Hard/Soft Data Fusion using Gaussian Process Classifiers

By Steven Reece[1], Stephen Roberts[1], David Nicholson[2], Chris Lloyd[2]

[1]Robotics Research Group, Dept. Engineering Science, Oxford University
{reece, sjrob}@robots.ox.ac.uk
[2]BAE Systems Advanced Technology Centre, Filton, Bristol
{david.nicholson2,chris.m.lloyd}@baesystems.com

Abstract

Modern applications of data fusion are rarely starved of data but they look more challenging because the data can be diverse (hard and soft), uncertain and ambiguous, and often swamped by irrelevant detail. This paper presents a mathematical framework for dealing with these issues. It manages diverse data by representing it in the common format of a kernel matrix. Uncertainty is managed by interpreting the kernels as the covariance matrices of Bayesian Gaussian Processes. Finally, irrelevant detail is managed by automatically detecting the relevant (or, conversely, the irrelevant) components within a multi-source dataset. The framework is illustrated by applying it to a synthetic vehicle-borne Improvised Explosive Device intent recognition scenario.

1. Introduction

With the ongoing proliferation of sensor types and sensing modalities in civilian and military settings, information fusion must be able to deal with large volumes of heterogeneous data. In particular, the 'hard' data associated with physical sensor devices has become increasingly augmented by 'soft' data associated with human sources. Often, these data types complement one another because hard sources make objective measurements of target states (e.g. identity, position) whereas soft sources make subjective judgements about target states (e.g. intent). Thus a combination of sources is likely to improve target state estimation in terms of overall completeness and accuracy [Hall & Jordan (2010)].

Hard-soft fusion is an emerging topic of both theoretical and applied importance within the information fusion community. This has been recognised by the recent publication of a textbook on the subject [Hall & Jordan (2010)], a special session at Fusion 2010 [Nicholson *et al* (2010)], a multi-million dollar Multidisciplinary University Research Initiative on network-based hard-soft information fusion funded by the US Army Research Office, and the creation of a large hard-soft dataset for promoting algorithm development and analysis [Pravia *et al* (2008)].

The complex target of interest in this paper is the 'pattern of life' associated with a counter-insurgency (COIN) scenario involving vehicle movements between specific buildings in a fictitious town. Based on surveillance reports from hard and soft sources stationed in the town, intelligence analysts would like to classify this pattern as a signature of either benign or hostile intent. For example, routine delivery of goods to a shopping mall in the former case or vehicle-borne delivery of an Improvised Explosive Device (IED) to the mall in the latter. The focus of this paper is the formulation and proof-of-principle

testing of a mathematical framework for representing and fusing heterogeneous data sources to provide inference in support of intelligence analysts.

A common parametric approach to tackle information fusion problems is to represent them as probabilistic graphical models, such as Kalman Filters, Hidden Markov Models, or richer Bayesian Networks [Das (2008)]. These models encode domain knowledge in the structure of the graph and employ Bayesian methods to fuse conditional probability data at the nodes. In practice, parametric models may be very difficult to construct and implement due to limited domain knowledge. This is of particular concern in the complex and asymmetric COIN domain.

A kernel-based mathematical framework is described here for representing disparate hard/soft data as well as performing inference with this data. This framework can be used to design regressor and classifier algorithms, with a specific focus in this paper on binary classification problems. The kernel-based approach is able to quantify the different forms of uncertainty inherent in the data. Further, as the approach is Bayesian it provides a principled way of quantifying uncertainty in the inferences. Finally, the approach is able to determine the relevant information sources within a potentially large array of data streams. Consequently, the intelligence analyst is able to assign limited sensing resources to the most informative data streams.

While non-parametric methods have a long history of application in data mining and machine learning, there is currently a gap for a framework that addresses all these requirements. The purpose of this paper is therefore to present such a framework and provide some initial proof-of-principle testing.

2. Kernels for Heterogeneous Data

Given multiple heterogeneous data inputs relating to a scenario, the first requirement is to represent them in a common mathematical format. This section describes a kernel representation of data which represents data independent of its type in the common format of a kernel matrix.

