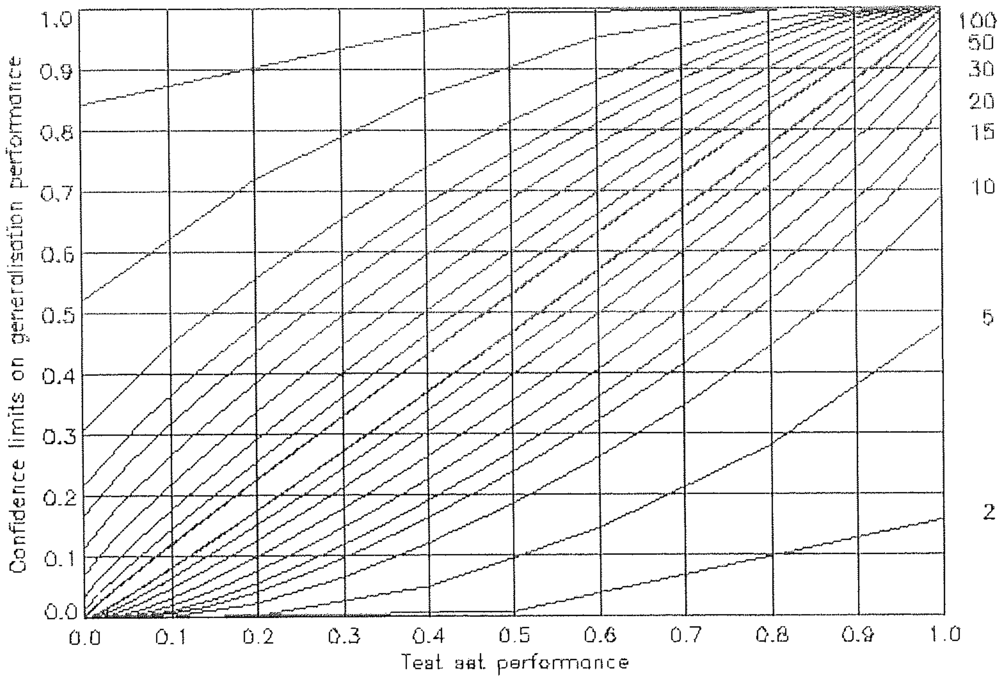

Figure 4-11: The actual 95% confidence intervals for generalisation performance.

The above results can be verified experimentally. Data from a pair of normal distributions separated by one standard deviation were generated. The Bayes error rate for these distributions is 30.8%. A nearest class mean (NCM) classifier was trained on 100 samples and tested on 1000 test sets each of 10, 20, 50 and 100 patterns. The error rates for the 1000 tests were recorded. Figure 4-12 shows the proportion of test results that fell below the specified value. Figure 4-13 shows an enlargement with the proportion 0.025 at the top. The lower confidence limits were found by linear interpolation between the nearest pair of experimental values - note that the curve is concave in this area and a linear interpolation is likely to give an underestimate. The upper limits were calculated in a similar manner. The performance values for which the appropriate proportion of test results failed to equal are given in the accompanying Table 4-4 for a Bayes' classifier with an error rate of 30.8%.

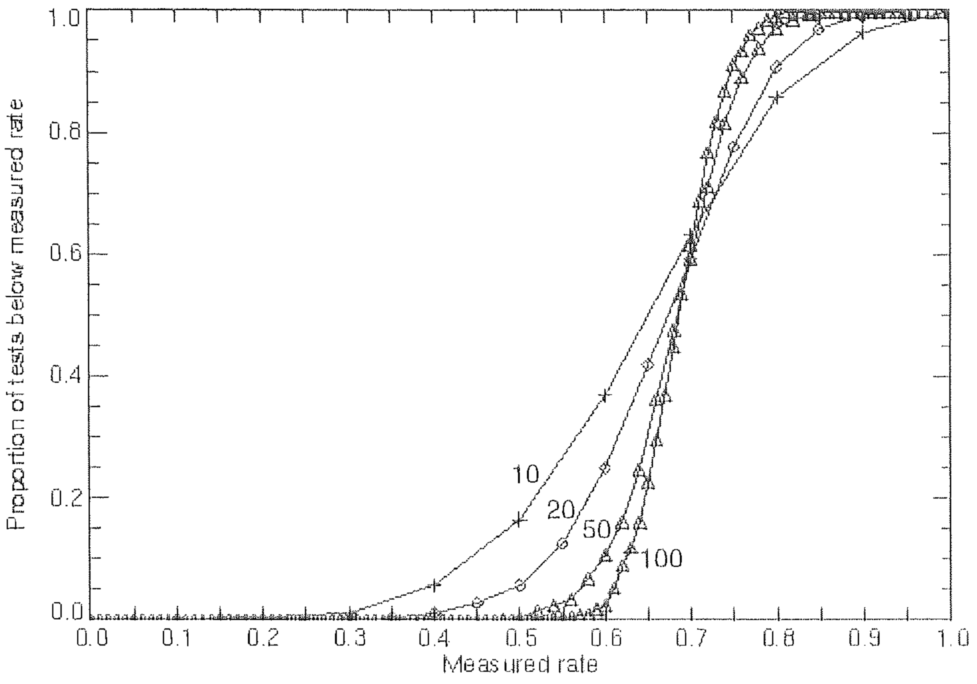

Figure 4-12: Experimental verification of the confidence interval formulae.

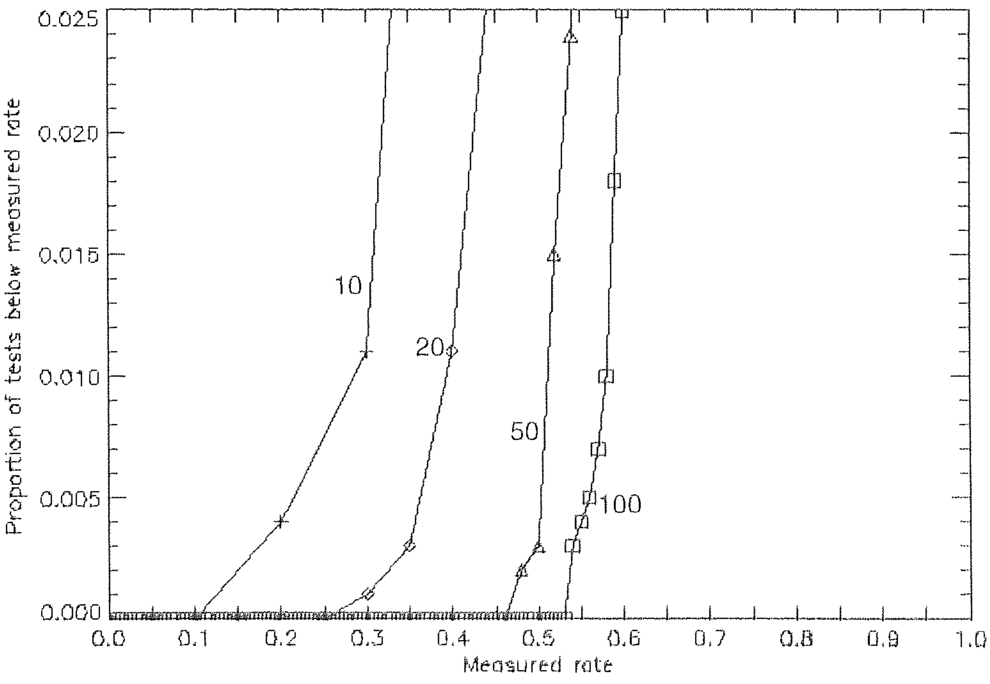

Figure 4-13: An expanded portion of the previous graph.

Sample size	Experimental range	Theoretical range
10	33%→93%	34%→93%
20	44%→87%	45%→88%
50	54%→81%	54%→81%
100	60%→78%	59%→78%

Table 4-4: The 95% confidence intervals for a Bayes' classifier.

4.4.4 The Generalisation / Confidence Dilemma

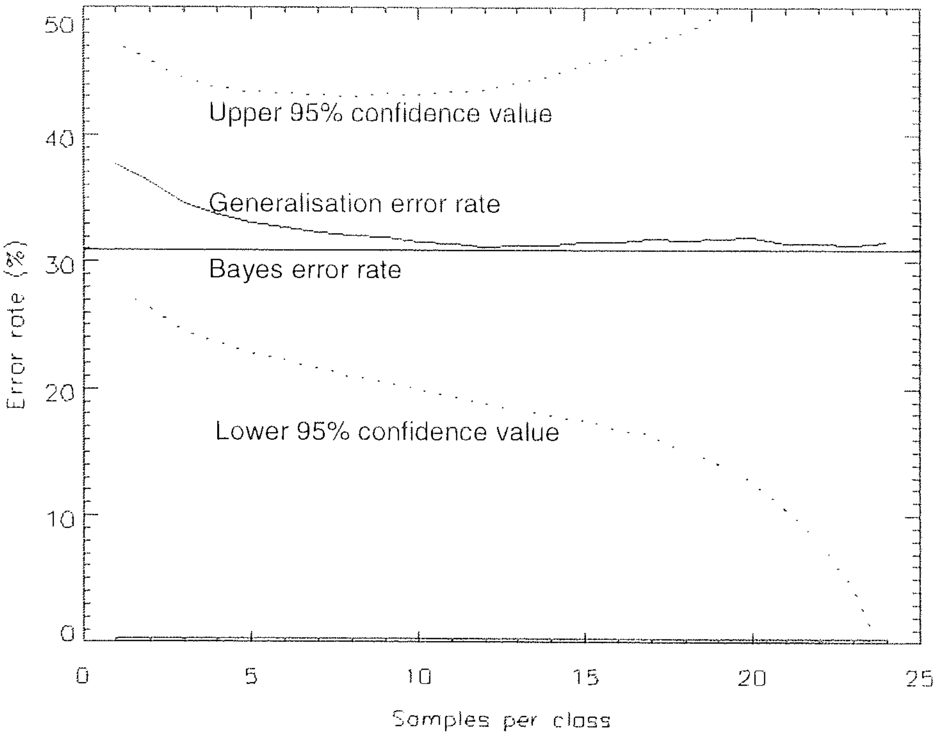

Figure 4-14: The value and confidence of the generalisation error rate.

The two preceding sections quantify two observations:

- ❑ the generalisation error rate is reduced by increasing the sample size of the design database
- ❑ the confidence in the generalisation error rate can be reduced by increasing the sample size of the test data

A dilemma faces the experimenter when there is a fixed number of samples to apportion between the design and test sets. Reducing the generalisation error rate increases the uncertainty in its value, and *vice versa*. The same issue is known as the bias / variance dilemma in the statistics literature. To illustrate the issue we consider a two-class recognition problem for normally distributed data with a separation of one standard deviation. Figure 4-14 shows the underlying error rate (30.9%) together with the estimated error rate (solid line) and the one standard deviation confidence interval for the generalisation error rate (dotted lines).

An appropriate proportion of samples should be allocated to the design and test sets according to circumstances. Unless otherwise stated we shall divide such datasets equally in the remaining experiments.

4.5 *Improved Fused Error Rate Estimation*

Using the ideas presented in the preceding sections we are now better able to estimate the error rate performance of the various decision-level fusion algorithms described at the beginning of this chapter. Let us take the same sensor characteristics described in 4.3.1 :

$$T_1 = \begin{pmatrix} 0.6 & 0.2 & 0.2 \\ 0.1 & 0.7 & 0.2 \end{pmatrix} \quad T_2 = \begin{pmatrix} 0.6 & 0.3 & 0.1 \\ 0.4 & 0.5 & 0.1 \end{pmatrix} \quad T_3 = \begin{pmatrix} 0.8 & 0.1 & 0.1 \\ 0.1 & 0.8 & 0.1 \end{pmatrix}$$

but that we shall further assume that these estimates were produced using 100 samples for T_1, T_2 and T_3. Since the number of samples is relatively large in each case we shall use the Gaussian approximation to the error rate distribution as described above. In this case the standard deviation in the first entry in the characteristic matrix for sensor 1 is:

$$\sigma = \sqrt{\frac{0.6 \times (1.0 - 0.6)}{100}} = \sqrt{0.0024} \approx 0.05$$

and similarly for the other entries.

To illustrate the effect of such errors in the characteristic matrices the following experiment was performed. The three sensor characteristic matrices shown above were used to produce a MAP decision fusion rule. Noise of various levels was then added to the elements of the characteristic matrices and the performance of the original fusion rule was computed for the noisy sensors. The error rate and deferral rate for the fusion centre was recorded for each experiment. Each experiment was repeated 1,000 times and the results averaged. Figure 4-15 shows the result of these experiments graphically. Shown solid is the deferral rate and dotted is the error rate.

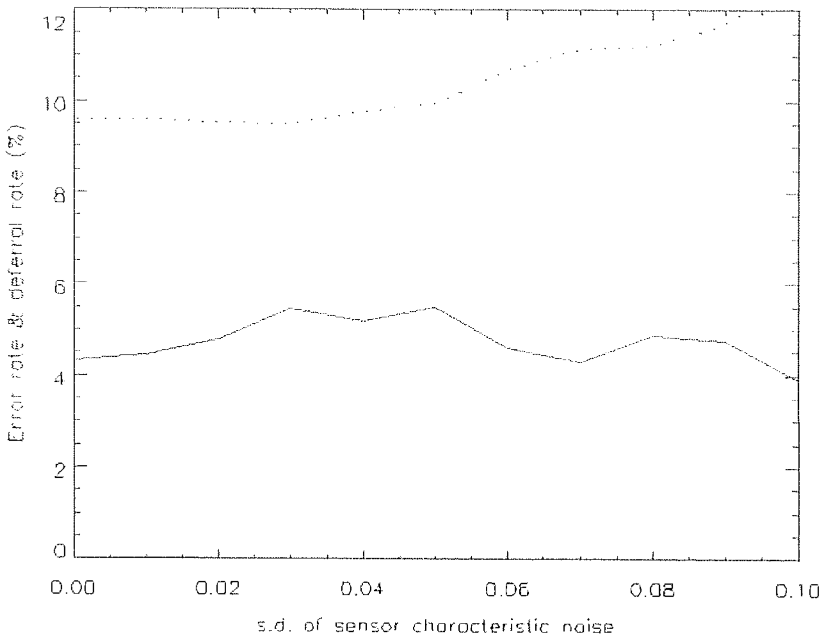

Figure 4-15: Decision fusion error rate and deferral rate for noisy sensors.

It can be seen that the fused error rate increases substantially while the deferral rate remains roughly constant. In the above experiment the decision threshold was held fixed at 0.6 which corresponded to an expected fusion matrix of:

$$T_F = \begin{pmatrix} 0.908 & 0.060 & 0.032 \\ 0.150 & 0.790 & 0.060 \end{pmatrix}$$

The corresponding error and deferral rates were 9.6% and 4.3% respectively. It is, of course, desirable that the deferral rate should be allowed to increase in order to maintain adequate performance in terms of the error rate. A second set of experiments was conducted to illustrate the rate at which the deferral rate alone rises with increased noise on the sensor characteristic matrices. To hold the error rate approximately constant a set of thresholds was evaluated under each condition and the threshold that led to the most appropriate error rate was selected. Figure 4-16 shows the error rate and deferral rate as a function of the standard deviation added to the sensor characteristic matrices. Again the solid line is the deferral rate and dotted line is the error rate (which in this case was held constant at 9.6% by adjustment of the deferral threshold). It can be seen that the increase in deferral rate is approximately linear in the range of noise levels evaluated in the experiments.

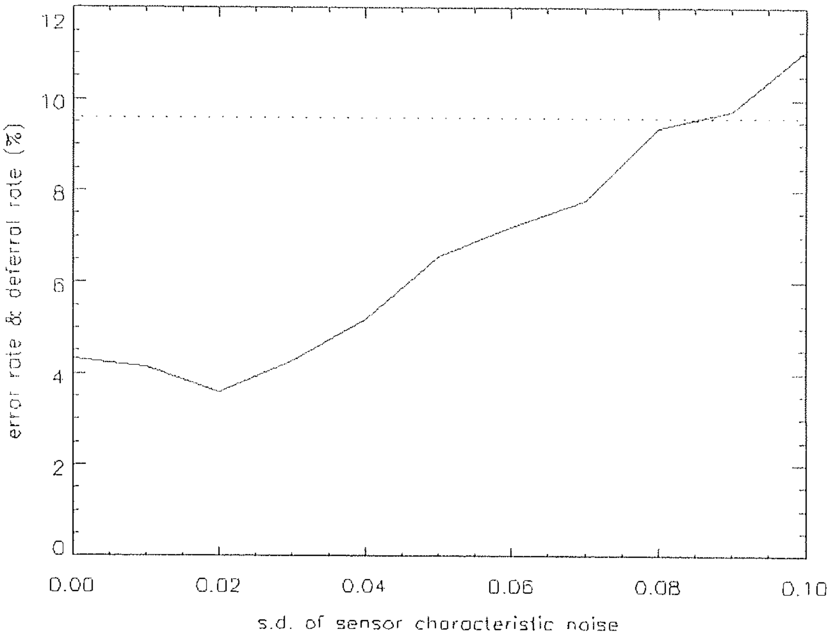

Figure 4-16: Decision fusion deferral rate at constant error rate for noisy sensors.

In practice it would not be possible to evaluate different threshold values for each noisy characteristic matrix since the matrix itself is unlikely to be available directly. We therefore require a method for setting the deferral threshold to maintain a desired error rate for noisy sensors. This can be accomplished by using an adaptive deferral threshold for the decision fusion rule, which monitors performance and changes in such a way as to maintain the expected fusion error rate at desired levels. In order to balance the requirement to obtain a good estimate of the error rate (which would necessitate a large number of observations) against the requirement to quickly set the threshold to an appropriate level we use a sequential probability ratio test. The method is described by way of a specific example in Appendix A.3.

Assume we have $p_1 = 0.105$, $p_2 = 0.095$ and $\varepsilon_1 = \varepsilon_2 = 0.1$ giving $a \approx 0.698$ and $b_1 = b_2 = 0.295$. Applying these tests to an incoming stream of labelled observations yields the behaviour illustrated in Figure 4-17.

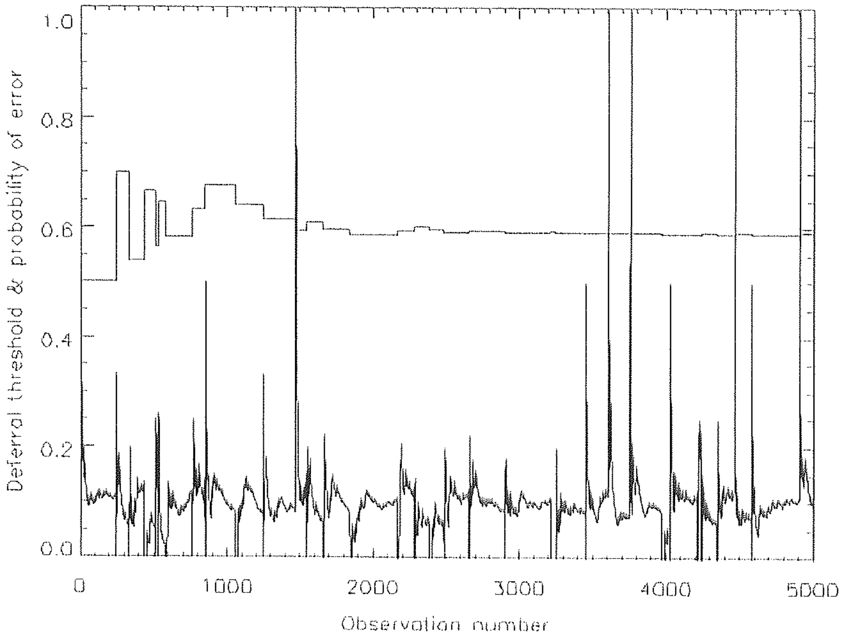

Figure 4-17: Adaptive threshold and error rate using a sequential likelihood ratio test.

In the above experiment the sensor characteristic matrices described earlier in the section were used (without noise). The deferral threshold was initialised at 0.5. The threshold was adjusted by adding or subtracting a threshold step as prescribed by the likelihood test. The threshold step was initially set to 0.25 and was divided by 1.25 each time it was used. It can readily be seen from the graph that the correct threshold is discovered after approximately 2,000 observations. Each discontinuity in the error plot corresponds to a change in deferral threshold.

4.6 *Chapter Discussion*

In this chapter have introduced the concept of the veto effect and shown that some data fusion architectures are susceptible to such an effect when a significant minority of the information sources provide erroneous decision information. We found that the centralised topology was more robust against the veto effect than the hierarchical approach also assessed. In the remainder of the chapter we described a *maximum a posteriori* approach to decision level fusion which makes use of the sensor characteristic matrices. We further showed that the performance of the fusion centre could be controlled by altering the decision threshold. In this way a desired fusion error rate could be maintained even when the underlying sensor characteristics were known only approximately. This scenario corresponds to a realistic system with poorly specified or unreliable sources.

Despite the success of this approach we have some misgivings about the suitability of decision level fusion in circumstances within which the sources are considered to be unreliable. Although the fused error rate can be controlled this is only achieved at the cost of an increase in the proportion of deferrals. It would be better to allow for the source unreliability in such a way that the error rate could be maintained without any increase in deferral rate. In the next chapter we begin to use the probability fusion level as a vehicle for handling this issue. We again focus on an important sub-problem and consider only those systems for which the deferral rate is zero. Adopting this problem reduces the interaction between the performance indicators. We shall see that an auxiliary, local performance indicator such as cross entropy provides a useful metric for assessing such schemes.

Chapter 5 Probabilistic Fusion

5.1 *Chapter Introduction*

In this chapter we turn our attention from decision fusion to probability fusion; the next level of abstraction down in the multi-level data fusion process. We shall motivate the use of probabilities for rational decision making and illustrate the problems introduced by inaccurate (or more specifically, overconfident) probabilities. We then describe the Bayesian fusion algorithm for conditionally independent probabilities and analyse the effect overconfidence has on the fused performance. The reasons why probabilistic fusion is attractive are that the performance is generally good, that the communication of probabilities requires low bandwidth (especially when quantisation is used) and that rational decision making methods based on probabilities are accessible.

We shall examine the issues of the veto effect resulting from unreliable sensor information as it is applicable to probability level fusion. It has been shown in the literature [41] that probabilistic information gives a good compromise between the high performance of feature based data fusion schemes and the low bandwidth of decision level fusion. Certain studies [161] have indicated that a considerable advantage is offered over decision level fusion even when only 1 bit of confidence information is transmitted with each decision. This was further extended by the present author and co-workers [32], who showed that some benefits were worthwhile up to 4 or 5 bits of information for each probability value.

5.2 *Decisions from Probabilities*

In order to make rational decisions automatically one must formulate the decision making process [83]. We define a decision, D, as being a choice from a set of N possible actions $A = \{a_1, a_2 \cdots a_N\}$ with N possibly being infinite. Associated with each action is an outcome $O = \{o_1, o_2, \cdots o_N\}$. An outcome may be mixed *i.e.* that:

$$\begin{aligned} o_1 \quad &\rightarrow \quad o_{1A} \text{ with } \Pr = p \\ &\rightarrow \quad o_{1B} \text{ with } \Pr = (1-p) \end{aligned}$$

Von Neumann and Morgenstern [166] define a quantity they label *utility* which assigns a numerical value to the desirability of a particular outcome as perceived by the decision making party. Therefore:

$$o_1 \text{ is preferred to } o_2 \equiv U(o_1) > U(o_2)$$

Note that the utility framework is not a model of the human decision-making process, rather a prescriptive model for automated decision-making. The utility of a mixed outcome (or its expected utility) is defined to be the sum of the utilities weighted by the probability that each of the outcomes will occur:

$$U(o_1) = P(o_{1A} \mid a_1)U(o_{1A}) + P(o_{1B} \mid a_1)U(o_{1B})$$

A set of utilities can be generated for a set of outcomes O by preferentially ordering the outcomes and arbitrarily setting the utility of any pair of outcomes from O and calculating all other utilities using a lottery model. In finding a utility for o_2 let us assume that the utilities for o_1 and o_3 are already available and that $U(o_1) \geq U(o_2) \geq U(o_3)$. Then if η is a uniform random number on the interval [0,1]:

$$o_2 = o_1 \text{ if } (\eta < p) \text{ and } o_3 \text{ otherwise}$$

which expresses the fact that the decision making party is indifferent about outcome o_2 and a lottery between outcomes o_1 and o_3 with the former occurring with probability p and the latter with probability *(1-p)*. This leads to a means for setting the utility of o_2

$$U(o_2) = pU(o_1) + (1-p)U(o_3)$$

The process is repeated until the utility of all outcomes has been set. Note that the arbitrary starting point means that utility is only a relative measure and is invariant under affine transformations. This is unimportant when a single party uses the framework to make decisions. If a decision is made such that the utility of the outcome is maximised then the decision making can be considered rational:

$$D(A) = a_i \text{ iff } U(o_i) \geq U(o_j) \forall j \neq i$$

Therefore, given a set of actions with associated outcomes it is possible to select that action which maximises the expected utility. For example let assume that a decision is to be made concerning a patient with a suspected diseased appendix. The actions are:

A={operate, don't operate}

The outcomes are mixed depending on the probability p that the patient actually does have a diseased appendix

$$O = \begin{Bmatrix} o_{1A} & o_{1B} \\ \\ o_{2A} & o_{2B} \end{Bmatrix}$$

o_{1A} = diseased appendix is removed with probability p

o_{1B} = patient is unnecessarily risked with probability $(1 - p)$

o_{2A} = burst appendix with probability p

o_{2B} = time and money saved with probability $(1 - p)$

The outcomes are first ordered, let us assume:

$$U(o_{2B}) > U(o_{1A}) > U(o_{1B}) > U(o_{2A})$$

We may set the best outcome to have a utility of 1 and the worst to have a utility of 0. Let us assume that the lottery procedure is followed and that the complete set of utilities is found

$$\begin{aligned} U(o_{2B}) &= 1.0 \\ U(o_{1A}) &= 0.9 \\ U(o_{1B}) &= 0.8 \\ U(o_{2A}) &= 0.0 \end{aligned}$$

Then the expected utility of each of the possible actions is

$$\begin{aligned}\langle U(a_1)\rangle &= 0.9p+0.8(1-p)\\ \langle U(a_2)\rangle &= 0.0p+1.0(1-p)\end{aligned}$$

In making the decision we need to assign a value to p. An approximation to this may be obtained automatically using pattern recognition algorithms (based on sensor and diagnostic attributes for instance). Let us say that the actual probability p that the patient has a diseased appendix is 0.2. Then

$$\begin{aligned}\langle U(a_1)\rangle &= 0.9\times0.2+0.8\times0.8 &= 0.82\\ \langle U(a_2)\rangle &= 0.0\times0.2+1.0\times0.8 &= 0.80\end{aligned}$$

So a rational decision would be to choose action a_1, *i.e.* to operate. However, the probability output by the pattern recognition system is only an approximation based on the statistical analysis of a finite amount of previous data. Let us assume that the probability output by the system is 0.1. The most likely diagnosis is still that the patient is healthy. However, 0.1 is an overconfident probability (too confident that the patient is healthy) which often occurs in real pattern recognition experiments. Now

$$\begin{aligned}\langle U(a_1)\rangle &= 0.9\times0.1+0.8\times0.9 &= 0.81\\ \langle U(a_2)\rangle &= 0.0\times0.1+1.0\times0.9 &= 0.90\end{aligned}$$

The rational decision based on this estimate of the probability is to not operate. However, this overconfident value for the estimated probability has not changed the actual probability (which remains at 0.2) and the utility associated with action a_2 is still 0.8 *i.e.* worse than the utility associated with the optimum action. Therefore, the accurate assignment of values to class conditional probabilities forms an essential and important prerequisite for any probability level data fusion system (or a means of fusing probabilities which is robust to such errors).

