Fully Bayesian Human-Machine Data Fusion for Robust Dynamic Target Surveillance and Characterization

by

Jeremy Muesing

B.S. in Aerospace Engineering, University of Colorado Boulder, 2017

A thesis submitted to the Faculty of the Graduate School of the University of Colorado in partial fulfillment of the requirements for the degree of Master of Science Ann and H.J. Smead Aerospace Engineering Sciences

2019

This thesis entitled: Fully Bayesian Human-Machine Data Fusion for Robust Dynamic Target Surveillance and Characterization written by Jeremy Muesing has been approved for the Ann and H.J. Smead Aerospace Engineering Sciences

Prof. Nisar Ahmed

Prof. Danielle Szafir

Date _____

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Muesing, Jeremy (M.S., Aerospace Engineering)

- Fully Bayesian Human-Machine Data Fusion for Robust Dynamic Target Surveillance and Characterization
- Thesis directed by Prof. Nisar Ahmed

This work examines the problem of rigorously characterizing and fusing observations provided by human operators with probabilistic information extracted by an automated data fusion system. in the context of dynamic multi-target track characterization for large-scale surveillance. This task is characterized by three major challenges. Firstly, stand-alone human operator observation errors are difficult to parameterize and calibrate a priori. Secondly, successive single target observations from one operator are, in general, not conditionally independent of one another. Finally, the decision of when a human operator should assist the automation depends on the observed operator error characteristics as well as the uncertainty of the automation's track characterizations, all of which must be calculated online. A new hierarchical fully Bayesian probabilistic model is developed to explicitly account both for uncertainties in 'human sensor' quality and Markovian conditional dependencies in successive target characterization reports. This model is used to perform online Bayesian inference via Gibbs sampling to simultaneously update the data fusion system's knowledge of human sensor characteristics and target type probabilities. The probabilistic model also allows for online Value of Information assessments to automatically query help from the human operator. Practical methods for approximating high-dimensional human sensor parameter posterior distributions via Dirichlet pdf moment-matching and parameter-tying are also developed, and shown to provide significant computational speedups for little/no noticeable information loss. These methods also demonstrate scalability with an increase in possible target types. Results with different combinations of simulated operator profiles and automated fusion system target characterization baselines show that fully Bayesian fusion can simultaneously improve machine-based characterization performance with variable human operator profiles, while accounting for uncertain operator characteristics.

Contents

Chapter

1	Intro	oduction	1
2	Background & Related Work		6
	2.1	Problem Description	6
	2.2	Related Work	7
3	Prol	olem Formation	13
	3.1	Probabilistic Graphical Model	13
	3.2	Bayesian Model Priors and Inference for Probabilistic Data Fusion	15
	3.3	Simulated Target and Ground Truth Data Generation	18
	3.4	Hidden Markov Models for Automated Probabilistic Track Characterization	22
4	Sim	ultaneous Target Characterization and Model Learning	25
	4.1	Human Sensor Parameter Priors and Dirichlet Distributions	25
	4.2	Gibbs Sampling	28
	4.3	Dirichlet Moment Matching for Transference of Learned Human Sensor Models	33
	4.4	Human Sensor Model Parameter Tying	34
5	Valu	ne of Information	36
	5.1	General VOI formulation for decision-making under uncertainty	37
	5.2	VOI querying for human-assisted target track characterization	38

6	Sim	ulation Results & Discussion	44
	6.1	Gibbs sampling inference for simulated operator	45
	6.2	Moment Matching	47
	6.3	Hidden Markov Model	48
	6.4	Inferring Human Sensor Parameters	49
		6.4.1 Naive independent observations model	49
		6.4.2 Markov dependency observation model	52
		6.4.3 Sensitivity to operator reliability/accuracy levels	53
	6.5	Value of Information	57
	6.6	Timing	64
	6.7	Classification	69
		6.7.1 Difference Between Models	69
		6.7.2 Effects of varying operator reliability/accuracy	71
		6.7.3 Sensitivity to number of target types	75
	6.8	Discussion	77
7	Con	clusion	79

Bibliography

81

vi

Tables

6.1	Average pass off frame for individual target types ("expert" operator)	58
6.2	Average pass off frame for individual target types (2nd operator)	59
6.3	Average pass off frame for individual target types (3rd operator)	60
6.4	Average pass off frame for individual target types ("novice" operator)	61
6.5	Average pass off frame for individual target types (10 targets "expert" operator)	62
6.6	Average total computation time in seconds for each simulation method over 100	
	target events. 30 runs	68

Figures

Figure

2.1	Notional operator interface for simulated scenario with synthetic data	12
3.1	Probabilistic graphical plate models: (a) independent model; (b) dependent model	16
3.2	(a) Signal intensity profiles for 'Cirriform' family. (b) Signal intensity profiles 'Cu-	
	mulonibiform' family	19
3.3	(a) Signal intensity profiles for 'Stratiform' family. (b) Signal intensity profiles 'Stra-	
	tocumuliform' family	20
3.4	Sensor signal intensity profiles (mean values and 2σ values shown): (a) $T = 5$; (b)	
	$T = 10. \dots \dots \dots \dots \dots \dots \dots \dots \dots $	21
4.1	Example tri-variate simplex Dirichlet distributions with α parameters (pdf values	
	shown in colorbar)	28
6.1	Signal, autocorrelation, and histogram of parameter-tied samples of θ_2	46
6.2	Moment matching example for: the parameter-tied true positive followed by a second	
	true positive observation; (b) the parameter-tied false positive followed by a true	
	positive observation.	47
6.3	HMM Target classification confusion matrices for (a) 5 targets and (b) 10 targets. $% \left(\left({{{\bf{x}}_{{\rm{s}}}} \right)^2 + \left({{{\bf{x}}_{$	48
6.4	Initial and final distributions for parameter-tied θ_1 values with simulated ground	
	truth values ("novice" operator).	50

6.5	Initial and final distributions for parameter-tied θ_2 values with simulated ground	
	truth values ("expert" operator)	51
6.6	Simulated individual operator confusion matrices: (a) "expert" operator; (b) 2nd	
	operator; (c) 3rd operator; (d) "novice" operator	54
6.7	Initial and final distributions for parameter-tied θ_2 values with simulated ground	
	truth values ("novice" operator).	55
6.8	Initial and final distributions for parameter-tied θ_2 values with simulated ground	
	truth values for 10 target simulation ("expert" operator)	56
6.9	Cumulative average frame where classification is passed from the HMM to the human	
	operator ("expert" operator)	57
6.10	Cumulative average frame where classification is passed from the HMM to the human	
	operator (2nd operator)	58
6.11	Cumulative average frame where classification is passed from the HMM to the human	
	operator (3rd operator).	59
6.12	Cumulative average frame where classification is passed from the HMM to the human	
	operator ("novice" operator).	60
6.13	Cumulative average frame where classification is passed from the HMM to the human	
	operator (10 targets "expert" operator)	61
6.14	Average computation time for independent, parameter tied, and full parameter ob-	
	servation models for 5000 gibbs samples and average computation time for moment	
	matching done after target classification	64
6.15	(a) Average computation time for moment matching when increasing to 10 target	
	types; (b) Average total computation times incorporating the average number of	
	observations required to classify a target and moment matching ("expert" operator).	65
6.16	Average total computation time for independent, parameter tied, and full parameter	
	observation models for 5000 gibbs samples and moment matching done after target	
	classification (a) 2nd operator and (b) 3rd operator.	66

6.17	Average total computation time for independent, parameter tied, and full parameter	
	observation models for 5000 gibbs samples and moment matching done after target	
	classification (a) "novice" operator and (b) 10 targets "expert" operator	67
6.18	Confusion matricies for independent, parameter tied, and full parameter observation	
	models. 30 runs of 100 target events each. "Expert" operator	69
6.19	Precision-Recall curves for independent, parameter tied, and full parameter obser-	
	vation models "expert" operator	70
6.20	Cumulative averages for HMM prior to being passed off to operator (red), HMM	
	without human intervention (purple), independent observations (green), parameter	
	tied observation (blue), and full parameter observations (orange) for (a) "expert"	
	operator; (b) Operator 2	71
6.21	Cumulative averages for HMM prior to being passed off to operator (red), HMM	
	without human intervention (purple), independent observations (green), parameter	
	tied observation (blue), and full parameter observations (orange) for (a) Opeartor 3;	
	(b) "novice" operator.	72
6.22	Confusion matricies for independent, parameter tied, and full parameter observation	
	models. 30 runs of 100 target events each. "Novice" operator	73
6.23	Precision-Recall curves for independent, parameter tied, and full parameter obser-	
	vation models "novice" operator.	74
6.24	Confusion matricies for independent, parameter tied, and full parameter observation	
	models (10 targets). 30 runs of 100 target events each. 10 target "expert" operator.	75
6.25	Cumulative averages for HMM prior to being passed off to operator (red), HMM	
	without human intervention (purple), independent observations (green), parameter	
	tied observation (blue), and full parameter observations (orange) for 10 targets	76
6.26	Precision-Recall curves for independent, parameter tied, and full parameter obser-	
	vation models. 10 targets "expert" operator.	77

