Sādhanā Vol. 29, Part 2, April 2004, pp. 145–174. © Printed in India

Bayesian and Dempster–Shafer fusion

SUBHASH CHALLA[1] and DON KOKS[2]

[1]Information and Communications Group, Faculty of Engineering, The University of Technology, Sydney, Australia
[2]Defence, Science and Technology Organisation (DSTO), Electronic Warfare and Radar Division, Adelaide, Australia
e-mail: schalla@eng.uts.edu.au; Don.Koks@dsto.defence.gov.au

Abstract. The Kalman Filter is traditionally viewed as a prediction–correction filtering algorithm. In this work we show that it can be viewed as a Bayesian fusion algorithm and derive it using Bayesian arguments. We begin with an outline of Bayes theory, using it to discuss well-known quantities such as priors, likelihood and posteriors, and we provide the basic Bayesian fusion equation. We derive the Kalman Filter from this equation using a novel method to evaluate the Chapman–Kolmogorov prediction integral. We then use the theory to fuse data from multiple sensors. Vying with this approach is the Dempster–Shafer theory, which deals with measures of "belief", and is based on the nonclassical idea of "mass" as opposed to probability. Although these two measures look very similar, there are some differences. We point them out through outlining the ideas of the Dempster–Shafer theory and presenting the basic Dempster–Shafer fusion equation. Finally we compare the two methods, and discuss the relative merits and demerits using an illustrative example.

Keywords. Kalman Filter; Bayesian Fusion; Dempster–Shafer Fusion; Chapman–Kolmogorov prediction integral.

1. Introduction

Data Fusion is a relatively new field with a number of incomplete definitions. Many of these definitions are incomplete owing to its wide applicability to a number of disparate fields. We use data fusion with the narrow definition of combining the data produced by one or more sensors in a way that gives a best estimate of the quantity we are measuring.

Current data fusion ideas are dominated by two approaches or paradigms. In this report we discuss these two philosophies that go to make up a large amount of analysis in the subject as it currently stands. We also give a brief and select review of the literature.

The oldest paradigm, and the one with the strongest foundation, is Bayes theory. This theory is based on the classical ideas of probability, and has at its disposal all of the usual machinery of statistics. We show that the Kalman Filter can be viewed as a Bayesian data fusion algorithm where fusion is performed over time. One of the crucial steps in such a formulation is the solution of the Chapman–Kolmogorov prediction integral. We present a

novel method to evaluate this prediction integral and incorporate it into the Bayesian fusion equations. We then put it to use to derive the Kalman Filter in a straightforward and novel way. We next apply the theory in an example of fusing data from multiple sensors. Again, the analysis is very straightforward and shows the power of the Bayesian approach.

Vying with the Bayes theory is the Dempster–Shafer theory, which is a recent attempt to allow more interpretation of what uncertainty is all about. The Dempster–Shafer theory deals with measures of "belief" as opposed to probability. In a binary problem, Dempster–Shafer theory introduces to the "zero" and "one" states that standard probability takes as exhausting all possible outcomes, other alternatives such as "unknown". We outline the ideas of the Dempster–Shafer theory, with an example given of fusion using the cornerstone of the theory known as Dempster's rule. Dempster–Shafer theory is based on the nonclassical idea of "mass" as opposed to the well-understood probabilities of Bayes theory; and although the two measures look very similar, there are some differences that we point out. We then apply the Dempster–Shafer theory to a fusion example, and point out the new ideas of "support" and "plausibility" that this theory introduces.

Although some of the theory of just how to do this is quite old and well established, in practice, many applications require a lot of processing power and speed: performance that only now is becoming available in this current age of faster computers with streamlined numerical algorithms. So fusion has effectively become a relatively new field.

A further fusion paradigm – not discussed here – is fuzzy logic, which in spite of all of the early interest shown in it, is not heavily represented in the current literature.

2. A review of data fusion literature

In this section we describe some of the ways in which data fusion is currently being applied in several fields. Since fusion ideas are currently heavily dependent on the precise application for their implementation, the subject has yet to settle into an equilibrium of accepted terminology and standard techniques. Unfortunately, the many disparate fields in which fusion is used ensure that such standardisation might not be easily achieved in the near future.

2.1 *Trends in data fusion*

To present an idea of the diversity of recent applications, we focus on the recent International Conferences on Information Fusion, by way of a choice of papers that aims to reflect the diversity of the fields discussed at these conferences. Our attention is mostly confined to the conferences, Fusion '98 and '99. The field has been developing rapidly, so that older papers are not considered purely for reasons of space. On the other hand, the latest conference, Fusion 2000, contains many papers with less descriptive names than those of previous years, that impart little information on what they are about. Whether this indicates a trend toward the abstract in the field remains to be seen.

Most papers are concerned with military target tracking and recognition. In 1998 there was a large number devoted to the theory of information fusion: its algorithms and mathematical methods. Other papers were biased toward neural networks and fuzzy logic. Less widely represented were the fields of finance and medicine, air surveillance and image processing.

The cross section changed somewhat in 1999. Although target-tracking papers were as plentiful as ever, medical applications were on the increase. Biological and linguistic models were growing, and papers concerned with hardware for fusion were appearing. Also appearing were applications of fusion to more of the everyday type of scenario: examples are traffic

analysis, earthquake prediction and machining methods. Fuzzy logic was a commonly used approach, followed by discussions of Bayesian principles. Dempster–Shafer theory seems not to have been favoured very much at all.

2.2 *Basic data fusion philosophy*

In 1986 the Joint Directors of Laboratories Data Fusion Working Group was created, which subsequently developed the data fusion process model (Hall & Garga 1999). This is a plan of the proposed layout of a generic data fusion system, and is designed to establish a common language and model within which data fusion techniques can be implemented.

The model defines relationships between the sources of data and the types of processing that might be carried out to extract the maximum possible information from it. In between the source data and the human, who makes decisions based on the fused output, there are various levels of processing.

Source preprocessing: This creates preliminary information from the data that serves to interface it better with other levels of processing.

Object refinement: The first main level of processing refines the identification of individual objects.

Situation refinement: Once individual objects are identified, their relationships to each other need to be ascertained.

Threat refinement: The third level of processing tries to infer details about the future of the system.

Process refinement: The fourth level is not so much concerned with the data, but rather with what the other levels are doing, and whether it is or can be optimised.

Data management: The housekeeping involved with data storage is a basic but crucial task, especially if we are dealing with large amounts of data or complex calculations.

Hall & Garga (1999) discuss this model and present a critique of current problems in data fusion. Their points in summary are as below.

- Many fused poor quality sensors do not make up for a few good ones.
- Errors in initial processing are very hard to correct down the line.
- It is often detrimental to use well-worn presumptions of the system: for example that its noise is Gaussian.
- Much more data must be used for training a learning algorithm than we might at first suppose. They quote (Hush & Horne 1993) as saying that if there are m features and n classes to be identified, then the number of training cases required will be at least of the order of between 10 and 30 times mn.
- Hall and Garga (1999) also believe that quantifying the value of a data fusion system is inherently difficult, and that no magic recipe exists.
- Fusion of incoming data is very much an ongoing process, not a static one.

Zou *et al* (2000) have used Dempster–Shafer theory in the study of reducing the range errors that mobile robots produce when they use ultrasound to investigate a specular environment. Such an environment is characterised by having many shiny surfaces, and as a result, there is a chance that a signal sent out – if it encounters several of these surfaces – will bounce

repeatedly; so that if and when it does return to the robot, it will be interpreted as having come from very far away. The robot thus builds a very distorted picture of its environment.

What a *Bayesian* robot does is build a grid of its surroundings, and assign to each point a value of "occupied" (by e.g. a wall) or "empty". These are mutually exclusive, so $p(\textit{occupied}) + p(\textit{empty}) = 1$. The Dempster–Shafer approach introduces a third alternative: "unknown", along with the idea of a "mass", or measure of confidence in each of the alternatives. Dempster–Shafer theory then provides a rule for calculating the confidence measures of these three states of knowledge, based on data from two categories: new evidence and old evidence.

The essence of Zou's work lies in building good estimates of just what the sensor measures should be. That is the main task, since the authors show that the results of applying Dempster–Shafer theory depend heavily on the choice of parameters that determine these measures. Thus for various choices of parameters, the plan built by the robot varies from quite complete but with additional points scattered both inside and outside of it (i.e. probabilities of detection and false alarm both high), to fairly incomplete, but without the extraneous extra points (corresponding to probabilities of detection and false alarm both low).

The final conclusion reached by Zou *et al* (2000) is that the parameter choice for quantifying the sensor measure is crucial enough to warrant more work being done on defining just what these parameters should be in a new environment. The Dempster–Shafer theory they used is described more fully in § 4.

Myler (2000) considers an interesting example of data fusion in which Dempster–Shafer theory fails to give an acceptable solution to a data fusion problem where it is used to fuse two irreconcilable data sets. If two sensors each have strongly differing opinions over the identity of an emitter, but agree very very weakly on a third alternative, then Dempster–Shafer theory will be weighted almost 100% in favour of that third alternative. This is an odd state of affairs, but one to which there appears to be no easy solution.

Myler accepts this and instead offers a measure of a new term he calls "disfusion": the degree to which there is agreement among sensors as to an alternative identity of the target that has not been chosen as the most likely one. If D is the number of dissenting sensors that disagree with the winning sensor, but agree with each other, and N is the total number of sensors fused, then the disfusion is defined as

$$\text{disfusion} \equiv D/(N-1). \tag{1}$$

Thus if all but one sensor weakly identify the target as some X, while the winning sensor identifies it as $Y \neq X$, then $D = N - 1$ and there is 100% disfusion. Myler contrasts this with "confusion", in which none of the sensors agree with any other. Clearly though, there are other definitions of such a concept that might be more useful in characterising how many sensors disagree, and whether they are split into more than one camp.

However, Myler's paper gives no quantitative use for disfusion, apart from advocating its use as a parameter that should prompt a set of sensors to take more measurements if the disfusion is excessive. This is certainly a good use for it, since we need to be aware that the high mass that Dempster–Shafer will attribute to an otherwise weak choice of target in the above example does not mean that Dempster–Shafer is succeeding in fusing the data correctly; and there needs to be an indicator built in to the fusion system to warn us of that.

Kokar *et al* (2000) bemoan the fact that at their time of writing (early 2000), data fusion had not lived up to its promises. They suggest that it needs to be approached somewhat differently to the current way, and have described various models that might provide a way forward. Their main suggestion is that a data fusion system should not be thought of so much

as a separate system that humans use to fuse data, but that rather we should be designing a complete human-automaton system with data-fusion capability in mind.