5.3 *Cross Entropy Scoring*

In the previous section the importance of accurate probabilities was made clear. Here we define a scoring measure that assesses such accuracy. In order to evaluate the accuracy of the class conditional probabilities produced by the various recognition methods, which are developed, we use the cross entropy score for pattern x:

$$E_{c(c|x)} = -P(c \mid x) \log Q(c \mid x)$$

Where $P(c|x)$ is the probability that the pattern x under consideration belongs to a particular class c and $Q(c|x)$ is the same quantity as estimated by the recogniser. The values of P and Q for all classes must sum to 1:

$$\sum_{i=1}^{C} P(c_i \mid x) = 1$$
$$\sum_{i=1}^{C} Q(c_i \mid x) = 1$$

and under such circumstances E_c is minimised when P=Q. To evaluate the cross entropy of a distribution this quantity is integrated over the distribution. If evaluating the cross entropy given a labelled test set, the values of P are usually either 0 or 1 since the pattern is often known to have belonged to a particular class.

5.4 *Independent Probability-Level Fusion*

By assuming conditional independence of the class probabilities we may greatly simplify the data fusion process for probabilities. If sensor data x and y are conditionally independent given class c we have:

$$P(x, y \mid c) = P(x \mid c)P(y \mid c)$$

It then follows (see Appendix A.4) that

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

the latter part of which is independent of the class c and so can be treated as an unknown constant which may be recovered by normalising over all classes. We shall proceed to use this result for the remaining treatment of probability level data fusion. It should be noted, however, that the assumption of conditional independence is not generally particularly accurate. The fusion of correlated probabilities has been dealt with separately in [15], [132], [133] and [134] and will not be addressed in this thesis.

5.5 *Overconfidence*

Human operators and the automated machines they use are often subject to overconfidence. Overconfidence occurs when the degree of belief associated with a hypothesis is greater than the evidence strictly supports. In the case of probability fusion the issue of overconfidence applies to the probabilities themselves. We have already seen that inaccurate probabilities can lead to irrational decisions. Overconfident probabilities are inaccurate probabilities which stem from the use of insufficient quantities of design data or when the test data and design data have different distributions and that difference was not allowed for.

5.5.1 Overconfidence Owing to Insufficient Design Data

Consider the following synthetic, two-class problem in which samples from each class are generated from a normal distribution. The separation between the means of the two classes is exactly one standard deviation and the Bayes error rate is therefore 30.9%.

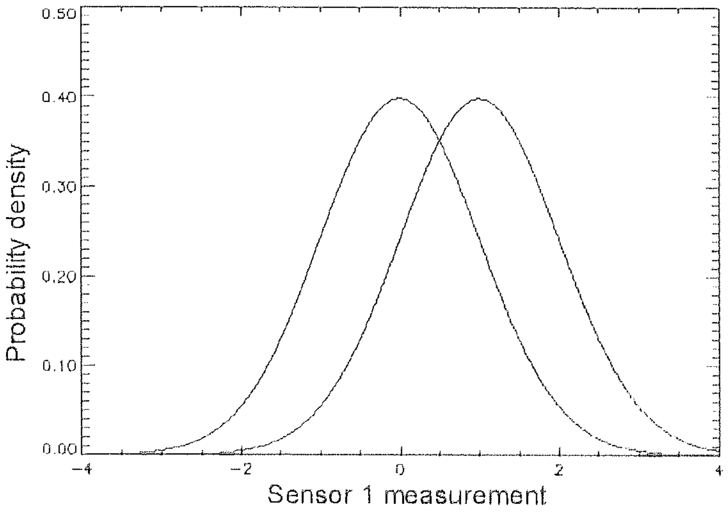

Figure 5-1: The normal distributions used to generate the data.

Figure 5-1 shows the underlying distributions of two classes in a synthetic NCTR problem. Data from each class follow unit normal distributions with means of zero and one.

Data from each class distribution were obtained and maximum likelihood estimates of the underlying distributions were calculated. Figure 5-2 shows these probability densities estimated from a small design dataset (in this extreme case comprising only 10 samples from each class). Note that the sample means and sample standard deviations vary quite considerably from the equivalent parameters of the underlying distributions. Figure 5-3 shows the conditional probability of class 2 that results from these distributions with the correct conditional probability superimposed. The probability values, which quickly saturate to zero outside the region occupied by the design data (the actual probability should actually be close to one for values of the measurement greater than 2.0).

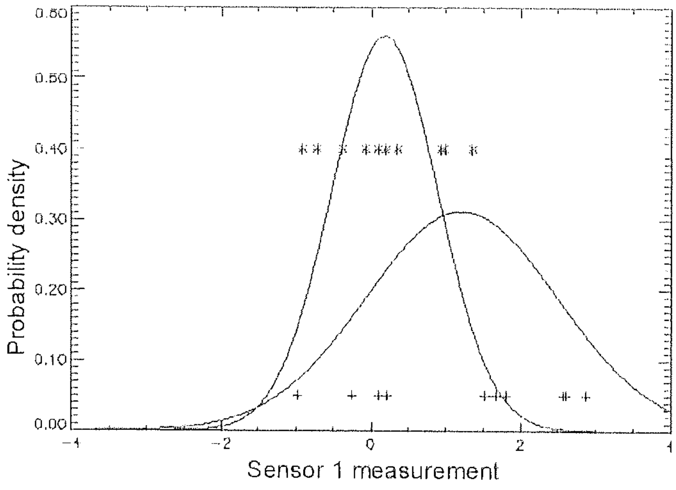

Figure 5-2: The samples from the first sensor and the maximum likelihood estimates of the probability density.

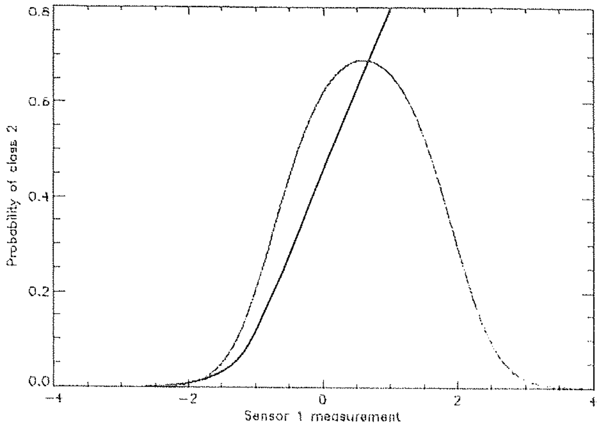

Figure 5-3: The maximum likelihood conditional probability of class 2.

In actual data fusion systems the sensor measurements are typically of much higher dimensionality and the model correspondingly more complex. In this case the number of samples required of a suitable design dataset is large and often expensive to obtain (sometimes prohibitively so). This source of overconfidence, however, is particularly amenable to distributed moderation, which will be covered in a later section.

5.5.2 Overconfidence Due to Unrepresentative Data

Even when large quantities of design data are available it is usually difficult to ensure that they are representative of data to be encountered in service. This may be because of changes in environment between the collection of the design data and the fielding of the system or through active measures of camouflage and deception by the target under scrutiny. We may assume that a greater variety of conditions will tend to broaden the distributions describing the likelihood of particular measurements. Possible effects of camouflage and deception are that the target distribution is broadened and moved closer to the non-target distribution.

5.5.3 Distribution Changes with Probability-Level Fusion

We desire that our fusion system should be robust to the data fused in service being drawn from a different distribution to that encountered during the design of the classifiers and the fusion rule. It is conceivable that in many situations some of the class distributions will remain the same, whereas others will differ from the design distributions. For example, if one class corresponds to background clutter and a second to targets of interest, then the background distribution may be quite constant whereas the target distribution would evolve owing to changed circumstances (equipment modifications or wearing out of machinery).

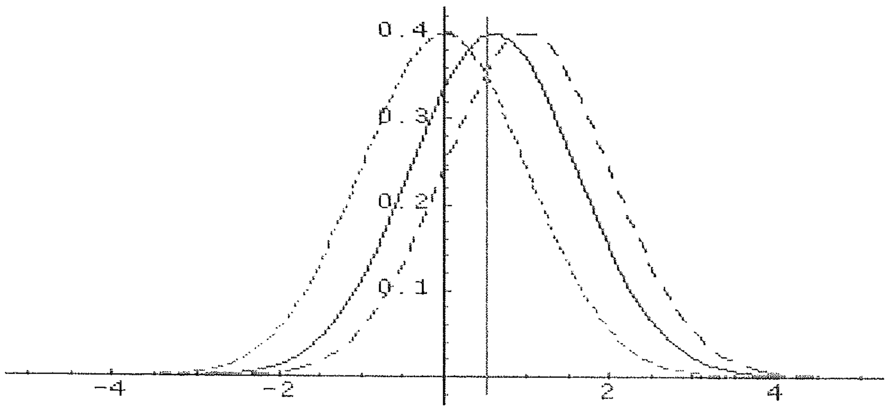

Figure 5-4: An example of changing sensor distributions by varying the mean.

This is illustrated in Figure 5-4 for a typical scenario. The dotted curve shows the probability density of samples from the first class (clutter), for both the design and test cases. The dashed line shows the same quantity for the second (target) class. This is shown as migrating towards the clutter class during the test phase (shown solid). The vertical line shows the Bayes' decision boundary produced during the design phase.

In the experiments described in this section we assume that data from just two classes (A and B) is normally distributed. Without loss of generality we further assume that data from class A follow a standard normal distribution with mean zero and unit standard deviation. The distribution for class B is assumed to have a mean greater than zero. We shall also set the prior to be equal, $P(A) = P(B) = \frac{1}{2}$. To assess the effects of changing distributions on a probability-level fusion process we shall generate data from multiple sensors. The sensors are assumed to be conditionally independent and have identical characteristics.

We shall use the following discriminant function for assigning test points to either class A or B which may be obtained by taking the logarithm of the ratio of the class conditional probabilities of the two classes:

$$g(\underline{\mathbf{x}}) = (\sigma_{des}{}^2 - 1)\sum_{i=1}^{n}\left(x_i + \frac{\mu_{des}}{\sigma_{des}{}^2 - 1}\right)^2 - 2\sigma_{des}{}^2 \log\left(\frac{P(A)}{P(B)}\right) - 2n\sigma_{des}{}^2 \log(\sigma_{des}) - \frac{n\mu_{des}{}^2\sigma_{des}{}^2}{(\sigma_{des}{}^2 - 1)}$$

assigning patterns to A if this function is positive and to B otherwise.

To quantify the amount of change between the design and test distributions we use the Kullback-Leibler number [102], [103] and [160]. In general the Kullback-Leibler number of two distributions is:

$$K_{1,2} = \int_{-\infty}^{\infty} dx \left(\ln \frac{P_1(x)}{P_2(x)} \right) P_1(x)$$

For two Gaussian distributions $N(\mu_1, \sigma_1^{\;2})$ and $N(\mu_2, \sigma_2^{\;2})$, the Kullback-Leibler number between the first and second distribution is given by:

$$K_{1,2} = \frac{1}{2}\left(\frac{\sigma_1^{\;2}}{\sigma_2^{\;2}} - 1 \right) + \frac{(\mu_1 - \mu_2)^2}{2\sigma_2^{\;2}} + \frac{1}{2}\log\left(\frac{\sigma_2^{\;2}}{\sigma_1^{\;2}} \right)$$

For example, we take class *B* distribution to have a design mean standard deviation of one and change this distribution during testing. For the test phase the standard deviation of *B* remained at one but the mean was altered to move closer to that of *A* as previously illustrated in Figure 5-4. This arrangement was analysed for a single sensor system and also for two and ten sensors using independent probability-level fusion as described earlier. Table 5-1 shows the error rate as a function of Kullback-Leibler number for each of the data fusion system. The same quantities are shown graphically in Figure 5-5. Note that the problem is getting harder (the distributions are less separated) and that this is exacerbated by using the wrong model to estimate the class probabilities. Fused classifiers are shown to mitigate this effect.

Actual mean	**Kullback-Leibler number**	**Error rate (1 sensor)**	**Error rate (2 sensors)**	**Error rate (10 sensors)**
1.0	0.000	30.9%	24.0%	5.7%
0.9	0.005	32.7%	26.3%	8.0%
0.8	0.020	34.5%	28.8%	11.4%
0.7	0.045	36.5%	31.4%	16.0%
0.6	0.080	38.4%	34.25	21.6%
0.5	0.125	40.4%	37.0%	27.8%

Table 5-1: The fused error rate system as a function of a design and test mean change.

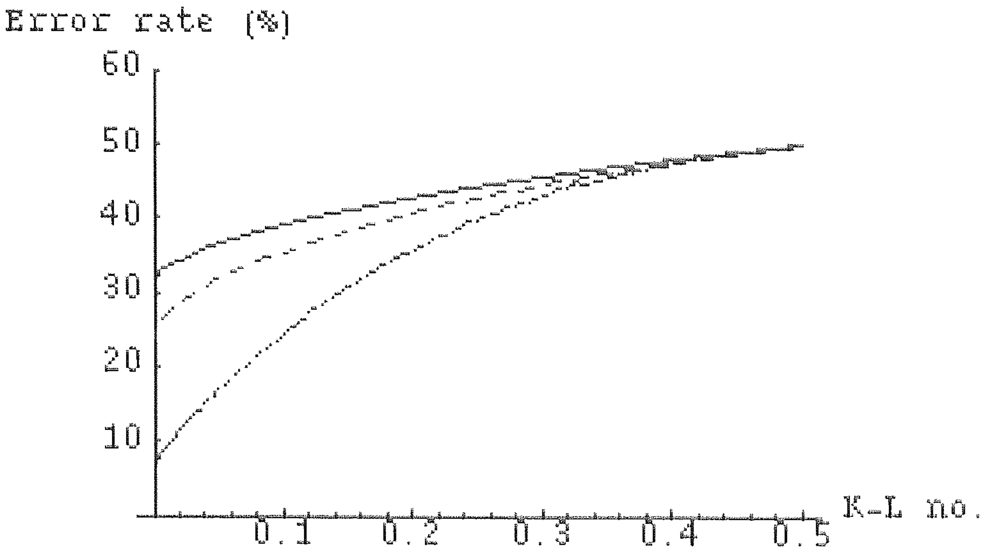

Figure 5-5: The fused error as a function of difference by varying the mean.

In this case the primary reason for the increase in fused error rate is the increasing overlap between the two distributions (and not the difference between the design and test distributions for class *B*). The difference in error rates for fusion systems using the original design distributions and using the correct test distributions is therefore of interest. Figure 5-6 shows this difference as percent increase in error rate as a function of the Kullback-Leibler number. Note that the initial degradation caused by the use of the wrong distributions is soon overtaken by the inevitable difficulty in separating the classes at all.

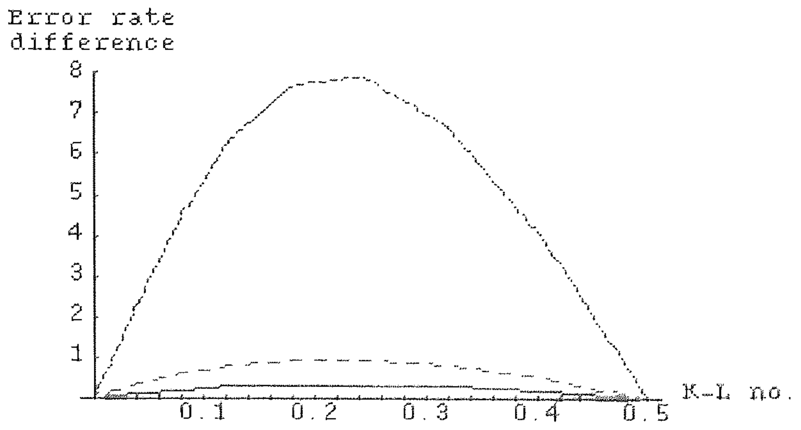

Figure 5-6: The difference in fused error rate produced by changing the mean.

The same Kullback-Leibler numbers could be produced in many different ways (not just by moving the means). A second analysis in which the mean for class B remained at one but the standard deviation increased as illustrated in Figure 5-7. The numerical results in

Table 5-2, and the graph in Figure 5-7, shows that the specific effect depends on the type as well as amount of difference between design and test distributions.

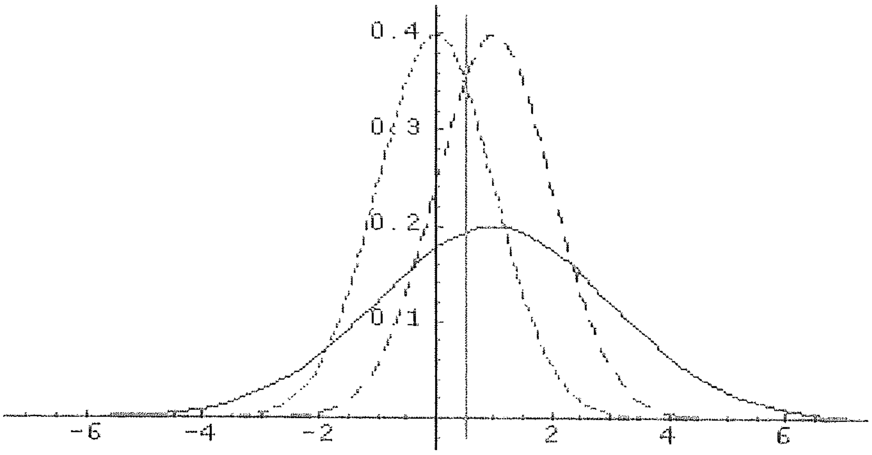

Figure 5-7: Changing sensor distributions by altering the standard deviation.

Actual standard deviation	**Kullback-Leibler number**	**Error rate (1 sensor)**	**Error rate (2 sensors)**	**Error rate (10 sensors)**
1.00	0.00	30.9%	24.0%	5.7%
1.25	0.04	32.7%	26.3%	8.0%
1.50	0.13	33.9%	27.9%	10.1%
2.00	0.32	35.5%	30.1%	13.6%

Table 5-2: Fused error rate produced by varying the standard deviation.

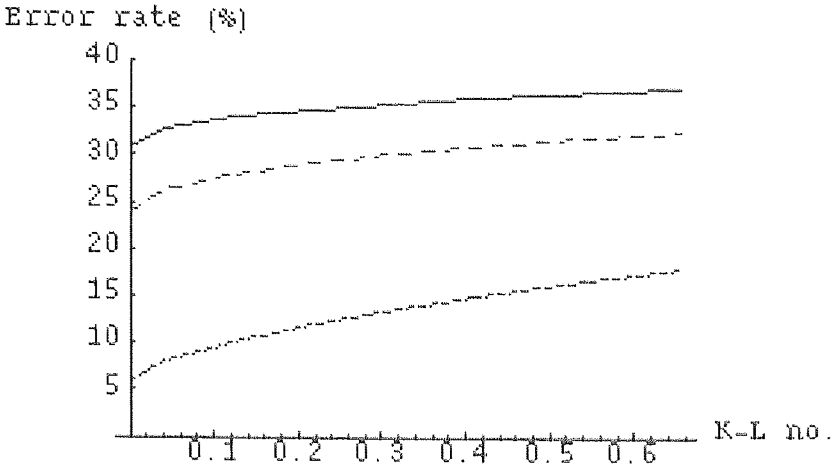

Figure 5-8: Fused error rate for as a function of Kullback-Leibler number.

It can be seen that the amount of degradation for the same Kullback-Leibler number is not as large for the broadened distribution as it was for the displaced distribution. However, unlike the earlier situation, as the test distribution for class B is broadened the intrinsic overlap between the two classes decreases; this is because the two classes have means which are quite close together. The task of separating the classes, therefore, eventually becomes easier. The degradation in fused performance is almost solely due to the inappropriate classifier being used.

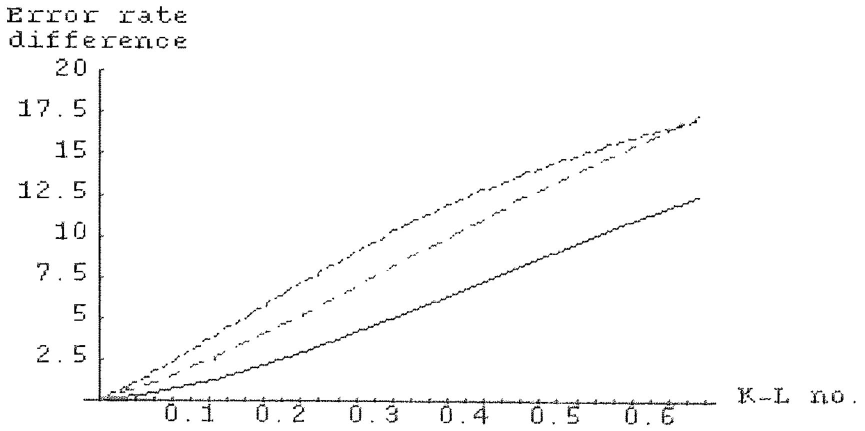

Figure 5-9: The difference in fused error altering the standard deviation.

Figure 5-9 is on a different vertical scale to Figure 5-6 but otherwise shows the same quantities. It should be noted that the initial increase in fused error rate in each case is approximately the same. The asymptotic behaviour depends on whether the mean or standard deviation was different between the design and test distributions.

5.5.4 Summary of Effect of Distribution Changes

It has been shown that under certain configurations of distributions a substantial degradation in fusion performance may be observed for quite small changes in the underlying distributions. We shall therefore address the technique of moderation since this down-weights the contribution of information that may have been derived from such distributions.

5.6 *Chapter Discussion*

We have seen that probability estimates can be used as the basis for rational decision making. Using a Bayesian probability fusion rule for conditionally independent sources we then examined the effect that inaccuracies play in the probability fusion process. It was demonstrated that the effect can be significant for relatively minor inaccuracies as might be caused by lack of design data or migration of the underlying distributions. The effect can be likened to the veto effect described in the chapter on decision fusion. In the next two chapters we shall examine methods for producing more accurate probability estimates and in Chapter 8 we describe an algorithm for allowing for inaccurate probabilities at the fusion centre itself.

Chapter 6 Analytic Moderation for Probability Fusion

6.1 *Chapter Introduction*

The production of accurate probabilities forms the basis of sound, rational decision making. All too often the probabilities produced by both manual and automatic methods are inaccurate. In most cases the probabilities are overconfident *i.e.* that the most likely class has too high a probability associated with it and all others are too low. This stems from a widespread underestimation of the effects that are produced by finite data samples. In this chapter we define the concepts of the *true posterior*, the *maximum likelihood / maximum a posteriori* estimates and compare the two in qualitative and quantitative terms. We take a generating model (referred to as *M*) with some parameters, θ_M, which generates some data, *D*. For example, the model might be a normal distribution with parameters μ and σ^2 corresponding to the population mean and population variance (not to be confused with the sample mean and sample variance which we would denote by $\hat{\mu}$ and $\hat{\sigma}^2$). If the parametric form of the generating model is known the posterior class conditional probability can be computed by marginalising over the unknown parameters, θ_M

$$P(C=c \mid x, D, M) = \int d\theta_M \;\; P(C=c \mid x, D, M, \theta_M) P(\theta_M \mid x, D, M)$$

i.e. the sum of particular class conditional probabilities for all possible values of the parameters weighted by the probability that that parameter set could have produced the observed data. Furthermore, if the form of the generating model, *M*, is not known then this must also be marginalised to obtain the correct posterior distribution:

$$P(C=c \mid x, D) = \int dM \int \mathrm{d}\theta_M \;\; P(C=c \mid x, D, M, \theta_M) P(\theta_M \mid x, D, M) P(M \mid x, D)$$

Since the class conditional probabilities tend to be less extreme using this paradigm they are often referred to as moderated probabilities. In many cases this integral is approximated by taking the maximum likelihood parameters of the maximum likelihood model (in practice this is often approximated by the model and its parameters which results in the most favourable performance when measured on test data):

$$P(\theta_M^* \mid x, D, M) = \text{argmax}_{\theta_M} P(\theta_M \mid x, D, M)$$

and

$$P(M^* \mid x, D) = \text{argmax}_M P(M \mid x, D)$$

In cases where the actual generating model is tested and the amount of data is large enough to ensure that the probability of the most likely set of parameters is dominant

$$P(\theta_M \mid x, D, M) \approx 0 \forall \theta_M \neq \theta_M^*$$

and

$$P(M \mid x, D) \approx 0 \forall M \neq M^*$$

then the use of the maximum likelihood parameters θ^* and the maximum likelihood model M^* will not incur a high penalty. However, for relatively small amounts of data, as found in many real pattern recognition problems, there will be a loss in accuracy. For applications where accuracy is critical then the maximum likelihood model should be discarded in favour of the true posterior (or a better approximation to it). The true *a posteriori* probability can, in some cases, differ significantly from the probability arrived at using the maximum likelihood parameters of the known distribution. However, in most cases an analytic solution for the true posterior is either impossible or computationally intractable. In these cases an approximation is made which results in a posterior distribution which exhibits some of the traits of the true posterior without necessitating the computational loads imposed by the exact solution.

We shall attack the double integral given above in two parts: firstly the integral (marginalisation) over model parameters and secondly the marginalisation over the models themselves. It will be shown that potential improvements over the best model with maximum likelihood parameters are available in both cases separately and in concert.

6.2 *Analytic Moderation for Normally Distributed Data*

In certain, simple, cases it is possible to analytically integrate the hidden parameter variables and to produce the true posterior given the data. The range of models for which this is possible is limited to Gaussian distributions (which form the basis of a range of automatic pattern recognition methods). We shall work through a few examples of simple generator distributions for which the true posterior distribution can be calculated analytically. We use

Bayes' rule to replace the class conditional probabilities with the data probability (suitably normalised) and work in this domain using a density model for each class. If the priors on the mean and variance are uniform and $1/_{\sigma^2}$ respectively, then it can be shown [16] that the moderated probabilities follow a *t*-distribution:

$$\frac{(x-\hat{\mu})}{\hat{\sigma}}\sqrt{\frac{N-1}{N+1}} \Rightarrow t_{N-1}$$

W. S. Gossett first discovered this result in 1908 and published under the *nom de plume* of "Student". The posterior of x for various values of N is shown in Figure 6-1.