Chapter 1

Introduction

As US defense networks continue to expand automated ingestion and processing of high volume remote sensing data, human data analysts and system operators will always need to be kept in the loop, since sophisticated machine learning algorithms for automated event detection, tracking, and data fusion do not perform perfectly in all situations and decisions are ultimately made through human chains of command. A principal challenge is that end users of these automated systems (while generally highly trained and knowledgeable about many aspects of the problem domain and sensing assets) are not themselves algorithm experts or engineers and are unaware of algorithmic limitations. Current systems do not allow users to directly interact with these algorithms to mitigate issues, e.g., by updating prior target location or behavior information online for time-sensitive operations. New technologies that better balance automation (e.g. based on machine learning algorithms for target characterization) and human oversight are needed to exploit the best of both worlds at various levels of the fusion pipeline. In contrast to current state of the art for deployable human-machine systems (where human interaction is very often treated in a post hoc manner), the next generation of automation must opportunistically leverage human reasoning abilities while still ensuring decision-making transparency and performance guarantees.

The goal of this research is to develop new fusion algorithms and interfaces that promote online collaborative human-machine perception for robust data analysis and fusion. The key idea is to allow operators to communicate directly with automated machine learning algorithms via user friendly graphical interfaces for real-time information exchange and data visualization. Specifically, such algorithms and interfaces should (i) let operators voluntarily push new information directly to automation, without exposing its inner workings and while accounting for uncertainties in human reporting; and (ii) let automation actively request useful information from analysts to boost performance via online querying. This allows the operator to act as an additional 'human sensor' capable of providing useful/timely information in situations that automated reasoning is otherwise unable to resolve satisfactorily on its own. These capabilities require a combination of probabilistic machine learning, data fusion, and human input processing algorithms into a unified framework that enables bidirectional information exchange between automation and operators for improved dynamic target analysis and decision-making under uncertainty. By using human reasoning to fill in automated machine perception/reasoning gaps, these efforts aim to measurably improve performance and robustness of state-of-the-art automated detection, tracking, and data fusion algorithms.

This paper looks at problem (i) (information push) in the context of a generic large-scale dynamic multi-target characterization problem, where a single human operator monitors events and objects of interest over a large-scale surveillance area simultaneously scanned by multiple sensing assets in real time. This work also investigates problem (ii) (information pull) where the fusion engine can access the accuracy of both the automation and the human operator in real time to request information in situations where it determines a human sensor would provide benefits over the automation alone. In this general set up, the human operator interacts with an automated data fusion pipeline to supervise the event/object detection, tracking and labeling process (e.g., by confirming/rejecting particular tracks as objects of interest/false alarms or new target types at the 'front end' of the automated pipeline, adjusting sensing parameters in the 'back end', etc.), as well as to make follow-up decision recommendations for specific objects/events of interest in a timely manner. This work focuses on the problem of how target type information provided by the human operator in real-time can be fused with uncertain target type characterization information produced by the automated fusion pipeline. In particular, this work considers how human operators can provide additional information in the form of positive or negative observations to enhance the target type probabilities produced by machine learning algorithms for each tracked object of interest in cases where the number of known possible target types is finite. This work specifically address the problem of modeling the error characteristics of operator observations, while simultaneously fusing these with automated probabilistic assessments of target type for online target characterization. To achieve this goal, this work address three major technical challenges.

Firstly, 'stand-alone' human operator observation error characteristics are generally difficult and cumbersome to parameterize and calibrate **a priori**. For instance, when the space of possible target types is large, then any corresponding parameterization of human observation errors must grow to accommodate all the possible ways the human could mischaracterize multiple similar target types. Furthermore, unlike 'hard' data from conventional sensors (radar, EO/IR, visual, etc.), 'soft' data provided by humans can have highly unusual properties that, despite being information rich, make them difficult to process and combine with other information sources. For instance, error probabilities associated with human observations may not have stationary or even unique statistical properties (even once processed into a standardized format). As such, successive observations that are streamed from a single human are also not expected to be independent and identically distributed (i.i.d.) relative to one another, making error model parameter estimation even more challenging.

Secondly, in real world scenarios, it is common for human reports to be dependent on other information that may already be present/fused elsewhere in a fusion pipeline. Without proper statistical modeling and blending of all information sources, this dependence leads to undesirable double-counting of information. For example, a human operator's target type observations may not be statistically independent of the target type assessments produced by the automated fusion pipeline.

Finally, both the information and time given by human operators are valuable. Having them continuously watch over ("babysit") the automation is not an efficient use of time, while eliminating their input altogether discards useful information. It is desirable to find an effective balance between automation and human reasoning while also accounting for individual differences between operators.

This work addresses these issues by developing a new fully Bayesian probabilistic model for Markovian online human-machine target characterization and data fusion which accounts for statistical dependencies between successive human observations, as well as between probabilistic machine and conditionally dependent human target assessments. This approach uses Dirichlet prior probability distributions to account for the frequency of human responses and to cope with parameter uncertainty due to training/calibration data being sparse in practical settings. It also uses Gibbs sampling-based approximate Bayesian inference to perform post-hoc fusion of probabilistic machine assessments and human target type assessments. The use of Dirichlet priors allows for computationally fast and efficient online Gibbs sampling for simultaneous target type and human parameter inference within the proposed Markovian soft data observation model structure. This further enables easy scaling to scenarios requiring simultaneous error modeling and data fusion with multiple operators. Additionally, parameter-tying significantly reduces the dimensionality of human model parameters; this reduced dimensionality also remains fixed even when increasing the problem size to account for more targets. A Value-of-Information framework is also developed to effectively query human observations to assist the fusion engine in situations where the automation cannot confidently produce a timely and accurate target characterization. Simulation results from the automated fusion target characterization baselines show that fully Bayesian fusion can simultaneously improve machine-based target characterization performance with variable levels of accuracy in human operator profiles, while accurately adapting to uncertain operator characteristics.

This paper is structured as follows. Chapter 2 details the general problem description for multi-target surveillance and tracking as well as related work on probabilistic modeling and fusion of human observations. Chapter 3 presents a framework to model and fuse various sources of information as well as introductions to problem specific setups. Chapter 4 presents the hierarchical Bayesian methodology for simultaneous data fusion and operator characterization problem. Chapter 5 discusses the use of Value-Of-Information and its application in scenarios with uncertain information about the reliability of the proposed data source. Chapter 6 presents our numerical simulation results, and conclusions are given in Chapter 7.

Chapter 2

Background & Related Work

2.1 Problem Description

This work considers how human operators can provide additional information in the form of positive/negative target type observations to enhance the target type probabilities produced by an automated probabilistic machine learning algorithm for each tracked object of interest in scenarios with a finite number of possible target types. In order to have a complete contextual picture for all tracks, the operator can access data associated with each track by visualizing various input/output data layers of the fusion pipeline, from raw sensor data feeds all the way up to automated track generation and labeling. However, for the purposes of the present work, each processing step of the fusion pipeline is effectively treated as a 'black box' whose inner workings cannot be accessed or altered by the operator. This paper considers a case with only a finite number of tracks. Each target track is associated with a track ID, a mean state vector and covariance produced by a tracking Kalman filter. The associated label probabilities are displayed to the operator, and the raw/processed data for those tracks are accessible to the operator.

Figure 2.1 shows a notional example interface built for a simulated tracking scenario with synthetic dynamic target data. The bottom right panel of the interface shows the input panel where operators can provide positive or negative observations to support/reject possible target classifications. Above this input panel is a bar graph representation for the automated pipeline's current probabilistic type assessment of a particular selected target track. The data associated with each target has many different features that an operator can examine through the interface to make their assessment of target type, (e.g. location, velocity profiles, intensity data) although this approach could easily scale to more sensor data and features. For simulation, these features were restricted to only intensity profiles.