This reference concentrates on describing various models for ways to accomplish this. The authors first describe a generic information-centred model that revolves around the flow of information in a system. Its highest levels are dealing with sensor data, down to the preliminary results of signal processing, through to extraction of relevant details from these, prediction of their states, and using these to assess a situation and plan a response. These levels are as described in the Joint Directors of Laboratories model at the beginning of this section.

Kokar's paper next describes a function-centred model. This is a cycle made up of four processes that happen in temporal sequence: collecting information, collating and sorting it to isolate the relevant parts, making a decision, and finally carrying out that decision. The results of this then influence the environment, which in turn produces more data for the cycle to begin anew. This model leads on quite naturally to an object-oriented approach, since it implies a need for objects to carry out these activities. The strength of this object-oriented approach is that it has the potential to make the code-writing implementation much easier.

Kokar *et al* (2000) emphasise the view that in many data fusion systems humans must interact with computers, so that the ways in which the various processes are realised need to take human psychology into account.

The three main methods of data fusion are compared in Cremer *et al* (1998). In this paper, the authors use Dempster–Shafer, Bayes and fuzzy logic to compare different approaches to land mine detection. Their aim is to provide a figure of merit for each square in a gridded map of the mined area, where this number is an indicator of the chance that a mine will be found within that grid square.

Each technique has its own requirements and difficulty of interpretation. For example, Dempster–Shafer and Bayes require a meaning to be given to a detection involving background noise. We can use a mass assigned to the background as either a rejection of the background, or as an uncertainty. The fuzzy approach has its difficulty of interpretation when we come to "defuzzify" its results: its fuzzy probabilities must be turned into crisp ones to provide a bottom line figure of merit.

Cremer *et al* (1998) do not have real mine data, so rely instead on a synthetic data set. They find that Dempster–Shafer and Bayes approaches outperform the fuzzy approach – except for low detection rates, where fuzzy probabilities have the edge. Comparing Dempster–Shafer and Bayes, they find that there is little to decide between the two, although Dempster–Shafer has a slight advantage over Bayes.

2.3 *Target location and tracking*

Sensor fusion currently finds its greatest number of applications in the location and tracking of targets, and in that sense it is probably still seen very much as a military technique that is gradually finding wider application.

Triesch (2000) describes a system for tracking the face of a person who enters a room and manoeuvres within it, or even walks past another person in that room. The method does not appear to use any standard theory such as Bayes or Dempster–Shafer. Triesch builds a sequence of images of the entire room, analysing each through various cues such as intensity profile, colour and motion continuity. To each metric are assigned a "reliability" and a "quality", both between zero and one, and set to arbitrary values to begin with. The data fusion algorithm is designed so that their values evolve from image to image in such a way that poorer metrics are given smaller values of reliability, and so are weighted less. Two-dimensional functions of the environment are then produced, one for each cue, where the function's value increases

in regions where the face is predicted to be. A sum of these functions, weighted with the reliabilities, then produces a sort of probability distribution for the position of the face.

Each cue has a "prototype vector": a representation of the face in the parameter space of that cue. This prototype is allowed to evolve in such a way as to minimise discordance in the cues' outputs. The rate of evolution of the prototype is determined by comparing the latest data with the current value of the prototype vector, as well as incorporating a preset time constant to add some memory ability to the system's evolution.

The results quoted by Triesch are spread across different regimes and cannot be described as conclusive. Although higher success rates are achieved when implementing their algorithm, the highest success occurs when the quality of each cue is constrained to be constant. Allowing this quality itself to evolve might be expected to give better results, but in fact it does not. Triesch posits that the reason for this anomalous result is that the dynamics of the situation, based as they are on a sequence of images, are not as continuous as they were assumed to be when the rules governing the system's evolution were originally constructed. He suggests that more work is needed to investigate this problem.

Schwartz (2000) has applied a maximum a posteriori (MAP) approach to the search for formations of targets in a region, using a model of a battlefield populated by a formation of vehicles. A snapshot taken of this battlefield yields a map which is then divided into a grid, populated by spots that might indicate a vehicle – or might just be noise. He starts with a set of templates that describe what a typical formation might look like (based on previously collected data about such formations). Each of these templates is then fitted digitally over the grid and moved around cell by cell, while a count is kept of the number of spots in each cell. By comparing the location of each spot in the area delineated by the template to the centroid of the spots in that template, it becomes possible to establish whether a particularly high density of spots might be a formation conforming to the template, or might instead just be a random set of elements in the environment together with noise, that has no concerted motion.

The MAP approach to searching for formations uses the Bayesian expression:

$$p(\text{formation} \mid \text{data}) = \frac{p(\text{data} \mid \text{formation})\ p(\text{formation})}{p(\text{data})}. \tag{2}$$

As mentioned in § 3, the MAP estimate of the degree to which a data set is thought to be a formation is the value of a parameter characterising the formation, that maximises $p(\text{formation} \mid \text{data})$. As is typical of Bayesian problems, the value of the prior $p(\text{formation})$ at best can only be taken to be some constant. Schwartz discusses statistical models for the placing of spots in the grid. His method does not involve any sort of evolution of parameters; rather it is simply a comparison of spot number with template shapes. Good quality results are obtained with – and require – many frames; but this is not overly surprising, since averaging over many frames will reduce the amount of noise on the grid.

Fuzzy logic is another method that has been used to fuse data. This revolves around the idea of a "membership function". Membership in a "crisp" set (i.e. the usual type of set encountered in mathematics) is of course a binary yes/no value; and this notion of a one or zero membership value generalises in fuzzy set theory to a number that lies between one and zero, that defines the set by how well the element is deemed to lie within it.

These ideas are applied by Simard *et al* (2000) of Lockheed Martin Canada and the Canadian Defence Research Establishment, along with a combination of other fusion techniques, to ship movements in order to build a picture of what vessels are moving in Canadian waters. The system they described as of 1999 is termed the Adaptive Fuzzy Logic Correlator (AFLC).

The AFLC system receives messages in different protocols relating to various contacts made, by both ground and airborne radars. It then runs a Kalman Filter to build a set of tracks of the various ships. In order to associate further contacts with known tracks, it needs to prepare values of the membership functions for electromagnetic and position parameters. For example, given a new contact, it needs to decide whether this might belong to an already-existing track, by looking at the distance between the new contact and the track. Of course, a distance of zero strongly implies that the contact belongs to the track, so we can see that the contact can be an element of a fuzzy set associated with the track, where the membership function should peak for a distance of zero.

Given surveillance data and having drawn various tracks from it, the system must then consult a database of known ships to produce a candidate that could conceivably have produced the track of interest. Electromagnetic data, such as pulse repetition frequency, can also be given a membership within different sets of emitters. The ideas of fuzzy sets then dictate what credence we give to the information supplied by various radar or surveillance systems. Comparing this information for many sensors reduces to comparing the membership function values for the various system parameters.

Once we have a candidate ship for any given track, we need to fuse incoming data by combining it with the data that already forms part of the track history. For example, the AFLC takes the last ten contacts made and forms the track history from these. Finally, the output of the AFLC is a map of the region of interest filled with tracks of ships, together with their identifications if these can be found in the ship database.

As the authors point out, the use of fuzzy logic is not without its problems when comparing different parameters. The membership function quantifying how close a new contact is to a track is not related to the membership function for say pulse repetition frequency, and yet these two functions may well need to be compared at some point. This comparison of apples with oranges is a difficulty, and highlights the care that we need to exercise when defining just what the various membership functions should be.

Kewley (1992) compares the Dempster–Shafer and fuzzy approaches to fusion, so as to decide which of a given set of emitters has produced certain identity attribute data. He finds that fuzzy logic gives similar results to Dempster–Shafer, but for less numerical work and complexity. Kewley also notes that while the Dempster–Shafer approach is not easily able to assimilate additional emitters after its first calculations have been done, fuzzy logic certainly can.

It's not apparent that there is any one approach we should take to fuse track data from multiple sensors. Watson *et al* (2000) discuss one solution they have developed: the optimal asynchronous track fusion algorithm (OATFA). They use this to study the tracking of a target that follows three constant velocity legs with two changes of direction in between, leading to its travelling in the opposite direction to which it started.

The authors base their technique on the Interacting multiple model algorithm (IMM). The IMM is described as being particularly useful for tracking targets through arbitrary manoeuvres, but traditionally it uses a Kalman Filter to do its processing. Watson *et al* (2000) suggest replacing the IMM's Kalman Filter with their OATFA algorithm (which contains several Kalman Filters of its own), since doing so produces better results than for the straight Kalman Filter case. They note, however, that this increase in quality tends to be confined to the (less interesting) regions of constant velocity.

The OATFA algorithm treats each sensor separately: passing the output from each to a dedicated Kalman Filter, that delivers its updated estimate to be combined with those of all of the other sensor/Kalman Filter pairs, as well as feeding back to each of the Kalman Filters.

Certainly the OATFA model departs from the idea that the best way to fuse data is to deliver it all to a central fusion engine: instead, it works upon each sensor separately. Typical results of the IMM-OATFA algorithm tend to show position estimation errors that are about half those that the conventional IMM produces, but space and time constraints make it impossible for the authors to compare their results with any other techniques.

Hatch *et al* (1999) describe a network of underwater sensors used for tracking. The overall architecture is that of a command centre taking in information at radio frequency, from a sublevel of "gateway" nodes. These in turn each take their data acoustically from the next sublevel of "master" nodes. The master nodes are connected (presumably by wires) to sensors sitting on the ocean floor.

The communication between command centre and sensors is very much a two-way affair. The sensors process and fuse some of their data locally, passing the results up the chain to the command centre. But because the sensors run on limited battery power, the command centre must be very careful with allocating them tasks. Thus, it sets the status of each ("process data", "relay it only up the chain", "sleep" or "die") depending on how much power each has. The command centre also raises or lowers detection thresholds in order to maintain a constant false alarm rate over the whole field; so that if a target is known to be in one region, then thresholds can be lowered for sensors in that region (to maximise detection probabilities), while being raised in other areas to keep the false alarm rate constant.

The processing for the sensors is done using both Kalman Filtering and a fuzzy logic-based α–β filter (with comparable results at less computational cost for the α–β filter). Fuzzy logic is also used to adapt the amount of process noise used by the Kalman Filter to account for target manoeuvres.

The paper gives a broad overview of the processing hierarchy without mentioning mathematical details. Rather, it tends to concentrate more on the architecture, such as the necessity for a two-way data flow as mentioned above.

2.4 *Satellite positioning*

Heifetz *et al* (1999) describe a typical problem involved with satellite-attitude measurement. They are dealing with the NASA Gravity Probe B, that was designed to be put into Earth orbit for a year or more in a precision measurement of some relativistic effects that make themselves felt by changes in the satellite's attitude.