A scenario may be described succinctly by a heterogeneous *feature vector*. A feature vector can contain hard and soft data, sequential events, relationships between events or virtually any information relating to the scenario. For example, the feature vector $\mathbb{V} =<$ 'Jim', 1604, 'large bag', 'soon afterwards' $>$ describes the scenario 'Jim arrived at 1604 with a large bag.He left soon afterwards'.

Let ϕ_i denote a function which extracts the i^{th} feature from the feature vector.

It is assumed that each feature vector contains the same sequence of data types. This allows individual features to be compared between feature vectors using distance measures tailored to specific types of data, whether it is nominal, ordinal, hard or soft. It is also assumed that each feature vector is complete and does not contain missing values. Relaxing these assumptions is a subject for future research.

Entire feature vectors may be compared by using kernels which transform individual features of different types into a common space. Each kernel, K, takes as input two values, $x_1 = \phi_i(\mathbb{V}_1)$ and $x_2 = \phi_i(\mathbb{V}_2)$, corresponding to the same feature in different feature vectors and returns a value indicating how close the two features actually are. There are various kernels for representing heterogeneous data types. An example of a kernel is the squared exponential kernel:

$$K_{SE}(x_1, x_2) = A \exp\left(-\frac{(x_1 - x_2)^2}{L^2}\right) \tag{2.1}$$

for any $x_1, x_2 \in \mathbb{R}$. The Squared Exponential kernel can be used to model smooth functions over the continuous quantitative inputs x. Here, $L \geq 0$ and $A \geq 0$ are hyperparameters and are called the *input scale* and the *output scale* respectively. They govern how correlated the output values are over neighbouring inputs.

Kernels can be combined in a number of ways, for example, by addition or by multiplication.

$$K_{new}(x_1, x_2) = K_1(x_1, x_2) + K_2(x_1, x_2) + \ldots + K_n(x_1, x_2) \tag{2.2}$$

$$K_{new}(x_1, x_2) = K_1(x_1, x_2) \times K_2(x_1, x_2) \times \ldots \times K_n(x_1, x_2). \tag{2.3}$$

In general, a kernel sum is appropriate for combining disjunctive inputs for which different outputs are correlated if *any* element of the input vectors is close. A kernel product is more appropriate when outputs are correlated only if the entire input vector is close.

Kernel combination may be used to place kernels over entire feature vectors. The feature kernel, $K_{\mathbb{V}}(\mathbb{V}_1, \mathbb{V}_2)$, measures the closeness of two feature vectors, $\mathbb{V}_1$ and $\mathbb{V}_2$. Firstly, choose an appropriate kernel, K_i, for each feature, $\phi_i(\mathbb{V})$. Then individual kernels are combined, for example, using the product rule into a single feature vector kernel:

$$K_{\mathbb{V}}(\mathbb{V}_1, \mathbb{V}_2) = \prod_i K_i(\phi_i(\mathbb{V}_1), \phi_i(\mathbb{V}_2)) \ . \tag{2.4}$$

The product of kernels is more appropriate for representing the COIN scenario investigated later in Section 4.

3. Gaussian Processes

The second requirement is to identify a mathematical approach to heterogeneous data fusion using kernels which offer a principled approach to the evaluation of uncertainty and risk.

The *Gaussian Process* (GP), Support Vector Machines and Kernel Density Estimators are all examples of kernel based methods. GPs were chosen as the basis for our mathematical framework because they provide a principled Bayesian approach to uncertainty and also impose an intuitive interpretation on the kernel, K. In GPs the kernel is often called the *covariance function*, as $K(x_1, x_2)$ is the *covariance* between the output values at x_1 and x_2.

A GP is often regarded as a "Gaussian distribution over functions" [Rasmussen & Williams (2006)]. It can be thought of as the generalisation of a Gaussian distribution over a finite vector space to a function space of infinite dimension. Just as a Gaussian is fully specified by its mean and covariance matrix, a Gaussian Process is fully described by its mean and covariance function K. Although our functions can be infinitely dimensional, GPs are used to infer, or predict, function values at a finite set of *test points* using the observed data. The training data $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ is drawn from a noisy process,

$$y_i = f(x_i) + \epsilon_i \tag{3.1}$$

where ϵ_i is independent identically distributed Gaussian noise with variance σ^2. For convenience both inputs and outputs are aggregated into $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_n\}$ respectively. The GP estimates the value of the function f at arbitrary test

points $X_* = \{x_{*1}, \ldots, x_{*m}\}$. The basic GP regression equations are given in [Rasmussen & Williams (2006)].