Consider the problem of determining the true target type label $X \in \{0, ..., T-1\}$ for a single selected target track over a fixed frame of data gathered by sensors, which is presented to a human operator. Prior to a set of new observations being provided by the human, the automated fusion pipeline produces a 'prior' probability P(X) over the T possible target types using a probabilistic classifier, such as a Hidden Markov Model [5] to associate raw and/or processed data signals obtained from the sensors to the most likely target profile according to some training database (other probabilistic algorithms could be used as well, provided they provide some measure of uncertainty on the final target type labels). Suppose also, the human then provides a series of Nassessment observations $O_1, O_2, ..., O_{N-1}, O_N$ for a given data frame, where for T target types, $O_i \in \{0^+, 1^+, ..., (T-1)^+, 0^-, 1^-, ..., (T-1)^-\}$, where 0^+ means 'selected target is of type 0', $0^$ means 'selected target is not of type 0', 1⁺ means 'selected target is of type 1', 1⁻ means 'selected target track is not of type 1', etc. The human may provide observations at any point they choose, or the automated fusion pipeline may also prompt the human to provide observations at a given data frame. Note that, for simplicity and ease of exposition, clutter/false targets and novel target types are not considered here. However, the Bayesian fusion methodology described in this work can in principle be extended to handle either of these phenomena.

2.2 Related Work

Probabilistic models and Bayesian reasoning provide a powerful general framework for augmenting automated reasoning and perception systems with 'soft data'—observations originating from human sources [14]—to complement 'hard data' from conventional sensors such as lidar, cameras, radar, etc. in partially observable environments. For instance, human pilots/payload specialists in wilderness search and rescue missions can interpret video feeds and electro-optical/IR data streams provided by small unmanned aircraft, and can spot important clues that help narrow down probable lost victim locations and movements [12]. Likewise, in large-scale surveillance for defense applications, dismounted soldiers can provide evidence on the whereabouts and behaviors of potential intruders moving across unsecure areas, which can be fused with hard data from UAV patrols to improve intruder detection and tracking performance [2, 1]. Soft data integration also lets human supervisors of automated systems stay 'in the loop' without overloading them with cognitively demanding dynamic planning tasks [19].

A key problem is: how should soft sensor data be formally integrated with hard data to augment automated algorithms? Soft data can be broadly related to either abstract phenomena that cannot be measured by robotic sensors (e.g. labels for object categories and behaviors) as well as measurable dynamical physical states that must be monitored constantly (object position, velocity, attitude, temperature, size, mass, etc.) [14]. This work focuses on the former, with the key assumption that *humans are not oracles* (where oracles have perfect knowledge). As with any other sensor data, human observations are subject to errors, limitations and ambiguities that must be modeled properly. This work aims to adapt widely used probabilistic sensor fusion, state estimation and machine learning algorithms so that soft data can be exploited with minimal effort on the part of the automated system and the human operator.

Existing approaches to Bayesian modeling and fusion techniques allow human sensors to directly interact [16, 6, 17, 3, 1, 30] with autonomous Bayesian state estimation and perception algorithms for dynamic target and event search, detection, localization, and tracking problems. In the context of target tracking with extended Kalman filters, for instance, ref. [16] assumes that humans provide numerical range and bearing measurement reports ('The target is at range 10 m, bearing 45 degrees'). Ref. [3] showed how to model and fuse flexible semantic natural language soft data to provide a broad range of positive/negative information for Bayesian state estimation, e.g. 'The target is parked near the tree in front of you', 'Nothing is next to the truck heading North'. These fusion algorithms are grounded in probabilistic reasoning and models, and thus enable 'plug and play' functionality—i.e., human sensors can directly plug into filters and algorithms already used by automated systems for state estimation, and automated systems can continue to function as usual even in the absence of human input.

Kaupp et al. [16] and Wang et al. [30] extend the core concept one step further to consider how uncertainties in human sensor model parameters can be accounted for in data fusion using maximum likelihood estimation, whereas Ahmed et al. [1] considers a fully Bayesian reasoning approach that places probabilistic priors on unknown human sensor parameters. Dani et al. [11], Mehta et al. [24], and Bishop et al. [4] have also considered alternative human-machine communication interfaces and probabilistic models for soft data fusion in dynamic target tracking problems. However, a key assumption for data modeling and fusion in all these works is that observations from a single observation remain i.i.d. and are further independent of other available information sources. The approach developed in this work does not assume i.i.d human sensor observations by allowing for the statistical dependencies between successive soft data inputs, thereby limiting the possible double-counting of soft information during the fusion process. Another key difference from the prior work in [1] is that the Bayesian reasoning approach developed here is applied in the context of multi-target characterization (as opposed to single target localization), and further considers how Bayesian inference approximations can be developed to plug into existing online probabilistic machine characterization algorithms.

In addition to formally modeling parameters and uncertainties in human sensors, the problem of queuing and initiating automation requests for human assistance is also considered from a formal Bayesian decision-theoretic perspective. Related recent work in this area can be tied to several different approaches. For example, ref. [29] developed an automatic target recognition (ATR) system and user interface for multi-UAV based surveillance, which provides helicopter command crews with simple symbolic target classification confidence indicators to prompt human input for challenging target tracks. Although simulation studies showed promising results for this simple prompting interface with human crews, the accuracy, reliability and workload of human sensors is not formally taken into account to prioritize ATR assistance requests, or to ensure robust/optimal target type data fusion. Building on the aforementioned applications of dynamic robotic target search and localization Burks, et al. [7, 8] recently developed optimal planning algorithms to enable active semantic sensing in human-autonomy teams via policy approximations for partially observable Markov decision processes (POMDPs). This approach allows queries to human sensors to be treated as explicit information-gathering actions that maximize long-term/non-myopic rewards that are explicitly tied to target localization and capture. Such approaches could be readily adapted to the dynamic target characterization/classification domain. However, due to the nontrivial expense required to obtain POMDP policies via approximate dynamic programming, these approaches currently do not scale well with features such uncertain non-i.i.d. human sensor data.

Earlier work by Kaupp, et al. examined the uncertain decision-making problem of when machine agents should poll human sensors using the concept of Value of Information (VOI) [17]. In VOI-based reasoning, machine agents determine whether querying a particular sensor (or other information source) will result in a better decision to be made by the agent in the long run. This is measured by the expected improvement in utility (i.e. cumulative sum or rewards for different expected problem states), minus the cost of activating the sensor and regardless of the sensor's observation. Whereas POMDP policies 'implicitly' account for VOI through finite/infinite horizon dynamic programming over probabilistic belief spaces, 'explicit' VOI reasoning approaches typically use online optimization and inference to make myopic (one-step horizon) decisions in uncertain contexts, which can also be easily represented by probabilistic graphical models (also known as influence diagrams) to exploit natural conditional dependence/independence relationships between different random variables. In [17], humans were tasked with assisting a simulated autonomous mobile robot equipped with camera and lidar with detecting and classifying movable/immovable obstacles (which can be pushed out of the way) as well as safe navigation routes. A directed probabilistic graphical model relating conditional dependencies between noisy human sensor inputs, robotic sensor data, navigation and querying actions, and overall task rewards/utilities was developed for the task, and combined with exact algorithms for probabilistic inference to enable a VOI-based human sensor polling process, which attempted to minimize the number of times human sensors were queried while maximizing safety and efficiency of robot navigation in a cluttered, uncertain environment. Due to the relatively small scale/size of the problem and the assumption of a small number of well-known human sensor parameters (which were identified and calibrated offline via training data), exact probabilistic inference algorithms could be used to quickly compute the required expectation operations over unknown true obstacle types, navigation states, and possible human sensor responses for VOI assessment in an online manner. However, these assumptions and techniques do not scale well to larger or more complex problem domains, especially when a large number of human sensor parameters is required and when these are highly uncertain or unknown a priori. In particular, exact inference is known to be NP-hard, and is all but intractable to implement for probabilistic models with complex dependencies and uncertain component conditional distributions. Ref. [21] considered an alternative deep supervised learning approach to VOI-based querying to mitigate some of these issues in the context of target search and localization, but requires known human sensor models, large amounts of training data, and is brittle to changes in problem specification. In this paper, computationally efficient approximate algorithms are explored together with hierarchical Bayesian modeling strategies to overcome these limitations.



Figure 2.1: Notional operator interface for simulated scenario with synthetic data.

Chapter 3

Problem Formation

This section describes the formal model and Bayesian data fusion problem for combined human-automation target characterization. Section 3.1 formulates the model for target characterization and human sensor parameter learning in terms of a hierarchical probabilistic graphical model, which is subsequently used in Section 3.2 to derive a Bayesian posterior distribution over the unknown target state for each track considered and unknown human sensor model parameters. Section 3.3 describes the process used for simulating target tracks for characterization, as well as for simulating human sensor inputs for ground truth model validation studies. Details of the hidden Markov model automated probabilistic target characterization approach used in this work, which provides the 'prior' distribution over target types for each track, are given in Chapter 4.4.