Their work is based around a Kalman Filter, but the nonlinearities involved mean that at the very least, an extended Kalman Filter is required. Unfortunately, the linearisation used in the extended Kalman Filter introduces a well-understood bias into two of the variables being measured. The authors are able to circumvent this difficulty by using a new algorithm Haupt *et al* (1996), that breaks the filtering into two steps: a Kalman Filter and a Gauss-Newton algorithm.

The first step, the Kalman Filter, is applied by writing trigonometric entities such as $\sin(\omega t + \delta)$ in terms of their separate $\sin\omega t$, $\cos\omega t$, $\sin\delta$, $\cos\delta$ constituents. Combinations of some of these constituents then form new variables, so that the nonlinear measurement equation becomes linear in those variables. Thus a linear Kalman Filter can be applied, and the state estimate it produces is then taken as a synthetic new measurement, to be fed to the Gauss-Newton iterator.

Although the paper was written before NASA's satellite was due for launch, the authors have plotted potentially achievable accuracies which show that in principle, the expected relative errors should be very small.

2.5 *Air surveillance*

Rodríguez *et al* (1998) discuss a proposal to fuse data in an air surveillance system. They describe a system whose centre is the Automatic Dependent Surveillance system, in which participating aircraft send their navigation details to Air Traffic Control for assistance in marshalling.

Since the proposed scheme uses a central control centre for fusion, it provides a good example of an attempt to fuse data in the way that preserves each sensor's individuality for as long as possible, which thus should lead to the best results. Air Traffic Control accepts each Automatic Dependent Surveillance system message and tries to associate it with an existing track. It does not do this on a message-by-message basis, but rather listens for some preset period, accumulating the incoming data that arrives during this time. Once it has a collection of data sets, it updates its information iteratively, by comparing these data sets with already-established tracks.

2.6 *Image processing and medical applications*

By applying information theory, Cooper & Miller (1998) address the problem of quantifying the efficacy of automatic object recognition. They begin with a library of templates that can be referenced to identify objects, with departures of an object's pose from a close match in this library being quantified by a transformation of that template. They require a metric specifying how well a given object corresponds to some template, regardless of that object's orientation in space.

This is done by means of "mutual information". They begin with the usual measures of entropy $S(x)$, $S(y)$ and joint entropy $S(x, y)$ in terms of expected values:

$$\begin{aligned} S(x) &= -E_x[\ln p(x)], \\ S(x, y) &= -E_x E_y[\ln p(x, y)]. \end{aligned} \tag{3}$$

Using these, the mutual information of x and y is defined as

$$I(x, y) = S(x) + S(y) - S(x, y). \tag{4}$$

If two random variables are independent, then their joint entropy is just the sum of their individual entropies, so that their mutual information is zero as expected. On the other hand, if they are highly matched, their mutual information is also high. The core of Cooper and Miller's paper is their calculation of the mutual information for three scenarios: two different sorts of visual mapping (orthographic and perspective projections), and the fusion of these. That is, they calculate the mutual information for three pairs of variables: one element of each pair being the selected template, and the other element being the orthographic projection, the perspective projection, and the fusion of the two projections.

For very low signal to noise ratios (SNRs), all three mutual informations are zero, meaning there is very little success in the object-template fits. All three informations climb as the SNR increases, tending toward a common upper limit of about 6·5 for the highest SNR values. The middle of the SNR range is where we see the interesting results. As hoped for, here the fused scenario gives the highest mutual information. Typical values in the middle of the SNR range (SNR $= 10$) are orthographic projection: 3·0, perspective projection: 3·8 and fused combination: 4·6.

Similar work has been done by Viola & Gilles (1996), who fuse image data by maximising the mutual information. In contrast to Cooper and Miller's work, they match different images

of the same scene, where one might be rotated, out of focus or even chopped up into several dozen smaller squares. They achieve good results, and report that the method of mutual information is more robust than competing techniques such as cross-correlation.

Fuzzy logic has been applied to image processing in the work of Debon *et al* (1999), who use it in locating the sometimes vague elliptical cross-section of the human aorta in ultrasound images. The situation they describe is that of an ultrasound source lowered down a patient's oesophagus, producing very noisy data that shows slices of the chest cavity perpendicular to the spine. The noise is due partly to the instrument, and partly to natural chest movements of the patient during the process. Within these ultrasound slices they hope to find an ellipse that marks the aorta in cross-section.

Rather than using the common approach of collecting and fusing data from many sensors, Debon *et al* (1999) use perhaps just one sensor that collects data, which is then fused with prior information about the scene being analysed. In this case the authors are using textbook information about the usual position of the aorta (since this is not likely to vary from patient to patient). This is an entirely reasonable thing to do, given that the same principle of accumulated knowledge is perhaps the main contributor for the well known fact that humans tend to be better, albeit slower, than computers at doing certain complex tasks.

The fuzzy model that the authors use allocates four fuzzy sets to the ultrasound image. These are sets of numbers allocated to each pixel, quantifying for example brightness and its gradient across neighbouring pixels. They then use these numbers in the so-called Hough transform, a method that can detect parametrised curves within a set of points.

The result of this fusion of library images of the aorta with actual data is that an ellipse is able to be fitted to an otherwise vague outline of the aorta in the ultrasound images. Inspection of the ultrasound images shows that this technique works very well.

A simpler approach to medical data fusion is taken by Zachary & Iyengar (1999), who describe a method for fusing data to reconstruct biological surfaces. They are dealing with three sets of data: namely, contour slices that result from imaging in three orthogonal planes. This is relatively new work, in the sense that medical imaging is usually done in a single plane.

Their approach to the problem does not actually analyse how well they are fusing the three sets of data. Their major effort lies in defining a good coordinate system within which to work, as well as giving care to ensuring that the sets of data are all scaled to match each other correctly. Although the resulting surfaces that are drawn through the points fit well, this has only been done in Zachary & Iyengar (1999) for a spherical geometry. However, the authors do describe having applied their method to ellipsoids and to some medical data.

2.7 *Intelligent internet agents*

Intelligent internet agents are also discussed in the literature, although somewhat infrequently. Strömberg (2000) discusses the makeup of a sensor management system in terms of two architectures: agent modelling and multi-level sensor management. His approach maintains that agents can be useful because, as an extension to the object oriented approach that is so necessary to modern programming, they allow a high degree of robustness and re-usability in a system. He points out that in a typical tracking problem, different modes of operation are necessary: fast revisits to establish a candidate track, with variable revisit times once the track is established. Agents are seen to be well suited to this work, since they can be left alone to make their own decisions about just when to make an observation.

2.8 *Business and finance*

An application of fusion to the theory of finance is described by Blasch (1998). He discusses the interaction between *monetary policy*, being concerned with money demand and income, and *fiscal policy*, the interaction between interest rates and income. The multiple sensors here are the various sources of information that the government uses to determine such indicators as changes in interest rates. However, these sources have differing update frequencies, from hourly to weekly or longer. The perceived need to update markets continually, means that such inputs are required to be combined in a way that acknowledges the different confidences in each.

Blasch (1998) quantifies the policies using a model with added Gaussian noise to allow the dynamics to be approximated linearly, with most but not all of his noise being white. Not surprisingly, he uses a Kalman Filter for the task, together with wavelet transforms introduced because of the different resolution levels being considered (since wavelets were designed to analyse models with different levels of resolution). An appreciation of Blasch's analysis requires a good understanding of fiscal theory, but his overall conclusion is that the Kalman Filter has served the model very well.

3. Bayesian data fusion

We will begin our presentation of Bayesian data fusion by first reviewing Bayes' theorem. To simplify the expressions that follow, we shorten the notation of $p(A)$ for the probability of some event A occurring to just (A): the "p" is so ubiquitous that we will leave it out entirely. Also, the probability that two events A, B occur is written as (A, B), and this can be related to the probability $(A|B)$ of A occurring given that B has already occurred:

$$(A, B) = (A|B)\,(B). \tag{5}$$

Now since $(A, B) = (B, A)$, we have immediately that

$$(A|B) = [(B|A)\,(A)]/(B). \tag{6}$$

If there are several events A_i that are distinguished from B in some way, then the denominator (B) acts merely as a normalisation, so that

$$(A|B) = [(B|A)\,(A)]/\left[\textstyle\sum_i (B|A_i)\,(A_i)\right]. \tag{7}$$

Equations (6) or (7) are known as Bayes' rule, and are very fruitful in developing the ideas of data fusion. As we said, the denominator of (7) can be seen as a simple normalisation; alternatively, the fact that the (B) of (6) can be expanded into the denominator of (7) is an example of the Chapman–Kolmogorov identity that follows from standard statistical theory:

$$(A|B) = \sum_i (A|X_i, B)(X_i|B), \tag{8}$$

which we use repeatedly in the calculations of this paper.

Bayes' rule divides statisticians over the idea of how best to estimate an unknown parameter from a set of data. For example, we might wish to identify an aircraft based on a set of

measurements of useful parameters, so that from this data set we must extract the "best" value of some quantity x. Two important estimates of this best value of x are:

Maximum likelihood estimate: The value of x that maximises (data$|x$)

Maximum a posteriori estimate: the value of x that maximises ($x|$data)

There can be a difference between these two estimates, but they can always be related using Bayes' rule.

A standard difficulty encountered when applying Bayes' theorem is in supplying values for the so-called *prior* probability (A) in equation (7). As an example, suppose several sensors have supplied data from which we must identify an aircraft, which might be a Bombardier Learjet, a Dassault Falcon, and so on. From (7), the chance that the aircraft is a Learjet on the available evidence is

$$(\text{Learjet}|\text{data}) = \frac{(\text{data}|\text{Learjet})\,(\text{Learjet})}{(\text{data}|\text{Learjet})\,(\text{Learjet}) + (\text{data}|\text{Falcon})\,(\text{Falcon}) + \ldots}. \tag{9}$$

It may well be easy to calculate (data|Learjet), but now we are confronted with the question: what is (Learjet), (Falcon) etc.? These are prior probabilities: the chance that the aircraft in question could really be for example a Learjet, irrespective of what data has been taken. Perhaps Learjets are not known to fly in the particular area in which we are collecting data, in which case (Learjet) is presumably very small.

We might have no way of supplying these priors initially, so that in the absence of any information, the approach that is most often taken is to set them all to be equal. As it happens, when Bayes' rule is part of an iterative scheme these priors will change unequally on each iteration, acquiring more meaningful values in the process.