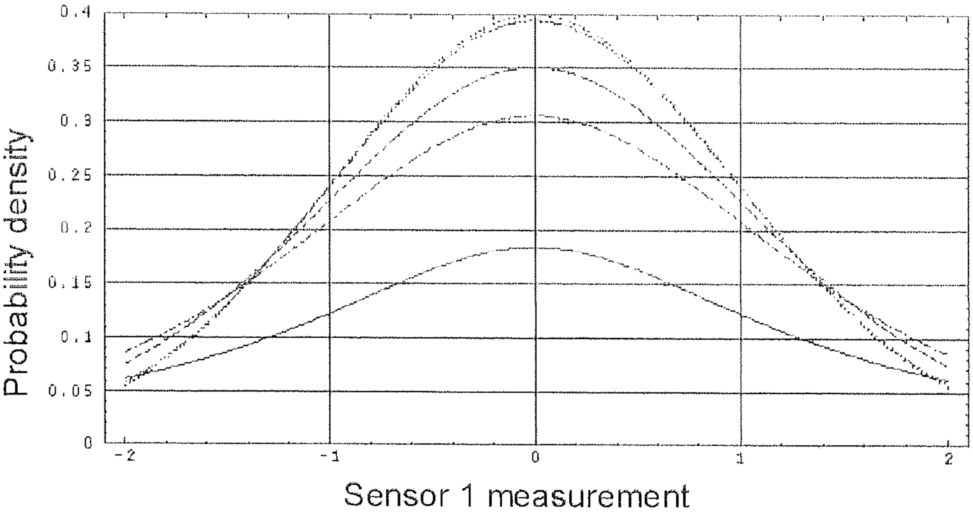

Figure 6-1: Posterior of a normally distributed variable.

Again, the class conditional probabilities follow easily from Bayes' rule:

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

and are illustrated in Figure 6-2 and Figure 6-3.

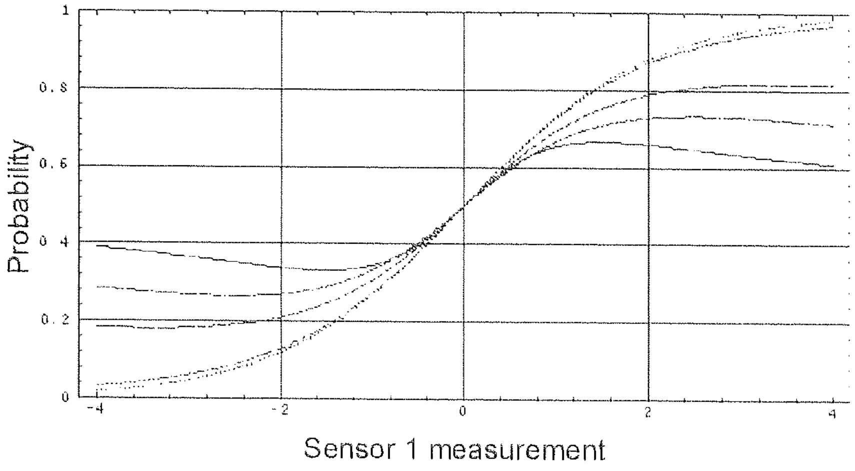

Figure 6-2: Posterior conditional probability for normally distributions.

Figure 6-2 shows the posterior conditional probability of a sample having been drawn from a normal distribution $N(\mu,\sigma^2)$ as opposed to $N(-\mu,\sigma^2)$ given that the sample means (located at $\pm\frac{1}{2}$) and the sample variances (equal to 1) were calculated from *N* samples with *N=2, 5, 10, 100* (shown solid, dot-dashed, dashed and dotted respectively). The unmoderated conditional density is shown dotted faintly and is visible alongside the plot for *N=100*.

Similar results may be obtained for various types of Gaussian distributions in multiple dimensions. Posterior distributions for spherical, right elliptical (in which the principal axes of the distribution are aligned with those of the co-ordinate system) and multivariate Gaussians result in various *t*-distributions.

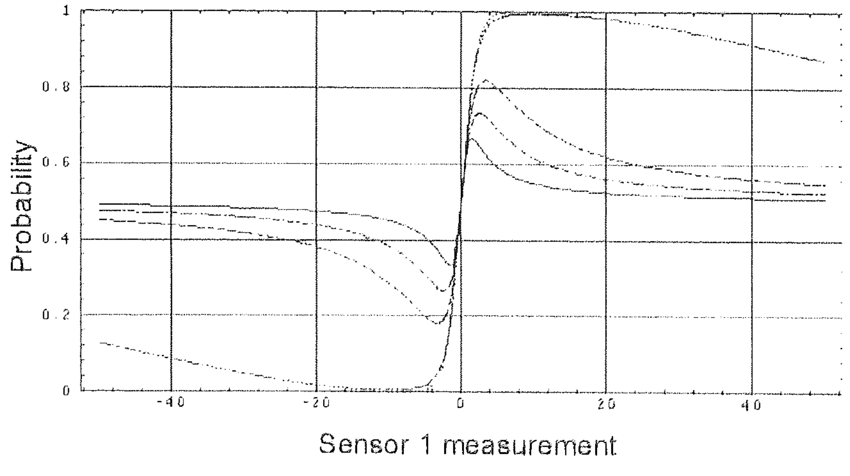

Figure 6-3: The same conditional probability as Figure 6-2 on a broader scale.

6.2.1 Experimental Results on Normally Distributed Data

The work of the previous two sections has provided us with techniques for moderating the probability estimates of Gaussian classifiers. A set of databases, sampled from Gaussian distributions, were used to evaluate the performance gains expected from utilising the moderated Gaussian classifiers rather than the unmoderated classifiers. Each database was segmented into two equal parts comprising a training database and a test database. The sample mean and sample variance for each attribute in the training data was used to implement a set of methods, which were used to classify the patterns in the test data. A tabular summary of the databases is show in

Table 6-1.

We will compare the accuracy of (fused) probabilities from moderated and unmoderated classifiers. One could use error rate but this does not reflect the accuracy of the class probabilities. We therefore also measure performance using cross entropy (as defined in section 5.2). Since the distributions overlap it is unclear what constitutes a good cross entropy score. We therefore, as a comparison, also provide the cross entropy error for a Bayes' classifier (with full knowledge of the underlying distributions).

Database	Number of dimensions	Number of classes	Bayes error rate	Bayes cross entropy
1D2C	1	2	30.9%	0.58
2D4C	2	4	30.9%	0.75
3D8C	3	8	30.9%	0.80

Table 6-1: Summary of the normally distributed evaluation datasets.

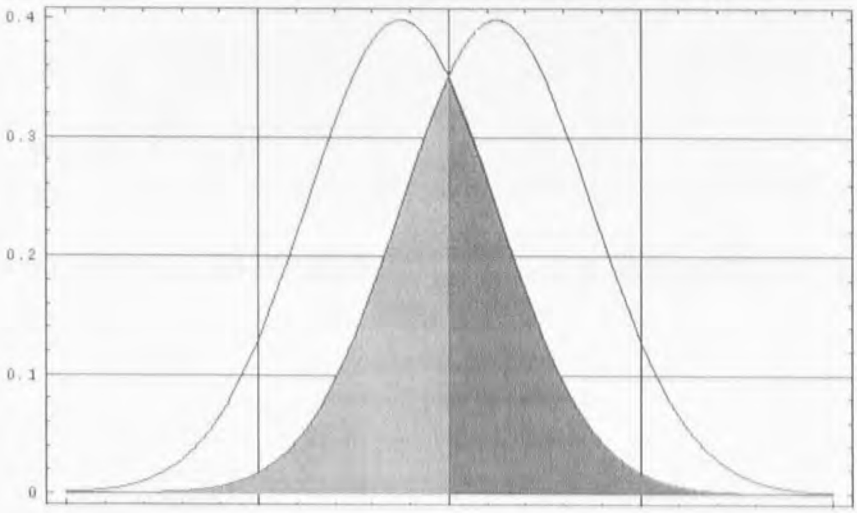

Figure 6-4: The 1-dimensional 2-class test problem.

The first set of experiments used 1-dimensional 2-class normally distributed data with population variance, $\sigma^2 = 1$, and population mean, $\mu = \pm\frac{1}{2}$, as illustrated in Figure 6-4. The Bayes' decision boundary is at *x=0* and the associated error rate can be found by evaluating the appropriate tail integrals. By symmetry arguments we may find the proportion of just one of the integrals in its respective Bayes decision region. Using one of the standard mathematical tables [2] we obtain:

$$\int_0^\infty dx \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-\frac{1}{2})^2}{2}} \approx 0.309$$

The same Bayes' classifier yields a cross entropy score of 0.58 since by symmetry:

$$E_c^* = -\sum_{i=1}^{2} \int_{-\infty}^{\infty} dx\, DP_i \log P_i = -2\int_{-\infty}^{\infty} dx\, DP \log P \approx 0.58$$

Where the data density D is given by:

$$D = \frac{N(\frac{1}{2},1) + N(-\frac{1}{2},1)}{2}$$

and the class probability P for one of the classes (arbitrarily the class with mean at $\frac{1}{2}$ is given by:

$$P = \frac{N(\frac{1}{2},1)}{N(\frac{1}{2},1) + N(-\frac{1}{2},1)}$$

with $N(\mu,\sigma^2)$ denoting a normal distribution with mean μ and variance σ^2 as defined in the equation:

$$P(x \mid \mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

If we calculate the class conditional probabilities from these estimates we obtain a moderated probability output as shown in Figure 6-5. Note that the probabilities are somewhat less certain than those produced by the maximum likelihood classifier. At no time does the probability of either class saturate to zero.

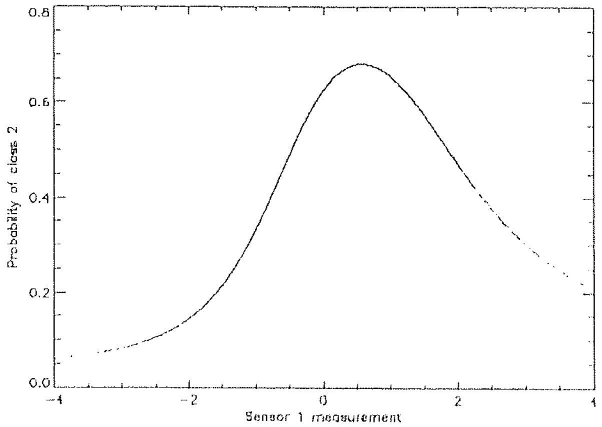

Figure 6-5: The conditional probability of the second class.

For the two-class problem we may calculate the expected performance at the fusion centre given the individual performance of the separate sensors if the data acquired by the sensors is normally distributed and conditionally independent. Given these assumptions the separation of the means in the joint measurement space, S_{xy}, can be recovered from the separations in each of the individual measurement axes, S_x and S_y, using Pythagoras' theorem thus:

$$S_{xy} = \sqrt{S_x^2 + S_y^2}$$

We shall illustrate the principle using some synthetic data having a (known) normal distribution. For these experiments we shall simulate two sensors each acquiring independent, normally distributed data on a target which belongs to one of two classes.

The mean of the measurement detected by each sensor depends on the target type, the standard deviation of the measurement is unity in each case and the means are separated by one standard deviation (see Figure 6-4). This configuration gives a Bayes error rate for each of the sensors separately of:

$$E = \frac{1}{\sqrt{2\pi\sigma^2}}\left(\int_{-\infty}^{\frac{1}{2}} dx\ e^{\frac{-(x-1)^2}{2\sigma^2}} + \int_{\frac{1}{2}}^{\infty} dx\ e^{\frac{-x^2}{2\sigma^2}} \right) \approx 31\%$$

and when acting in concert yields a Bayes error rate of:

$$E = \frac{1}{\sqrt{2\pi\sigma^2}}\left(\int_{-\infty}^{\frac{\sqrt{2}}{2}} dx\ e^{\frac{-(x-\sqrt{2})^2}{2\sigma^2}} + \int_{\frac{\sqrt{2}}{2}}^{\infty} dx\ e^{\frac{-x^2}{2\sigma^2}} \right) \approx 24\%$$

The performance of the separate sensors was estimated by classifying a large sample of 10,000 patterns from each class and counting the number that were allocated to the correct class. Table 6-2 shows these performance values.

Data Source	Error rate
Underlying distribution	31%
Underlying joint distribution	24%
Mean of sensors 1 & 2	33%
Fused sensors 1 and 2	30%

Table 6-2: The error rates for the individual and fused sensors.

Note that the average performance of each of the separate sensors is somewhat less than predicted from the underlying distributions. The fused error rate, however, is much worse than could have been obtained. This is largely due to the veto effect.

Table 6-3 shows the performance values for the same sample when employing moderated probabilities. The error rates for the maximum likelihood classifier are included for comparison. It clearly shows that the maximum likelihood error rate can be lowered by the use of appropriately moderated class conditional probabilities.

Data Source	Method	Error rate
Joint distribution	*a priori*	24%
Fused sensors 1&2	ML	30%
Fused sensors 1&2	Moderated	26%

Table 6-3: The fused error rates including moderated conditional probabilities.

In the experiments a training set was created with *N* samples from each class. These were used to calculate the sample means and sample variances, which in turn were used with the algorithms described above to classify the 10,000 test patterns. To obtain a value for the cross entropy score and classification rate for a particular value of *N*, the average of 10 runs was used with each of the training and test sets being independently generated for each run. The training and test sets for different values of *N* were also independent.

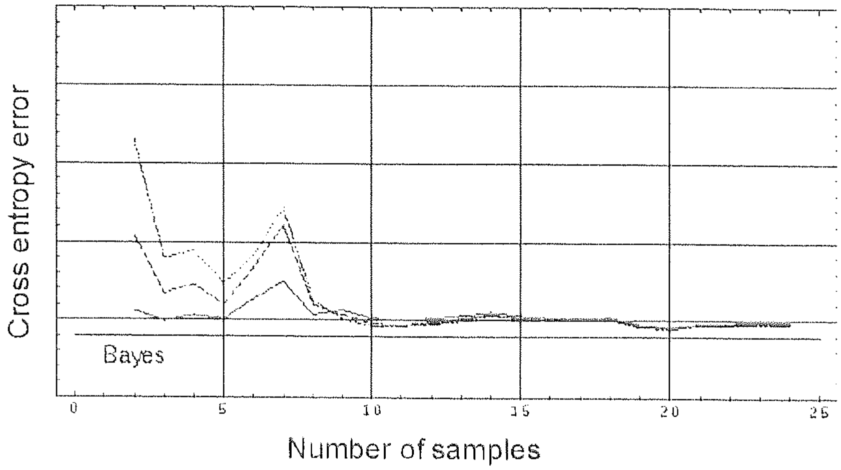

Figure 6-6: Average cross entropy error for the 1-dimensional 2-class test data.

The accompanying graph shown in Figure 6-6 illustrates the advantage of using the moderation techniques. The graph shows the cross entropy error for each of the three techniques (since the spherical and elliptical methods are equivalent in one dimension) as a function of the number of patterns in each of the two classes. Shown as a horizontal line is the cross entropy error produced by a Bayes classifier. The local fluctuations are due to the particular locations of training sample points being either representative or unrepresentative. The trend is for a lower cross entropy error. The *t*-distribution method having consistently lower cross entropy error than the corrected Gaussian or the unmoderated Gaussian techniques. The gain in performance is most noticeable for small training sample sizes (number of patterns per class, *N<10*).

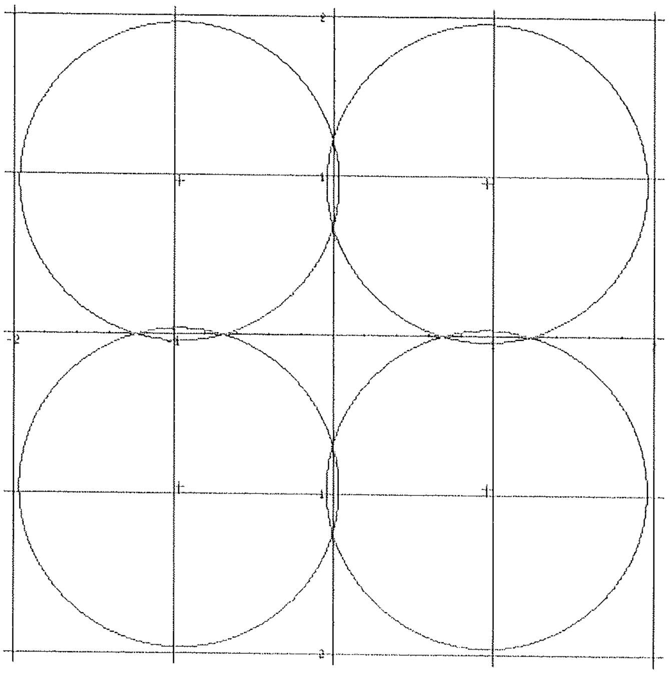

Figure 6-7: The contours at 1 standard deviation for the four class 2-dimensional problem.

The second set of experiments involved the 4-class 2-dimensional Gaussian data illustrated in Figure 6-7. The data consisted of four 2-dimensional spherical normal distributions with variance 1.0 and means at $(\pm 0.961, \pm 0.961)$. The Bayes decision boundaries are clearly aligned with the axes and it can be shown that the Bayes error rate for this set of distributions is also 30.9% since using the same symmetry arguments as before:

$$\int_{0}^{\infty}\int_{0}^{\infty} dx\, dy \frac{1}{2\pi} e^{-\frac{(x-0.961)^2+(y-0.961)^2}{2}} \approx 0.309$$

The same Bayes' classifier yields a cross entropy score of 0.75 since:

$$E_c^* = -\sum_{i=1}^{4} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} dx\, dy\, DP_i \log P_i = -4 \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} dx\, dy\, DP \log P \approx 0.75$$

Where the data density D is given by:

$$D=\frac{N(\{0.961,0.961\},1)+N(\{-0.961,0.961\},1)+N(\{0.961,-0.961\},1)+N(\{-0.961,-0.961\},1)}{4}$$

and the class probability P for one of the classes, arbitrarily the class with mean at (0.961,0.961), is given by:

$$P=\frac{N(\{0.961,0.961\},1)}{N(\{0.961,0.961\},1)+N(\{-0.961,0.961\},1)+N(\{0.961,-0.961\},1)+N(\{-0.961,-0.961\},1)}$$

with $N(\{\mu_x,\mu_y\},\sigma^2)$ denoting a spherical normal distribution with mean $\mu=\{\mu_x,\mu_y\}$ and variance σ^2.

As before, N training samples from each of the four classes were generated for estimation purposes as well as an independent set of 10,000 test patterns for evaluation. Each value of N used a different training and test set and the results for each method are the average of 10 runs. The results are summarised in Figure 6-8 for the six methods tested.

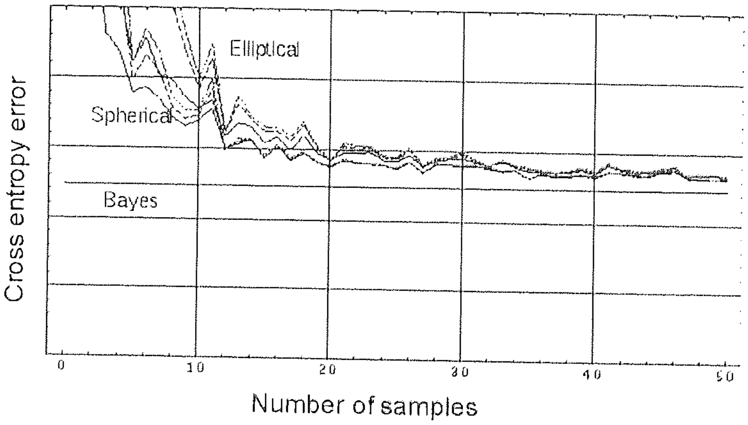

Figure 6-8: Average cross entropy error for the 2-dimensional 4-class test data.

The performance gains are now more noticeable particularly for the elliptical classifiers, which demonstrate the robustness of the moderation technique to error in the model selection. For values of N up to about 10 the fully moderated elliptical classifier is actually producing more accurate probabilities that the unmoderated spherical classifier. This is despite making incorrect assumptions about the underlying distributions (which are, of course, spherical). In other words, for this problem at small sample sizes, it is better to do the correct thing with the wrong model than the wrong thing with the correct model.

Experiments with 8 class 3-dimensional Gaussian Data

The third set of experiments used the 3-dimensional 8-class normally distributed data illustrated in Figure 6-9. The variance for the spherical distribution of each class was 1.0 and the means were located at $\{\pm 1.197, \pm 1.197, \pm 1.197\}$. The Bayes error rate is once again 30.9% since:

$$\int_0^\infty \int_0^\infty \int_0^\infty dx\, dy\, dz \frac{1}{(2\pi)^{\frac{3}{2}}} e^{\frac{(x-1.197)^2+(y-1.197)^2+(z-1.197)^2}{2}} \approx 0.309$$

In these experiments the extension to right elliptical distributions was used. The Bayes classifier giving an error rate of 30.3% and a cross entropy score of 0.57 for the test set used in these experiments.

Once again the spherical classifiers are better than the elliptical classifiers with the *t*-distribution classifier having lower error than either the flattened Gaussian or the maximum likelihood Gaussian in all cases.

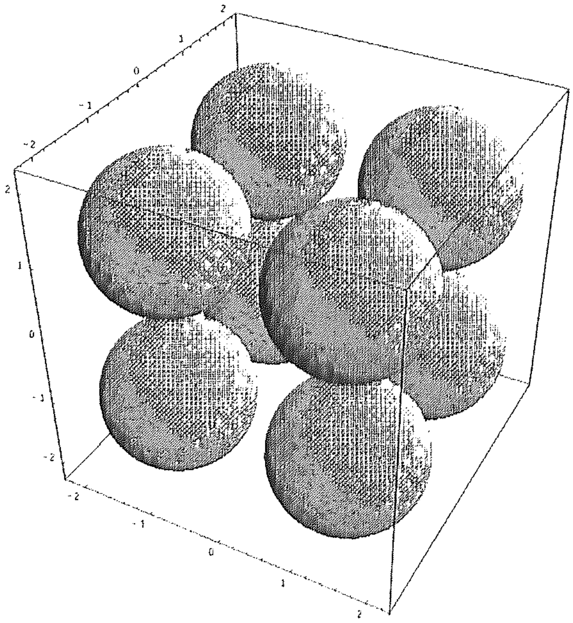

Figure 6-9: The 1 standard deviation spheres for the 3-dimensional, 8-class test data.

Although the three spherical methods are essentially the same after about 20 patterns per class the moderated elliptical methods are notably better than their unmoderated counterparts until after 40 patterns per class. The results are summarised graphically in Figure 6-10.

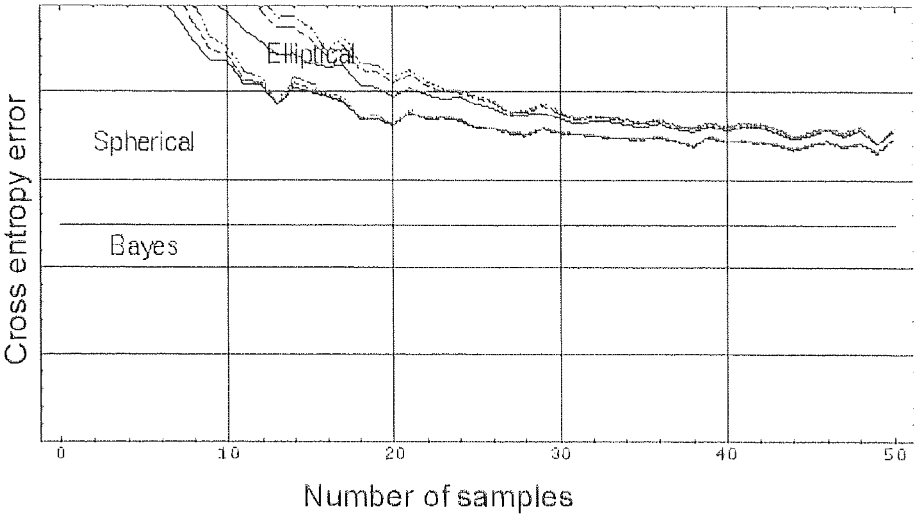

Figure 6-10: Average cross entropy for the 3-dimensional 8-class test data.

6.2.2 Summary of Experiments with Gaussian Data

It is noted that the performance gains expected of the analytically moderated Gaussian methods when applied to normally distributed data are significant. The improvement over unmoderated Gaussian classifiers is greatest for small amounts of training data.

6.2.3 Experimental Results on Real Data

The same six methods developed in the preceding sections were also evaluated on data obtained from measurements of actual quantities in the real world. The datasets were mainly obtained from the UCI Repository of Machine Learning Databases and Domain Theories at the University of California in Irvine [128]. These datasets are widely used for benchmarking by the machine learning community. The data is no longer normally distributed and the following results give an indication of the relative robustness of the various techniques. As well as providing experimental results with the classifier methods listed in the previous sections it is intended that these databases form a core set of evaluation data for use with the existing and original recognition algorithms used in the multilevel information processing studies. A tabular summary of the real world datasets used in these experiments is given in Table 6-4.

Database	Dimensions	Classes	Total samples	Training samples
Vowels	2	10	1520	190-760
Irises	4	3	150	50-75
Glasses	9	2	163	82
Breast cancer	9	2	699	350

Table 6-4: Summary of the real world datasets used in the moderation experiments.

Details of the datasets used can be found in Appendix B.

Results on Peterson and Barney Vowel Formant Data

The moderated and unmoderated Gaussian classifier methods described in the earlier sections were used to classify the Peterson and Barney vowel formant data[1]. For these experiments the data was divided into a training set and a test set by assigning the first and each subsequent odd numbered pattern to the training set and the second and each even numbered pattern to the test set. There were therefore 760 patterns in each of these sets. Three sets of experiments were carried out. In the first set the whole training set was used to calculate the sample means and variances. In the second experiment half of the training set was used and in the third a quarter of the training set was used. Therefore the training sets consisted of 760, 380 and 190 patterns in total or 76, 38 and 19 patterns from each class. The results are shown in
Table 6-5.

Method	Training patterns	Error rate	Cross entropy
Spherical ML Gaussian	760	42.5%	1.09
Elliptical ML Gaussian	760	28.7%	0.81
Multivariate ML Gaussian	760	22.1%	0.73
Spherical posterior distribution	760	42.5%	1.09
Elliptical posterior distribution	760	28.9%	0.81
Multivariate posterior distribution	760	22.2%	0.73
Spherical ML Gaussian	380	44.2%	2.27
Elliptical ML Gaussian	380	29.6%	1.47
Multivariate ML Gaussian	380	23.4%	1.21
Spherical posterior distribution	380	43.9%	1.95
Elliptical posterior distribution	380	29.5%	1.25
Multivariate posterior distribution	380	23.2%	1.07
Spherical ML Gaussian	190	42.4%	3.32
Elliptical ML Gaussian	190	31.4%	2.19
Multivariate ML Gaussian	190	25.9%	1.55
Spherical posterior distribution	190	42.2%	2.14
Elliptical posterior distribution	190	31.4%	1.41
Multivariate posterior distribution	190	25.3%	1.19

Table 6-5: Results on the Peterson and Barney vowel formant data.