Each kernel has a set of hyperparameters, θ, which include the input and output scales. These hyperparameters can be inferred through Bayes' rule. The hyperparameters are usually given a vague prior distribution $Pr(\theta)$.

3.1. *The Binary Gaussian Process Classifier*

The key problem addressed in this paper is how to classify feature vectors into one of two classes denoted $y = 0$ or $y = 1$, indicating a hostile or benign event, for example. The GP regressor considered so far can be extended to classification problems [Rasmussen & Williams (2006)]. As identical scenarios are re-encountered they may exhibit behaviours occasionally associated with each class. For example, a small van driven by a nervous driver may only occasionally be hostile, but not necessarily all of the time. To accommodate a distribution over class labels for each scenario we model the probability of class membership, $Pr(y = 1 \mid x)$ for each scenario x. For binary classification $Pr(y = 0 \mid x) = 1 - Pr(y = 1 \mid x)$.

The scenarios, x, are expressed as feature vectors and form the inputs of our GP classifier. The outputs are the probability distributions $Pr(y = 1 \mid x)$. Since the output is a probability function then the output space must be confined to the range $[0, 1]$. Following [Rasmussen & Williams (2006)] we choose to map from the Gaussian process latent regressor function, f, onto $y \in [0, 1]$ via the sigmoid function, $g(f(x))$:

$$g(f(x)) = \frac{1}{1 + \exp(-sf(x))} \tag{3.2}$$

where $s > 0$ is the sigmoid *sensitivity*. Also, following [Rasmussen & Williams (2006)], we define the latent function so that it has a direct probabilistic interpretation:

$$Pr(y = 1 \mid x) = g(f(x)) \ . \tag{3.3}$$

The sigmoid serves a further purpose, it allows near step-wise changes in probability across class boundaries whilst requiring only smoothly varying functions in the latent space. Thus, we can use simple off-the-shelf covariance functions, K, such as the squared-exponential, without the need to define change-points or declare kernel non-stationarities at the class boundaries.

The sigmoid function approximately maps a normal distributed latent function into a Beta distributed class probability, $Pr(y = 1 \mid x)$, for each scenario, x:

$$\mathbb{B}(g(f(x))) \frac{dg(f(x))}{df(x)} \approx \mathbb{N}(f(x); m_x, P_x) \tag{3.4}$$

where $Pr(y = 1 \mid x) = g(f(x))$ and m_x and P_x are the latent function mean and variance for some normal distribution at x.

Let the training samples, D, be binary samples drawn for various scenarios. If we now focus on those samples of scenario x alone and the impact that they have on the posterior distributions, $Pr(y = 1 \mid x)$ and $Pr(y = 1 \mid x')$, for different scenarios, $x' \neq x$. These samples comprise N_x occurrences of scenario x where n_x of these samples are assigned class label $y(x) = 1$ and $N_x - n_x$ are assigned class label $y(x) = 0$. The posterior Beta distribution over $Pr(y = 1 \mid x)$ can be inferred using Bayes rule.

As $f(X)$ is a Gaussian process but the sigmoid function, g, is non-linear, the posterior $Pr(f(X^*) \mid z(X))$ can be inferred using the Extended Kalman Filter (EKF). Within the EKF implementation the latent function values of the test points,$X^* = \{x_1^*, \ldots, x_L^*\}$,

are stacked to form the L *state vector*, $f(X^*)$. The prior mean over $f(X^*)$ is chosen to be zero as this induces the appropriate prior mean of 0.5 for the class probabilities $Pr(y = 1 \mid x^*)$ for all $x^* \in X^*$. The EKF initial state covariance, K, is exactly the latent function Gaussian process covariance kernel. This is the feature vector product kernel described in Section 2.