3.1 Probabilistic Graphical Model

Let $X^t \in \{0, ..., T-1\}$ be the true target type for track t. For simplicity, X^t will be referred to as X when the context for single track characterization is clear. Although X is unknown, information such as the sensor measurement intensity signature or velocity vectors can inform the likelihood of different targets types. Such data are processed and fed into a automated probabilistic classifier to produce a 'prior' probability P(X) over the T possible target types for a given track t. Any probabilistic machine learning model can be used to produce P(X); as described later, hidden Markov models (HMMs) are used in this work to account for the time dependencies in the processed sensor data and the various target states or 'maneuver stages' commonly seen in tracking scenarios. Further, let $O_k^t \in \{0^+, 1^+, ..., (T-1)^+, 0^-, 1^-, ..., (T-1)^-\}$ (or more simply O_k) be the k-th observed target characterization from the series of positive/negative inputs provided by the human operator for track t, as described in Chapter 2.1. Note that P(X) does not change as the operator provides inputs for track t (i.e. the track is handed off to the operator until the operator releases the track again).

Consider next the probabilistic parameters defining the 'human sensor' model, i.e. the likelihood that the operator provides a positive/negative observation of a given target type. In general, this depends on the true (unknown) target type, as well as on previous human observations. Let $\theta_1 = P(O_k|X)$ be the likelihood distribution for a single independent observation O_k , and $\theta_2 = P(O_{k+1}|O_{k-m:k-1}, X)$ be the likelihood distribution for conditionally dependent observations O_{k+1} that follow after some observation sequence $O_{k-m:k}$, for $m \ge 1$. Note that $\sum_{O_k} \theta_1 = 1$ for any fixed X, and $\sum_{O_{k+1}} \theta_2 = 1$ for any fixed X and $O_{k-m:k}$. The likelihoods θ_1 and θ_2 are modeled here as discrete conditional probability tables (CPTs), though other parametric models could also be used. It is assumed any human operator is completely characterized by their observation likelihood parameters. These must be identified offline or learned online for each operator to account for observation accuracy when performing fusion to update target type probabilities, as well as to determine when the operator should be asked for assistance.

Figure 3.1 shows two different probabilistic graphical models for the fusion problem. The shaded nodes in these figures denote observed random variables or controlled variables, while unshaded nodes represent latent random variables. Superscripted variables within the large rectangle (plate) correspond to a particular target track t, which is to be characterized. Subscripts within the plate denote discrete event/time indices. Variables outside the plate denote common parameters across all target track characterization events, while variables inside the plate represent repeated model structures across all tracks. The models in Figure 3.1 contain blocks representing rewards R (green) and actions A (red) for each track t. The rewards are values earned by the system for correctly classifying targets, where each track is considered in some predetermined ordered sequence. The sole action $A \in \{\text{Ask operator for assistance'}\}$ is based on the current belief in the human

sensor parameters (θ_1 and θ_2). Using these values, the automation can determine whether and at what point to pass track t to the human operator for classification. This has no influence on what the human will eventually say, only that the human will provide observations to assist in classifying the target for track t if the action is taken. The decision making process for asking the operator for assistance is detailed in Chapter 5. For the remainder of this section and in Chapter 6, the focus will be on developing the model and data fusion approximations required to infer X and the human sensor model parameters, assuming observations $\{O_k^t\}_{k=1}^{N^t}$ are already available.

Fig. 3.1 (a) reflects the naive 'independent' observation model most often used in the soft data fusion literature, which ignores dependencies between operator observations and thus only considers $\theta_1 = P(O_k|X)$ for each observation instance k. Fig. 3.1 (b) shows the proposed 'dependent' observation model, which accounts for first-order Markov dependencies between successive operator observations such that m = 1 and $\theta_2 = P(O_k|O_{k-1}, X)$. The dependent model requires more parameters, and introduces dependencies between variables that make probabilistic inference and decision making significantly harder than for the naive independent model. In Chapter 4, the tradeoffs between computational cost and target classification accuracy for both models will be assessed using truth model simulations with synthetic tracking data and different operator models. In the remainder of this section and Chapters 4 and 5, only the 'dependent' model will be considered.

3.2 Bayesian Model Priors and Inference for Probabilistic Data Fusion

The goal of data fusion is to update the prior target type distribution $P(X^t)$ for each track t with operator observations $\{O_k^t\}_{k=1}^{N^t}$, while also accounting for uncertainties in θ_1 and θ_2 . First consider a Bayesian posterior distribution over X, which can be derived from the graph model in Fig. 3.1 (b) assuming fixed θ_1 and θ_2 parameters,

$$P(X|O_{1:N}) = \frac{P(X,O_{1:N})}{P(O_{1:N})} = \frac{P(O_{1:N}|X)P(X)}{P(O_{1:N})}$$
(3.1)

where the term $P(O_{1:N}|X)$ represents the joint data likelihood for the observed human data in the current data frame for track t. Using the first-order Markov dependency property, the joint



Figure 3.1: Probabilistic graphical plate models: (a) independent model; (b) dependent model.

observation likelihood term is

$$P(O_{1:N}|X) = P(O_1|X) \prod_{k=1}^{N-1} P(O_{k+1}|O_k, X) = \theta_1 \prod_{k=1}^{N-1} \theta_2^{k+1|k},$$
(3.2)

where $\theta_2^{k+1|k}$ is the likelihood model parameter which represents $P(O_{k+1}|O_k, X)$. Eq. (3.2) could be used to learn θ_1 and θ_2 parameters via maximum likelihood estimation [5], when X^t is known for a given set of calibration tracks. However, it can be difficult and expensive in practice to obtain a sufficiently large number of observations and ground-truth labeled tracks to obtain accurate point estimates for θ_1 and θ_2 . When X^t is unknown, estimation of θ_1 and θ_2 via maximum likelihood becomes more challenging, since the marginal likelihood function $P(O_{1:N})$ with respect to these parameters is in general non-convex and has no simple closed-form solution. The expectationmaximization (EM) algorithm could be used to alternately infer unknown X^t values and estimate parameters until convergence to a local likelihood maximum [15], but again does not produce reliable results with sparse or limited training sample sizes. To account for the uncertainty in θ_1 and θ_2 in such situations while enabling online inference of X, a fully Bayesian approach is considered instead which seeks a joint posterior distribution over θ_1, θ_2 , and X given $O_{1:N}$.

To this end, suitable prior distributions must be specified over θ_1 and θ_2 to capture the uncertainty in these parameters in light of limited a priori information and/or calibration data. Since the outcome space of the random variables X, O_1 and O_2 are all discrete/categorical in nature, θ_1 and θ_2 can be modeled by conditional probability tables (CPTs) for multinomial distributions. Consider, for instance, $N^t = 2$ observations where $\theta_1 = P(O_1|X)$ and $\theta_2 = P(O_2|O_1, X)$. Therefore, θ_1 and θ_2 are modeled by arrays of positive real numbers whose rows must sum to 1 for a particular configuration of conditioning variables. For instance, θ_1 must be a $2T \times T$ array of such values. Each row j represents the CPT for $P(O_1|X = j)$, and entry (m, j) of θ_1 represents the conditional probability $P(O_1 = m|X = j)$, for $m \in \{1, ..., 2T\}$ (indexed 1-to-1 to the outcome space for O_1) and $j \in \{0, ..., T - 1\}$ (possible true target types). Likewise, θ_2 must be a $2T \times 2T \times T$ array to represent the full CPT for $P(O_2|O_1, X)$. For $N^t > 2$, a time invariant parameter tying assumption is imposed which forces $P(O_k|O_{k-1}, X)$ to be identical for any $k \ge 2$, so that $\theta_2 = \theta_3 = ...\theta_N$, thus greatly reducing the number of parameters required to model the human operator.