3.1 *Single sensor tracking*

As a first example of data fusion, we apply Bayes' rule to *tracking*. Single sensor tracking, also known as filtering, involves a combining of successive measurements of the state of a system, and as such it can be thought of as a fusing of data from a single sensor over time as opposed to *sensor set*, which we leave for the next section. Suppose then that a sensor is tracking a target, and makes observations of the target at various intervals. Define the following terms:

$$\begin{aligned} x_k &= \text{target state at "time" } k \text{ (iteration number } k); \\ y_k &= \text{observation made of target at time } k; \\ Y_k &= \text{set of all observations made of target up to time } k \\ &= \{y_1, y_2, \ldots, y_k\}. \end{aligned} \tag{10}$$

The fundamental problem to be solved is to find the new estimate of the target state $(x_k|Y_k)$ given the old estimate $(x_{k-1}|Y_{k-1})$. That is, we require the probability that the target is something specific given the latest measurement and all previous measurements, given that we know the corresponding probability one time step back. To apply Bayes' rule for the set Y_k, we separate the latest measurement y_k from the rest of the set Y_{k-1} – since Y_{k-1} has already been used in the previous iteration – to write $(x_k|Y_k)$ as $(x_k|y_k, Y_{k-1})$. We shall swap the two terms x_k, y_k using a minor generalisation of Bayes' rule. This generalisation is easily shown

by equating the probabilities for the three events (A, B, C) and (B, A, C) expressed using conditionals as in (5):

$$(A, B, C) = (A|B, C)\,(B|C)\,(C); \tag{11}$$
$$(B, A, C) = (B|A, C)\,(A|C)\,(C); \tag{12}$$

so that Bayes' rule becomes

$$(A|B, C) = \frac{(B|A, C)\,(A|C)}{(B|C)}. \tag{13}$$

Before proceeding, we note that since only the latest time k and the next latest $k-1$ appear in the following expressions, we can simplify them by replacing k with 1 and $k-1$ with 0. So we write

$$\overbrace{(x_1|Y_1)}^{\text{“conditional density”}} = (x_1|y_1, Y_0) = \frac{\overbrace{(y_1|x_1, Y_0)}^{\text{“likelihood”}}\ \overbrace{(x_1|Y_0)}^{\text{“predicted density”}}}{\underbrace{(y_1|Y_0)}_{\text{normalisation}}}. \tag{14}$$

There are three terms in this equation, and we consider each in turn.

The *likelihood* deals with the probability of a measurement y_1. We will assume the noise is "white", meaning uncorrelated in time,[1] so that the latest measurement does not depend on previous measurements. In that case the likelihood (and hence normalisation) can be simplified:

$$\text{likelihood} = (y_1|x_1, Y_0) = (y_1|x_1). \tag{15}$$

The *predicted density* predicts x_1 based on *old* data. It can be expanded using the Chapman–Kolmogorov identity:

$$\text{predicted density} = (x_1|Y_0) = \int \mathrm{d}x_0\ \underbrace{(x_1|x_0, Y_0)}_{\text{“transition density”}}\ \overbrace{(x_0|Y_0)}^{\text{result from previous iteration (“prior”)}}. \tag{16}$$

We will also assume the system obeys a Markov evolution, implying that its current state directly depends only on its previous state, with any dependence on old measurements encapsulated in that previous state. Thus the transition density in (16) can be simplified to $(x_1|x_0)$, changing that equation to

$$\text{predicted density} = (x_1|Y_0) = \int \mathrm{d}x_0\,(x_1|x_0)\,(x_0|Y_0). \tag{17}$$

Lastly, the *normalisation* can be expanded by way of Chapman–Kolmogorov, using the now-simplified likelihood and the predicted density:

$$\text{normalisation} = (y_1|Y_0) = \int \mathrm{d}x_1\,(y_1|x_1, Y_0)\,(x_1|Y_0) = \int \mathrm{d}x_1\,(y_1|x_1)\,(x_1|Y_0). \tag{18}$$

Finally then, (14) relates $(x_1|Y_1)$ to $(x_0|Y_0)$ via (15)–(18), and our problem is solved.

[1] Such noise is called white because a Fourier expansion must yield equal amounts of all frequencies.

An example – Deriving the Kalman Filter: As noted above, the Kalman Filter is an example of combining data over time as opposed to sensor number. Bayes' rule gives a very accessible derivation of it based on the preceding equations. Our analysis actually requires two matrix theorems given in appendix A. These theorems are reasonable in that they express Gaussian behaviour that's familiar in the one dimensional case. Refer to appendix A to define the notation $N(x;\,\mu,\,P)$ that we use.

In particular, (A.5) gives a direct method for calculating the predicted probability density in (17), which then allows us to use the Bayesian framework (Ho 1964) to derive the Kalman Filter equation. A derivation of the Kalman Filter based on Bayesian belief networks was proposed recently by Krieg (2002). However, in both these papers the authors do not solve for the predicted density (17) directly. They implicitly use a "sum of two Gaussian random variables is a Gaussian random variable" argument to solve for the predicted density. While alternative methods for obtaining this density by using characteristic functions exist in the literature, we consider a direct solution of the Chapman–Kolmogorov equation as a basis for the predicted density function. This approach is more general and is the basis of many advanced filters, such as particle filters. In a linear Gaussian case, we will show that the solution of the Chapman–Kolmogorov equation reduces to the Kalman predictor equation. To the best of our knowledge, this is an original derivation of the prediction integral, (26).

First, assume that the target is unique, and that the sensor is always able to detect it. The problem to be solved is: given a set Y_k of measurements up until the current time k, estimate the current state x_k; this estimate is called $\hat{x}_{k|k}$ in the literature, to distinguish it from $\hat{x}_{k|k-1}$, the estimate of x_k given measurements up until time $k-1$. Further, as above we will simplify the notation by replacing $k-1$ and k with 0 and 1 respectively. So begin with the expected value of x_1:

$$\hat{x}_{1|1} = \int \mathrm{d}x_1\; x_1(x_1|Y_1). \tag{19}$$

From (14) and (15) write the conditional density $(x_1|Y_1)$ as

$$(x_1|Y_1) = \frac{\overbrace{(y_1|x_1)}^{\text{likelihood}}\;\overbrace{(x_1|Y_0)}^{\text{predicted density}}}{\underbrace{(y_1|Y_0)}_{\text{normalisation}}}. \tag{20}$$

We need the following quantities.

Likelihood $(y_1|x_1)$*:* This is derived from the measurement dynamics, assumed linear:

$$y_1 = Hx_1 + w_1, \tag{21}$$

where w_1 is a noise term, assumed Gaussian with zero mean and covariance R_1. Given x_1, the probability of obtaining a measurement y_1 must be equal to the probability of obtaining the noise w_1:

$$(y_1|x_1) = (w_1) = (y_1 - Hx_1) = N(y_1 - Hx_1;\,0,\,R_1) = N(y_1;\,Hx_1,\,R_1). \tag{22}$$

Predicted density $(x_1|Y_0)$*:* Using (17), we need the transition density $(x_1|x_0)$ and the prior $(x_0|Y_0)$. The transition density results from the system dynamics (assumed linear):

$$x_1 = Fx_0 + v_1 + \text{perhaps some constant term}, \tag{23}$$

where v_1 is a noise term that reflects uncertainty in the dynamical model, again assumed Gaussian with zero mean and covariance Q_1. Then just as for the likelihood, we can write

$$(x_1|x_0) = (v_1) = (x_1 - Fx_0) = N(x_1 - Fx_0;\, 0,\, Q_1) = N(x_1;\, Fx_0,\, Q_1). \quad (24)$$

The prior is also assumed to be Gaussian:

$$(x_0|Y_0) = N(x_0;\, \hat{x}_{0|0},\, P_{0|0}). \quad (25)$$

Thus from (17) the predicted density is

$$\begin{aligned}(x_1|Y_0) &= \int \mathrm{d}x_0\ N(x_1;\, Fx_0,\, Q_1)\ N(x_0;\, \hat{x}_{0|0},\, P_{0|0}) \\ &\overset{(A.5)}{=} N(x_1;\, \hat{x}_{1|0},\, P_{1|0}),\end{aligned} \quad (26)$$

where

$$\begin{aligned}\hat{x}_{1|0} &\equiv F\hat{x}_{0|0}, \\ P_{1|0} &\equiv FP_{0|0}F^T + Q_1.\end{aligned} \quad (27)$$

Normalisation $(y_1|Y_0)$*:* This is an integral over quantities that we have already dealt with:

$$\begin{aligned}(y_1|Y_0) &= \int \mathrm{d}x_1\ \underbrace{(y_1|x_1)}_{(3.18)}\ \underbrace{(x_1|Y_0)}_{(3.22)} \\ &= \int \mathrm{d}x_1\ N(y_1;\, Hx_1,\, R_1)\ N(x_1;\, \hat{x}_{1|0},\, P_{1|0}) \\ &\overset{(A.5)}{=} N(y_1;\, H\hat{x}_{1|0},\, S_1),\end{aligned} \quad (28)$$

where

$$S_1 \equiv HP_{1|0}H^T + R_1. \quad (29)$$

Putting it all together, the conditional density can now be constructed through equations (20, 22, 26, 28):

$$\begin{aligned}(x_1|Y_1) &= \frac{N(y_1;\, Hx_1,\, R_1)\ N(x_1;\, \hat{x}_{1|0},\, P_{1|0})}{N(y_1;\, H\hat{x}_{1|0},\, S_1)} \\ &\overset{(A.3)}{=} N(x_1;\, X_1,\, P_{1|1}),\end{aligned} \quad (30)$$

where

$$\begin{aligned}K &\equiv P_{1|0}H^T\left(HP_{1|0}H^T + R_1\right)^{-1}, \quad \text{(used in next lines)}, \\ X_1 &\equiv \hat{x}_{1|0} + K(y_1 - H\hat{x}_{1|0}), \\ P_{1|1} &\equiv (1 - KH)\,P_{1|0}.\end{aligned} \quad (31)$$

Finally, we must calculate the integral in (19) to find the estimate of the current state given the very latest measurement:

$$\hat{x}_{1|1} = \int dx_1 \; x_1 N(x_1;\, X_1,\, P_{1|1}) = X_1, \tag{32}$$

a result that follows trivially, since it is just the calculation of the mean of the normal distribution, and that is plainly X_1.

This then is the Kalman Filter. Starting with $\hat{x}_{0|0}$, $P_{0|0}$ (which must be estimated at the beginning of the iterations), and Q_1, R_1 (really Q_k, R_k for all k), we can then calculate $\hat{x}_{1|1}$ by applying the following equations in order, which have been singled out in the best order of evaluation from (27, 31, 32):

$$\begin{aligned}
P_{1|0} &= F P_{0|0} F^T + Q_1 \\
K &= P_{1|0} H^T \left(H P_{1|0} H^T + R_1 \right)^{-1} \\
P_{1|1} &= (1 - K H)\, P_{1|0} \\
\hat{x}_{1|0} &= F \hat{x}_{0|0} \\
\hat{x}_{1|1} &= \hat{x}_{1|0} + K(y_1 - H \hat{x}_{1|0})
\end{aligned} \tag{33}$$

The procedure is iterative, so that the latest estimates $\hat{x}_{1|1}$, $\hat{P}_{1|1}$ become the old estimates $\hat{x}_{0|0}$, $\hat{P}_{0|0}$ in the next iteration, which always incorporates the latest data y_1. This is a good example of applying the Bayesian approach to a tracking problem, where only one sensor is involved.