The results show that the elliptical methods produce better classifications with more accurate probabilities than the spherical methods. This is to be expected since the data is quite clearly non-spherical. It is also apparent that the moderated Gaussian methods show very little gain for the experiments with the whole training set. However, the moderated techniques show a marked improvement in probability accuracy for the experiments with half of the training set even though the classification rate is essentially unaffected. For the experiment with one quarter of the training set the moderated techniques now display a significant accuracy gain. The classification rate remains unchanged. Even though the classification accuracy is not improved the improvement in probability accuracy indicated by the cross entropy score is worthwhile (either for decision making purposes as described earlier, or as part of a later fusion scheme).

Results on Iris Data

The experiments on the Iris data again analysed the effects of sample size. In the first experiment the odd numbered patterns were assigned to the training data and the even numbered patterns to the test data. This resulted in 75 patterns in each database with 25 from each class. In the second set of experiments every third pattern which appeared in the training and test sets was discarded. This resulted in 50 patterns in each database. The results are summarised inTable 6-6.

The results show that there is very little difference between the methods for this problem. On the full training data there is no change in either classification rate or cross entropy between the moderated and unmoderated methods. The elliptical classifiers doing slightly better than the spherical classifiers in this case. For the smaller training database the results are also very similar. However, the moderated classifiers do produce slightly more accurate probability estimates than the unmoderated classifiers.

Method	Training patterns	Error rate	Cross entropy
Spherical ML Gaussian	75	6.7%	0.13
Elliptical ML Gaussian	75	4.0%	0.11
Multivariate ML Gaussian	75	3.2%	0.10
Spherical posterior distribution	75	6.7%	0.13
Elliptical posterior distribution	75	4.0%	0.11
Multivariate posterior distribution	75	3.2%	0.10
Spherical ML Gaussian	50	6.3%	0.14
Elliptical ML Gaussian	50	4.0%	0.13
Multivariate ML Gaussian	50	3.4%	0.11
Spherical posterior distribution	50	6.3%	0.13
Elliptical posterior distribution	50	4.0%	0.11
Multivariate posterior distribution	50	3.4%	0.10

Table 6-6: Results on the iris data.

Results on Glass Data

Method	Training Patterns	Error Rate	Cross Entropy
Spherical ML Gaussian	82	46.9%	2.28
Elliptical ML Gaussian	82	38.3%	2.96
Multivariate ML Gaussian	82	37.2%	3.12
Spherical posterior distribution	82	46.9%	21.0
Elliptical posterior distribution	82	39.5%	2.52
Multivariate posterior distribution	82	37.2%	2.87

Table 6-7: Results on the glass data.

The experimental results show that all of the Gaussian based classifiers do not perform as well as the result previously reported with nearest neighbour [16]. Also apparent is that no reduction in error rate is obtained by using the moderated methods. However, a small improvement in cross entropy score and hence accuracy of probabilities is obtained with the *t*-distribution based technique.

Results on Breast Cancer Diagnosis Data

Method	Training Patterns	Error Rate	Cross Entropy
Spherical ML Gaussian	350	3.2%	0.69
Elliptical ML Gaussian	350	4.0%	1.18
Multivariate ML Gaussian	350	4.3%	1.29
Spherical posterior distribution	350	3.7%	0.74
Elliptical posterior distribution	350	4.6%	1.12
Multivariate posterior distribution	350	4.5%	1.21

Table 6-8: Results on the breast cancer diagnosis database.

The results show that, on this database, the moderated techniques do not produce increases in probability accuracy (in fact, the accuracy gets slightly worse although the change is not statistically significant). This is probably due to the discrete attribute values being highly non-Gaussian. Rule based methods would probably be more suited to this database.

6.3 *Analytic Moderation for Other Distributions*

Here we develop a Bayesian approach to dealing with noisy data. Initially we restrict our analyses to problems in which the density model is either Gaussian or a Gaussian mixture. Furthermore, we assume that the sensor noise is Gaussian in form but do not make any assumptions about the amount of sensor noise. We make appropriate approximations to ensure that the prescribed algorithms are computationally tractable.

Two important problems arise from the very nature of measuring the attribute values themselves. Values are not generally measured to infinite precision, nor are sensors usually noise-free. As a result the values associated with particular attributes might more correctly be viewed as samples from a distribution, the location and dispersion of which depends on the sensing process (see Figure 6-11).

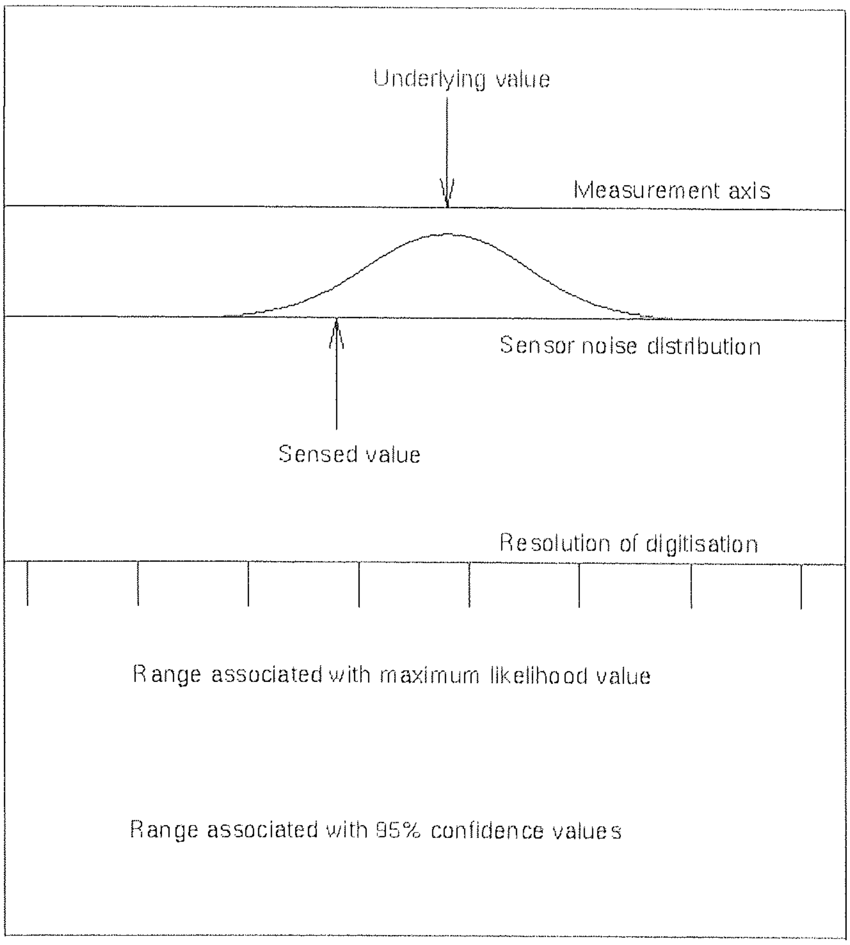
Underlying value
Measurement axis
Sensor noise distribution
Sensed value
Resolution of digitisation
Range associated with maximum likelihood value
Range associated with 95% confidence values

Figure 6-11: Schematic representation of the sources of noise in a sensor.

In some cases the dispersion might be large, and the usual approximation, which uses delta functions for the distribution, is clearly not appropriate. In the extreme case of infinite dispersion, the data value can be considered as missing altogether. The missing data problem constitutes a significant subset of the noisy data problem and warrants separate investigation in more detail (see [87] for example). Herein we address the more general noisy data problem for a restricted class of density models [16].

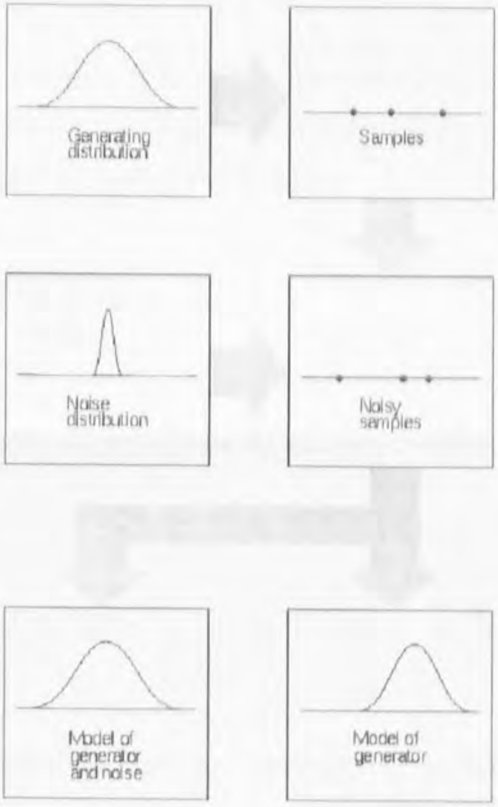

Figure 6-12: The separation between the underlying generator and the noise generator.

6.3.1 Allowing for Unknown Sensor Noise

We shall tackle the problem from a Bayesian standpoint [37] and [107]. For the moment we shall deal exclusively with problems of one dimension. Problems of multiple dimensions in which the separate variables are independent follow trivially. The generalisation to multidimensional spherical distributions is relatively straightforward whereas the multivariate case is somewhat more involved. For practical purposes we may assume a functional form for the sensor noise - in this case we shall assume the noise is zero-mean Gaussian. However, we do not know the sensor noise variance, s^2, so have to marginalise over this variable [16] and [91]. Note that all of the probabilities are conditioned on our Gaussian assumptions about the model - we shall proceed to omit this from our equations for the sake of visual clarity.

$$P(x \mid D) = \int_0^\infty ds^2 \; P(x \mid D, s^2) P(s^2 \mid D)$$

The first of the terms in the integral is merely the probability derived from the particular model under consideration, estimated from the data and using the appropriate value of s^2. The second term can be re-written using Bayes' rule to give:

$$P(s^2 \mid D) = \frac{P(D \mid s^2) P(s^2)}{P(D)}$$

the denominator simply being an appropriate normalising constant which is found by further marginalisation:

$$P(D) = \int_0^\infty ds^2 \; P(D \mid s^2) P(s^2)$$

This gives us our canonical formula for incorporating sensor noise into our pattern recognition density models:

$$P(x \mid D) = \int_0^\infty ds^2 \; P(x \mid D, s^2) \frac{P(D \mid s^2) P(s^2)}{\int_0^\infty ds^2 \; P(D \mid s^2) P(s^2)}$$

The first term will depend on the particular model we intend to use. The second term depending only on the evidence [116] for each possible value of sensor noise variance. Of course we are required to measure this from our data and in doing so will have to use some form of model. This model need not be the same as the final model (indeed it is perhaps desirable for it not to be so).

Two plausible (and simple) models for calculating this quantity come to mind; a Gaussian estimate and a sum of Gaussian kernels. We further require that the estimates of $P(D \mid s^2)$ should not unfairly favour small values of s^2 merely because we take the pragmatic step of using the **same data** for estimating the density model and the noise value. This is particularly important for a kernel estimate but also affects the Gaussian estimate. In either case we require an estimate of the evidence for a particular data point produced by omitting that data point from the density model. The problem can be stated thus - if we measure the sample mean and variance, $\hat{\mu}$ and $\hat{\sigma}^2$, from a sample of measurement values, what is the sample mean and variance, $\hat{\mu}_j$ and $\hat{\sigma}_j^2$, of the same set with the *j*th element omitted? Appendix A.7 shows that this does not have a simple form. Therefore, although the Gaussian model seems at first sight to provide us with a suitably tractable method of incorporating the model evidence, it transpires that the proper leave-one-out evidence is somewhat complex. In the ensuing analyses we shall use a leave-one-out estimate for $P(D|s^2)$ based on a Gaussian kernel density estimate:

$$P(d_i \mid s^2) = \frac{1}{N-1}\sum_{j \neq i} \frac{1}{\sqrt{2\pi s^2}} e^{-\frac{(d_i - d_j)^2}{2s^2}}$$

Assuming that the d_i are independent measurements we may write:

$$\begin{aligned} P(D \mid s^2) &= \prod_i P(d_i \mid s^2) \\ &= \prod_i \frac{1}{N-1}\sum_{j \neq i} \frac{1}{\sqrt{2\pi s^2}} e^{-\frac{(d_i - d_j)^2}{2s^2}} \end{aligned}$$

We are now forced to make a further assumption in order to simplify the necessary calculations and ensure that the integration is tractable. We assume that the sum is dominated by its largest component and can be replaced by that component alone. We shall firstly digress to assess the likely impact of this approximation.

We take a practical standpoint and use computer simulations to assess the factors involved.[2] We perform the following experiment; we sample *N* points independently from a uniform

[2] Many thanks to Dr. Richard Glendinning who observed that, for certain cases, this problem may be accessible analytically. For now we shall press on experimentally.

distribution contained within the d dimensional unit hypercube. We calculate the contribution of a Gaussian kernel centred on each point at the origin. Various heuristics have been suggested for setting the kernel [153]. In our experiments, the variance of the kernel depends on N and d in such a way that the standard deviation is approximately commensurate with the expected distance between data points. For this experiment we use:

$$\sigma^2(N,d) = \left(\frac{1}{N^{\frac{1}{d}} + 1} \right)^2$$

We plot the proportion of the density estimate contributed by the nearest kernel for various values of N (between 1 and 20) and d (1, 5 and 10). Each value is averaged over 1,000 experiments. The graph is shown in Figure 6-13.

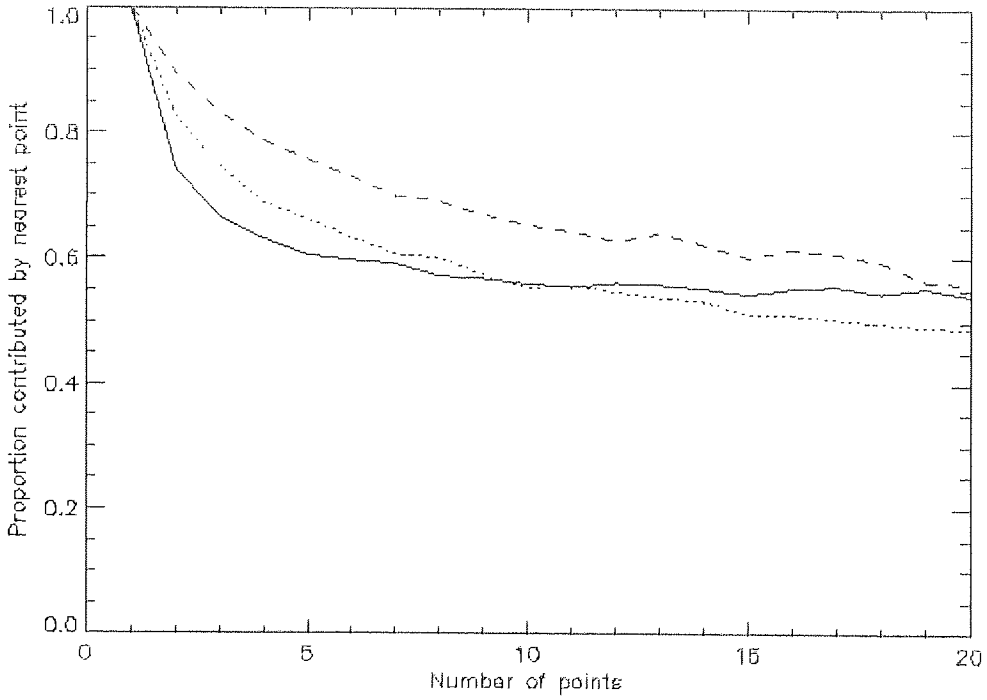

Figure 6-13: Proportion of the density estimate contributed by the nearest kernel.

It is noted that the portion of the density estimate contributed by the nearest point decreases with increased sample packing (as expected). However the rate at which this proportion falls away is greatest in one dimension and is considerably reduced in 5 and 10 dimensions. In nearly all cases the nearest kernel contributes half of the total kernel sum (or more) at the origin for up to 20 points.

What is the likely effect on the resulting density estimate? Since we are only taking the nearest kernel into account, the contribution from more distant kernels is ignored and unfair weighting is given to narrow kernels. Once incorporated into the final estimate this will result in a density estimate that is insufficiently smooth. One possible way to circumvent this problem is to give some extra weight to smoother estimates. The contribution of the nearest kernel tends to half of the total density. Furthermore, we can partially correct for the bias by doubling the estimate for the local variance, $\hat{\sigma}_L^2$. This, somewhat heuristic, bias correction is included in all subsequent experiments.

Defining d_i^* to be the nearest data point to d_i we have

$$\frac{1}{N-1}\sum_{j\neq i}\frac{1}{\sqrt{2\pi s^2}}e^{\frac{(d_i-d_j)^2}{2s^2}} \approx \frac{1}{\sqrt{2\pi s^2}}e^{-\frac{(d_i-d_i^*)^2}{2s^2}}$$

This approximation results in

$$\begin{aligned} P(D\,|\,s^2) &\approx \prod_i \frac{1}{\sqrt{2\pi s^2}}e^{-\frac{(d_i-d_i^*)^2}{2s^2}} \\ &= \left(\frac{1}{\sqrt{2\pi s^2}}\right)^N e^{-\frac{\sum_i (d_i-d_i^*)^2}{2s^2}} \end{aligned}$$

We propose a prior (which is admittedly improper) of s^2 of $P(s^2)=\frac{\kappa}{s^2}$[3] and replace the calculable quantity $\sum_i (d_i-d_i^*)^2$ with the term 2ψ thus giving the required integral

$$\int_0^\infty ds^2\; P(D\,|\,s^2)P(s^2) \approx \int_0^\infty ds^2\; s^{-N-2}e^{-\frac{\psi}{s^2}}$$

We then compare this with the standard equation of the Gamma distribution (itself a superset of the Chi-square distribution):

[3] It is noted, by the author and others, that this prior is a matter for debate and that other priors would result in subtly different posterior distributions.

$$P(x)=\frac{\beta^{\alpha}}{\Gamma(\alpha)}x^{\alpha-1}e^{-\beta x}$$

we therefore have

$$\begin{aligned}\int_0^{\infty} ds^2\ P(D\,|\,s^2)P(s^2) &\approx \kappa(2\pi)^{-\frac{N}{2}}\Gamma\left(\frac{N}{2}\right)\psi^{-\frac{N}{2}}\\ &= \kappa(2\pi\psi)^{-\frac{N}{2}}\Gamma\left(\frac{N}{2}\right)\end{aligned}$$

Returning to our original formulation we now have

$$\begin{aligned}P(x\,|\,D) &= \int_0^{\infty} ds^2\ P(x\,|\,D,s^2)\frac{P(D\,|\,s^2)P(s^2)}{\int_0^{\infty} ds^2\ P(D\,|\,s^2)P(s^2)}\\ &\approx \int_0^{\infty} ds^2\ P(x\,|\,D,s^2)\frac{(2\pi s^2)^{-\frac{N}{2}}e^{-\frac{\psi}{s^2}}\kappa s^{-2}}{\kappa(2\pi\psi)^{-\frac{N}{2}}\Gamma\left(\frac{N}{2}\right)}\\ &= \int_0^{\infty} ds^2\ P(x\,|\,D,s^2)\frac{s^{-N-2}\psi^{\frac{N}{2}}e^{-\frac{\psi}{s^2}}}{\Gamma\left(\frac{N}{2}\right)}\\ &= \frac{\psi^{\frac{N}{2}}}{\Gamma\left(\frac{N}{2}\right)}\int_0^{\infty} ds^2\ P(x\,|\,D,s^2)s^{-N-2}e^{-\frac{\psi}{s^2}}\end{aligned}$$

By inserting in the appropriate equation for $P(x|D,s^2)$ and evaluating the above integral, this result gives us a method of allowing for Gaussian sensor noise of zero mean and unknown variance for a variety of density estimate models. Of course the integration will not be analytically tractable in all cases. However, for some simple models we may evaluate this quantity with little approximation. We begin with a few definitions; the sample mean, $\hat{\mu}$, is normally defined thus:

$$\hat{\mu}=\frac{1}{N}\sum_{i=1}^{N}d_i$$

Which presupposes that the d_i are exactly measurable. As before there is likely to be some noise associated with these measurements and it is desirable to allow for this in our analysis. If we assume that the underlying values corresponding to the measured data can be modelled as a Gaussian kernel estimate with means at the data values and variance s^2 we may extend our definition of sample mean thus:

$$\hat{\mu} = \int_{-\infty}^{\infty} dx\, x \sum_{i=1}^{N} \frac{1}{\sqrt{2\pi s^2}} e^{-\frac{(x-d_i)^2}{2s^2}}$$

We may rearrange this slightly to obtain:

$$\hat{\mu} = \frac{1}{N\sqrt{2\pi s^2}} \sum_{i=1}^{N} \int_{-\infty}^{\infty} dx\, x e^{-(\frac{x}{s\sqrt{2}} - \frac{d_i}{s\sqrt{2}})^2}$$

We can effect a simple change of variables by letting $y = \frac{x}{s\sqrt{2}}$, giving $x = s\sqrt{2}y$ and $dx = s\sqrt{2}dy$. This leads to

$$\hat{\mu} = \frac{1}{N\sqrt{2\pi s^2}} \sum_{i=1}^{N} 2s^2 \int_{-\infty}^{\infty} dy\, y e^{-(y - \frac{d_i}{s\sqrt{2}})^2}$$

This integral is now of a standard form contained in [75]:

$$\int_{-\infty}^{\infty} dy\, y^n e^{-(y-\beta)^2} = \frac{\sqrt{\pi} H_n(i\beta)}{2^n i^n}$$

where $H_n(p)$ is the nth order Hermite polynomial of p.

We have $n=1$ and $\beta = \frac{d_i}{s\sqrt{2}}$. Noting that $H_1(p)=2p$ leads to

$$\hat{\mu} = \frac{1}{N\sqrt{2\pi s^2}} \sum_{i=1}^{N} \frac{2s^2 \sqrt{\pi} i d_i}{s\sqrt{2} 2i}$$

which simplifies to

$$\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} d_i$$

That is to say the same sample mean as was obtained without the noise. This is perhaps to be expected since the noise is symmetric. We now find an expression for the sample variance, traditionally defined as

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

(note that this is not an unbiased estimate of the population variance) and extended to cover noisy data by

$$\hat{\sigma}^2 = \int_{-\infty}^{\infty} dx\,(x - \hat{\mu})^2 \frac{1}{N}\sum_{i=1}^{N}\frac{1}{\sqrt{2\pi s^2}} e^{-\frac{(x-d_i)^2}{2s^2}}$$

Rearranging as before gives

$$\hat{\sigma}^2 = \frac{1}{N\sqrt{2\pi s^2}}\sum_{i=1}^{N}\int_{-\infty}^{\infty} dx\,(x - \hat{\mu})^2 e^{-(\frac{x}{s\sqrt{2}} - \frac{d_i}{s\sqrt{2}})^2}$$

We use the same change of variable as earlier to give

$$\hat{\sigma}^2 = \frac{1}{N\sqrt{2\pi s^2}}\sum_{i=1}^{N}\int_{-\infty}^{\infty} dy\; s\sqrt{2}(s\sqrt{2}y - \hat{\mu})^2 e^{-(y - \frac{d_i}{s\sqrt{2}})^2}$$

Expanding gives

$$\hat{\sigma}^2 = \frac{1}{N\sqrt{2\pi s^2}}\sum_{i=1}^{N}\int_{-\infty}^{\infty} dy\; s\sqrt{2}(2s^2y^2 - 2\sqrt{2}\hat{\mu}sy + \hat{\mu}^2)e^{-(y - \frac{d_i}{s\sqrt{2}})^2}$$

which can be written thus

$$\hat{\sigma}^2 = \frac{1}{N\sqrt{2\pi s^2}} \sum_{i=1}^{N} \left(2\sqrt{2}s^3 \int_{-\infty}^{\infty} dy\ y^2 e^{-(y-\frac{d_i}{s\sqrt{2}})^2} - 4\hat{\mu}s^2 \int_{-\infty}^{\infty} dy\ ye^{-(y-\frac{d_i}{s\sqrt{2}})^2} + s\sqrt{2}\hat{\mu}^2 \int_{-\infty}^{\infty} dy\ e^{-(y-\frac{d_i}{s\sqrt{2}})^2} \right)$$

Again using [75] and noting that $H_0(p)=1$, $H_1(p)=2p$ and $H_2(p)=4p^2$-2 we have:

$$\hat{\sigma}^2 = \frac{1}{N\sqrt{2\pi s^2}} \sum_{i=1}^{N} \left(2\sqrt{2}s^3\sqrt{\pi}\left(\frac{d_i^2}{2s^2} - \frac{1}{2}\right) - 4\hat{\mu}s^2\sqrt{\pi}\frac{d_i}{s\sqrt{2}} + s\sqrt{2}\hat{\mu}^2\sqrt{\pi} \right)$$

which simplifies to

$$\hat{\sigma}^2 = \frac{1}{N}\left(\sum_{i=1}^{N} d_i^2 + Ns^2 - 2\hat{\mu}\sum_{i=1}^{N} d_i + N\hat{\mu}^2 \right)$$

This gives

$$\hat{\sigma}^2 = \frac{1}{N}\sum_{i=1}^{N}(d_i - \hat{\mu})^2 + s^2$$

that is to say that the variances add.

In the next section we present the more flexible and analytically simpler noisy Gaussian kernel density estimate.

6.3.2 Noisy Gaussian Kernel Density Models

The same principle may be applied to the related method in which the density model is built up using the sum of a set of Gaussian kernel functions, centred on the data points [137]. We define a kernel density estimate thus:

$$P(x \mid D, s^2) = \frac{1}{N}\sum_i \frac{1}{\sqrt{2\pi s^2}} e^{-\frac{(x-d_i)^2}{2s^2}}$$

where s^2 is the same sensor noise variance described above, *i.e.* that the sensor noise is the **only** noise in the system. Such a kernel density estimate using Gaussian kernels and a bandwidth (standard deviation) selected manually to give an appropriately smooth estimate is shown in Figure 6-14.

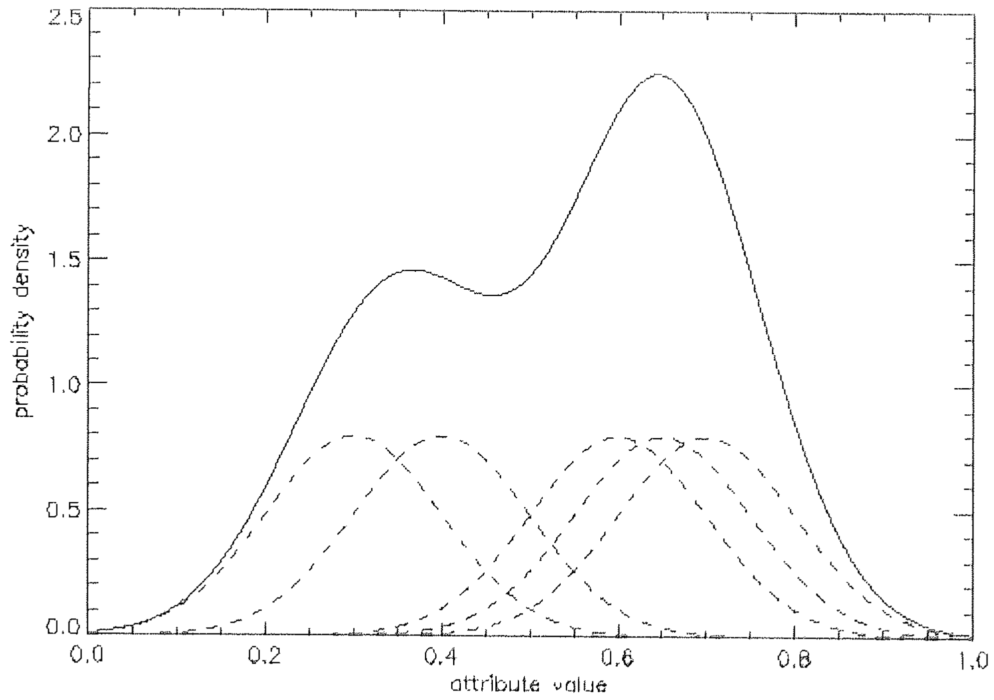

Figure 6-14: An example kernel density estimate using Gaussian kernels.