If one assumes suitable model parameter prior pdfs $p(\theta_1)$ and $p(\theta_2)$ for the CPTs in θ_1 and θ_2 , then the posterior distribution for fully Bayesian inference over a single target track with the first-order Markov dependent human observation model becomes

$$P(X,\theta_1,\theta_2|O_{1:N}) = \frac{P(X)\theta_1 p(\theta_1) \prod_{k=1}^{N-1} \theta_2^{k+1|k} p(\theta_2)}{P(O_{1:N})}.$$
(3.3)

If this result is marginalized over θ_1 and θ_2 to account for ignorance in the true values of these parameters (which are now treated as random variables), then the posterior Bayesian data fusion update for X becomes

$$P(X|O_{1:N}) = \frac{\int P(X, O_{1:N}, \theta_1, \theta_2) d\theta_1 d\theta_2}{\sum_X \int P(X, O_{1:N}, \theta_1, \theta_2) d\theta_1 d\theta_2}$$
(3.4)
= $\frac{\int P(O_{1:N}|X, \theta_1, \theta_2) P(X) p(\theta_1) p(\theta_2) d\theta_1 d\theta_2}{P(O_{1:N})} = \frac{P(X) \int \theta_1 \prod_{k=1}^{N-1} \theta_2^{k+1|k} p(\theta_1) p(\theta_2) d\theta_1 d\theta_2}{P(O_{1:N})}$ (3.5)

If X and θ_2 are marginalized instead from eq. (3.5),

$$P(\theta_1|O_{1:N}) = \frac{\sum_X \int P(X, O_{1:N}, \theta_1, \theta_2) d\theta_2}{\int \sum_X \int P(X, O_{1:N}, \theta_1, \theta_2) d\theta_1 d\theta_2}$$
(3.6)
= $\frac{\sum_X P(X) \int P(O_{1:N}|X, \theta_1, \theta_2) p(\theta_1) p(\theta_2) d\theta_2}{P(O_{1:N})} = \frac{\sum_X P(X) \int \theta_1 \prod_{k=1}^{N-1} \theta_2^{k+1|k} p(\theta_1) d\theta_2}{P(O_{1:N})}.$

(3.7)

Likewise, if X and θ_1 are marginalized from eq. (3.5),

$$P(\theta_2|O_{1:N}) = \frac{\sum_X \int P(X, O_{1:N}, \theta_1, \theta_2) d\theta_1}{\int \sum_X \int P(X, O_{1:N}, \theta_1, \theta_2) d\theta_1 d\theta_2}$$
(3.8)
$$= \frac{\sum_X P(X) \int P(O_{1:N}|X, \theta_1, \theta_2) p(\theta_1) p(\theta_2) d\theta_1}{P(O_{1:N})} = \frac{\sum_X P(X) \int \theta_1 \prod_{k=1}^{N-1} \theta_2^{k+1|k} p(\theta_1) d\theta_1}{P(O_{1:N})}.$$
(3.9)

The marginal posteriors in eqs. (3.7) and (3.9) allow for display/assessment of operator reliability and performance through interpretation of the statistics of θ_1 and θ_2 (the joint posterior for θ_1 and θ_2 can also be obtained, more generally). In particular, these distributions carry information about the operator's true positive, false positive, true negative and false negative rates for each possible target type.

Note that $P(X, \theta_1, \theta_2 | O_{1:N})$ is in general analytically intractable, due to the complex conditional dependencies between the human sensor observation sequence, the unknown prior model parameters, and target type X. These dependencies are also evident from the 'd-connectedness' [27] of these variables in probabilistic graphical model in Fig. 3.1 (b), since X and the θ parameters share $O_{1:N}$ variables as children. However, as will be discussed in Chapter 4.2, computationally efficient approximations to these posterior distributions can nevertheless be obtained via suitable approximate inference algorithms that exploit local conditional dependencies within the graphical model in Fig. 3.1 (b). Unlike maximum likelihood point estimation, the (approximate) posterior distribution for X, θ_1 , and θ_2 provide the full landscape of possible outcomes that jointly 'explain' $O_{1:N}$, and can thus avoid getting trapped by local likelihood maxima. Furthermore, since Bayesian inference inherently supports recursive data fusion even with hierarchical variable dependencies [23], it is well-suited to online processing of new streams of soft human observation data for multiple target tracks.

3.3 Simulated Target and Ground Truth Data Generation

Synthetic tracking data was generated to approximate track detection and dynamic target profiles seen in real world application scenarios involving space-based tracking and characterization.



Figure 3.2: (a) Signal intensity profiles for 'Cirriform' family. (b) Signal intensity profiles 'Cumulonibiform' family.

These data sets were used to train the probabilistic classification algorithm described in Chapter 3.4, and will be used in Chapter VI to assess the probabilistic modeling and inference techniques to be developed in Chapters IV and V. In order for the data to remain unrestricted and publishable, units, exact physical specifications, and specific target types are not assigned.

Synthetic tracking data was generated to approximate track detection and dynamic target profiles seen in a real world application scenario. Though discussions with industry contacts, it was expressed that target tracking scenarios often involve targets that are similar to one another based on the country of origin or by the overall type of target. While the exact information is speculation, this lead to the creation of five 'families' of targets. Each family contains five more profiles that provide slight differences from one another both in intensity and velocity profiles referred to as 'genuses'. Four of the families are shown in Figures 3.2 and 3.3.

For this work, data sets for T = 5 similar target type track profiles were generated in order to model scenarios where the automation would require human assistance. Five typical signal return intensity profiles are shown in Figure 3.4 (a) each with four maneuvering states e.g. take off, staging, etc. While the average profiles are simple to differentiate, real scenarios involve sensors that must contend with atmospheric noise. In addition to the basic intensity profile shapes, white Gaussian



Figure 3.3: (a) Signal intensity profiles for 'Stratiform' family. (b) Signal intensity profiles 'Stratocumuliform' family.

noise is added as well as signal attenuation due to external/uncontrollable environmental effects on the sensors, e.g. due to weather. The introduction of noise makes these targets considerably harder to classify. In scenarios where targets have not been seen frequently or where additive noise can make differentiation difficult, contextual information from a human operator proves necessary for clarification. To demonstrate scalability of the proposed model and inference approximations to larger scale tracking scenarios, 5 additional target profiles were also generated, as shown in Figure 3.4 (b) for a total of T = 10 types. The original five targets from Figure 3.4 (a) are the same, but the new target types are designed to provide an even larger amount of overlap, giving both the automation and the human operator a more challenging characterization scenario. Both scenarios contain the same amount of Gaussian noise and atmospheric attenuation both during training and testing.

Note that, in addition to the generation of data for each of the different target types, data for the human operator observations can also be generated to simulate instances of the simultaneous data fusion and operator characterization problem using 'ground truth' human sensor models. To this end, CPTs can be created for both θ_1 and θ_2 and treated as ground truth, along with the true target types for simulated target track profiles that are processed by the probabilistic classification



Figure 3.4: Sensor signal intensity profiles (mean values and 2σ values shown): (a) T = 5; (b) T = 10.

algorithm which produces P(X). When a target type is randomly chosen for a ground truth simulation, a sequence of observations form the human can then also generated from the associated probability vectors in the CPTs. This methodology makes a few assumptions. First, observations are generated according to the same model proposed in Figure 3.1 (b) for human operators. Second, likelihood values are assumed for what a human operator might say under similar circumstances. As mentioned earlier, since it is assumed that no clutter targets or new/unknown target types exist (i.e. no targets from outside T = 5/10 bounded target types), this leads to a finite number of possible observations to model for any given operator. This method's main advantage is the ability to compare the 'ground truth' probability vectors (i.e. the actual θ_1 and θ_2 values that generated $O_{1:N}$ observation data) to the θ_1 and θ_2 values inferred from observation data via Bayesian data fusion. This ground truth data generation and testing technique is used in Chapter 6 to assess and compare the performance of the proposed models and inference approximations under known operating conditions with varying amounts of data that are inexpensive to obtain.

3.4 Hidden Markov Models for Automated Probabilistic Track Characterization

A hidden Markov model (HMM) is used in this work to provide the 'prior' distribution over target types P(X) for each track t. While the proposed Bayesian data fusion framework allows for any machine-learning algorithm to provide the prior distribution, HMMs provide a convenient choice here for several reasons. Firstly, HMMs are well-established for classification of time series data, which may be described by continuous or discrete features [28]. Secondly, HMMs naturally provide probabilities that can be used to model independent priors for target characterization before human operator data is provided. Thirdly, HMMs provide a convenient theoretical abstraction for representing the outputs of other memory-based probabilistic time-series classifiers, e.g. long-short term memory deep neural networks [26]. As such, it is assumed that a target type probability vector P(X) associated with a particular target track t is naturally produced by the automated machine classification element of the tracking and data fusion pipeline (using processed sensor signal data and other low-level tracking data), and is recursively updated by this element until the track is 'handed off' to the human operator.

In this work, the synthetic track data described in Chapter 3.3 are used to train an HMM for each possible target type $X \in \{0, ..., T-1\}$. Let S_j^X represent a discrete (integer valued) target type-dependent latent target signal state at discrete time j (e.g. which can correspond to target take off events, staging events, separation/maneuver events, etc.), and Y_j represent a vector of sensed noisy target signal values processed by the fusion pipeline at time j (signal intensity, raw position return, raw velocity return, etc.). Note that, given a particular target type, the associated signal state S_j^X for that target type encodes memory that can be used to predict the values Y_j . Thus, different target types X need not have the same number of possible signal states S_j^X , and need not have the same physical interpretations for S_i^X (if any).