3.2 *Fusing data from several sensors*

Figure 1 depicts a sampling of ways to fuse data from several sensors.

Centralising the fusion combines all of the raw data from the sensors in one main processor. In principle this is the best way to fuse data in the sense that nothing has been lost in preprocessing; but in practice centralised fusion leads to a huge amount of data traversing the network, which is not necessarily practical or desirable. Preprocessing the data at each sensor reduces the amount of data flow needed, while in practice the best setup might well be a hybrid of these two types.

Bayes' rule serves to give a compact calculation for the fusion of data from several sensors. Extend the notation from the previous section, with time as a subscript, by adding a superscript to denote sensor number:

$$\begin{aligned}
\text{Single sensor output at indicated time step} &= y^{\text{sensor number}}_{\text{time step}}, \\
\text{all data up to and including time step} &= Y^{\text{sensor number}}_{\text{time step}}.
\end{aligned} \tag{34}$$

Fusing two sensors: The following example of fusion with some preprocessing shows the important points in the general process. Suppose that two sensors are observing an aircraft, whose signature ensures it is either one of the jet-powered Bombardier Learjet and Dassault Falcon, or perhaps the propeller-driven Cessna Caravan. We will derive the technique here for the fusing of the sensors' preprocessed data.

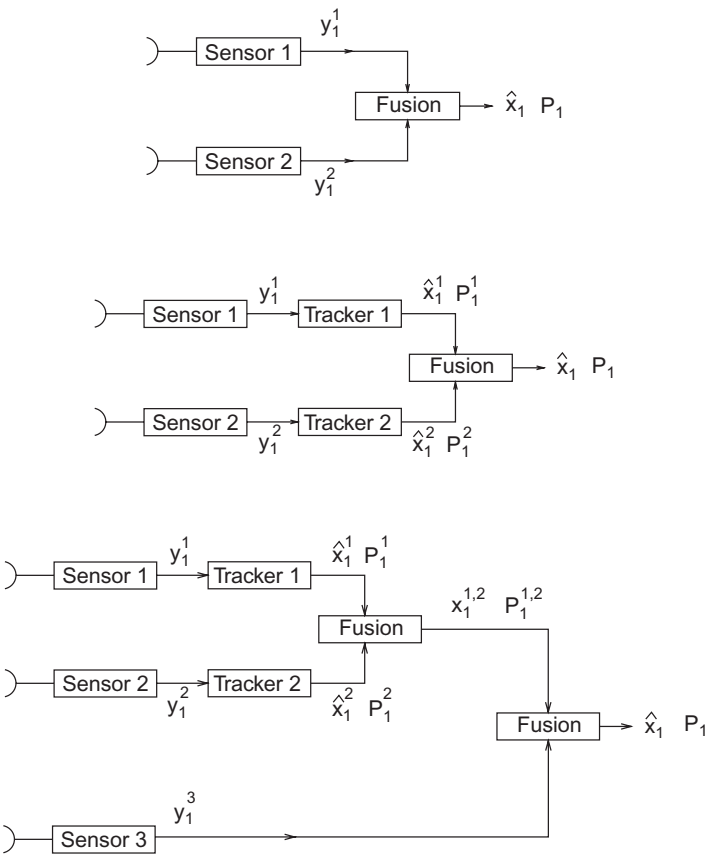

Figure 1. Different types of data fusion: centralised (top), centralised with preprocessing done at each sensor (middle), and a hybrid of the two (bottom).

Sensor 1's latest data set is denoted Y_1^1, formed by the addition of its current measurement y_1^1 to its old data set Y_0^1. Similarly, sensor 2 adds *its* latest measurement y_1^2 to its old data set Y_0^2. The relevant measurements are in table 1. Of course these are not in any sense raw data. Each sensor has made an observation, and then preprocessed it to estimate what type the aircraft might be, through the use of tracking involving that observation and those preceding it (as described in the previous section).

As can be seen from the old data, Y_0^1 Y_0^2, both sensors are leaning towards identifying the aircraft as a Learjet. Their latest data, y_1^1 y_1^2, makes them even more sure of this. The fusion node has allocated probabilities for the fused sensor pair as given in the table, with e.g. 0·5 for the Learjet. These fused probabilities are what we wish to calculate for the latest data; the 0·5, 0·4, 0·1 values listed in the table might be prior estimates of what the aircraft could reasonably be (if this is our first iteration), or they might be based on a previous iteration using old data. So for example if the plane is known to be flying at high speed, then it probably is not the Caravan, in which case this aircraft should be allocated a smaller prior probability than the other two.

Now how does the fusion node combine this information? With the aircraft labelled x, the fusion node wishes to know the probability of x being one of the three aircraft types, given the

Table 1. All data from sensors 1 and 2 in § 3·2.

Sensor 1	Sensor 2
Old data	
$(x = \textit{Learjet} \mid Y_0^1) = 0{\cdot}4$	$(x = \textit{Learjet} \mid Y_0^2) = 0{\cdot}6$
$(x = \textit{Falcon} \mid Y_0^1) = 0{\cdot}4$	$(x = \textit{Falcon} \mid Y_0^2) = 0{\cdot}3$
$(x = \textit{Caravan} \mid Y_0^1) = 0{\cdot}2$	$(x = \textit{Caravan} \mid Y_0^2) = 0{\cdot}1$
New data	
$(x = \textit{Learjet} \mid Y_1^1) = 0{\cdot}70$	$(x = \textit{Learjet} \mid Y_1^2) = 0{\cdot}80$
$(x = \textit{Falcon} \mid Y_1^1) = 0{\cdot}29$	$(x = \textit{Falcon} \mid Y_1^2) = 0{\cdot}15$
$(x = \textit{Caravan} \mid Y_1^1) = 0{\cdot}01$	$(x = \textit{Caravan} \mid Y_1^2) = 0{\cdot}05$
Fusion node has:	
$(x = \textit{Learjet} \mid Y_0^1\, Y_0^2) = 0{\cdot}5$	
$(x = \textit{Falcon} \mid Y_0^1\, Y_0^2) = 0{\cdot}4$	
$(x = \textit{Caravan} \mid Y_0^1\, Y_0^2) = 0{\cdot}1$	

latest set of data: $(x|Y_1^1\, Y_1^2)$. This can be expressed in terms of its constituents using Bayes' rule:

$$\begin{aligned}(x \mid Y_1^1\, Y_1^2) &= (x \mid y_1^1\, y_1^2\, Y_0^1\, Y_0^2) \\ &= \frac{(y_1^1\, y_1^2 \mid x,\ Y_0^1\, Y_0^2)\ (x \mid Y_0^1\, Y_0^2)}{(y_1^1\, y_1^2 \mid Y_0^1\, Y_0^2)}.\end{aligned} \tag{35}$$

The sensor measurements are assumed independent, so that

$$(y_1^1\, y_1^2 \mid x,\ Y_0^1\, Y_0^2) = (y_1^1 \mid x,\ Y_0^1)\ (y_1^2 \mid x,\ Y_0^2)\,. \tag{36}$$

In that case, (35) becomes

$$(x \mid Y_1^1\, Y_1^2) = \frac{(y_1^1 \mid x,\ Y_0^1)\ (y_1^2 \mid x,\ Y_0^2)\ (x \mid Y_0^1\, Y_0^2)}{(y_1^1\, y_1^2 \mid Y_0^1\, Y_0^2)}. \tag{37}$$

If we now use Bayes' rule to again swap the data y and aircraft state x in the first two terms of the numerator of (37), we obtain the final recipe for how to fuse the data:

$$\begin{aligned}(x \mid Y_1^1\, Y_1^2) &= \frac{(x \mid Y_1^1)\ (y_1^1 \mid Y_0^1)}{(x \mid Y_0^1)} \cdot \frac{(x \mid Y_1^2)\ (y_1^2 \mid Y_0^2)}{(x \mid Y_0^2)} \cdot \frac{(x \mid Y_0^1\, Y_0^2)}{(y_1^1\, y_1^2 \mid Y_0^1\, Y_0^2)} \\ &= \frac{(x \mid Y_1^1)\ (x \mid Y_1^2)\ (x \mid Y_0^1\, Y_0^2)}{(x \mid Y_0^1)\ (x \mid Y_0^2)} \times \text{normalisation}.\end{aligned} \tag{38}$$

The necessary quantities are listed in table 1, so that (38) gives

$$\begin{aligned}
\left(x = \text{Learjet} \mid Y_1^1\, Y_1^2\right) &\propto \frac{0{\cdot}70 \times 0{\cdot}80 \times 0{\cdot}5}{0{\cdot}4 \times 0{\cdot}6},\\
\left(x = \text{Falcon} \mid Y_1^1\, Y_1^2\right) &\propto \frac{0{\cdot}29 \times 0{\cdot}15 \times 0{\cdot}4}{0{\cdot}4 \times 0{\cdot}3},\\
\left(x = \text{Caravan} \mid Y_1^1\, Y_1^2\right) &\propto \frac{0{\cdot}01 \times 0{\cdot}05 \times 0{\cdot}1}{0{\cdot}2 \times 0{\cdot}1}.
\end{aligned} \tag{39}$$

These are easily normalised, becoming finally

$$\begin{aligned}
\left(x = \text{Learjet} \mid Y_1^1\, Y_1^2\right) &\simeq 88{\cdot}8\%\\
\left(x = \text{Falcon} \mid Y_1^1\, Y_1^2\right) &\simeq 11{\cdot}0\%\\
\left(x = \text{Caravan} \mid Y_1^1\, Y_1^2\right) &\simeq 0{\cdot}2\%.
\end{aligned} \tag{40}$$

Thus for the chance that the aircraft is a Learjet, the two latest probabilities of 70%, 80% derived from sensor measurements have fused to update the old value of 50% to a new value of 88·8%, and so on as summarised in table 2. These numbers reflect the strong belief that the aircraft is highly likely to be a Learjet, less probably a Falcon, and almost certainly not a Caravan.