We apply our earlier result to obtain

$$\begin{aligned} P(x \mid D) &\approx \frac{\psi^{\frac{N}{2}}}{\Gamma\left(\frac{N}{2}\right)} \int_0^\infty ds^2 \, \frac{1}{N} \sum_i \frac{1}{\sqrt{2\pi s^2}} e^{-\frac{(x-d_i)^2}{2s^2}} s^{-N-2} e^{-\frac{\psi}{s^2}} \\ &= \frac{\psi^{\frac{N}{2}}}{N\sqrt{2\pi}\,\Gamma\left(\frac{N}{2}\right)} \sum_i \int_0^\infty ds^2 \; e^{-\frac{(x-d_i)^2}{2s^2}} s^{-N-3} e^{-\frac{\psi}{s^2}} \\ &= \frac{\psi^{\frac{N}{2}}}{N\sqrt{2\pi}\,\Gamma\left(\frac{N}{2}\right)} \sum_i \int_0^\infty ds^2 \; e^{-\frac{(x-d_i)^2-2\psi}{2s^2}} s^{-N-3} \end{aligned}$$

Now this is of a form we have seen before - the Gamma distribution of the previous section. We may immediately write down our solution thus:

$$\int_0^\infty ds^2 \; e^{-\frac{(x-d_i)^2-2\psi}{2s^2}} s^{-N-3} = \Gamma\left(\frac{N+1}{2}\right) \varphi^{\frac{-N-1}{2}}$$

where $\varphi = \dfrac{(x-d_i)^2}{2} + \psi$ giving the required probability estimate as

$$P(x \mid D) \approx \frac{\psi^{\frac{N}{2}} \Gamma\left(\frac{N+1}{2}\right)}{N\sqrt{2\pi}\,\Gamma\left(\frac{N}{2}\right)} \sum_i \frac{1}{\varphi^{\frac{N+1}{2}}}$$

Re-expanding the constants ψ and φ gives

$$P(x \mid D) \approx \frac{\Gamma\left(\frac{N+1}{2}\right)\left(\frac{\sum_j (d_j - d_j^*)^2}{2}\right)^{\frac{N}{2}}}{N\sqrt{2\pi}\,\Gamma\left(\frac{N}{2}\right)} \sum_i \frac{1}{\left(\frac{(x-d_i)^2}{2} + \frac{\sum_j (d_j - d_j^*)^2}{2}\right)^{\frac{N+1}{2}}}$$

$$= \frac{\Gamma\left(\frac{N+1}{2}\right)\left(\sum_j (d_j - d_j^*)^2\right)^{\frac{N}{2}}}{N\sqrt{\pi}\,\Gamma\left(\frac{N}{2}\right)} \sum_i \frac{1}{\left((x-d_i)^2 + \sum_j (d_j - d_j^*)^2\right)^{\frac{N+1}{2}}}$$

this may be rewritten as

$$P(x \mid D) \approx \frac{\Gamma\left(\frac{N+1}{2}\right)\left(\sum_j (d_j - d_j^*)^2\right)^{\frac{N}{2}}}{N\sqrt{\pi}\,\Gamma\left(\frac{N}{2}\right)\left(\sum_j (d_j - d_j^*)^2\right)^{\frac{N+1}{2}}} \sum_i \frac{1}{\left(\frac{(x-d_i)^2}{\left(\sum_j (d_j - d_j^*)^2\right)^{\frac{N}{2}}} + 1\right)^{\frac{N+1}{2}}}$$

$$= \frac{\Gamma\left(\frac{N+1}{2}\right)}{N\sqrt{\sum_j (d_j - d_j^*)^2 \pi}\,\Gamma\left(\frac{N}{2}\right)} \sum_i \left(\frac{(x-d_i)^2}{\left(\sum_j (d_j - d_j^*)^2\right)^{\frac{N}{2}}} + 1\right)^{\frac{-N-1}{2}}$$

Note the striking resemblance with the *t*-distribution

$$P(x \mid D) = \frac{\Gamma\left(\frac{N}{2}\right)}{\hat{\sigma}\sqrt{(N+1)\pi}\,\Gamma\left(\frac{N-1}{2}\right)}\left(\frac{(x-\hat{\mu})^2}{(N+1)\hat{\sigma}^2} + 1\right)^{-\frac{N}{2}}$$

which is further enhanced if one regards $\hat{\varsigma}^2 = \frac{1}{N}\sum_i (d_i - d_i^*)^2$ as an estimate of the average local variance (a term roughly comparable to the $\hat{\sigma}^2$ term in the *t*-distribution).

$$\begin{aligned} P(x \mid D) &\approx \frac{\Gamma\left(\frac{N+1}{2}\right)}{N\hat{\varsigma}\sqrt{N\pi}\,\Gamma\left(\frac{N}{2}\right)} \sum_i \left(\frac{(x-d_i)^2}{N\hat{\varsigma}^2}+1\right)^{\frac{-N-1}{2}} \\ &= \frac{1}{N}\sum_i \frac{\Gamma\left(\frac{N+1}{2}\right)}{\hat{\varsigma}\sqrt{N\pi}\,\Gamma\left(\frac{N}{2}\right)} \left(\frac{(x-d_i)^2}{N\hat{\varsigma}^2}+1\right)^{\frac{-N-1}{2}} \end{aligned}$$

that is to say a sum of *t*-distributions.

This may not seem too surprising since we require

$$P(x \mid D) = \int_0^\infty ds^2 \sum_i N(d_i, s^2)$$

where $N(\mu,\sigma^2)$ denotes a Gaussian distribution of mean μ and variance σ^2. Now we may take the summation out of the integral obtaining

$$P(x \mid D) = \sum_i \int_0^\infty ds^2\; N(d_i, s^2)$$

The predicted posterior, *P(x|D)*, is therefore in the form of a sum of *t*-distributions as proved more rigorously above. What we now have is a parameter-free, Bayesian, noisy kernel density estimate to compare with traditional Gaussian kernel density estimate. We shall proceed to illustrate just such a comparison in the next section.

6.3.3 Illustration of the Bayesian Noisy Kernel Estimate

To illustrate this result and to compare it to other, classical, kernel density estimates we introduce a synthetic problem. Data is generated from a known distribution - let us say a Gaussian distribution, $N(\mu,\sigma^2)$. N_t points are sampled from the distribution for training purposes and a kernel density estimate is formed. The resultant model is evaluated using a global entropy based measure of the suitability of the model:

$$E_G = -\int_{-\infty}^{\infty} dx\; P(x) \log \hat{P}(x)$$

where $P(x)$ denotes the underlying (correct) distribution and $\hat{P}(x)$ the estimate of that distribution. In practice we find it sufficiently accurate to approximate this integral with a summation over an appropriate range of test points sampled from the underlying distribution

$$E_G \approx -\sum_{i=1}^{N_e} \log \hat{P}(x_i)$$

where N_e test points x are selected evenly over an interval which covers the distributions. In the following experiments N_e=*200* and the interval is centred on the mean of the underlying distribution and extends 3 standard deviations in both directions. Two models are employed; the noisy kernel model developed above and the standard Parzen estimator with Gaussian kernels. The width of the kernels in the Parzen model is required to be set by the user. There are a number of suggested heuristics for determining this kernel width [153] which reduce the unwanted burden of searching for the best bandwidth but which do not, generally, give quite such good results. In order to give least disadvantage to the standard method we therefore accept the computational expense of the search for optimum bandwidth.

For each experimental run we build 20 standard models of various bandwidths (ranging from 0.1 to 2.0 in steps of 0.1) and evaluate the entropy error on a cross validation set (consisting of half the training dataset) for each model. We select the best bandwidth found thus and use it to build a density estimate on the whole dataset. Figure 6-15 shows the cross validation entropy error during the bandwidth search for one of the experiments

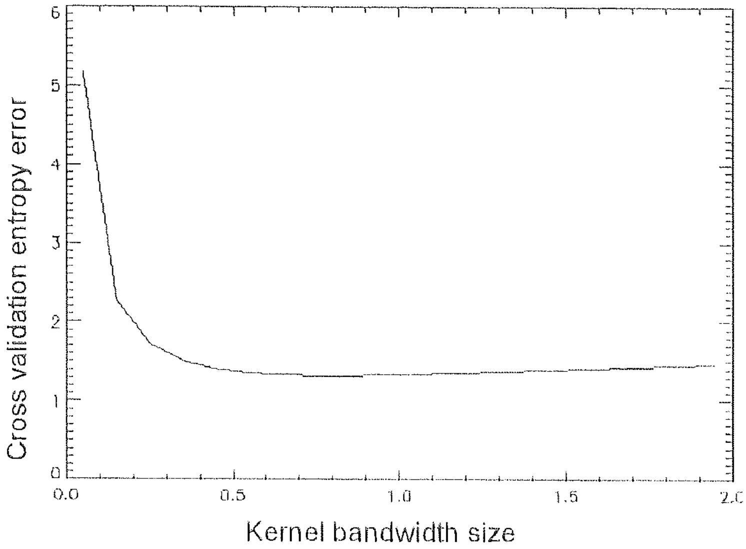

Figure 6-15: The cross validation entropy error of the standard kernel estimate.

We use a Gaussian underlying distribution with zero mean and unit variance and conduct experiments with N_t between 2 and 10.

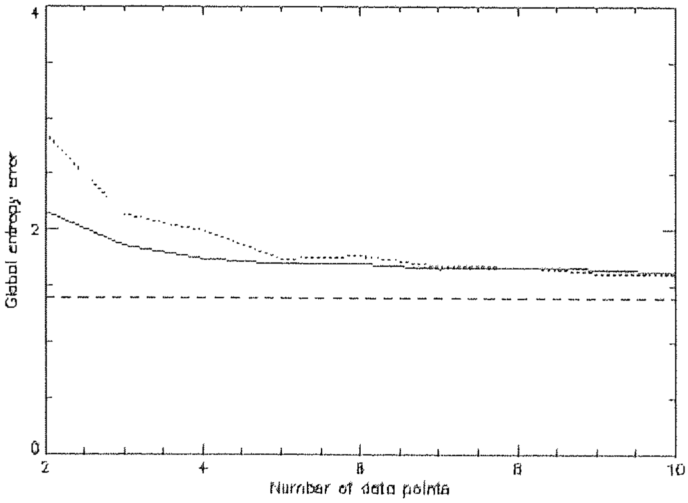

Figure 6-16: Global entropy error for standard and Bayesian kernel estimates.

Figure 6-16 shows the global entropy error for the standard method, the Bayesian method and the underlying distribution. We note that the new Bayesian density model provides a more accurate global density estimate in almost all one-dimensional experiments with fewer than 10 data points. Both techniques, however, improve slowly thereafter. In order to assess in which regions the Bayesian method is better we use another error criterion, employed extensively in the literature. We measure the entropy error at the mean of the underlying distribution (which in our case is at the origin):

$$E_\mu = P(0)\log\hat{P}(0)$$

Figure 6-17 shows the central entropy error for the same set of experiments as the previous graph.

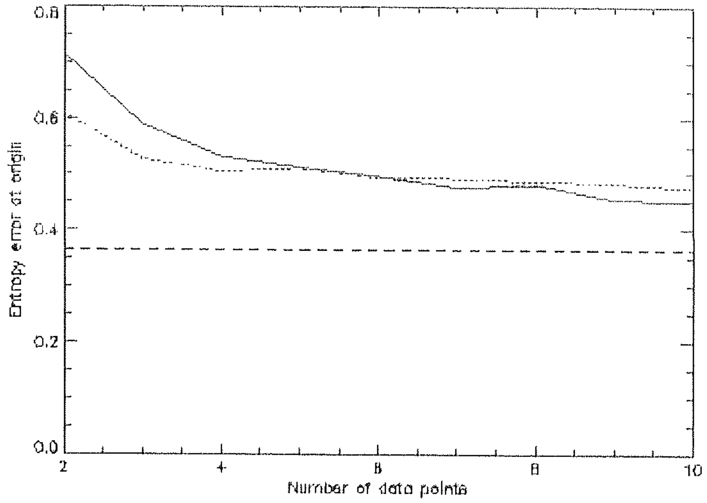

Figure 6-17: Central entropy for standard and Bayesian kernel estimates.

It is noted that the central density estimate is approximately as good for both the Bayesian scheme and the standard technique (the standard technique being slightly better for the first few trials and thereafter slightly worse). This seems to indicate that the improvement in accuracy is mainly afforded in the tails of the distribution and not in those areas where most samples will be found.

6.4 *Chapter Discussion*

We have shown that the simple expedient of marginalising of model parameters can yield conditional probability estimates that are more accurate than taking the maximum likelihood values. The effect is most pronounced in the tails of the distributions when outlying observations are unlike those used during the design process. The benefits of this approach are clearly justified using the arguments put forward in Chapter 5 . The analytic approach taken in the Chapter 6 , however, is limited to those distributions for which the necessary integrals are tractable. In each case the classifier is based on a parametric density model and does not make good use of discriminative information. In the next chapter we introduce a number of heuristic approaches to probability moderation that are applicable to density models and discriminators alike.

Chapter 7 Heuristic Moderation for Probability Fusion

7.1 *Chapter Introduction*

In this chapter we develop three approaches to heuristic probability moderation:

1. Semi-analytic approaches – in which a discriminative classifier is described which is composed of Gaussians and is therefore able to be (approximately) moderated using the results of the previous chapter
2. Discounting approaches – in which the classifier output is discounted by mixing with the prior distributions
3. More general mixing approaches where a set of classifiers are averaged to produce the moderated output (itself a form of data fusion)

We spend most time developing the semi-analytic technique but include the other approaches for completeness.

7.2 *Semi-analytic Moderation*

The treatment of moderation in the previous section relies on the classifier model being used to be Gaussian. In this case the moderated probabilities can be calculated directly from the estimated model. However, such data modelling classifiers do not always produce such good performance as discriminative classifiers such as the multi-layer perceptron (MLP). In this section we shall develop a discriminative version of the Gaussian classifier which may be regarded as a variant on the MLP but is nonetheless amenable to analytic moderation.

It is the logistic function in a single layer perceptron (or the output layer of a MLP) that finally produces the class conditional probabilities [148]. This function has close ties with the Gaussian distribution (see Appendix A.5). We therefore note that for a particular, simple MLP classifier as shown in Figure 7-1, trained discriminatively, produces for a specific pair of input distributions, precisely the same output as a Gaussian density model.

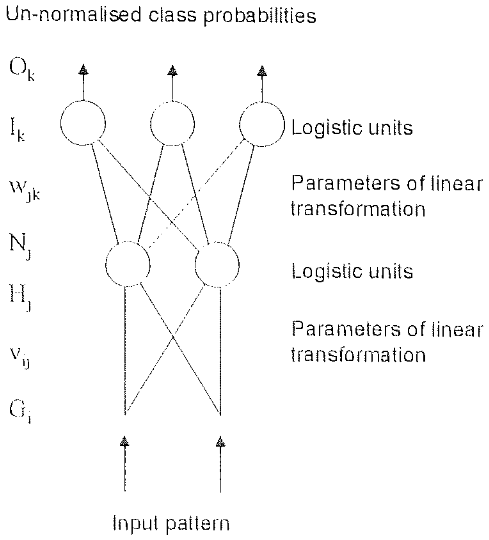
Un-normalised class probabilities
O_k
I_k
w_{jk}
N_j
H_j
v_{ij}
G_i
Logistic units
Parameters of linear transformation
Logistic units
Parameters of linear transformation
Input pattern

Figure 7-1: The standard multi-layer perceptron classifier.

This observation leads us to postulate a network architecture which is trained discriminatively but which outputs class conditional probabilities similar to Gaussian density models for a wider range of (multiclass) distributions.

7.2.1 Discriminative Gaussian Classifier

A hybrid data model which comprises a Gaussian density model which is trained discriminatively (*i.e.* to separate the classes) has two benefits. The classification performance is high and moderation is possible analytically using the results from previous work [16]. The classifier proposed consists of a set of Gaussian distributions, one for each class with the error criterion being applied to the posterior class probabilities as computed by Bayes' rule. The network was developed from work done by the author in 1987 [10] and continued in collaboration with Bridle in 1989 [38] and [39]. As in the standard MLP formulation, the input vector *{G_i}* is first subjected to a linear transformation:

$$H_j = \sum_i v_{ij} G_i$$

and the output N_j of the first layer units mirror their input:

$$N_j = H_j$$

The squared Euclidean distance of these transformed inputs is then measured to a set of reference vectors *{m_{jk}}*:

$$I_k = -\sum_j (m_{jk} - N_j)^2$$

this relates to the exponent of a Gaussian distribution with variance $\frac{1}{2}$ so that the Gaussian response and normalisation can be applied simultaneously giving:

$$O_k = \frac{e^{I_k}}{\sum_k e^{I_k}}$$

Figure 7-2 shows a diagrammatic representation of the network. The input pattern is linearly transformed and the response of a set of Gaussian distributions measured. The responses are then normalised.

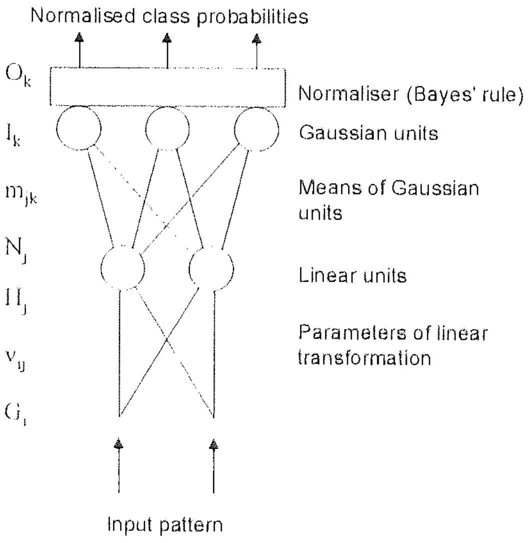

Figure 7-2: The discriminative Gaussian classifier network.

Bridle [38] pointed out that the relative-entropy scoring criterion, is particularly suitable for this network architecture as the computation of the derivative of the criterion function with respect to each of the parameters is considerably simplified (see Appendix A.6).

With these derivatives we may proceed to iteratively optimise the weight values using an appropriate gradient-based non-linear optimisation algorithm (such as gradient descent or conjugate gradient descent [170]). However, we may now initialise our weight values sensibly (rather than using small random values as is the case for the standard MLP). We may initialise the first layer linear transformation to the identity transformation (weights set to $v_{jk} = \delta_{jk}$) and the reference points to the means of the data associated with the corresponding class.

7.3 *Discounted Moderation*

A basic form of moderation may be used for partially circumventing the problems described in the previous section. In this case we discount the probabilities derived from the design data, D, using prior information. Two strategies are employed; linear discounting or winner-takes-all discounting. In either case the class B likelihood under test conditions, $P(x \mid D, B')$, is a combination of the likelihood prescribed by the design data, $P(x \mid D, B)$, and the class B prior, $P(x \mid B)$.

7.3.1 Linear Discounting

For linear discounting the combination rule is a linear mixing parameter, λ ($0 \le \lambda \le 1$):

$$P(x \mid \lambda, D, B) = \lambda P(x \mid D, B) + (1 - \lambda) P(x \mid B)$$

The extreme value, $\lambda = 1$, corresponds to using the original design likelihood as described in the previous section. Use of $\lambda = 0$ corresponds to using the prior likelihood only. Figure 7-3 illustrates the distributions being mixed and the resulting discounted distribution. Note that the discounted distribution has heavier tails than the design distribution.

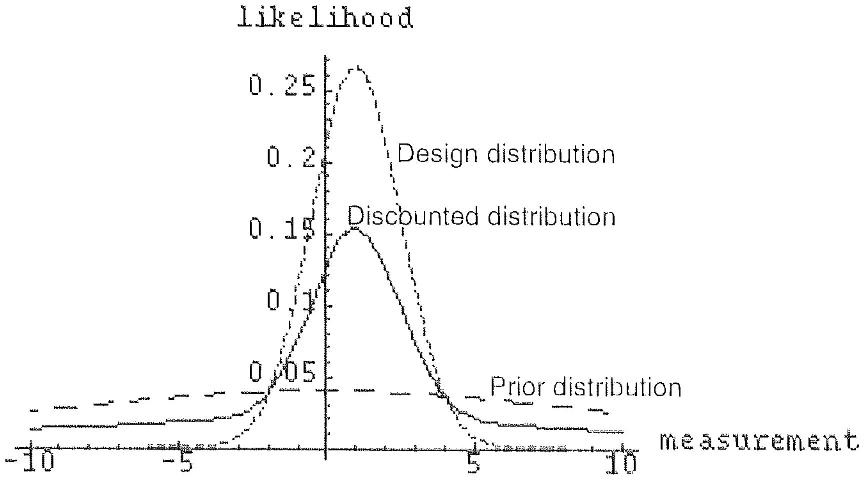

Figure 7-3: The likelihood functions for design, prior distribution linear discounting.

The value of λ may be found using a uniform one-dimensional search using a small sample of retuning data (assumed to be available at the start of testing). In practice each sensor in the fusion system would require a different value of λ. However, for the purpose of illustration we assume that the sensors are identical. In this case the estimation of λ may be pooled, thereby increasing accuracy.

7.3.2 Winner-Takes-All Discounting

One of the distracting facets of the linear discounting model is the requirement to find a value for the mixing parameter, λ. The winner-takes-all discounting model is parameter free and therefore does not require optimisation. In this model the maximum likelihood from either the design distributions or the prior is used for classification (with appropriate normalisation to ensure that the probabilities integrate to one):

$$P(x \mid M, D, B) \propto \max\{P(x \mid D, B), P(x \mid B)\}$$

Figure 7-4 shows the application of winner-takes-all discounting to the same distributions shown for the linear discounting (Figure 7-3). Again, the heavier tails of the discounted distribution are clearly visible.

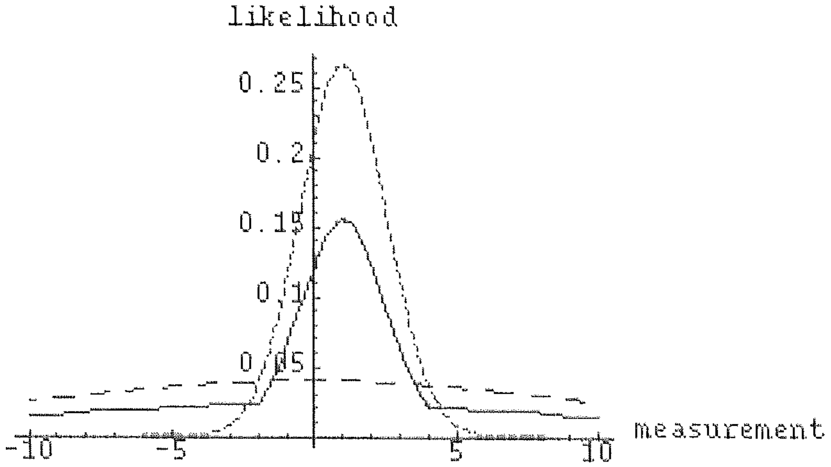

Figure 7-4: Likelihood functions for winner-takes-all discounting.

The performance of both the linear discounting and the winner-takes-all discounting methods were assessed. A probability-level fusion system comprising ten identical sensors was simulated. For each sensor the class A distribution was a standard normal and the class *B* design distribution was a Gaussian with unit standard deviation and a mean of one. The prior distribution was taken to be a Gaussian with mean zero and standard deviation 10. For each value of λ the fused error rate was calculated using a Monte Carlo simulation in which 50,000 samples were drawn from each of the two classes. The optimum value of λ was also calculated using a line search technique based on just 10 labelled samples from each class (the retuning dataset). This retuning experiment was repeated 5,000 times and a histogram of the best value for λ was built up. The results are shown in Figure 7-5 for a test distribution for class *B* with altered mean of *N(0.8,1)*. In this case neither technique reduces the fused error rate. The optimum value of λ is 1.0 (*i.e.* no inclusion of the prior distribution) which was found in over a quarter of the simulations (mean value of $\lambda = 0.93$). Using the mean value for λ the fused error rate was found to be 13.1% compared to 11.5% obtained without discounting (this difference is statistically significant).

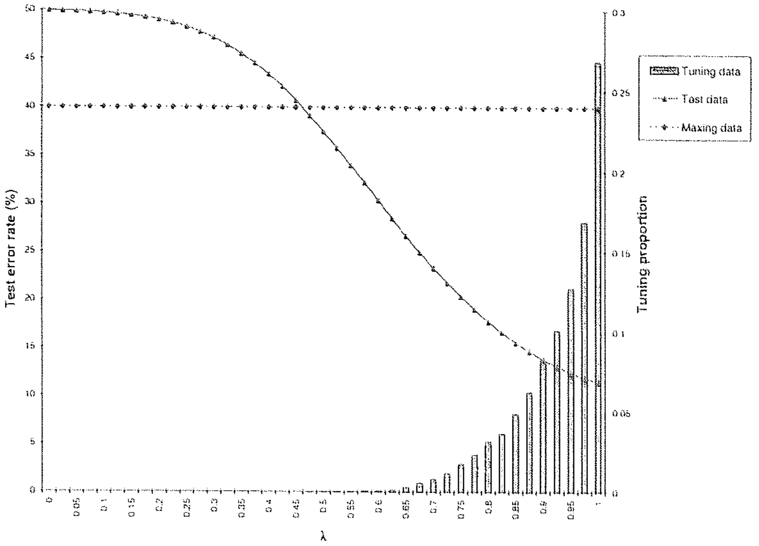

Figure 7-5: Fused error rate as a function of λ for distribution *$N(0.8, 1^2)$*.

The same experiment was repeated for a broadened test distribution for class *B* of *$N(1, 2^2)$*. In this case, most values for λ above 0.25 for the linear discounting method gave better performance than that obtained not using discounting. The optimum value for λ was approximately 0.8 which resulted in a fused error rate of 6.3%. The winner-takes-all discounting method also reduced the error rate (to 10.4%). Both of these improvements are statistically significant.