Given a set of processed sensor signals $Y_j\}_{j=1}^{Z^t}$ for track t and a hypothesized target track type X, the HMM forward-backward algorithm can be used to recursively estimate and update the current signal state probability distribution online, based on the probability of transitioning from one signal state to another $P_X(S_j^X|S_{j-1}^X)$ and the likelihood $P_X(Y_j|S_j^X)$ of seeing the observed signal features Y_j given a hypothesized signal state S_j^X and target type X [28],

$$P_X(S_j^X, Y_{1:j}) = \alpha_X(S_j^X) = p_X(Y_j|S_j^X) \sum_{\substack{S_{j-1}^X\\j=1}} P_X(S_j^X|S_{j-1}^X) \alpha_X(S_{j-1}^X).$$
(3.10)

The state transition distribution $P_X(S_j^X|S_{j-1}^X)$ and observation likelihood functions $P_X(Y_j|S_j^X)$ for each target type X can be easily estimated from training data via maximum likelihood, if the true target types are known for each track t in the training data set. However, in the data fusion problem, the target signal state S_j^X at a particular time (sensing frame) j is not the primary object of interest. Rather, S_j^X is instrumental in computing the actual quantity of interest, namely likelihood $P_X(Y_{1:Z^t})$ that the associated target signal sequence $\{Y_j\}_{j=1}^{Z^t}$ for track t belongs to a particular target type X,

$$P_X(Y_{1:Z^t}) = \sum_{\substack{S_{Z^t}^X \\ Z^t}} P_X(S_{Z^t}^X, Y_{1:Z^t}) = \sum_{\substack{S_{Z^t}^X \\ Z^t}} \alpha_X(S_j^X).$$
(3.11)

This likelihood in turn can then be normalized and compared across all target types X via Bayes' rule to produce P(X) for track t, after all z^t frames have been processed by the HMM for each target type (assuming uniform priors for X), such that

$$P(X) \equiv P(X|Y_{1:z^t}) \propto P_X(Y_{1:Z^t}).$$
(3.12)

To prevent numerical underflow for long data sequences $Y_{1:Z^t}$, normalization to produce P(X) is accomplished using a combination of the 'normalized' forward-backward algorithm [5] and log-sum of exponentials renormalization [22]. This technique can also be implemented in an online manner such that P(X) is recursively updated as new sensor inputs Y_j are processed for track t. The resulting P(X) distribution can then be used to classify targets if any particular target hypothesis is distinct enough from others (e.g. past a certain confidence threshold), or else it can be used to decide when track t should be passed to an operator to further resolve the target type hypotheses. In the latter case, P(X) acts as the 'prior' for the probabilistic graphical model for each track t to ensure statistically consistent fusion between machine-processed sensor signal data $Y_{1:Z^t}$ and soft human sensor observations $O_{1:N^t}$. For the purposes of this work, the HMMs for each target type were only trained with $Y_{1:Z^t}$ representing simulated intensity measurements (c.f. Fig. 3.4), but other signal features such as position, velocity, etc. could also be used more generally. Detailed baseline HMM classification rates for each target type are provided in Chapter 4, where these are also compared to classification performance with fused human operator inputs.

Chapter 4

Simultaneous Target Characterization and Model Learning

This chapter describes the details behind the Bayesian modeling and approximate inference approach for performing online recursive fusion of observed operator inputs $O_{1:N^t}$ for each target track t, assuming that the prior P(X) has been generated by the HMM-based automated classification algorithm prior to hand off to the operator and that the joint posterior distribution $P(X, \theta_1, \theta_2 | O_{1:N^t})$ is desired. To this end, four specific technical issues are motivated and addressed in turn: specification of the CPT parameter priors $p(\theta_1)$ and $p(\theta_2)$ (Section 4.1); approximate inference to obtain $P(X, \theta_1, \theta_2 | O_{1:N^t})$ in an online manner (Section 4.2); approximation of the marginal posterior parameter pdfs $p(\theta_1 | O_{1:N^t})$ and $p(\theta_2 | O_{1:N^t})$ to 'reset' the human sensor parameter priors $p(\theta_1)$ and $p(\theta_2)$ for each new target track t + 1 following soft data fusion and human parameter learning for tracks 1 : t (Section 4.3); and human sensor parameter set reduction for efficient computation (Section 4.4).

4.1 Human Sensor Parameter Priors and Dirichlet Distributions

Recall that the θ_1 and θ_2 parameters correspond to the values for the CPTs describing $P(O_1|X)$ and $P(O_{k+1}|O_k, X)$ for $k \ge 1$, and that each row of a CPT is a probability vector of nonnegative real values that must sum to 1. A natural family of parameter prior probability density functions to consider in this case for defining $p(\theta_1)$ and $p(\theta_2)$ corresponds to the well-known Dirichlet distribution, which is a member of the so-called exponential family of distributions. As will be shown in Secs. IV.B and IV.C, the Dirichlet pdf provides several nice features for approximating the joint posterior distribution over X, θ_1 and θ_2 , as well as for inducing 'updated priors' that embed information learned about human operator sensor parameters whenever new target tracks are processed.

The Dirichlet distribution, also known as the multivariate Beta distribution, is a distribution over a probability simplex of dimension K-1, where K is the number of discrete exclusive outcomes being modeled by an unknown K-dimensional probability vector $\boldsymbol{\mu} = [\mu_1, ..., \mu_K]^T$ [5],

$$p(\boldsymbol{\mu}|\boldsymbol{\alpha}) = Dir(\boldsymbol{\mu}|\boldsymbol{\alpha}) = \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_1)\dots\Gamma(\alpha_K)} \prod_{k=1}^K \mu_k^{\alpha_k - 1}, \quad \text{where} \quad \sum_k \mu_k = 1$$
(4.1)

where Γ is the gamma function, $\boldsymbol{\alpha} = [\alpha_1, ..., \alpha_K]^T$ is the so-called 'concentration' or 'pseudo-count' hyperparameter vector of the distribution, $\alpha_0 = \sum_{i=1:K} \alpha_i$ and the unknown probabilities $\mu_i \ge 0$ for all categories *i* must sum to 1.

An example of a 3-category simplex is shown in Figure 4.1. Note that a sample realization μ drawn from the Dirichlet pdf represents a K-dimensional probability vector. The distribution can be completely described by the α hyperparameter. As the α_i elements change, so does the pdf in the direction of the largest α_i parameters. This essentially reallocates a portion of the probability mass to category *i*, such that μ still remains a proper probability vector. Note that the 3 axis do not correspond directly to each dimension of the α hyperparameter. Instead α corresponds to each of the 3 points on the displayed triangle. An increase in the corresponding α_i reallocates probability mass towards that corner proportional to the ratio of all α_i values.

While a 3-category Dirichlet pdf can easily be shown in a 2D model, the human-automation fusion problem considered here requires much higher dimensionality. In particular, the CPTs for $P(O_1|X)$ and $P(O_{k+1}|O_k, X)$ require a 10-dimensional or 20-dimensional Dirichlet per X and O_k configuration, i.e. to describe all possible positive and negative observations about T = 5 or T = 10target types under all possible conditioning configurations. Therefore, a total of $2T^2$ Dirichlet pdfs of dimension K = 2T are required under this prior pdf model for $p(\theta_2)$, while 2T Dirichlet pdfs of dimension K = 2T are required for $p(\theta_1)$. The scalability of the Dirichlet prior model will be
revisited in Chapter 4.4 in the context of improving computational efficiency for inference and human sensor parameter learning.

The Dirichlet prior is sufficiently flexible to model and encode prior information about human sensor responses over a wide range of important operator characteristics. For example, α_i values can be selected to encode whether particular operators are inconsistent or consistent in their responses, leading to a wider/narrower concentration of the pdf for different target types X and prior responses O_k . The hyperparameters can also encode whether operators generally provide reliable or unreliable observations, leading to larger/smaller priors over corresponding θ values for correct/incorrect target characterization responses O_1 and O_{k+1} given X.