Three or more sensors: The analysis that produced equation (38) is easily generalised for the case of multiple sensors. The three sensor result is

$$\left(x \mid Y_1^1\, Y_1^2\, Y_1^3\right) = \frac{\left(x \mid Y_1^1\right)\left(x \mid Y_1^2\right)\left(x \mid Y_1^3\right)\left(x \mid Y_0^1\, Y_0^2\, Y_0^3\right)}{\left(x \mid Y_0^1\right)\left(x \mid Y_0^2\right)\left(x \mid Y_0^3\right)} \times \text{normalisation}, \tag{41}$$

and so on for more sensors. This expression also shows that the fusion order is irrelevant, a result that also holds in Dempster–Shafer theory. Without a doubt, this fact simplifies multiple sensor fusion enormously.

4. Dempster–Shafer data fusion

The Bayes and Dempster–Shafer approaches are both based on the concept of attaching weightings to the postulated states of the system being measured. While Bayes applies a

Table 2. Evolution of probabilities for the various aircraft.

		Latest sensor probs:		
Target type	Old value	Sensor 1	Sensor 2	New value
Learjet	50%	70%	80%	88·8%
Falcon	40%	29%	15%	11·0%
Caravan	10%	1%	5%	0·2%

more "classical" meaning to these in terms of well known ideas about probability, Dempster–Shafer (Dempster 1967, 1968; Shafer 1976; Blackman & Popoli 1999) allow other alternative scenarios for the system, such as treating equally the sets of alternatives that have a nonzero intersection: for example, we can combine all the alternatives to make a new state corresponding to "unknown". But the weightings, which in Bayes' classical probability theory are probabilities, are less well understood in Dempster–Shafer theory. Dempster–Shafer's analogous quantities are called *masses*, underlining the fact that they are only more or less to be understood as probabilities.

Dempster–Shafer theory assigns its masses to all of the subsets of the entities that comprise a system. Suppose for example that the system has 5 members. We can label them all, and describe any particular subset by writing say "1" next to each element that is in the subset, and "0" next to each one that isn't. In this way it can be seen that there are 2^5 subsets possible. If the original set is called S then the set of all subsets (that Dempster–Shafer takes as its start point) is called 2^S, the *power set*.

A good example of applying Dempster–Shafer theory is covered in the work of Zou *et al* (2000) discussed in § 2·2. Their robot divides its surroundings into a grid, assigning to each cell in this grid a mass: a measure of confidence in each of the alternatives "occupied", "empty" and "unknown". Although this mass is strictly speaking not a probability, certainly the sum of the masses of all of the combinations of the three alternatives (forming the power set) is required to equal one. In this case, because "unknown" equals "occupied or empty", these three alternatives (together with the empty set, which has mass zero) form the whole power set.

Dempster–Shafer theory gives a rule for calculating the confidence measure of each state, based on data from both new and old evidence. This rule, Dempster's rule of combination, can be described for Zou's work as follows. If the power set of alternatives that their robot builds is

$$\{\text{occupied, empty, unknown}\} \quad \text{which we write as} \quad \{O, E, U\}, \tag{42}$$

then we consider three masses: the bottom-line mass m that we require, being the confidence in each element of the power set; the measure of confidence m_s from sensors (which must be modelled); and the measure of confidence m_o from old existing evidence (which was the mass m from the previous iteration of Dempster's rule). As discussed in the next section, Dempster's rule of combination then gives, for elements A, B, C of the power set:

$$m(C) = \left[\sum_{A\cap B=C} m_s(A)m_o(B)\right] \Bigg/ \left[1 - \sum_{A\cap B=\varnothing} m_s(A)m_o(B)\right]. \tag{43}$$

Apply this to the robot's search for occupied regions of the grid. Dempster's rule becomes

$$m(O) = \frac{m_s(O)m_o(O) + m_s(O)m_o(U) + m_s(U)m_o(O)}{1 - m_s(O)m_o(E) - m_s(E)m_o(O)}. \tag{44}$$

While Zou's robot explores its surroundings, it calculates $m(O)$ for each point of the grid that makes up its region of mobility, and plots a point if $m(O)$ is larger than some preset confidence level. Hopefully, the picture it plots will be a plan of the walls of its environment.

In practice, as we have already noted, Zou and coworkers 2000 did achieve good results, but the quality of these was strongly influenced by the choice of parameters determining the sensor masses m_s.

4.1 *Fusing two sensors*

As a more extensive example of applying Dempster–Shafer theory, focus again on the aircraft problem considered in § 3·2. We will allow two extra states of our knowledge:

(1) The "unknown" state, where a decision as to what the aircraft is does not appear to be possible at all. This is equivalent to the subset {Learjet, Falcon,}.
(2) The "fast" state, where we cannot distinguish between a Learjet and a Falcon. This is equivalent to {Learjet, Falcon}.

Suppose then that two sensors allocate masses to the power set as in table 3; the third column holds the final fused masses that we are about to calculate. Of the eight subsets that can be formed from the three aircraft, only five are actually useful, so these are the only ones allocated any mass. Dempster–Shafer also requires that the masses sum to one over the whole power set. Remember that the masses are not quite probabilities: for example if the sensor 1 probability that the aircraft is a Learjet was really just another word for its mass of 30%, then the extra probabilities given to the Learjet through the sets of fast and unknown aircrafts would not make any sense.

These masses are now fused using Dempster's rule of combination. This rule can in the first instance be written quite simply as a proportionality, using the notation of (34) to denote sensor number as a superscript:

$$m^{1,2}(C) \propto \sum_{A \cap B = C} m^1(A)\, m^2(B). \tag{45}$$

We will combine the data of table 3 using this rule. For example the Learjet:

$$\begin{aligned} m^{1,2}(\mathit{Learjet}) \propto\ & m^1(\mathit{Learjet})\, m^2(\mathit{Learjet}) + m^1(\mathit{Learjet})\, m^2(\mathit{Fast}) \\ & + m^1(\mathit{Learjet})\, m^2(\mathit{Unknown}) + m^1(\mathit{Fast})\, m^2(\mathit{Learjet}) \\ & + m^1(\mathit{Unknown})\, m^2(\mathit{Learjet}) \\ =\ & 0{\cdot}30 \times 0{\cdot}40 + 0{\cdot}30 \times 0{\cdot}45 + 0{\cdot}30 \times 0{\cdot}03 + 0{\cdot}42 \times 0{\cdot}40 \\ & + 0{\cdot}10 \times 0{\cdot}40 \\ =\ & 0{\cdot}47. \end{aligned} \tag{46}$$

The other relative masses are found similarly. Normalising them by dividing each by their sum yields the final mass values: the third column of table 3. The fusion reinforces the idea

Table 3. Mass assignments for the various aircraft.

Sensor 1 allocates a mass m^1, while sensor 2 allocates a mass m^2

Target type	Sensor 1 (mass m^1)	Sensor 2 (mass m^2)	Fused masses (mass $m^{1,2}$)
Learjet	30%	40%	55%
Falcon	15%	10%	16%
Caravan	3%	2%	0·4%
Fast	42%	45%	29%
Unknown	10%	3%	0·3%
Total mass	100%	100%	100% (correcting for rounding errors)

Table 4. A new set of mass assignments, to highlight the "fast" subset anomaly in table 3.

Target type	Sensor 1 (mass m^1)	Sensor 2 (mass m^2)	Fused masses (mass $m^{1,2}$)
Learjet	30%	50%	63%
Falcon	15%	30%	31%
Caravan	3%	17%	3·5%
Fast	42%		2%
Unknown	10%	3%	0·5%
Total mass	100%	100%	100%

that the aircraft is a Learjet and, together with our initial confidence in its being a fast aircraft, means that we are more sure than ever that it is not a Caravan. Interestingly though, despite the fact that most of the mass is assigned to the two fast aircraft, the amount of mass assigned to the "fast" type is not as high as we might expect. Again, this is a good reason not to interpret Dempster–Shafer masses as probabilities.

We can highlight this apparent anomaly further by reworking the example with a new set of masses, as shown in table 4. The second sensor now assigns no mass at all to the "fast" type. We might interpret this to mean that it has no opinion on whether the aircraft is fast or not. But, such a state of affairs is no different numerically from assigning a zero mass: as if the second sensor has a strong belief that the aircraft is not fast! As before, fusing the masses of the first two columns of table 4 produces the third column. Although the fused masses still lead to the same belief as previously, the 2% value for $m^{1,2}(\text{Fast})$ is clearly at odds with the conclusion that the aircraft is very probably either a Learjet or a Falcon. So masses certainly are not probabilities. It might well be that a lack of knowledge of a state means that we should assign to it a mass higher than zero, but just what that mass should be, considering the possibly high total number of subsets, is open to interpretation. However, as we shall see in the next section, the new notions of support and plausibility introduced by Dempster–Shafer theory go far to rescue this paradoxical situation.

Owing to the seeming lack of significance given to the "fast" state, perhaps we should have no intrinsic interest in calculating its mass. In fact, knowledge of this mass is actually not required for the final normalisation,[2] so that Dempster's rule is usually written as an equality:

$$m^{1,2}(C) = \frac{\sum\limits_{A\cap B=C} m^1(A)\, m^2(B)}{\sum\limits_{A\cap B\neq\varnothing} m^1(A)\, m^2(B)} = \frac{\sum\limits_{A\cap B=C} m^1(A)\, m^2(B)}{1-\sum\limits_{A\cap B=\varnothing} m^1(A)\, m^2(B)}. \tag{47}$$

[2]The normalisation arises in the following way. Since the sum of the masses of each sensor is required to be one, it must be true that the sum of all products of masses (one from each sensor) must also be one. But these products are just all the possible numbers that appear in Dempster's rule of combination (45). So this sum can be split into two parts: terms where the sets involved have a nonempty intersection and thus appear somewhere in the calculation, and terms where the sets involved have an empty intersection and so don't appear. To normalise, we'll ultimately be dividing each relative mass by the sum of all products that do appear in Dempster's rule, or – perhaps the easier number to evaluate – one minus the sum of all products that don't appear.

Dempster–Shafer in tracking: A comparison of Dempster–Shafer fusion (47) and Bayes fusion (38) shows that there is no time evolution in (47). But we can allow for it after the sensors have been fused, by a further application of Dempster's rule, where the sets A, B in (47) now refer to new and old data. Zou's robot is an example of this sort of fusion from the literature, as discussed in the beginning of § 4.

4.2 *Three or more sensors*

In the case of three or more sensors, Dempster's rule might in principle be applied in different ways depending on which order is chosen for the sensors. But it turns out that because the rule is only concerned with set intersections, the fusion order becomes irrelevant. Thus three sensors fuse to give

$$m^{1,2,3}(D) = \frac{\sum\limits_{A\cap B\cap C=D} m^1(A)m^2(B)m^3(C)}{\sum\limits_{A\cap B\cap C\neq\varnothing} m^1(A)m^2(B)m^3(C)} = \frac{\sum\limits_{A\cap B\cap C=D} m^1(A)m^2(B)\ m^3(C)}{1-\sum\limits_{A\cap B\cap C=\varnothing} m^1(A)m^2(B)m^3(C)}, \tag{48}$$

and higher numbers are dealt with similarly.