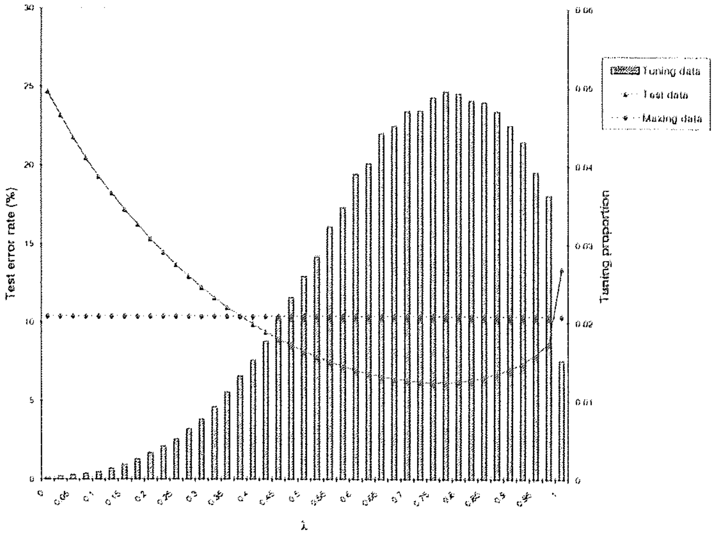

Figure 7-6: Fused error rate as a function of λ for distribution *$N(0.8, 2^2)$*.

Figure 7-7 shows the fused error rates for a system of four sensors as a function of the Kullback-Leibler number for the two discounting techniques and the standard approach. It can be seen that the winner-takes-all method produces an improvement in fused error rate over a range of conditions. The linear discounting method gives as good or better performance than the design distributions in all cases.

In these experiments a prior of $N(0,10^2)$ was used. The sensitivity of the approaches to choice of prior was also investigated. In further experiments the mean of the prior was varied from 0→1 and was found to make no significant difference to the results obtained. The standard deviation of the prior was also varied from 5→100. Altering the width of the prior did have an effect on performance with fused error rate gradually increasing towards that of the design distribution classifier. For most plausible values of the prior, however, using the discounting methods lead to a significant improvement in the fused performance

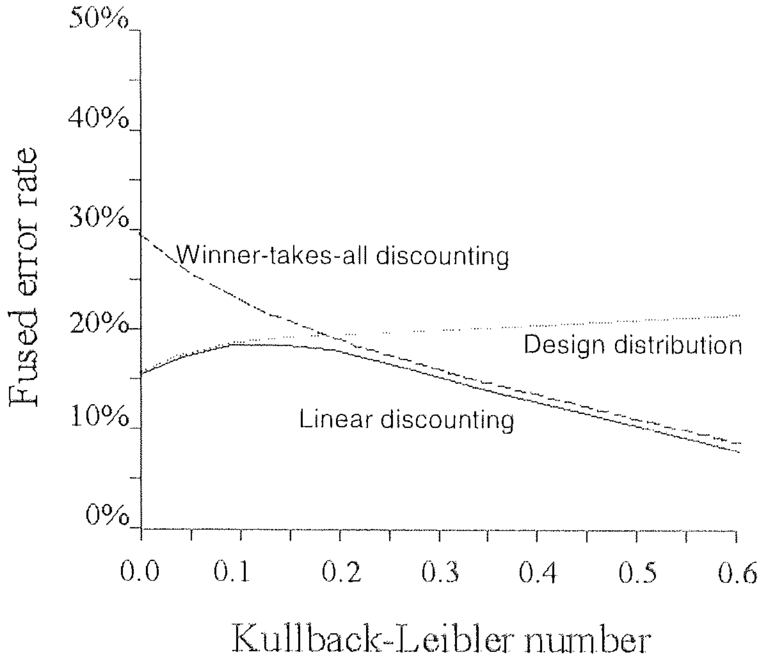

Figure 7-7: Fused error rate for the design and discounted models.

7.4 Mix Moderation

We have already observed the Bayesian posterior probability of a particular class given a set of measurements and a corpus of training data:

$$P(C=c \mid x, D)=\int dM \int \mathrm{d}\theta_M \; P(C=c \mid x, D, M, \theta_M) P(\theta_M \mid x, D, M) P(M \mid x, D)$$

The analytic moderation of previous sections has addressed the uncertainty in the parameter values of the selected model. In this section we address the uncertainty in the selection of the model itself. Ignoring the parameter values for the moment we therefore require a solution to the integral:

$$P(C=c \mid x, D)=\int dM \;\; P(C=c \mid x, D, M) P(M \mid x, D)$$

Since it will not, in general, be feasible to integrate over **all** possible models, only approximate solutions to this integral are addressed. The simple approach favoured here, though not an accurate approximation introduces the philosophy of not taking the single best model. It replaces the integral over all possible models with the sum over all models that have been evaluated:

$$\int dM \;\; P(C=c \mid x, D, M) P(M \mid x, D) \approx \sum_i P(C=c \mid x, D, M_i) P(M_i \mid x, D)$$

If a sufficient variety of models, $M_{i,}$ are chosen to provide some degree of coverage of the space of all models, then this is likely to provide a more accurate probability than taking the single best model [163]. Two main issues present themselves:

- ❑ What range of models should be developed?
- ❑ How should the probability of the model given the data be evaluated?

7.4.1 Classifier Combination by Averaging

A simple method for allocating the model probabilities (or at least the relative probabilities since we expect the distribution to integrate to unity) is to set them to be the same and equal to $1/_{N_M}$, where N_M denotes the number of models used. Since this is likely to give undue weight to poor models it is proposed that the best N^*_M models are selected (based on some performance measure such as cross entropy or percent correct) and used in the summation with equal weight:

$$\int dM \quad P(C=c \mid x, D, M)P(M \mid x, D) \approx \frac{1}{N^*_M}\sum_i P(C=c \mid x, D, M_i)$$

We therefore average the class conditional probabilities output by the various classifiers. This introduces some degree of robustness to overfitting, particularly if a good selection of models is used. It does, however, fail to give extra weight to those models that are fitting the data well. This may be alleviated by weighting the models according to some simple function of the error.

7.4.2 Classifier Combining by Likelihood Modelling

We retain the basic model for combining conditional probabilities:

$$\int dM \quad P(C=c \mid x, D, M)P(M \mid x, D) \approx \sum_i P(C=c \mid x, D, M_i)P(M_i \mid x, D)$$

but use a different method for assessing the model probability. In this case obtained from the data itself using Bayes' rule:

$$P(M_i \mid D) = \frac{P(D \mid M_i)P(M_i)}{P(D)}$$

Now $P(D \mid M_i)$ can be obtained from the model and the data since, assuming independence of individual data points, D_j:

$$P(D \mid M_i) = \prod_j P(D_j \mid M_i)$$

We may reasonably ignore the probability of the data, *P(D)*, since it will be independent of the model and can be allowed for by scaling the final weights. The prior on the models $P(M_i)$ should reflect the simplicity (or otherwise) of model M_i. We may set $P(M_i)$ to be inversely proportional to the number of free parameters in the model. This approach relies on each classifier producing $P(D_j \mid M_i)$ as an intermediate stage in the production of the class conditional probability. There are many such classifiers that could usefully be combined in this way. However, there are many discrimination-based classifiers, which estimate the class conditional probability directly and never produce the data likelihood. These classifiers are useful in that they are optimised to reduce the mis-classification rate, which is the quantity ultimately of interest to the decision-maker.

7.5 *Chapter Discussion*

We have seen that, even when the marginalisation integrals are intractable, there remain courses of action for ensuring that the probabilities used in a data fusion centre are not unduly confident. The discriminative Gaussian classifier is trained discriminatively and is capable of producing as good a classification performance as other simple discriminative techniques. It offers the additional advantage, however, of being amenable to moderation using the analytic techniques introduced in Chapter 6 . For the sake of completeness two other approaches were described which achieve similar moderating results by mixing the classification probabilities with those from the prior distributions or other classifiers. This work is ongoing.

Chapter 8 Centralised Moderation

8.1 *Chapter Introduction*

In this section a method is developed which allows moderation to be carried out at the fusion centre when access to the separate sensor processors is denied. This centralised algorithm shows that less certainty at the sensor level can lead to increased infallibility at the fusion centre and therefore higher overall data fusion performance. In this case transformations are applied at the fusion centre which can correct for inappropriate sensor processing. We select a Chebyshev transform for this purpose and demonstrate that the coefficients of this transform may be optimised using ground truth labelled data. The principle is illustrated using synthetic data.

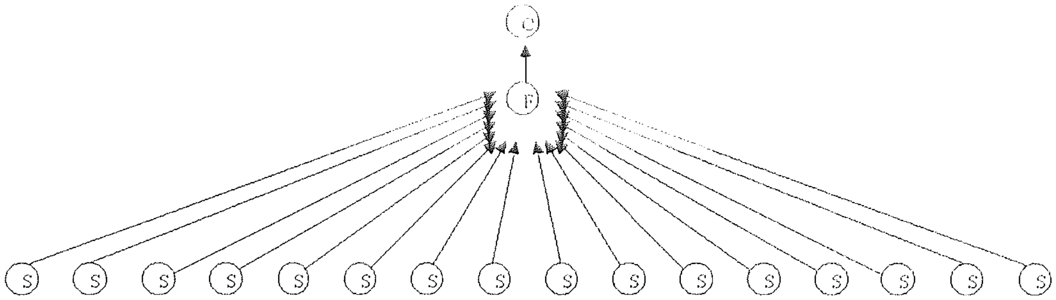

Figure 8-1: The centralised moderation data fusion architecture.

We now consider the situation in which a probability-level fusion centre is presented with class conditional probabilities from a set of sensors that have not been appropriately moderated. We shall show that it is possible for some moderation to occur at the fusion centre using a set of adaptive transforms that may be learnt from training data.

Consider the case where a sensor outputs maximum likelihood estimates of the class conditional probabilities but the fusion centre is aware of the desired, moderated probabilities. Figure 8-2 shows the mapping of maximum likelihood probabilities (on the horizontal axis) to moderated probabilities (on the vertical axis) for the samples from sensor 1 under consideration in the previous sections. Note that the mapping is not single valued but that a trend is clearly visible. We shall exploit this trend by designing a transformation which the fusion centre applies to incoming (unmoderated) probabilities which converts them to moderated probabilities.

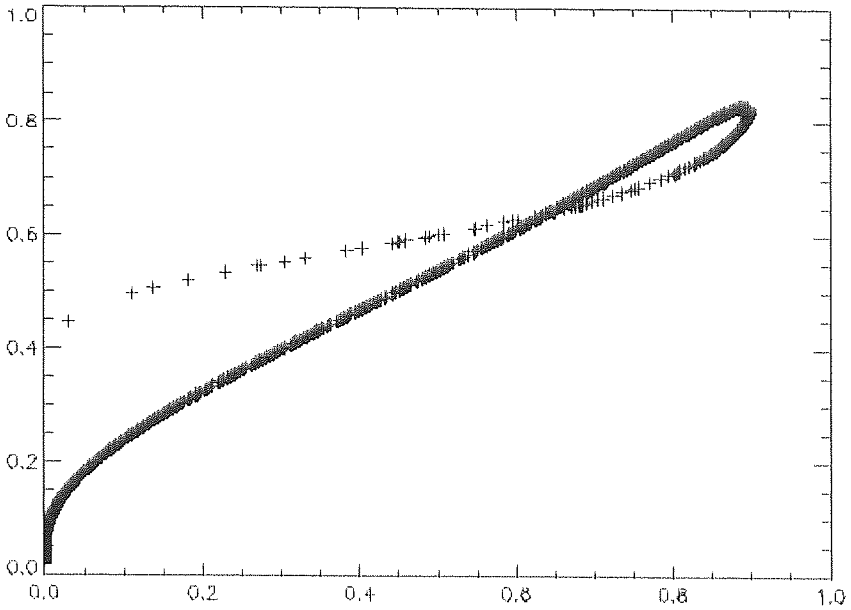

Figure 8-2: Maximum likelihood versus moderated conditional probabilities.

We shall use a transformation on the interval *[0,1]* and select rescaled versions of the Chebyshev polynomials as a basis set for this purpose (although they are orthogonal we do not use this property in this instance):

$$\begin{aligned}
T_0(\frac{x+1}{2}) &= 1 \\
T_1(\frac{x+1}{2}) &= x \\
T_2(\frac{x+1}{2}) &= 2x^2 - 1 \\
T_3(\frac{x+1}{2}) &= 4x^3 - 3x \\
T_4(\frac{x+1}{2}) &= 8x^4 - 8x^2 + 1 \\
T_5(\frac{x+1}{2}) &= 16x^7 - 20x^3 + 5x
\end{aligned}$$

We have initially used the first six such polynomials $T_0 T_1 T_2 T_3 T_4$ and T_5 as our basis set. The fusion process now proceeds as follows. First the incoming probabilities, P_{ij}, corresponding to the conditional probability of class i estimated by sensor j, are transformed using the basis set described above:

$$P'_{ij} = \sum_{k\in\{0,1,2,3,4,5\}} t_{kj} T_k\left(\frac{P_{ij}+1}{2}\right)$$

where t_{kj} denotes the weight applied to the kth basis function for sensor j. These transformed probabilities are multiplied and renormalised in the standard way:

$$P'_i = \frac{\prod_j P'_{ij}}{\Pr(i)^{N-1}}$$

where the prior probability of the ith class is given by $\Pr(i)$ and the number of sensors is N. To determine the parameters of the transformation we define a sum squared error criterion between the fused, transformed probabilities and the ground truth, D_i:

$$E = \frac{1}{2}\sum_i (D_i - P''_i)^2$$

where

$$P''_i = \frac{P'_i}{\sum_j P'_j}$$

and calculate the partial derivatives of the error criterion with respect to each of the parameters:

$$\frac{\partial E}{\partial P''_i} = P''_i - D_i$$

and

$$\frac{\partial P''_i}{\partial P'_i} = \frac{\sum_j \left(P'_j - P'_i\right)}{\left(\sum_j P'_j\right)^2}$$

then

$$\frac{\partial P_i^{'}}{\partial P_{ij}^{'}} = \frac{\prod_k P_{ik}^{'}}{\Pr(i) P_i^{'}}$$

and finally

$$\frac{\partial P_{ij}^{'}}{\partial t_{kj}} = T_k\left(\frac{P_{ij}+1}{2}\right)$$

The hill-climbing gradient for each parameter can now be obtained using the chain rule and the parameters iteratively updated using a training database. For our experiments we changed the parameters in proportion to the gradient with a constant of proportionality equal to 0.01.

8.2 *Centralised Moderation with Insufficient Design Data*

This procedure was carried out for the same sensor data that has been analysed in previous sections with a new set of 1,000 sensor samples. After 1,000 iterations the parameters of each of the sensor transformations had ceased changing. The transformation found for the first sensor is shown in Figure 8-3.

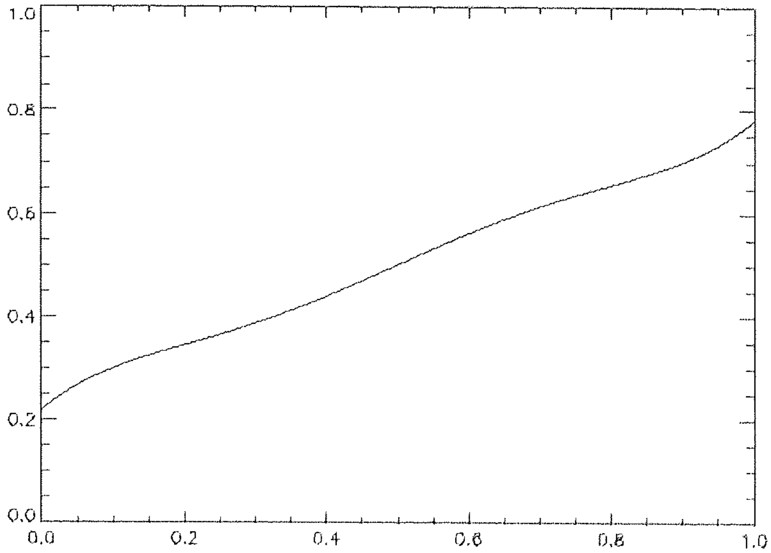

Figure 8-3: The adapted transformation for mapping maximum likelihood probabilities.

Note that the trend is to moderate the probabilities prior to fusion. Using this transformation we calculated the error rates using the same evaluation database used previously. Results are shown in Table 8-1.

Data Source	Method	Error rate
Joint distribution	*a priori*	24%
Fused sensors 1&2	ML	30%
Fused sensors 1&2	moderated	26%
Fused sensors 1&2	transformed	26%

Table 8-1: Fused error rates for moderated and transformed conditional probabilities.

Observe that the fused error rate has been reduced to the same value that would have been obtained had Bayesian probability moderation been carried out at the individual sensors. The veto effect has been eliminated despite the presence of (initially) overconfident class conditional probability estimates.

8.3 *Centralised Moderation with Unrepresentative Design Data*

To assess the applicability of the centralised moderation algorithm on probabilities that result from unrepresentative design data a second set of experiments was performed. The design data is the same as before comprising two classes each having a unit normal distribution with means at zero and one respectively. We assume that there is sufficient design data to accurately estimate these distributions. The evaluation data, however, is somewhat different. The data from class 1 is drawn from the same unit normal centred at the origin as estimated from the design data. The distribution of the second class is broader than the design distribution. This is illustrated in Figure 8-4. Both the design and evaluation distributions are the same for each of the two sensors. The optimisation of the parametric moderating transformation proceeded exactly as before except that 2,000 iterations were employed in this case.

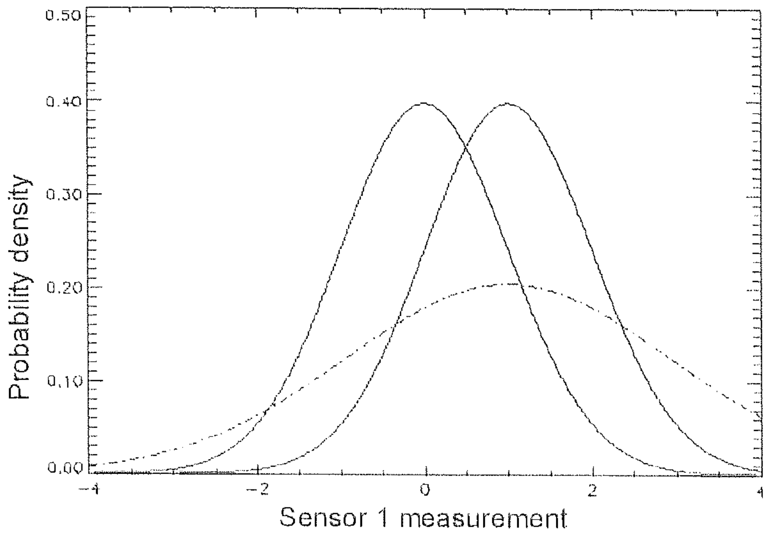

Figure 8-4: The design and underlying distributions for each of the two sensors.

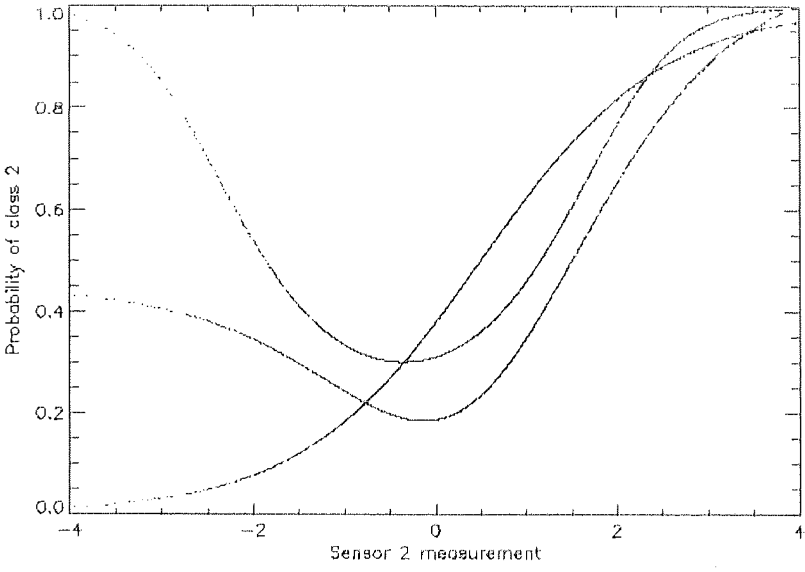

Figure 8-5: The conditional probability of class 2 for one of the sensors.

Figure 8-5 shows the probability of class 2 derived from three sources when the underlying distribution has twice the standard deviation of the design distribution. The sigmoid shows the probability according to the design distributions as produced by each of the sensors. The highest curve shows the probability of class 2 given the (unobservable) underlying distributions. Part way between these is the transformed probability produced by the centralised moderation algorithm.

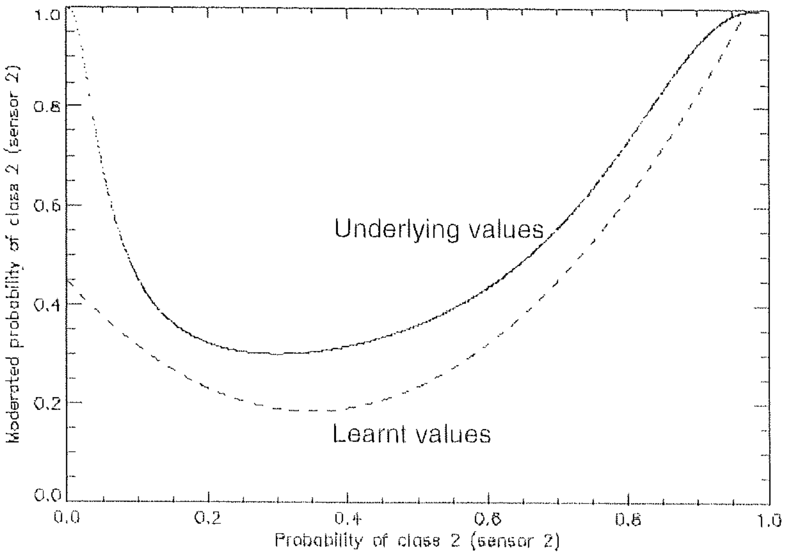

Figure 8-6: The mapping of raw sensor derived probabilities to moderated probabilities.

The mapping of design distribution probabilities to underlying probabilities is shown in Figure 8-6 as the upper curve. Also shown is the transformation found following the optimisation process.

The same experiment was performed for a variety of underlying distribution with different standard deviations. The error rate of the fusion system based on the underlying distributions falls with increased dispersion as evidenced by the lowest of the three curves in Figure 8-7. The fused error rate based on the design distributions gets worse with the increasing difference between the underlying and design distributions. The middle curve, which closely tracks the lower error rate, shows the performance for the centralised moderation algorithm described above.

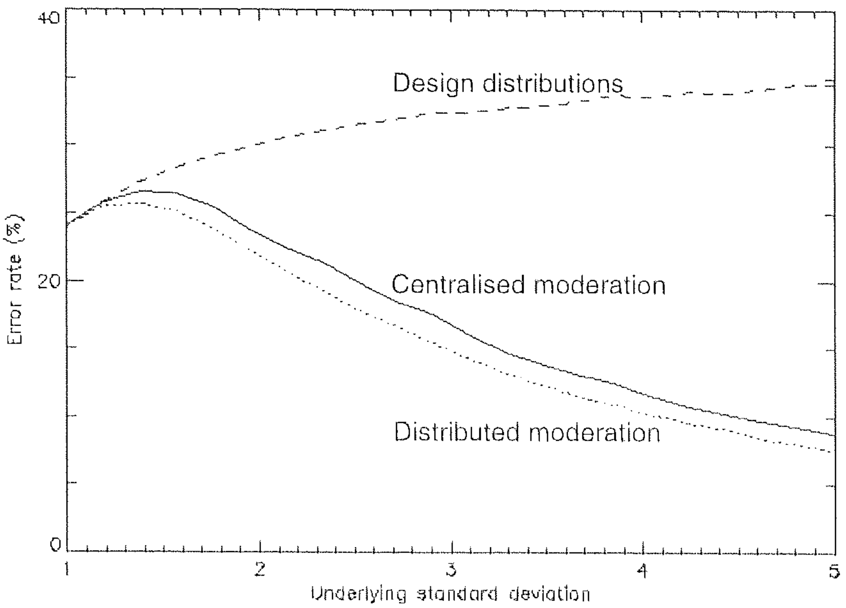

Figure 8-7: The fused error rates as a function of the standard deviation of class 2.

8.4 *Chapter Discussion*

In this chapter we described the concept of centralised moderation where the fusion centre itself performs the moderating transformation to the conditional probabilities from the separate sources. An algorithm was presented which learned such a transformation given only the labels of the design data and not the correctly moderated probabilities from each source. The resulting transformation was seen to provide a plausible mapping from unmoderated to moderated probabilities and was shown to recover the moderated fused performance for simple synthetic problems. When applied to real data the technique was shown to make improvements over unmoderated fusion, which were both statistically and operationally significant.

Conclusions

The thesis began with the question:

> *"How does one obtain effective fusion which is robust to the quality of the identification information being fused?"*

During the course of the research it became clear that such a question was not well defined for many of the profusion of data fusion systems currently deployed (largely in the defence domain). A major failing of the original objective was its breadth. In order to constrain the problem sufficiently to allow analysis and computer experimentation, it was first necessary to map the field of data fusion. No single source provided the required framework for approaches, architectures and functional topologies. In the first part of the thesis an abbreviated description of such frameworks were provided. It was concluded that a multi-stage, cyclic model best described the data fusion process. The model identified high-level fusion (fusion of decisions or probabilities) as being of practical relevance and theoretical interest. The framework allowed us to hone the question as it applied to decision level fusion in multi-level data fusion systems using a centralised architecture. The development and assessment of data fusion algorithms that improve or allow for information quality was the subject of the second part of the thesis.

Part two began by examining decision level fusion. An adaptive technique introduced was able to maintain a desired error rate in the presence of poor quality information. The method used a sequential error rate test coupled with a simple adaptive deferral threshold. However, it was thought that such a scheme was not making best use of the data available and we therefore turned our attention to probability level fusion. We showed that probability fusion could be used as the basis for rational decision making and that the accuracy of the probabilities being fused was of importance.

In chapters 6, 7 and 8 we developed the idea of probability moderation. Three approaches were assessed (analytic, heuristic and centralised). In each case significant improvements were found to be possible. The analytic approach was found to give good results when the underlying distributions were Gaussian (or nearly so). The utility of the approach was also demonstrated on a discriminative Gaussian classifier network. Extensions of the analytic approach to a richer variety of classifiers are left for future work. The heuristic approaches were motivated by the need for widely applicable, simple techniques and were shown to give adequate results in some cases. The final technique of centralised moderation was a novel approach to the moderation problem. It used non-linear transformations for each source, coupled though the subsequent probability fusion. The transformations were learned from training data in much the same way as a neural network. It has subsequently been applied to a number of defence-related classification problems and found to offer operationally dramatic performance gains. The technique is now being further refined and evaluated against other problems.

We conclude that high-level data fusion is a useful, low-bandwidth technique for increasing the performance of identification systems as long as measures are taken to ensure that the source information is appropriately accurate. Several methods for addressing this accuracy and / or allowing for it in the fusion process were presented.