The hierarchical Bayesian modeling approach also naturally allows for observation data provided by operators to be leveraged in learning θ_1 and θ_2 via Bayesian inference. To this end, one particularly convenient property of the Dirichlet distribution is that multiplication of the Dirichlet parameter priors $p(\theta_1)$ and $p(\theta_2)$ by the multinomial CPTs $P(O_1|X)$ and $P(O_{k+1}|O_k, X)$, respectively, returns Dirichlet posterior pdfs over θ_1 and θ_2 given X and $O_{1:N^t}$,

$$p(\theta_{1}|X, O_{1}) \propto p(\theta_{1})P(O_{1}|X) \propto Dir(\theta_{1}|\boldsymbol{\alpha^{1-}})P(O_{1}|X) \propto Dir(\theta_{1}|\boldsymbol{\alpha^{1+}}),$$
(4.2)
$$p(\theta_{2}|X, O_{2:N^{t}}) \propto p(\theta_{2}) \prod_{k=1}^{N^{t}-1} P(O_{k+1}O_{k}|X) \propto Dir(\theta_{1}|\boldsymbol{\alpha^{2-}}) \prod_{k=1}^{N^{t}-1} P(O_{k+1}O_{k}|X) \propto Dir(\theta_{2}|\boldsymbol{\alpha^{2+}}),$$
(4.3)

where α^{1-} and α^{2-} denote the prior Dirichlet hyperparameters, and α^{1+} and α^{2+} denote the posterior Dirichlet hyperparameters. This property makes the Dirichlet a conjugate prior for multinomial CPTs. This in turn leads to the 'pseudo-count' interpretation of the α hyperparameters, which can be directly updated from $\alpha^{1-} \rightarrow \alpha^{1+}$ by observed positive/negative target type counts in O_1 when X is known (and likewise for $\alpha^{2-} \rightarrow \alpha^{2+}$ with respect to the observations in $O_{2:N^t}$ when X is known- see [5]).

Note that the conjugacy property does **not** extend to the marginal posterior pdfs in eqs. (3.7) and (3.9), which in general have much more complex representation due to conditional dependencies. Furthermore, the dependencies between observations in the probabilistic graphical model imply that



Figure 4.1: Example tri-variate simplex Dirichlet distributions with α parameters (pdf values shown in colorbar).

the marginal posterior distribution for X in eq. (3.5) also has dependencies on θ_1 and θ_2 (as well as the X values for other targets). However, from the same graphical model, it is straightforward to find the posterior of θ_1 and θ_2 given X and $O_{1:N}$ in closed-form when Dirichlet priors are used; likewise, the posterior of X given θ_1 , θ_2 and $O_{1:N}$ can also be obtained in closed form. These considerations naturally lend themselves to efficient Markov Chain Monte Carlo sampling-based inference approximations for estimating the left-hand side of eq.(3.3), thus providing a tractable mechanism for solving the simultaneous data fusion and operator characterization problem.

4.2 Gibbs Sampling

Recall that simultaneous data fusion for inference of the target type X and operator sensing characteristics θ_1 and θ_2 given $O_{1:N^t}$ requires approximation of eq. (3.3), the joint posterior distribution of these variables. Note that although this distribution is analytically intractable (and hence cannot be sampled 'directly'), it can be evaluated point-wise (i.e. up to the normalizing constant given by $P(O_{1:N^t})$ at any given configuration of X, θ_1 , and θ_2 when P(X), $p(\theta_1)$ and $p(\theta_2)$ are specified as multinomial and Dirichlet pdfs, respectively, and when the sequence of human sensor observations $O_{1:N^t}$ is known. This means that samples can be drawn from eq. (3.3) via 'indirect' Monte Carlo techniques that exploit the ability to evaluate the posterior in a point-wise manner [9].

A Gibbs sampling Monte Carlo approximation is employed to this end. Gibbs sampling is

a form of sequential Markov Chain Monte Carlo (MCMC) sampling. In MCMC, samples from a proposal distribution are constructed based on the current sample state (i.e. where each sample in a sequence is dependent on the previous sample), so that the resulting Markov chain which generates the sequence has the desired posterior distribution (from which i.i.d. samples are to be drawn) as its stationary distribution. This can be ensured, for instance, via the well-known Metropolis-Hastings algorithm, which assesses the probability of accepting or rejecting a proposed random move from the current sample state to a new sample state according to relative likelihoods given by a proposal distribution and the desired posterior (both of which only need to be evaluated up to their respective normalizing constants). Gibbs sampling is a special case of Metropolis-Hastings MCMC for multivariate distributions, in which samples from the proposal distribution are always accepted and never rejected, since the proposal distribution is constructed from lower-dimensional 'local conditional' distributions relative to the desired full posterior distribution. As such, Gibbs sampling is highly efficient compared to other MCMC approaches, where high sample rejection rates lead to slow convergence. A complete overview and derivation of MCMC and Gibbs sampling can be found in [5, 20, 9].

The main idea behind Gibbs sampling is to perform conditional simulation of the unobserved variables in Figure 3.1. Note that, since all target observations across all tracks are conditionally dependent on the same unknown operator parameters θ_1 and θ_2 , it follows that the unknown target types X are also conditionally dependent across all frames (owing to the d-separation property of the graphical model). This dependency would typically require that all target IDs be inferred and updated simultaneously along with the unknown operator parameters to obtain the most accurate estimates for the target types **given all available operator data**, even when using an approximate inference method. This requirement is relaxed so that the target type posteriors for target t and operator parameter posteriors only need to be obtained given all information available up to the point at which target t is being processed. That is, after processing target t - 1 after processing target t. However, it is desired to carry forward new information about the operator in the form of

updated beliefs about θ_1 and θ_2 parameters after processing target t - 1, so that this is a suitable prior over θ_1 and θ_2 when target t is processed (and likewise as we go from target t to t + 1, etc.).

As such, to simplify the inference process, the sampling estimates are implemented moving only forward in time to 'filter' observations $O_{1:N^t}$ as they are received for track t. In other words, each target is processed one at a time in a pre-determined sequence, such that target type probabilities P(X) are updated only once as they are accessed in sequence, and such that operator parameters are always updated following each target processing frame. Chapter 4.3 describes a procedure for carrying forward the information obtained via the approximate posteriors over θ_1 and θ_2 from target t to the priors over θ_1 and θ_2 for target t + 1. This process only relies on the output of Gibbs sampling for each target, and thus does not alter its implementation for the single target case.

Next, consider the iterative conditional probability distributions needed for Gibbs sampling updates over a given 'frame' of N^t human sensor observations for a single target track t, $O_{1:N^t}$. Assume that this target has prior P(X), while the operator has Dirichlet priors $p(\theta_1)$ and $p(\theta_2)$ at this point. Then, for sampling iterations $i = 1, \dots, N_S$, samples X^{i+1} , θ_1^{i+1} and θ_2^{i+1} are alternately drawn from the conditional posteriors,

$$X^{i} \sim P(X|O_{1:N}, \theta_{1}^{\text{curr}}, \theta_{2}^{\text{curr}}) \propto P(X)\theta_{1}^{\text{curr}} \prod_{j=1}^{N-1} \theta_{2}^{\text{curr}}$$

$$(4.4)$$

$$\theta_1^i \sim P(\theta_1 | O_{1:N}, X^{\text{curr}}) \propto \theta_1 p(\theta_1) = p(O_1 | X^{\text{curr}}, \theta_1) p(\theta_1), \tag{4.5}$$

$$\theta_2^i \sim P(\theta_2 | O_{1:N}, \theta_1^{\text{curr}}, X^{\text{curr}}) \propto \prod_{k=1}^{N^t - 1} \theta_2^{k+1|k} p(\theta_2) = \prod_{k=1}^{N^t - 1} p(O_{k+1} | O_k, X^{\text{curr}}, \theta_1^{\text{curr}}, \theta_2) p(\theta_2), \quad (4.6)$$

where θ_1^{curr} , θ_2^{curr} , and X^{curr} are the "current" (most recent) values of θ_1 , θ_2 , and X (prior to each being resampled). Although X, θ_1 , and θ_2 are conditionally dependent on one another, in Gibbs sampling each unknown variable is sampled while holding the other unknown variables constant. After a sufficient number of samples N_S has been obtained, the posterior distribution for the target type X and each element of the CPTs for θ_1 and θ_2 can be obtained empirically

$$p(X,\theta_1,\theta_2|O_{1:N^t}) \approx \frac{1}{N_s} \sum_{i=1}^{N_s} \delta([X^i,\theta_1^i,\theta_2^i], [X,\theta_1,\theta_2]).$$
(4.7)

This approximation can be obtained following a sufficient "burn-in" period to allow the underlying MCMC chain to mix, and only retaining samples after every few iterations to avoid biasing the samples due to their sequential dependencies.