4.3 *Support and plausibility*

Dempster–Shafer theory contains *two new ideas* that are foreign to Bayes theory. These are the notions of *support* and *plausibility*. For example, the support for the aircraft being "fast" is defined to be the total mass of all states implying the "fast" state. Thus

$$\text{spt}(A) = \sum_{B\subseteq A} m(B). \tag{49}$$

The support is a kind of loose lower limit to the uncertainty. On the other hand, a loose upper limit to the uncertainty is the plausibility. This is defined, for the "fast" state, as the total mass of all states that don't contradict the "fast" state. In other words:

$$\text{pls}(A) = \sum_{A\cap B\neq\varnothing} m(B). \tag{50}$$

The supports and plausibilities for the masses of table 3 are given in table 5.

Table 5. Supports and plausibilities associated with table 3.

	Sensor 1		Sensor 2		Fused masses	
Target type	Spt	Pls	Spt	Pls	Spt	Pls
Learjet	30%	82%	40%	88%	55%	84%
Falcon	15%	67%	10%	58%	16%	45%
Caravan	3%	13%	2%	5%	0·4%	1%
Fast	87%	97%	95%	98%	99%	∼100%
Unknown	100%	100%	100%	100%	100%	100%

Interpreting the probability of the state as lying roughly somewhere between the support and the plausibility gives the following results for what the aircraft might be, based on the fused data. There is a good possibility that it's a Learjet; a reasonable chance that it's a Falcon; almost no chance of its being a Caravan, which goes hand in hand with the virtual certainty that the aircraft is fast. Finally, the last implied probability might look nonsensical: it might appear to suggest that there is a 100% lack of knowledge of what the aircraft is, despite all that has just been said. But that's not what it says at all. What it does say is that there is complete certainty that the aircraft's identity is unknown. And that is quite true: the aircraft's identity *is* unknown. But what is also meant by the 100% is that there is complete certainty that the aircraft is *something*, even if we cannot be sure what that something is. Even so, we have used such assumptions as

$$\{\text{Learjet}\} \cap \text{Unknown} = \{\text{Learjet}\}, \tag{51}$$

which is not necessarily true, because we cannot be sure that the Unknown set does contain a Learjet. Dempster–Shafer theory treats the Unknown set as a superset, which is why we have assumed it contains a Learjet. But this vagueness of just what is meant by an "Unknown" state can and does give rise to apparent contradictions in Dempster–Shafer theory.

5. Comparing the Dempster–Shafer and Bayes theories

The major difference between these two theories is that Bayes works with probabilities, which is to say rigorously-defined numbers that reflect how often an event will occur if an experiment is performed a large number of times. On the other hand, Dempster–Shafer theory considers a space of elements that each reflect not what Nature chooses, but rather the state of *our knowledge* after making a measurement. Thus, Bayes does not use a specific state called "unknown emitter type" – although after applying Bayes theory, we might well have no clear winner, and will decide that the state of the emitter is best described as unknown. On the other hand, Dempster–Shafer certainly does require us to include this "unknown emitter type" state, because that can well be the state of *our knowledge* at any time. Of course the plausibilities and supports that Dempster–Shafer generates also may or may not give a clear winner for what the state of the emitter is, but this again is distinct from the introduction into that theory of the "unknown emitter type" state, which is always done.

The fact that we tend to think of Dempster–Shafer masses somewhat nebulously as probabilities suggests that we should perhaps use real probabilities when we can, but Dempster–Shafer theory doesn't demand this.

Both theories have a certain initial requirement. Dempster–Shafer theory requires masses to be assigned in a meaningful way to the various alternatives, including the "unknown" state; whereas Bayes theory requires prior probabilities – although at least for Bayes, the alternatives to which they're applied are all well defined. One advantage of using one approach over the other is the extent to which prior information is available. Although Dempster–Shafer theory doesn't need prior probabilities to function, it does require some preliminary assignment of masses that reflects our initial knowledge of the system.

Dempster–Shafer theory also has the advantage of allowing more explicitly for an undecided state of our knowledge. In the military arena, it can of course sometimes be far safer to be undecided about what the identity of a target is, than to decide wrongly and act accordingly with what might be disastrous consequences.

Dempster–Shafer also allows the computation of the additional notions of support and plausibility, as opposed to a Bayes approach which is restricted to the classical notion of probabilities only. On the other hand, while Bayes theory might be restricted to more classical notions (i.e. probability), the pedigree of these gives it an edge over Dempster–Shafer in terms of being better understood and accepted.

Dempster–Shafer calculations tend to be longer and more involved than their Bayes analogues (which are not required to work with all the elements of a set); and despite the fact that earlier literature (e.g. Cremer *et al* 1998 and Braun 2000) indicates that Dempster–Shafer can sometimes perform better than Bayes theory, Dempster–Shafer's computational disadvantages do nothing to increase its popularity.

Braun (2000) has performed a Monte Carlo comparison between the Dempster–Shafer and Bayes approaches to data fusion. The paper begins with a short overview of Dempster–Shafer theory. It simply but clearly defines the Dempster–Shafer power set approach, along with the probability structure built upon this set: basic probability assignments, belief- and plausibility functions. It follows this with a simple but very clear example of Dempster–Shafer formalism by applying the central rule of the theory, the Dempster combination rule, to a set of data.

What is not at all clear is precisely which sort of algorithm Braun is implementing to run the Monte Carlo simulations, and how the data is generated. He considers a set of sensors observing objects. These objects can belong to any one of a number of classes, with the job of the sensors being to decide to which class each object belongs. Specific numbers are not mentioned, although he does plot the number of correct assignments versus the total number of fusion events for zero to 2500 events.

The results of the simulations show fairly linear plots for both the Dempster–Shafer and Bayes approaches. The Bayes approach rises to a maximum of 1700 successes in the 2500 fusion instances, while the Dempster–Shafer mode attains a maximum of 2100 successes – which would seem to place it as the more successful theory, although Braun (2000) does not say as much directly. He does produce somewhat obscure plots showing finer details of the Bayes and Dempster–Shafer successes as functions of the degree of confidence in the various hypotheses that make up his system. What these show is that both methods are robust over the entire sensor information domain, and generally where one succeeds or fails the other will do the same, with just a slight edge being given to Dempster–Shafer as compared with the Bayes approach.

6. Concluding remarks

Although data fusion still seems to take tracking as its prototype, fusion applications are beginning to be produced in numerous other areas. Not all of these uses have a statistical basis however; often the focus is just on how to fuse data in whichever way, with the question of whether that fusion is the best in some sense not always being addressed. Nor can it always be, since very often the calculations involved might be prohibitively many and complex. Currently too, there is still a good deal of philosophising about pertinent data fusion issues, and the lack of hard rules to back this up is partly due to the difficulty in finding common ground for the many applications to which fusion is now being applied.

Appendix A. Gaussian distribution theorems

The following theorems are special cases of the one-dimensional results that the product of Gaussians is another Gaussian, and the integral of a Gaussian is also another Gaussian.

The notation is as follows. Just as a Gaussian distribution in one dimension is written in terms of its mean μ and variance σ^2 as

$$N(x;\,\mu,\sigma^2) \equiv \frac{1}{\sigma\sqrt{2\pi}} \exp \frac{-(x-\mu)^2}{2\sigma^2}, \tag{A.1}$$

so also a Gaussian distribution in an n-dimensional vector x is denoted through its mean vector μ and covariance matrix P in the following way:

$$N(x;\,\mu,\,P) \equiv \frac{1}{|P|^{1/2}(2\pi)^{n/2}} \exp \frac{-1}{2}(x-\mu)^T P^{-1}(x-\mu) = N(x-\mu;\,0,\,P). \tag{A.2}$$

Theorem 1.

$$\frac{N(x_1;\,\mu_1,\,P_1)\;N(x_2;\,Hx_1,\,P_2)}{N(x_2;\,H\mu_1,\,P_3)} = N(x_1;\,\mu,\,P), \tag{A.3}$$

where

$$\begin{aligned} K &= P_1 H^T (H P_1 H^T + P_2)^{-1}, \\ \mu &= \mu_1 + K(x_2 - H\mu_1), \\ P &= (1 - KH)P_1. \end{aligned} \tag{A.4}$$

The method of proving the above theorem is relatively well known, being first shown in Ho (1964) and later appearing in a number of texts. However, the proof of the next theorem which deals with the Chapman–Kolmogorov theorem is not that well known.

Theorem 2.

$$\int_{-\infty}^{\infty} dx_1\; N(x_1;\,\mu_1,\,P_1)\;N(x_2;\,Fx_1,\,P_2) = N(x_2;\,\mu,\,P), \tag{A.5}$$

where

$$\begin{aligned} \mu &= F\mu_1, \\ P &= F P_1 F^T + P_2. \end{aligned} \tag{A.6}$$

Here, we present a proof of the above theorem by directly solving the integral. Note that in Gaussian integrals, P_1 and P_2 are symmetric, which means their inverses will be too – a fact that we will use often in our derivation.

The left hand side of (A.5) is

$$\begin{aligned} \int_{-\infty}^{\infty} \mathrm{d}x_1\; N(x_1;\,\mu_1,\,P_1)\;N(x_2;\,Fx_1,\,P_2) &= \frac{1}{(2\pi)^{n/2}\,|P_1|^{1/2}\,(2\pi)^{n/2}\,|P_2|^{1/2}} \\ \times \int \exp \frac{-1}{2}\left[(x_1-\mu_1)^T\,P_1^{-1}(x_1-\mu_1) + (x_2-Fx_1)^T P_2^{-1}(x_2-Fx_1)\right]\,\mathrm{d}x_1. \end{aligned} \tag{A.7}$$

Write the integrand on the right hand side as $e^{-E/2}$, so that

$$E = (x_1 - \mu_1)^T P_1^{-1}(x_1 - \mu_1) + (x_2 - Fx_1)^T P_2^{-1}(x_2 - Fx_1). \tag{A.8}$$

If we define

$$\begin{aligned} A &= x_2 - F\mu_1, \\ B &= x_1 - \mu_1, \end{aligned} \tag{A.9}$$

then it follows that $x_2 - Fx_1 = A - FB$, in which case

$$\begin{aligned} E &= B^T P_1^{-1} B + (A - FB)^T P_2^{-1}(A - FB) \\ &= B^T P_1^{-1} B + A^T P_2^{-1} A - B^T F^T P_2^{-1} A - A^T P_2^{-1} FB + B^T F^T P_2^{-1} FB. \end{aligned} \tag{A.10}$$