The sample class counts, $z_x = n_x/N_x$, for M feature vectors $x \in X$ in the training set are also stacked, $z(X)$.

3.2. *Automatic Relevance Determination*

Determining the relevant features is crucial to successful classification. If we include irrelevant features in the feature vector then class uncertainty will increase. This arises when different feature values for irrelevant features increase the distance between, otherwise close, feature vectors. However, excluding relevant features also causes class uncertainty to increase. This arises because subtle distinctions between classes will be lost if relevant data is excluded. Fortunately, identifying the relevant features is a side effect of the GP classifier algorithm.

Following [Neal (1996)] when a feature is irrelevant its input scale will be inferred to be very large. For very large input scales the covariance will be independent of that input, effectively removing it from the feature vector.

4. Illustrative COIN Scenario

The COIN scenario used to illustrate the application of our mathematical framework is a vehicle-borne IED threat detection scenario, designed originally to test parametric modelling techniques [Grande *et al* (2008)]. The scenario involves a number of building facilities and a number of vehicles in transit between those facilities carrying out various functions: transport, collection, and delivery. If the intent of the participants of these collective events is hostile, they culminate in the delivery of an IED to a shopping mall. However, it is more likely that the collective events are benign and result in the delivery of garden and electrical supplies to the mall for reasons of routine business and commerce.

It is assumed that this urban 'hotspot' has been under observation by the authorities for some time and certain reports have been logged. Moreover, each of these reports is labelled according to a forensic analysis which determined whether it formed part of a hostile or benign scenario. Consequently, there is a set of labelled exemplars for each scenario class with which to train the GP classifier to recognise new scenarios.

The simulated dataset for this scenario consisted of a mix of hard (quantitative) and soft (qualitative) report types. In total there are ten observables associated with the scenario, seven hard and three soft. The hard data elements refer to time stamps associated with the various vehicle arrival, departure and wait times during the scenario. The soft elements refer to the sizes of the vehicles {very small, small, medium, large, very large} and the emotional state of the driver of the vehicle departing the depot {calm, nervous}.

The scenarios were created in such a way that only four of the ten observable features distinguish whether they are hostile or non-hostile. For example, a medium-sized vehicle is always used to convey an IED payload to the mall and its wait-time at the depot is constant rather than proportional to the size of the vehicle. The observables were generated with additive noise, blurring these subtle distinctions even further. The relevance probability determined by our framework for each source is displayed in Figure 1 for different numbers of labelled input scenarios. This figure shows the proportion of test instances where each source was identified as relevant.

Figure 2 displays representative results for the classification of the training and test

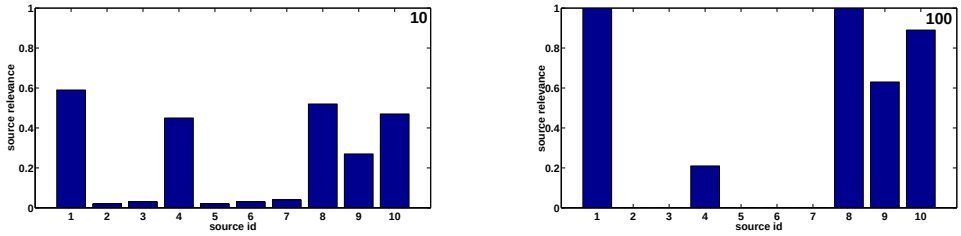

FIGURE 1. Source relevance probabilities inferred by our framework for different numbers of training inputs (indicated in the top right corner of each figure). The (unknown) true relevant sources in the scenarios that were simulated are 1 & 10.