List of References

This abridged bibliography contains some of the key references to data fusion and related topics from what is an enormous, and still expanding, open literature. Since much of the funding for data fusion research is derived from defence sources this represents a tiny fraction of the total output of the data fusion community. References are ordered alphabetically by surname of first author. Where a first author is cited more than once the individual references are ordered chronologically. Wherever possible full reference information is provided. Some of the papers and reports obtained during the course of this research, however, are internal working papers – these are generally not obtainable without contacting the authors directly.

1. M. Abdulghafour, J. Goddard and M. Abidi, "Non-deterministic Approaches to Data Fusion", *SPIE 1383 - Sensor Fusion III, pp. 596-610* (1991).

2. M. Abramowitz and I. Stegun, "Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables", *Dover Publications, Inc.* (1965).

3. S. Akkihal and G. Wise, "Data Fusion with a Faulty Sensor", *Proceedings of the American Control Conference, pp. 638-642* (1995).

4. C. Arndt, "Sensor Data Fusion", *unpublished University of Siegen report* (1997).

5. J. Austin, M. Bedworth, M. Bernhardt, P. Greenway, C. Harris, R. Johnston, A. Little, D. Lowe and M. Markin, "Technology Foresight on Data Fusion and Data Processing", *The Royal Aeronautical Society*, (1997).

6. W. Baek and S. Bommareddy, "Optimal M-ary Data Fusion with Distributed Sensors", *IEEE Aerospace and Electronic Systems*, Volume 31, Number 3, pp. 1150-1152 (1995).

7. A. Barker, "Neural Network Applications to Sensor Data Fusion", *PhD Thesis, University of Virginia,* (1989).

8. O. Basir and H. Shen, "Sensory Data Integration: A Team Consensus Approach", *Proceedings of IEEE conference on Robotics and Automation* pp. 1683-1688 (1992).

9. M. Beckerman, "A Bayes – Maximum Entropy Method for Multi-Sensor Data Fusion", *Proceedings of IEEE – Robotics and Automation, pp. 1668-1674* (1992).

10. M. Bedworth, "Using Error Back Propagation: Some Alternatives to Logistic Networks", *RSRE Research Note SP4/75* (1988).

11. M. Bedworth and A. Heading, "Classification of Ship Silhouettes by Pattern Recognition and Data Fusion", *DRA Technical Report CSE1/227* (1992).

12. M. Bedworth and A. Heading, "Detection and Identification of the Vapour Fingerprints of Plastic Explosives using Pattern Recognition and Data Fusion", *DRA Technical Report CSE1/229* (1992).

13. M. Bedworth, "On the Quantity and Quality of Data and Pattern Recognition", *RSRE Memorandum 4712,* (1992).

14. M. Bedworth, M. Cooper and A. Heading, "Identification of Ships from Data Fusion of Infrared Images", *DRA Technical Report CSE1/226* (1992).

15. M. Bedworth and A. Heading, "The Importance of Models in Bayesian Data Fusion", *Proceedings of IEEE conference on Control Applications, pp. 410-414* (1992).

16. M. Bedworth, "Probability Moderation for Multilevel Information Processing", DRA Technical Report DRA/CIS(SE1)/651/8/M94.AS03BP032/1 (1992).

17. M. Bedworth, "Sensor Selection Methods for Data Fusion Systems", *DRA Technical Report TP5/85/4/31* (1993).

18. M. Bedworth, "Statement of Airpower", *DRA Technical Report 651/8/HIP/GIP/RP5* (1996).

19. M. Bedworth, "Multilevel Information Processing: Final Report", *DERA Report DERA/CIS/CIS5/CR97221/1* (1997).

20. M. Bedworth, "Less Certain, More Reliable", *Proceedings of FUSION'98, Las Vegas, pp. 572-580* (1998).

21. M. Bedworth and J. O'Brien, "Sensor Reliability Modelling for Ground Vehicle Reognition", *Proceedings of NATO/IRIS Conference on Data Fusion, Quebec,* (1998).

22. M. Bedworth, "A Neural Network Approach to Feature-Level Fusion", *Proceedings of Data and Information Symposium IDC'99, Adelaide, pp. 597-604* (1999).

23. M. Bedworth, "Multi-Sensor Data Fusion for Ship Discrimination using Sensor Reliability Models", *Proceedings of Data and Information Symposium IDC'99, Adelaide* (1999).

24. M. Bedworth, "Defining Decision-Level Fusion Rules using Small Samples", *to be published in Proceedings of SPIE AeroSense - Sensor Fusion: Architectures, Algorithms and Applications III* (1999).

25. M. Bedworth, J. Llinas and J. O'Brien, "The Necessity of International Collaboration in Data Fusion and a Mechanism for Easing the Process", *submitted to FUSION'99, Sunnyvale* (1999).

26. M. Bedworth and J. O'Brien, "The Omnibus Model: A New Architecture for Data Fusion", *submitted to FUSION'99, Sunnyvale* (1999).

27. S. Belur and B. Dasarathy, "Optimal Features-in Features-out (FEI-FEO) Fusion for Decisions in Multisensor Environments", *SPIE 3376 - Sensor Fusion: Architectures, Algorithms and Applications II* (1998).

28. P. Bergeron, J. Couture, J. Duquet, M. Macieszczak and M. Mayrand, "A New Knowledge-based system for the Study of Situation and Threat Assessment in the Context of Naval Warfare", *Proceedings of FUSION'98, pp. 388-395* (1998).

29. A. Berler and S. Shimony, "Bayes Networks for Sensor Fusion in Occupancy Grids", *Technical Report of the Ben Gurion University of the Negev* (1997).

30. J. Black, "Fusion of Infrared and Visible-Light Images", *Proceedings of Command Information Systems, Oxford* (1994).

31. J. Black, "Fusion of Infrared and Visible-Light Images", *Proceedings of Battlefield Systems International, Chertsey* (1994).

32. J. Black and M. Bedworth, "Quantisation for Probability-level Fusion on a Bandwidth Budget", *SPIE 3376 - Sensor Fusion: Architectures, Algorithms and Applications II, pp. 152-160* (1998).

33. R. Blum, S. Kassam and H. Poor, "Distributed Detection with Multiple Sensors: Advanced Topics", *Proceedings of IEEE,* Volume 85, Number 1, pp. 64-79 (1997).

34. D. Booth, N. Thacker and J. Mayhew, "Data Fusion using an MLP", *RIPRREP/1000/79/80* (1990).

35. E. Bosse and J. Simard, "Identity and Attribute Information Fusion using Evidential Reasoning", *SPIE 3067 – Sensor Fusion: Architectures, Algorithms and Applications, pp. 38-49* (1997).

36. J. Boyd, "A Discourse on Winning and Losing", *slides of Maxwell AFB lecture* (1987).

37. G. Box and G. Tiao, "Bayesian Inference and Statistical Analysis", *John Wiley & Sons* (1992).

38. J. Bridle, "Probabilistic Interpretation of Feedforward Classification Network Outputs with Relationships to Statistical Pattern Recognition", *in Neuro-computing: Algorithms, Architectures and Applications, Springer-Verlag* (1989).

39. J. Bridle, "Training Stochastic Model Recognition Algorithms as Networks can lead to Maximum Mutual Information Estimation of Parameters", *in Advances in Neural Information Processing Systems* 2 (1989).

40. D. Broomhead and D. Lowe, "Multivariable Function Interpolation and Adaptive Networks", *Complex Systems 2, pp. 321-355* (1988).

41. D. Buede and E. Waltz, "Benefits of Soft Sensors and Probabilistic Fusion", *SPIE 1096 – Signal and Data Processing of Small Targets, pp. 309-320* (1989).

42. K. Byrd, B. Smith, D. Allen, N. Morris, C. Bjork and K. Deal-Giblin, "Intelligent Processing Techniques for Sensor Fusion", *SPIE 3376 - Sensor Fusion: Architectures, Algorithms and Applications II, pp. 2-15* (1998).

43. D. Campos and J. Llinas, "Possibilities for Improved Formal Processing Methods for Situation Assessment", *Proceedings of FUSION'98, pp. 462-469* (1998).

44. Z. Chair and P. Varshney, "Optimal Data Fusion in Multiple Sensor Detection Systems", *IEEE Aerospace and Electronic Systems,* Volume 21, Number 1, pp. 98-101 (1986).

45. C. Chatfield, "Problem Solving A Statistician's Guide", *Chapman and Hall,* (1988).

46. H. Chernoff and L. Moses, "Elementary Decision Theory", *John Wiley and Sons, Inc.* (1959).

47. C. Chong, "Distributed Architectures for Data Fusion", *Proceedings of FUSION'98, pp. 84-92* (1998).

48. C. Clopper and E. Pearson, "The use of Confidence or Fiducial Limits Illustrated in the case of the Binomial", *Biometrika,* Volume 26, pp. 404-413 (1934).

49. M. Cooper and M. Miller, "Information Gain in Object Recognition Via Sensor Fusion", *Proceedings of FUSION'98, pp. 143-148* (1998).

50. T. Cover, "Geometrical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition", *IEEE Transactions on Electronic Computers,* Volume14 (1965).

51. T. Cover and P. Hart, "Nearest Neighbour Classification", *IEEE Transactions on Information Theory,* Volume 13 (1967).

52. J. Crowley and Y. Demazeau, "Principles and Techniques for Sensor Data Fusion", *Signal Processing,* Volume 32, Number 1, pp. 5-27 (1993).

53. B. Dasarathy, "Nosing Around the Neighbourhood, A New System Structure and Classification Rule for Recognition in Partially Exposed Environments", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Volume 2, Number 1 (1980).

54. B. Dasarathy, "Paradigms for Information Processing in Multisensor Environments", *SPIE 1306 – Sensor Fusion III, pp. 69-80* (1990).

55. B. Dasarathy, "Sensor Fusion: Architectures, Algorithms and Applications", *SPIE short course SC39* (1997).

56. B. Dasarathy, "Sensor Fusion Potential Exploitation – Innovative Architectures and Illustrative Applications", *Proceedings of IEEE,* Volume 85, Number 1, pp. 24-38 (1997).

57. B. Dasarathy, "Asymmetric Fusion Strategies for Target Detection in Multisensor Environments", *SPIE 3067 – Sensor Fusion: Architectures, Algorithms and Applications, pp. 26-37* (1997).

58. B. Dasarathy, "A Trainable Decisions-in Decisions-out (DEI-DEO) Fusion System", *SPIE 3376 - Sensor Fusion: Architectures, Algorithms and Applications II* (1998).

59. B. Dasarathy and S. Townsend, "GIFTS: A Guide to Intelligent Fusion Technology Selection", *Proceedings of FUSION'98, pp. 65-72* (1998).

60. K. Demirbas, "MAP Approach to Object Recognition with Distributed Sensors", *IEEE Aerospace and Electronic Systems*, Volume 24, Number 3 pp. 309-313 (1988).

61. P. DeVijver and J. Kittler, "Pattern Recognition: A Statistical Approach", *Prentice Hall International Press* (1983).

62. M. Druzdzel and L. van der Gaag, "Elicitation of Probabilities for Belief Networks: Combining Qualitative and Quantitative Information", *Proceedings of 11th Annual Conference on Uncertainty in Artificial Intelligence, pp. 141-148* (1995).

63. R. Duda and P. Hart, "Pattern Classification and Scene Analysis", *John Wiley & Sons* (1973).

64. J. Duquet, P. Bergeron, D. Blodgett, J. Couture, M. Macieszczak, M. Mayrand, B. Chalmers and S. Paradis, "Functional and Real-time Requirements of a Multisensor Data Fusion (MSDF) Situation and Threat Assessment (STA) Resource Management (RM) System", *SPIE 3067 – Sensor Fusion: Architectures, Algorithms and Applications* (1997).

65. P. Edwards, "Data Fusion in the Land Battle Environment", *AGARD CP-440*, 16, pp. 16-1-16.9 (1988).

66. S. Farmer and M. Bedworth, "Pattern Processing Methods for the IMSA System", *DERA Technical Report DRA/CIS(SE1)/651/HIP/GIP/RP11/1* (1997).

67. R. Fisher, "The Use of Multiple Measurements in Taxonomic Problems", *Annual Eugenics*, Volume 7, Part II, pp.179-188 (1936).

68. D. Foley, "Considerations of Sample Size and Feature Size", *IEEE Transactions on Information Theory*, Volume 18, Number 5, pp 618-626 (1972).

69. K. Fukunaga and D. Kessell, "Estimation of Classification Error", *IEEE Transactions on Computers*, Volume 20, Number 12, pp 1521-1527 (1971).

70. K. Fukunaga and R. Hayes, "Effects of Sample Size in Classifier Design", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 11, Number 8 (1989).

71. J. Gebhardt and R. Kruse, "Information Source Modelling for Consistent Data Fusion", *Proceedings of FUSION'98, pp. 27-34* (1998).

72. E. Geraniotis and Y. Chan, "Robust Data Fusion for Multisensor Detection Systems", *IEEE Transactions on Information Theory*, Volume 36, Number 6, pp. 1265-1279 (1990).

73. J. Ghosh, S. Beck and C. Chu, "Evidence Combination Techniques for Robust Classification of Short-duration Oceanic Signals", *University of Texas, Austin Technical Report* (1992).

74. W. Gossett, "The Probable Error of the Mean", *Biometrika*, Volume 6, Number 1 (1908).

75. I. Gradshteyn and M. Ryshik, "Table of Integrals, Series and Products", *Academic Press Inc.* (1980).

76. P. Greenway, R. Deaves and D. Bull, "Communications Management in Decentralised Data Fusion Systems", *Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 8242 (1996).

77. D. Hall and J. Llinas, "Data Fusion and Multisensor Correlation", *Technology Training Corporation course*, (1985)

78. D. Hall, "Mathematical Techniques in Multisensor Data Fusion", *Artech House* (1992).

79. D. Hall, "Data Fusion", *H. Silver Associates course* (1995).

80. D. Hall and J. Llinas, "An Introduction to Multisensor Data Fusion", *Proceedings of IEEE*, Volume 85 Number 1, pp. 6-23 (1997).

81. P. Hart, "The Condensed Nearest Neighbour Rule", *IEEE Transactions on Information Theory,* Volume 14 (1968).

82. A. Heading and M. Bedworth, "Data Fusion for Object Classification", *Systems, Man and Cybernetics,* (1991).

83. A. Heading, "Decision Making and Data Fusion", *DRA Milestone report 92.85.04.29,* (1993).

84. W. Highleyman, "The Design and Analysis of Pattern Recognition Experiments", *Bell Systems Technical Journal,* pp.723-744 (1962).

85. T. Ho, J. Hull and S. Srihari, "On Multiple Classifier systems for Pattern Recognition", *Proceedings of IEEE, pp. 84-87* (1992).

86. I. Hoballah and P. Varshney, "Distributed Bayesian Signal Detection", *IEEE Transactions on Information Theory,* Volume 35, Number 5, pp. 995-1000 (1989).

87. B. Horn, "Classification of Incomplete Datasets", *DRA Technical Report DRA/CIS(SE1)/651/8/RP5/1.0* (1995).

88. R. Hummel and L. Manevitz, "Combining Bodies of Dependent Information", *Proceedings of IJCAI, pp. 1015-1017* (1987).

89. S. Hutchins, J. Morrison and R. Kelly "Decision Support for Tactical Decision Making Under Stress", *Proceedings of 2nd International Symposium on Command and Control Research and Technology, pp. 204-215* (1996).

90. F. Incropera and D. De Witt, "Fundamentals of Heat Transfer", *John Wiley* (1981).

91. H. Jeffreys, "The Theory of Probability", *Oxford University Press* (1939).

92. F. Jensen, "An Introduction to Bayesian Networks", *UCL Press* (1995).

93. W. Jones, "Air Conditioning Engineering", *Arnold* (1973).

94. R. Kain, "Mean Accuracy of Pattern Recognisers with Many Pattern Classes", *IEEE Transactions on Information Theory*, pp 424-425 (1969).

95. H. Kalayeh and D. Landgrebe, "Predicting the Required Number of Training Samples", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 5, Number 6, pp 664-667 (1983).

96. M. Kam, Q. Zhu and W. Gray, "Optimal Data Fusion of Correlated Local Decisions in Multiple Sensor Detection Systems", *IEEE Aerospace and Electronic Systems*, Volume 28, Number 3, pp. 916-919 (1992).

97. R. Kelly, S. Hutchins and J. Morrison, "Decision Processes and Team Communications with a Decision Support System", *Proceedings of 2nd International Symposium on Command and Control Research and Technology* (1996).

98. L. Klein, "Sensor and Data Fusion Concepts and Applications", *SPIE Tutorial Text T14* (1993).

99. M. Kokar and K. Kim, "Review of Multisensor Data Fusion Architectures and Techniques", *Proceedings of IEEE Symposium on Intelligent Control* pp. 261-266 (1993).

100. W. Krebs, D. Scribner, G. Miller, J. Ogawa and J. Schuler, "Beyond Third Generation: A Sensor Fusion Targeting FLIR Pod for the F/A-18", *SPIE 3376 - Sensor Fusion: Architectures, Algorithms and Applications II, pp.129-140* (1998).

101. R. Krzysztofowicz and D. Long, "Fusion of Detection Probabilities and Comparison of Multisensor Systems", *Systems, Man and Cybernetics* Volume 20, Number 3 pp. 605-677 (1990).

102. S. Kullback and R. Leibler, "On Information and Sufficiency", *Annals of Mathematical Statistics*, Volume 22, pp. 79-86 (1951).

103. S. Kullback, "Information Theory and Statistics", *John Wiley & Sons* (1959).

104. R. Lake, "Distributed Event Driven Architectures for Evolutionary Sensor Fusion", *SPIE 3376 - Sensor Fusion: Architectures, Algorithms and Applications II, pp. 81-87* (1998).

105. W. Lakin and J. Miles, "IKBS in Multisensor Data Fusion", *DRA Technical Report* (1984).

106. W. Lakin and J. Miles, "An AI Approach to Data Fusion and Situation Assessment", *in Advances in Command, Control and Communications Systems, Peregrinus* (1987).

107. P. Lee, "Bayesian Statistics: An Introduction", *Oxford University Press* (1989).

108. M. Liggins, C. Chong, I. Kadar, M. Alford, V. Vamicola and S. Thomopoulos, "Distributed Fusion Architectures and Algorithms for Target Tracking", *Proceedings of IEEE* Volume 85 Number 1, pp. 95-107, (1997).

109. W. Lincoln and J. Skrzypek, "Synergy of Clustering Multiple Back Propagation Networks", *Advances in Neural Information Processing Systems 2, pp. 650-657* (1989).

110. R. Linn, D. Hall and J. Llinas, "A Survey of Multisensor Data Fusion Systems", *SPIE 1470 – Data Structures and Target Classification, pp.13-29* (1991).

111. J. Llinas, "Formal Methods of Automated Reasoning for Situational Awareness", *SPIE 3067 – Sensor Fusion: Architectures, Algorithms and Applications, pp. 62-71* (1997).

112. J. Llinas, "A Survey of Techniques for CIS Data Fusion", *Science Applications Technical Report* pp. 77-84 (1997).

113. R. Luce and H. Raiffa, "Games and Decisions", *John Wiley and Sons* (1957)

114. R. Luo and M. Kay, "Multisensor Integration and Fusion in Intelligent Systems", *Systems, Man and Cybernetics* Volume 19, Number 5 pp. 901-931 (1989).

115. R. Luo and M. Kay, "Data Fusion and Sensor Integration", *in Data Fusion in Robotics and Machine Intelligence, Academic Press, Chapter* 2 (1992).

116. D. MacKay, "Bayesian Methods for Adaptive Models", *personal communication* (1991).

117. D. MacKay, "Bayesian Model Comparison and Backprop Nets", *Advances in Neural Information Processing Systems,* Volume 4, pp 839-846 (1991).

118. P. Mahalanobis, "On The Generalised Distance in Statistics", *Proceedings of the National Institute of Science Calcutta,* Volume 12 pp. 49-55 (1936).

119. R. Mahler, "Random Sets as a Foundation for General Data Fusion", *Paramax Systems Corporation Technical Report,* (1997).

120. J. Manyika and H. Durrant-White, "An Information Theoretic Approach to Management in Decentralised Data Fusion", *SPIE 1828 - Sensor Fusion V* (1992).

121. D. Marsay, "A Way Ahead for Strategic and Tactical Data Fusion Systems", *DRA Divisional Memorandum IS1/69* (1992).

122. D. Marsay, "Information Fusion for Intelligence", *personal communication* (1996).

123. K. Marsh and J. Richardson, "Fusion of Multisensor Data", *Journal of Robotics Research,* Volume 7, Number 6 (1988).

124. D. McMichael, "Radar Emitter Location using Bayesian Methods", *DRA Research Note CSE1/242* (1993).

125. D. McMichael, "Data Fusion for Vehicle-Borne Mine Detection", *Proceedings of IEE Eurel International Conference on The Detection of Abandoned Land Mines, pp.167-171* (1996).

126. D. McMichael, "Estimating Structure of Bayesian Networks using Mixtures", *Proceedings of FUSION'98, pp. 109-115* (1998).

127. D. Meister, K. Hipel and M. De, "Coalition Formation Metrics for Decision Making", *IEEE Systems, Man and Cybernetics* Volume 3, pp. 2017-2022 (1991).

128. P. Murphy, "Repository of Machine Learning Databases and Domain Theories", ftp://ftp.ics.uci.edu/pub/machine-learning-databases (1999).

129. P. Nahin and J. Pokoski, "NCTR Plus Sensor Fusion Equals IFFN", *Aerospace and Electronic Systems,* Volume 16, Number 3 pp. 320-337 (1990).

130. N. Nandhakumar and J. Aggarwal, "Integrated Analysis of Thermal and Visual Images for Scene Interpretation", *IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 10, Number 4, pp. 469-481* (1988).

131. N. Nandhakumar, "Robust Physics-Based Analysis of Thermal and Visual Imagery", *Journal of the Optical Society of America, Volume 11, Number 11, pp. 2981-2989* (1994).

132. J. O'Brien, "An Algorithm for the Fusion of Correlated Probabilities", *Proceedings of FUSION'98 Las Vegas, pp. 565-571* (1998).

133. J. O'Brien and M. Bedworth, "Correlated Probability Fusion for the Recognition of Ground Vehicles", *Proceedings of NATO/IRIS Conference on Data Fusion, Quebec,* (1998).

134. J. O'Brien, "Correlated Probability Fusion for Multiple Class Discrimination", *Proceedings of Data and Information Fusion symposium IDC'99, Adelaide, pp.571-578* (1999).

135. H. Pan and D. McMichael, "Fuzzy Causal Probabilistic Networks – A New Ideal and Practical Inference Engine", *Proceedings of FUSION'98, pp. 101-108* (1998).

136. A. Papoulis, "Probability, Random Variables and Stochastic Processes", *McGraw-Hill* (1991).

137. E. Parzen, "On Estimation of a Probability Density Function and Mode", *Annals of Mathematical Statistics,* Volume 33, pp. 1065-1076 (1962)

138. J. Pearl, "Probabilistic Reasoning in Intelligent Systems", *Morgan Kaufmann* (1988).

139. D. Penny, "The Automatic Management of Multi-Sensor Systems", *Proceedings of FUSION'98, pp. 748-756* (1998).

140. A. Pete, K. Pattipati and C. Rossano, "Distributed Binary Detection with Different Local Hypotheses", *IEEE Systems, Man and Cybernetics* Volume 3, pp. 2023-2028 (1991).

141. G. Peterson and H. Barney, "Control Methods used in the Study of Vowels", *Journal of the Acoustical Society of America,* Volume 24 (1952).

142. A. Pinz and R. Bartl, "Information Fusion in Image Understanding", *Proceedings of the 11th IAPR, pp.366-370* (1992).

143. J. Rajapakse and R. Acharya, "Multisensor Data Fusion with Hierarchical Neural Networks", *SUNY Buffalo Technical Paper,* (1997).

144. N. Rao, "To Fuse or Not to Fuse: Fuser Versus Best Classifier", *SPIE 3376 - Sensor Fusion: Architectures, Algorithms and Applications II, pp. 25-34* (1998).

145. N. Rao, "A Fusion Method that Performs Better Than Best Sensor", *Proceedings of FUSION'98, pp. 19-26* (1998).

146. S. Raudys and V. Pikelis, "On Dimensionality, Sample Size, Classification Error and Complexity of Classification Algorithm in Pattern Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Volume 2, Number 3, pp. 242-252 (1980).

147. J. Richardson and K. Marsh, "Fusion of Multisensor Data", *International Journal of Robotics Research,* Volume 7, Number 6, pp. 78-96 (1988).

148. D. Rumelhart, G. Hinton and R. Williams, "Learning Representations by Backpropagating Errors", *Nature,* Volume 323 (1986).

149. K. Scheerer, "Airborne Multisensor System for the Autonomous Detection of Landmines", *Proceedings of IEE Eurel International Conference on The Detection of Abandoned Land Mines, pp. 183-187* (1996).

150. A. Schreiber, B. Wielinga, and J. Breuker (editors), "KADS: A Principled Approach to Knowledge-Based System Development", *Volume 11 of Knowledge-Based Systems Book Series, Academic Press, London,* (1993).

151. E. Shahbazian, L. Gagnon, J. Duquet, M. Macieszczak and P. Valin, "Fusion of Imaging and Non-imaging Data for Surveillance Aircraft", *SPIE 3067 – Sensor Fusion: Architectures, Algorithms and Applications, pp. 179-189* (1997).

152. E. Shahbazian, J. Duquet and P. Valin, "A Blackboard Architecture for Incremental Implementation of Data Fusion Applications", *Proceedings of FUSION'98, pp. 455-460* (1998).

153. B. Silverman, "Density Estimation for Statistics and Data Analysis", *Chapman and Hall* (1986).

154. A. Steinberg, C. Bowman and F. White, "Revisions to the JDL Data Fusion Model", *NATO/IRIS Conference on Data Fusion,* (1998).

155. L. Stewart and P. McCarty, "The Use of BBNs to Fuse Continuous and Discrete Information for Target Recognition, Tracking and Situation Assessment", *AGARD-337, pp. 161-166* (1996).

156. Z. Tang, K. Pattipati and D. Kleinman, "An Algorithm for Determining the Decision Thresholds in a Distributed Detection Problem", *IEEE Systems, Man and Cybernetics* Volume 3, pp. 928-930 (1989).

157. O. Taylor and J. MacIntyre, "Adaptive Local Fusion Systems for Novelty Detection and Diagnostics in Condition Monitoring", *SPIE 3376 - Sensor Fusion: Architectures, Algorithms and Applications II, pp. 210-218* (1998).

158. D. Teneketzis and P. Varaiya, "The Decentralised Quickest Detection Problem", *Proceedings of IEEE, Automation and Control,* Volume 29, Number 7, pp. 641-644 (1984).

159. R. Tenney and N. Sandell, "Detection with Distributed Sensors", *IEEE Aerospace and Electronic Systems,* Volume 17, Number 4, pp. 501-510 (1981).