Group the first and last terms to write

$$E = B^T \left(P_1^{-1} + F^T P_2^{-1} F\right) B + A^T P_2^{-1} A - B^T F^T P_2^{-1} A - A^T P_2^{-1} FB. \tag{A.11}$$

It will be convenient to introduce two new matrices:

$$\begin{aligned} M^{-1} &= P_1^{-1} + F^T P_2^{-1} F, \\ P &= P_2 + F P_1 F^T. \end{aligned} \tag{A.12}$$

Note that because P_1 and P_2 are symmetric, so will M and M^{-1} also be, which we make use of frequently. The first term in (A.11) then becomes

$$E = B^T M^{-1} B + A^T P_2^{-1} A - B^T F^T P_2^{-1} A - A^T P_2^{-1} FB. \tag{A.13}$$

We can simplify E by first inverting P. The very useful matrix inversion lemma[3] gives

$$P^{-1} = \left(P_2 + F P_1 F^T\right)^{-1} = P_2^{-1} - P_2^{-1} F M F^T P_2^{-1}, \tag{A.14}$$

which rearranges trivially to give

$$P_2^{-1} = P^{-1} + P_2^{-1} F M F^T P_2^{-1}. \tag{A.15}$$

We now insert this last expression into the second term of (A.13), giving

$$\begin{aligned} E &= B^T M^{-1} B + A^T P^{-1} A + A^T P_2^{-1} F M F^T P_2^{-1} A - B^T F^T P_2^{-1} A - A^T P_2^{-1} FB \\ &= \left(B - M F^T P_2^{-1} A\right)^T M^{-1} \left(B - M F^T P_2^{-1} A\right) + A^T P^{-1} A. \end{aligned} \tag{A.16}$$

Defining $\mu_2 = \mu_1 + M F^T P_2^{-1} A$ produces $B - M F^T P_2^{-1} A = x_1 - \mu_2$, in which case

$$E = (x_1 - \mu_2)^T M^{-1} (x_1 - \mu_2) + A^T P^{-1} A. \tag{A.17}$$

[3]This says that for matrices a, b, c, d of appropriate size and invertibility:

$$(a + bcd)^{-1} = a^{-1} - a^{-1} b \left(c^{-1} + d a^{-1} b\right)^{-1} d a^{-1}$$

Hence the right hand side of (A.7) becomes

$$\frac{e^{\frac{-1}{2}A^T P^{-1} A}}{(2\pi)^{n/2}\,|P_1|^{1/2}\,(2\pi)^{n/2}\,|P_2|^{1/2}} \int \exp\frac{-1}{2}\left[(x_1-\mu_2)^T\,M^{-1}\,(x_1-\mu_2)\right] \mathrm{d}x_1. \tag{A.18}$$

This is a great improvement over (A.7), because the integration variable x_1 only appears in a simple Gaussian integral, and so can be integrated out. But before doing that integration, we will show that the normalisation factors can be simplified, by means of the following fact:

$$|P_1\,P_2| = |P\,M|. \tag{A.19}$$

To prove this fact, we first begin to rewrite P in terms of M, P_1 and P_2:

$$\begin{aligned} P &= F P_1 F^T + P_2 \\ &= \left(F P_1 F^T P_2^{-1} + 1\right)\,P_2. \end{aligned} \tag{A.20}$$

It will prove useful to factor out F, but unfortunately because F is in general not square and so not invertible, we cannot just introduce factors of F^{-1} to effect this. However, it's quite sufficient to make use of a "right inverse", through introducing a factor of $F F^T (F F^T)^{-1}$, since this is always well defined. In that case

$$\begin{aligned} P &= \left(F P_1 F^T P_2^{-1} + 1\right)\,F F^T\,(F F^T)^{-1}\,P_2 \\ &= \left(F P_1 F^T P_2^{-1} F + F\right)\,F^T\,(F F^T)^{-1}\,P_2 \\ &= F P_1 \left(F^T P_2^{-1} F + P_1^{-1}\right)\,F^T\,(F F^T)^{-1}\,P_2 \\ &= F P_1\,M^{-1}\,F^T\,(F F^T)^{-1}\,P_2. \end{aligned} \tag{A.21}$$

If we now multiply both sides by M and then take the determinant of each, we obtain

$$\begin{aligned} |P\,M| &= |F\,P_1\,M^{-1}\,F^T\,(F F^T)^{-1}\,P_2\,M| \\ &= |F|\,|P_1|\,|M|^{-1}\,|F^T|\,|F F^T|^{-1}\,|P_2|\,|M| \\ &= |P_1\,P_2|, \end{aligned} \tag{A.22}$$

since the various determinants cancel. QED. This fact then enables the Gaussian integral over x_1 in equation (A.18) to be easily set equal to 1; and so we arrive at a simple expression for equation (A.7):

$$\begin{aligned} \int_{-\infty}^{\infty} dx_1\;N(x_1;\,\mu_1,\,P_1)\;N(x_2;\,F x_1,\,P_2) &= \frac{e^{\frac{-1}{2}A^T P^{-1} A}}{(2\pi)^{n/2}\,|P|^{1/2}} \\ &= \frac{1}{(2\pi)^{n/2}\,|P|^{1/2}} \exp\frac{-1}{2}\left[(x_2-F\mu_1)^T\,P^{-1}\,(x_2-F\mu_1)\right] \\ &= N(x_2;\,F\mu_1,\,P). \end{aligned} \tag{A.23}$$

This completes the proof.

References

Blackman S, Popoli R 1999 *Design and analysis of modern tracking systems* (Boston: Artech House)

Blasch E 1998 Decision making in multi-fiscal and multi-monetary policy measurements. *Proc. Int. Conf. on Multisource-Multisensor Information Fusion (Fusion '98)* 1: 285–292

Braun J 2000 Dempster–Shafer theory and Bayesian reasoning in multisensor data fusion, *Sensor Fusion: Architectures, Algorithms and Applications IV; Proc. SPIE* 4051: 255–266

Cooper M, Miller M 1998 Information gain in object recognition via sensor fusion. *Proc. Int. Conf. on Multisource-Multisensor Information Fusion (Fusion '98)* 1: 143–148

Cremer F, den Breejen E, Schutte K 1998 Sensor data fusion for anti-personnel land mine detection. *Proc. EuroFusion 98* 55–60

Debon R, Solaiman B, Cauvin J-M, Peyronny L, Roux C 1999 Aorta detection in ultrasound medical image sequences using Hough transform and data fusion. *Proc. 2nd Int. Conf. on Information Fusion (Fusion '99)* 1: 59–66

Debon R, Solaiman B, Roux C, Cauvin J-M, Robazkiewicz M 2000 Fuzzy fusion and belief updating. Application to esophagus wall detection on ultrasound images. *Proc. 3rd Int. Conf. on Information Fusion (Fusion 2000)* 1: TuC5_17–TuC5_23

Dempster A P 1967 Upper and lower probabilities induced by a multivalued mapping. *Ann. Math. Stat.* 38: 325–339

Dempster A P 1968 A generalization of Bayesian inference, *J. R. Stat. Soc.* B 30: 205–247.

Hall D, Garga A 1999 Pitfalls in data fusion (and how to avoid them). *Proc. 2nd Int. Conf. on Information Fusion (Fusion '99)* 1: 429–436

Hatch M, Jahn E, Kaina J 1999 Fusion of multi-sensor information from an autonomous undersea distributed field of sensors. *Proc. 2nd Int. Conf. on Information Fusion (Fusion '99)* 1: 4–11

Haupt G, Kasdin N, Keiser G, Parkinson B 1996 Optimal recursive iterative algorithm for discrete nonlinear least-squares estimation. *J. Guidance, Control Dynam.* 19: 643–649

Heifetz M, Keiser G 1999 Data analysis in the gravity probe B relativity experiment. *Proc. 2nd Int. Conf. on Information Fusion (Fusion '99)* 2: 1121–1125

Ho Y C 1964 A Bayesian approach to problems in stochastic estimation and control. *IEEE Trans. Autom. Control* AC-9: 333

Hush D, Horne B 1993 Progress in supervised neural networks: what's new since Lippman? *IEEE Signal Process. Mag.* 10(1): 8–39

Kewley D J 1992 Notes on the use of Dempster–Shafer and fuzzy reasoning to fuse identity attribute data, Defence Science and Technology Organisation, Adelaide. Technical memorandum SRL-0094-TM

Kokar M, Bedworth M, Frankel C 2000 A reference model for data fusion systems. *Sensor fusion: Architectures, algorithms and applications IV; Proc. SPIE* 4051: 191–202

Krieg M L 2002 A Bayesian belief network approach to multi-sensor kinematic and attribute tracking. *Proc. Conf. on Information, Decision and Control (IDC2002)*

Myler H 2000 Characterization of disagreement in multiplatform and multisensor fusion analysis. *Signal processing Sensor fusion, and target recognition IX; Proc. SPIE* 4052: 240–248

Rodríguez F, Portas J, Herrero J, Corredera J 1998 Multisensor and ADS data integration for en-route and terminal area air surveillance. *Proc. Int. Conf. on Multisource-Multisensor Information Fusion (Fusion '98)* 2: 827–834

Shafer G 1976 *A mathematical theory of evidence* (Princeton, NJ: University Press)

Schwartz S 2000 Algorithm for automatic recognition of formations of moving targets. *Sensor fusion: Architectures, algorithms and applications IV; Proc. SPIE* 4051: 407–417

Simard M-A, Lefebvre E, Helleur C 2000 Multisource information fusion applied to ship identification for the recognised maritime picture. *Sensor fusion: Architectures, algorithms and applications IV; Proc. SPIE* 4051: 67–78

Strömberg D 2000 A multi-level approach to sensor management. *Sensor fusion: Architectures, algorithms and applications IV; Proc. SPIE* 4051: 456–461

Triesch J 2000 Self-organized integration of adaptive visual cues for face tracking. *Sensor fusion: Architectures, algorithms and applications IV; Proc. SPIE* 4051: 397–406

Viola P, Gilles S 1996 at: http://www-rocq.inria.fr/ gilles/IMMMI/immmi.html The report is by Gilles who uses Viola's work, and is entitled "Description and experimentation of image matching using mutual information"

Watson G, Rice T, Alouani A 2000 An IMM architecture for track fusion. *Signal processing, sensor fusion, and target recognition IX; Proc. SPIE* 4052: 2–13

Zachary J, Iyengar S 1999 Three dimensional data fusion for biomedical surface reconstruction. *Proc. 2nd Int. Conf. on Information Fusion (Fusion '99)* 1: 39–45

Zou Y, Ho Y K, Chua C S, Zhou X W 2000 Multi-ultrasonic sensor fusion for autonomous mobile robots. *Sensor fusion: Architectures, algorithms and applications IV; Proc. SPIE* 4051: 314–321