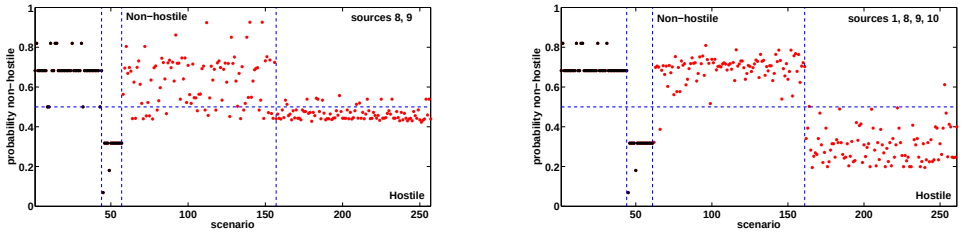

FIGURE 2. Improved classification performance as a result of fusing more relevant sources

scenarios when two and and four (all) of the relevant vectors are fused. These figures plot, for each distinct scenario numbered on the horizontal axis, the probability of being classified non-hostile on the vertical axis. Ignoring the classification of the training scenarios (left of scenario 60), the classification of the test scenarios (right of scenario 60) should fall into two blocks on the leading diagonal. The proportion of off-diagonal (misclassified) scenarios is notably reduced as more relevant sources are fused by the classifier.

Figure 3 displays the classification results in the case when two relevant features are fused and also when another relevant feature and five irrelevant features are combined. For this test dataset there are no misclassifications when only the two relevant features are combined. However, inclusion of the irrelevant features (despite the inclusion of an extra relevant feature) introduces numerous misclassifications.

5. Conclusion

A novel kernel-based approach for heterogeneous (hard and soft) data fusion has been developed and applied to a synthetic COIN problem. Kernels allow disparate information types to be represented in a common covariance space. The Gaussian Processes kernel method provides a rigorous probabilistic basis on which to manage uncertainty and as a side-effect allows relevant sources to be automatically detected.

This work was supported by BAE Systems Strategic Capability Solutions. We acknowledge further support from EPSRC project EP/I011587/1. A more detailed discussion of the material presented here can be found in [Reece *et al* (2011)].

REFERENCES

S. DAS 2008 High-Level Data Fusion, *Artech House*, 2008.
D. GRANDE, G. LEVCHUK, W. STACY, AND M. KRUGER 2008 Identification of adversarial

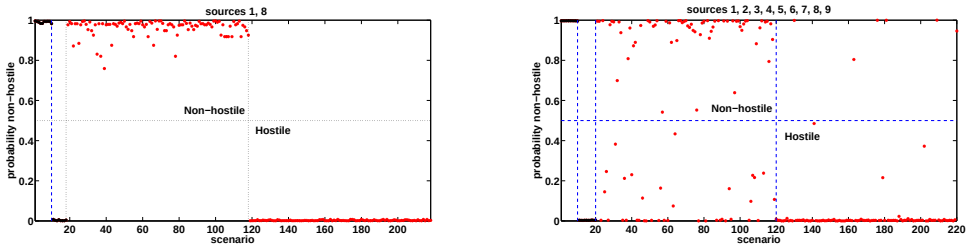

FIGURE 3. Improved classification performance as a result of excluding irrelevant sources.

activities: profiling latent use of facilities from structural data and real-time intelligence, In *Proceedings of the 13th International Command and Control Research and Technology Symposium Seattle, WA*, 2008.

D. L. HALL AND J. J. JORDAN 2010 Human-Centered Information Fusion, *Artech House*, 2010.

R. M. NEAL 1996 Bayesian Learning for Neural Networks, *Lecture Notes in Statistics 118. Springer, New York*, 1996.

D. NICHOLSON, C. LLOYD, M. WILLIAMS 2010 Information Fusion Algorithms and Analysis for an Exemplar Detection of Intent Problem, In *Proceedings of the 13th International Conference on Information Fusion (Fusion 2010), Edinburgh*, 2010.

M. A. PRAVIA, R.K. PRASANTH, P.O. ARAMBEL, C. SIDNER, AND C. Y. CHONG Generation of a fundamental data set for hard/soft information fusion, In *Proceedings of the 11th International Conference on Information Fusion (Fusion 2008), Cologne*, 2008.

C. E. RASMUSSEN, C. K. I. WILLIAMS Gaussian Processes for Machine Learning, *The MIT Press*, 2006.

S. REECE, S. ROBERTS, D. NICHOLSON, C. LLOYD 2011 Determining Intent using Hard/Soft Data and Gaussian Process Classifiers, In *Proceedings of the 14th International Conference on Information Fusion (Fusion 2011), Chicago, USA*, 2010.