160. C. Therrien, "Decision Estimation and Classification" *John Wiley & Sons* (1989).

161. S. Thomopoulos, R. Viswanathan and D. Bougoulias, "Optimal Decision Fusion in Multiple Sensor Systems", *IEEE Aerospace and Electronic Systems,* Volume 23, Number 5, pp. 644-653 (1987).

162. S. Thomopoulos, N. Okello, I. Kadar and L. Lovas, "Design of a Multisensor Data Fusion System for Target Recognition", *SPIE 1955-3, pp. 1-13* (1990).

163. K. Tumer and J. Ghosh, "Bayes Error Rate Estimation through Classifier Combining", *University of Austin report TX78712-1084* (1996).

164. P. Verlinde, G. Maitre and E. Mayoraz, "Decision Fusion Using a Multi-Linear Classifier", *Proceedings of FUSION'98, pp. 47-53* (1998).

165. R. Viswanathan and P. Varshney, "Distributed Detection with Multiple Sensors: Fundamentals", *Proceedings of IEEE,* Volume 85, Number 1, pp. 54-63 (1997).

166. J. Von Neumann and O. Morgenstern, "Theory of Games and Economic Behaviour", *Princeton University Press* (1944).

167. W. Wang, P. Luh, D. Serfaty and D. Kleinman, "Hierarchical Team Co-ordination in Dynamic Decision Making", *IEEE Systems, Man and Cybernetics* Volume 3, pp. 2041-2047 (1991).

168. E. Waltz and D. Buede, "Data Fusion and Decision Support for Command and Control", *Systems, Man and Cybernetics* Volume 16, Number 6 pp. 865-879 (1986).

169. R. Watrous, "Current Status of Peterson-Barney Vowel Formant Data", *Journal of the Acoustical Society of America,* Volume 89, Number 5 (1991).

170. A. Webb, D. Lowe and M. Bedworth, "A Comparison of Nonlinear Optimisation Strategies for Feedforward Adaptive Layered Networks", *RSRE Memorandum 4157* (1988).

171. D. Whitaker, "Information Fusion Research at DRA", *personal communication* (1996).

172. D. Whitaker, "NATO Tactical Data Links", *personal communication* (1996).

173. G. Wilson, "Some Aspects of Data Fusion", *Proceedings of IEE International Conference on C3 Theory and Applications,* pp. 99-105 (1992).

174. L. Xu, A. Krzyzak and C. Suen, "Methods of Combining Multiple Classifiers and their Application to Handwriting Recognition", *IEEE Systems, Man and Cybernetics* Volume 22, Number 3, pp.418-435 (1992).

175. C. Yu and P. Varshney, "Decision Fusion using Channels with Communications Constraints", *SPIE 3067 – Sensor Fusion: Architectures, Algorithms and Applications, pp. 94-105* (1997).

176. B. Zhu, N. Ansari and E. Hou, "An Adaptive Fusion Model for Distributed Detection Systems", *SPIE 1828 – Sensor Fusion V, pp. 332-341* (1992).

Appendix A: Derivations of Key Results

A.1 *Heat Flux Model*

The quantity W_{abs} is given by the equation:

$$W_{abs} = \alpha_s W_s \cos\theta$$

where

$$W_s = \frac{A}{e^{B/\sin\alpha}}$$

W_s is the radiation flux falling on a surface that is perpendicular to the sun, α_s is the solar absorptivity of the surface and θ is the angle between the normal to the facet and the direction to the sun, A and B are time-of-year parameters [93] and a is the angular altitude of the Sun.

Assuming that the rest of environment has the same emmisivity and the environment has the same temperature as the air, the net heat flux lost by radiation given by the Stefan-Boltzmann law is:

$$W_{rad} = \varepsilon_0 \sigma (T_s^4 - T_{air}^4)$$

Where ε_0 is the emmisivity of the material and σ is the Stefan-Boltzmann constant, T_s is the surface temperature and T_{air} the ambient air temperature. The heat flux lost by convection is approximated by the equation:

$$W_{cv} = h(T_s - T_{air})$$

Where h is the heat transfer coefficient. The quantity h is dependent on the speed, temperature and thermo-physical properties of air.

A.2 Approximation to Binomial Confidence Limits

We use the approximation to the inverse of the incomplete beta function based on that of Hastings [2]. We give only the lower confidence limit, L_l, the upper confidence limit follows from symmetry: $L_u(x) = 1 - L_l(x)$. In this approximation N denotes the total number of patterns in the test set and E_T the proportion of test samples mis-classified. The value of y_p is chosen depending on the confidence level - the value of $y_p = 1.96$ used here corresponds to the 95% confidence limit. The approximation holds when at least one test pattern is either correctly or incorrectly classified

$$L_l \approx \frac{NE_T}{NE_T + (N - NE_t + 1)e^{2w}}$$

$$w = w_1 - w_2 w_3$$

$$w_1 = \frac{y_p \sqrt{h + \lambda}}{h}$$

$$w_2 = \frac{1}{h_2} - \frac{1}{h_1}$$

$$w_3 = \lambda + \frac{5}{6} - \frac{2}{3h}$$

$$h = \frac{2}{h_3}$$

$$h_1 = 2NE_T - 1$$

$$h_2 = 2N - 2NE_T + 1$$

$$h_3 = \frac{1}{h_1} + \frac{1}{h_2}$$

$$\lambda = \frac{y_p^2}{6} - \frac{1}{2}$$

$$y_p = 1.96$$

If no test patterns were mis-classified then:

$$L_l = 0$$

and if all test pattern were mis-classified then:

$$L_l = \sqrt[N]{0.025}$$

A.3 *Sequential Hypothesis Testing*

Assume that we require the fused error rate to lie in the interval $[p_1, p_2]$. Furthermore that we are prepared to accept an equal probability of ε that the estimated error rate could be above or below this interval. We use an interval test of the form:

error rate too high	if	$\hat{e} \geq a + b$
error rate too low	if	$\hat{e} \leq a - b$
more observations required	if	$a - b < \hat{e} < a + b$

where $\hat{e}$ is our estimate of the error rate a is a test quantity for the estimate and $2b$ is the size of the acceptable range for the estimate.

Define a likelihood ratio test based on N samples with E errors (using the data D):

$$\lambda = \frac{P(D \mid p = p_1)}{P(D \mid p = p_2)} = \frac{p_1{}^E (1 - p_1)^{N-E}}{p_2{}^E (1 - p_2)^{N-E}} = \frac{p_1{}^E (1 - p_1)^N (1 - p_2)^E}{p_2{}^E (1 - p_2)^N (1 - p_1)^E} = \left(\frac{1 - p_1}{1 - p_2}\right)^N \left(\frac{p_1/(1 - p_1)}{p_2/(1 - p_2)}\right)^E$$

Assume that an interval test with limits k_1 and k_2 exists for this quantity:

$k_2 \leq \lambda \leq k_1$

this leads to

$$k_2 \left(\frac{1 - p_2}{1 - p_1}\right)^N \leq \left(\frac{p_1/(1 - p_1)}{p_2/(1 - p_2)}\right)^E \leq k_1 \left(\frac{1 - p_2}{1 - p_1}\right)^N$$

taking logarithms leads to:

$$\log k_2 + N \log\left(\frac{1 - p_2}{1 - p_1}\right) \leq E \log\left(\frac{p_1/(1 - p_1)}{p_2/(1 - p_2)}\right) \leq \log k_1 + N \log\left(\frac{1 - p_2}{1 - p_1}\right)$$

dividing throughout by $N \log\left(\frac{p_1/(1 - p_1)}{p_2/(1 - p_2)}\right)$ gives

$$\frac{\log\left(\dfrac{1-p_2}{1-p_1}\right)}{\log\left(\dfrac{p_1/(1-p_1)}{p_2/(1-p_2)}\right)}+\frac{\log k_2}{N\log\left(\dfrac{p_1/(1-p_1)}{p_2/(1-p_2)}\right)}\le\frac{E}{N}\le\frac{\log\left(\dfrac{1-p_2}{1-p_1}\right)}{\log\left(\dfrac{p_1/(1-p_1)}{p_2/(1-p_2)}\right)}+\frac{\log k_1}{N\log\left(\dfrac{p_1/(1-p_1)}{p_2/(1-p_2)}\right)}$$

which is in the correct form for an interval test on the estimated error rate since $\hat{e}=E/N$. As described above we use ideal value a and range $2b$:

$$\begin{array}{ccc}
\text{error rate too high} & \text{if} & \hat{e}\ge a+b \\
\text{error rate too low} & \text{if} & \hat{e}\le a-b \\
\text{more observations required} & \text{if} & a-b<\hat{e}<a+b
\end{array}$$

where

$$a=\frac{\log\left(\dfrac{1-p_2}{1-p_1}\right)}{\log\left(\dfrac{p_1/_{1-p}}{p_2/_{1-p_2}}\right)}$$

and it can be shown (for example in [46]) that

$$b\approx\frac{\log\left(\dfrac{1-\varepsilon}{\varepsilon}\right)}{\log\left(\dfrac{p_1/_{1-p}}{p_2/_{1-p_2}}\right)}$$

gives a likelihood test which is optimal in the sense that it tests the hypothesis that the deferral threshold should be changed using a minimum number of observations.

A.4 *Independent Probability Fusion*

First we write the required joint conditional probability as a function of the data likelihoods using Bayes' rule:

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

we then use our assumption of conditional independence to decompose the right hand side into contributions from the separate sensors:

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

since the fused conditional probability should be written in terms of separate conditional probabilities (PRI-PRO probability-in / probability-out fusion using the expanded Dasarathy data fusion model shown in Table 3-4) we apply Bayes' rule once more to obtain:

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

and finally, collecting terms we obtain:

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

the latter part of which is independent of the class c and so can be treated as an unknown constant which may be recovered by normalising over all classes. This result is easily extendible to the case of multiple sensors. For N sensors we have the general result:

$$P(c \mid \{x_1 \cdots x_N\}) = \frac{\prod_{i=1}^{N} P(c \mid x_i)}{P(c)^{N-1}} \times \frac{\prod_{i=1}^{N} P(x_i)}{P(\{x_1 \cdots x_N\})}$$

and again, the second term can be recovered by normalising over classes.

A.5 *Logistic Function and Gaussians*

For a pair of Gaussians of unit variance and means at $\pm\frac{1}{2}$ we have:

$$P_{\frac{1}{2}} = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-\frac{1}{2})^2}{2}}$$

and

$$P_{\frac{1}{2}} = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x+\frac{1}{2})^2}{2}}$$

Now the probability of the class (arbitrarily with mean at $\frac{1}{2}$) can be found thus:

$$\begin{aligned}
P(C_{\frac{1}{2}}) &= \frac{P_{\frac{1}{2}}}{P_{\frac{1}{2}} + P_{-\frac{1}{2}}} \\
&= \frac{e^{-\frac{(x-\frac{1}{2})^2}{2}}}{e^{-\frac{(x-\frac{1}{2})^2}{2}} + e^{-\frac{(x+\frac{1}{2})^2}{2}}} \\
&= \frac{1}{1+e^{-\frac{(x+\frac{1}{2})^2}{2}+\frac{(x-\frac{1}{2})^2}{2}}} \\
&= \frac{1}{1+e^{-\frac{x}{2}-\frac{x}{2}}} \\
&= \frac{1}{1+e^{-x}}
\end{aligned}$$

which is the logistic function used in the multi-layer perceptron.

A.6 *Gradient Calculation for Discriminative Gaussian MLP*

The relative-entropy scoring criterion, is particularly suitable for this network architecture as the computation of the derivative of the criterion function with respect to each of the parameters is considerably simplified.

$$J = -\sum_{p}\sum_{k} T_{pk} \log O_{pk}$$

where k is used to index the network output corresponding to each class for pattern p. We first require the derivative of the criterion function with respect to the I_k. Since this depends on all the units in the final layer we have, by use of the chain rule:

$$\frac{\partial J}{\partial I_k} = \sum_{h} \frac{\partial J}{\partial O_h} \frac{\partial O_h}{\partial I_k}$$

Now

$$\frac{\partial J}{\partial O_h} = -\sum_{g} T_g \log O_g (h \in \{g\}) = -\frac{T_h}{I_h}$$

and

$$\frac{\partial O_h}{\partial I_k} = O_h(\delta_{hk} - O_k)$$

where δ_{hk}, the Kronecker-δ, takes a value of unity when h=k and zero otherwise. Now

$$\begin{aligned}
\sum_{h} \frac{\partial J}{\partial O_h} \frac{\partial O_h}{\partial I_k} &= \sum_{h} -\frac{T_k}{O_k} O_k(\delta_{hk} - O_h) \\
&= -\sum_{h} T_h \delta_{hk} + O_k \sum_{h} T_h \\
&= -T_k + O_k \\
&= O_k - T_k
\end{aligned}$$

Finally we have the required derivative for the reference point weights

$$\begin{aligned}
\frac{\partial J}{\partial w_{jk}} &= \frac{\partial J}{\partial I_k} \frac{\partial I_k}{\partial w_{jk}} \\
&= 2(O_k - T_k)(O_j - w_{jk})
\end{aligned}$$

Now the first layer of linear transformation weights can be computed using a similar approach to the standard MLP formulation:

$$\frac{\partial J}{\partial w_{ij}} = \frac{\partial J}{\partial I_j}\frac{\partial I_j}{\partial w_{ij}}$$

where

$$\frac{\partial I_j}{\partial w_{ij}} = O_i$$

and

$$\frac{\partial J}{\partial I_j} = \frac{\partial J}{\partial O_j}\frac{\partial O_j}{\partial I_j}$$

Now

$$\frac{\partial O_j}{\partial I_j} = 1$$

and

$$\frac{\partial J}{\partial O_j} = \sum_k \frac{\partial J}{\partial I_k}\frac{\partial I_k}{\partial O_j}$$

which computes to

$$\frac{\partial J}{\partial O_j} = \sum_k 2(O_k - T_k)(w_{jk} - O_j)$$

which is noted to be $-\sum_k {\partial J}/{\partial w_{jk}}$ which we computed in an earlier equation. Therefore

$$\frac{\partial J}{\partial w_{ij}} = O_i - \sum_k \frac{\partial J}{\partial w_{jk}}$$

A.7 *Leave-one-out Sample Mean and Variance*

The sample mean is defined as:

$$\hat{\mu} = \tfrac{1}{N}\sum_{i=1}^{N} x_i$$

and with the *j*th element omitted this becomes:

$$\hat{\mu}_j = \tfrac{1}{N-1}\left(\sum_{i=1}^{N} x_i - x_j\right)$$

which leads directly to:

$$\hat{\mu}_j = \tfrac{1}{N-1}\left(N\hat{\mu} - x_j\right)$$

Now the sample variance is defined as:

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

and with the *j*th value omitted as

$$\hat{\sigma}_j^2 = \frac{1}{N-1}\left(\sum_{i=1}^{N} x_i^2 - x_j^2\right) - \hat{\mu}_j^2$$

which expands to:

$$\begin{aligned}\hat{\sigma}_j^2 &= \frac{1}{N-1}\sum_{i=1}^{N} x_i^2 - \frac{x_j^2}{N-1} - \left(\frac{N\hat{\mu}}{N-1} - \frac{x_j}{N-1}\right)^2 \\ &= \frac{1}{N-1}\sum_{i=1}^{N} x_i^2 - \frac{x_j^2}{N-1} - \frac{1}{(N-1)^2}\left(N^2\hat{\mu}^2 - 2N\hat{\mu}x_j + x_j^2\right)\end{aligned}$$

which simplifies to

$$\begin{aligned}
\hat{\sigma}_j^2 &= \frac{1}{N-1}\sum_{i=1}^{N} x_i^2 - \frac{N}{(N-1)^2}\left(N\hat{\mu}^2 - 2\hat{\mu}x_j + \frac{N-1}{N}x_j^2 + \frac{x_j^2}{N}\right) \\
&= \frac{1}{N-1}\sum_{i=1}^{N} x_i^2 - \frac{N}{(N-1)^2}\left(N\hat{\mu}^2 - 2\hat{\mu}x_j + x_j^2\right) \\
&= \frac{1}{N-1}\sum_{i=1}^{N} x_i^2 - \frac{N}{(N-1)^2}\hat{\mu}^2 - \frac{N}{(N-1)^2}(x_j - \hat{\mu})^2
\end{aligned}$$

which can be rewritten in terms of $\hat{\sigma}^2$ as

$$\hat{\sigma}_j^2 = \frac{N}{N-1}\left(\hat{\sigma}^2 - \frac{(x_j - \hat{\mu})^2}{N-1}\right)$$

Now, assuming independence, the leave-one-out data evidence can be written down thus

$$\begin{aligned}
P(D) &\approx \prod_{i=1}^{N} P(d_i \mid \{D - d_i\}) \\
&= \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\left(\frac{N}{N-1}\left(\hat{\sigma}^2 - \frac{(d_i - \hat{\mu})^2}{N-1}\right)\right)}}\, e^{-\frac{\left(d_i - \frac{1}{N-1}(N\hat{\mu} - d_i)^2\right)}{2\left(\frac{N}{N-1}(\hat{\sigma}^2 - \frac{(d_i - \hat{\mu})^2}{N-1})\right)}}
\end{aligned}$$

This does not have a simple form.

Appendix B: Descriptions of Experimental Databases

Many of the experimental datasets used during the course of the research were obtained from the University of California, Irvine repository of machine learning databases and domain theories. The data is obtainable via FTP from:

ftp://ftp.ics.uci.edu/pub/machine-learning-databases

B.1 *Peterson and Barney Vowel Formant Data*

The Peterson and Barney vowel data was originally created by Gordon Peterson and Harold Barney [141] in 1952 to study the characteristics of vowels as spoken by 76 American men (33), women (28) and children (15). Four attributes were measured corresponding to the frequencies in *Hz* of the fundamental voicing and the first, second and third formants (resonances of the vocal tract) during a portion of the steady state vowel sound. The data has since been appropriated by the pattern recognition and machine learning community and used for vowel classification using the second and third attributes only (first and second formants) since the data is then easily visualised and most of the discriminative information resides in these features. Numerous results are available based on the data, particularly in the neural networks literature. An initial data analysis of this vowel formant data reveals the characteristics shown in Table B-1.

Attribute	Variance ratio
Formant 0	0.01
Formant 1	2.76
Formant 2	5.03
Formant 3	0.63

Table B-1: The characteristics of the Peterson and Barney vowel formant data.

The source of this data has a somewhat convoluted history. Peterson and Barney appear to have mislaid their original computer readable data and the only versions available were three data files keyed in from a printout of the magnetic tape before it's destruction. Several discrepancies existed in these data files which were tracked down by Raymond Watrous as being transcription errors. Watrous corrected the data using collateral information and made the, now definitive, data available at the University of Pennsylvania, USA [169]. This supersedes the data sourced from Lippmann who obtained the data from Bill Huang who is reputed to have digitised it manually from a diagram of the original data. Due to the questionable origin of some of these datasets the results presented in other papers may not be directly comparable to those presented here. However, the definitive database used here is accessible to readers should they wish to perform comparative experiments.

The F1/F2 data consists of 1520 records associated with the 76 speakers uttering each of the ten vowel sounds twice in succession. Speakers 1→33 are adult males, 34→61 are adult females and speakers 61→76 are children. Of the child speakers 62, 63, 65, 66, 67, 68, 73 and 76 are female.

The data has ten classes of vowels with two attributes per pattern. The distribution of patterns from each class is approximately a bivariate Gaussian. Classifiers making this assumption tend to do well on this data.

Figure B-1 shows the overall scatter of the data and the one standard deviation ellipses associated with a bivariate Gaussian distribution estimated for each class.

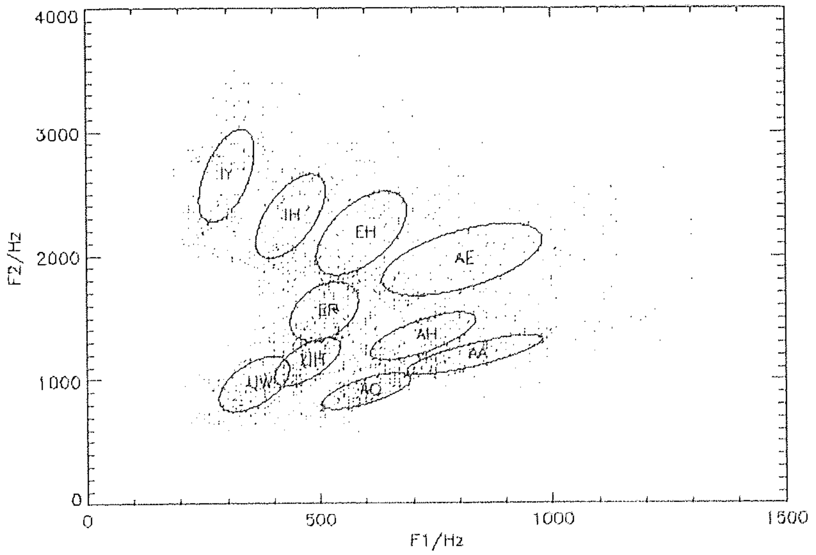

Figure B-1: One standard deviation ellipses for the Peterson and Barney vowel data.

B.2 *Iris Data*

Originally created by R. A. Fisher [67] using measurements made by Dr. E. Anderson and used by numerous researchers including [53] and [63], the Iris database was obtained from the UCI Repository of Machine Learning Databases and Domain Theories. The Iris classification problem is a small task, which is relatively easy to obtain high performance rates with.

The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. Types *Iris Setosa* and *Iris Versicolour* were found in the same natural colonies whereas *Iris Virginica* was measured at a different location. One class (*Iris Setosa*) is linearly separable from the other classes, which are not linearly separable from each other.

Class index	Class name	1-from-N code
1	*Iris Setosa*	0 0 1
2	*Iris Versicolour*	0 1 0
3	*Iris Virginica*	1 0 0

Table B-2: Class attributions for the Iris database.

There are four attributes per pattern. The first attribute is the sepal length in cm, the second is the sepal width, also in cm, the third and fourth the petal length and width in cm. All attributes are measured to an accuracy of 1 mm.

Attribute	Minimum	Maximum	Mean	Standard deviation	Class correlation
Sepal length	4.3	7.9	5.84	0.83	0.78
Sepal width	2.0	4.4	3.05	0.43	0.42
Petal length	1.0	6.9	3.76	1.76	0.95
Petal width	0.1	2.5	1.20	0.76	0.96

Table B-3: Attribute characteristics for the Iris database.

Table B-3 details a set of statistics regarding the attribute values of the iris data. Figure B-2 shows the data from the three classes projected onto the two most discriminative attribute axes.

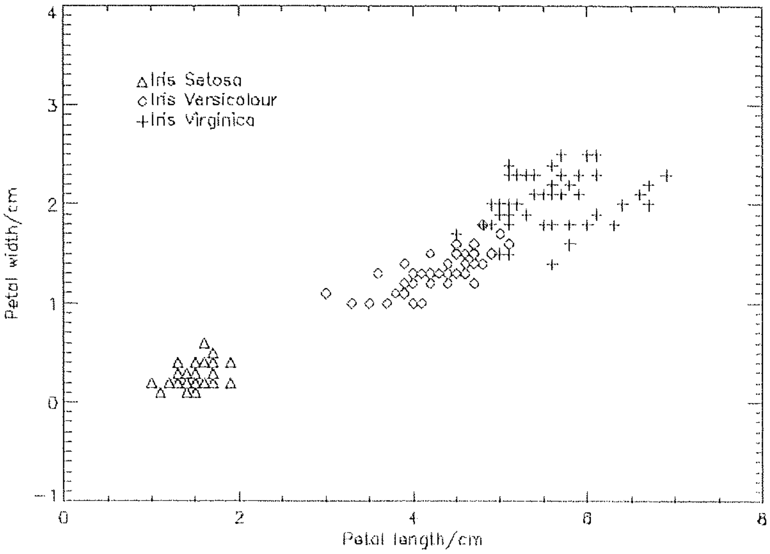

Figure B-2: The iris data projected onto the two most discriminative axes.

B.3 *Glass Data*

The glass identification database was created by B. German of Central Research Establishment, Home Office Forensic Science Service, Aldermaston. The classification of glass fragments into types having being float-processed or not could be used to assist criminological investigations. Six types of glass were analysed: float or non-float building windows, vehicle glass (all float-processed), container, tableware and headlamp glass. For the experiments reported here only the building and vehicle glass was used of which float processed building windows accounted for 70 samples, non-float processed building windows for 76 samples and float processed vehicle windows for 17 samples. Both of the float-processed types were combined into a single class. There were therefore 163 patterns, which were divided into training and test files by assigning all odd numbered patterns to the training database (82 patterns in all) and all even numbered patterns to the test database (81 patterns in all).

Attribute	Minimum	Maximum	Mean	Standard deviation	Variance ratio
Refractive index	1.51	1.53	1.52	0.00	0.00
Sodium	10.73	14.86	13.20	0.35	0.02
Magnesium	0.00	4.48	3.29	0.79	0.10
Aluminium	0.29	2.12	1.28	0.10	0.15
Silicon	69.81	74.45	72.59	0.41	0.00
Potassium	0.00	1.10	0.48	0.05	0.04
Calcium	7.08	16.19	8.92	1.88	0.01
Barium	0.00	3.15	0.03	0.06	0.01
Iron	0.00	0.37	0.07	0.01	0.01

Table B-4: Attribute characteristics for the Glass database.

Previous use has included nearest neighbour and rule based systems that obtained as low as 15% error rate.

B.4 *Breast Cancer Diagnosis*

Dr. William Wolberg of the University of Wisconsin Hospitals, Madison collected several sets of data concerning the analyses of cell samples. The measured attributes are potentially useful in diagnosing malignant breast cancers. The data set contains 699 instances from 8 clinical case groups analysed between January 1989 and November 1991. The data is arranged chronologically. Previous usage has concentrated on the first group of 367 instances.

For each case there are 9 attributes which are scored manually on a scale of 1 to 10. For the purposes of these experiments these scores were scaled by 0.1. Associated with each case is a diagnosis of the sample as being representative of either a benign or malignant tumour. Of the 699 samples in the database 458 are classed as benign and the remaining 241 malignant.

Attribute	**Minimum**	**Maximum**	**Mean**	**Standard deviation**	**Variance ratio**
Clump thickness	0.1	1.0	0.44	0.08	1.14
Uniformity of cell size	0.1	1.0	0.31	0.09	1.84
Uniformity of cell shape	0.1	1.0	0.32	0.09	1.91
Marginal adhesion	0.1	1.0	0.28	0.08	0.85
Single epithelial cell size	0.1	1.0	0.32	0.05	0.81
Base nuclei	0.1	1.0	0.36	0.13	1.82
Bland chromatin	0.1	1.0	0.34	0.06	1.31
Normal nucleoli	0.1	1.0	0.29	0.09	0.93
Mitoses	0.1	1.0	0.16	0.03	0.19

Table B-5: Attribute characteristics for the Wisconsin breast-cancer database.

Attributes 1 → 6 have missing values which were not recorded are unavailable. For the purposes of these experiments these were replaced by the intermediate scaled value of 0.5.