An example of how this works in practice with the Dirichlet distributions is as follows. The prior distribution P(X) over target types is first presented to the operator for a given target track (as constructed from the HMM for machine-only automated data sequence classification). The operator provides a first observation about one of the target types. The posterior for each target type (i.e. for each of the T realizations of X) is computed using a sample parameter value θ_1^{CUTT} taken from the prior Dirichlet distributions over θ_1 via equation (4.4), without the use of θ_2 due to the single observation provided. The posterior over target types are then normalized and temporarily stored. A sample is then taken from this normalized posterior to instantiate a new value for X^{CUTT} . Given X^{CUTT} and the previous observation O_1 , the corresponding hyperparameters α^1 describing the Dirichlet distribution for the appropriate portions of θ_1 (i.e. the part of the CPT consistent with the observed human observations and sampled target type X^{CUTT}) are updated. A random sample from this Dirichlet pdf is then taken and stored; this serves as the new θ_1^{CUTT} value (i.e. updated CPT for the observation likelihood) for the next sampling iteration, which resamples the current value of X after re-computing the posterior in eq. (4.4) and so on.

Continuing the example, next suppose a new observation O_2 is obtained; the previously computed distribution over target type X is then reset to the prior p(X) and a new posterior for X is computed with the new sampled θ_1 value. In turn the hyperparameters α^1 are also reset during each iteration before re-sampling θ_1 with the new sampled X value. A third step is then added to the sampling process, by using the prior Dirichlet pdf over θ_2 to sample θ_2^{curr} . Then, eq. (4.4) is used once again to sample a new X^{curr} . Just as before, a sampled X value serves as the 'current" value, and then both θ_1 and θ_2 are sampled via eqs. (4.5)-(4.6) according to the available observations O_2 and O_1 , and their sample values stored. The sampling cycle then repeats starting with eq. (3.3). The same logic is then applied as new observations O_3 , ... O_{N^t} are obtained for target track t, until N_s samples have been obtained. The stored target type samples and marginal posterior distributions (which can be formed by storing and averaging the sampling distributions in eq.(4.4)) can be displayed to the operator to make further decisions for track t. As described next in Chapter 4.3, the θ_1 and θ_2 samples can be used to form new priors for the unknown operator parameters in fusion updates for the next track t + 1 to be processed.

A few additional remarks on the Gibbs sampling approximate inference strategy are in order. Firstly, it may be pointed out that Gibbs sampling is nominally a computationally expensive process that is not typically used for online estimation applications. However, for the motivating application considered here, target track information would be presented to the operator on the order of tens of seconds. This not only permits adequate time for operators to provide additional track information, but also permits the use of more sophisticated inference approximations and computing capabilities. In addition, the conjugacy property of the Dirichlet pdf ensures that the conditional posteriors in eqs.(4.4)-(4.6) can be computed and sampled easily for Gibbs updates, and that information about each individual operator can be efficiently summarized between target track updates using sufficient statistics (see moment matching section below). Modeling the CPT priors with Dirichlet distributions thus allows the otherwise intractable simultaneous fusion and operator characterization problem to be solved for each individual operator in an online manner, while providing Bayesian updates for target classification.

Secondly, Dirichlet distribution allows for sampling over parameters that are not directly observed. Altering the α_i hyperparameter values for a single category of a Dirichlet distribution changes each corresponding marginal Beta distribution that comprises the Dirichlet. When an observation is given, not only is that parameter re-sampled, but all parameters describing that event are as well. For example, the row corresponding to X = 1 and $O_{n-1} = 2$ is a CPT with a Dirichlet prior where a single α_i value could be altered by conditioning on X and O_{n-1} , but all 2T CPT parameters are re-sampled on each Gibbs iteration. This maintains the benefits of modeling the uncertainty CPT probability vectors, while also improving the ability to explore the full parameter space on each Gibbs iteration.

Finally, another aspect to consider in the context of Bayesian data fusion is the possibility of

'double-counting' information obtained from human observations $O_{1:N}$ when updating P(X). To avoid this, every time a new human observation O_k is given for track t, the whole approximate inference computation must be restarted from the same priors P(X), $p(\theta_1)$, and $p(\theta_2)$ for track t. This is done until the approximate posterior target type probability $P(X|O_{1:N^t})$ (estimated via trivial marginalization of eq. (4.7)) reaches a certain threshold to be 'released'. At this point, the final approximate posterior distributions for θ_1 and θ_2 (also estimated via trivial marginalization of eq. (4.7)) can be used to 'bootstrap' the priors $p(\theta_1)$ and $p(\theta_2)$ for the next track t + 1 to be processed.

4.3 Dirichlet Moment Matching for Transference of Learned Human Sensor Models

The samples generated by the Gibbs sampler results in N_s values of θ_1 and θ_2 drawn from intermediate Dirichlet pdfs (where one Dirichlet pdf is present for each CPT in the observation likelihood model). Each set of samples model the posterior Dirichlet distributions to be used for subsequent event classifications. One of the major advantages of modeling conditional probability vectors as Dirichlet distributions is that they are closed form. In order to retain the closed-form solution and ability to sample from Dirichlet distributions on processing of subsequent target tracks with the same human operator, Dirichlet pdf moment matching is employed [25] to transform the posterior in eqs. (3.7) and (3.9) into updated prior pdfs $p(\theta_1)$ and $p(\theta_2)$ for the next target to be processed. A Dirichlet distribution can be completely described by its hyper parameter values α . In order to accurately model samples, the maximum likelihood estimate of α must also maximize $l_{Dir}(\boldsymbol{\mu}|\boldsymbol{\alpha}) = \prod_i p(\mu_i|\alpha_i)$. Given an initial vector of $\boldsymbol{\alpha}$'s (prior), one can find a fixed point solution by iteratively solving the following coupled nonlinear equations where $\boldsymbol{\psi}$ represents the digamma function, $p_{\mu k}$ represents the kth sampled probability, and N_s represents the number of samples.

$$\boldsymbol{\psi}(\alpha_i^{new}) = \boldsymbol{\psi}(\sum_i \alpha_i^{old}) + \log \bar{\mu}_i \tag{4.8}$$

where
$$\log \bar{\mu}_i = \frac{1}{N} \sum_{k}^{N_s} \log p_{\mu k}$$
 (4.9)

In this problem, there are 2T categories (positive and negative for each target type). In addition, the number of samples will be the same for all categories of the distribution as detailed in the Gibbs sampling section. Solving for α^{new} for each category gives the posterior Dirichlet distribution in closed form that is completely described by the α hyperparameters. These are then used for the next target processing frame as the priors over distributions $p(\theta_1)$ and $p(\theta_2)$.

4.4 Human Sensor Model Parameter Tying

The probabilistic model in Figure 3.1 (b) attempts to explicitly account for successive dependencies between human operator observations, rather than assuming these observations are independent. One major issue is that this generally leads to an exponentially larger number of parameters to consider as more observations are given about the same object. For example, with T = 5 target types and 2T possible positive/negative observations per target type, each observation increases the number of required parameters by a factor of 10, with 50 parameters on the 1st observation, 500 on the 2nd, 5000 on the 3rd etc. Even in the simplified time-invariant and trackinvariant first order Markov dependence model proposed, one must keep track of two consecutive observations, giving 500 parameters to infer for T = 5. In general, the total number of parameters to estimate in θ_1 and θ_2 (without the first order Markov dependency assumption) for T possible target types is $K = 2T + T(2T)^{|O|}$, where |O| is the maximum number of observations provided about any target. As stated previously, the dependencies of observations are cut off per the first order Markov dependence assumption, which leads to only $K = 2T + 2T^3$ parameters to consider for any number of observations provided by the operator.

This still presents two problems: 1) it quickly becomes difficult to infer parameters for observations that may never be seen in training data or deployment; and 2) approximate inference can become computationally infeasible. To reduce the problem dimension, an additional parameter tying assumption is used, which considers four basic types of human observations relative to the true (unknown) target types: true positive, false negative, false positive, and true negative. This constraint leads to only 4 parameters for θ_1 and $4^2 = 16$ total parameters for θ_2 (true negative followed by false positive, true negative followed by true negative, etc.), regardless of how many target types T are possible. Note that this also drastically reduces the dimensions of the Dirichlet pdf priors and posteriors for Bayesian inference. As a result, the mixing time and efficiency of Gibbs sampling are also greatly enhanced for approximate inference of X, θ_1 and θ_2 .

However, this approach does lose information about interdependencies between target types and human accuracy. For example, if target type 1 has a very similar profile to target type 4 while all other target profiles looked very different from each other, the probability of an operator providing the same observations for tracks of types 1 and 4 would be no different than for tracks 2,3 and 5. However, a reasonably good automated probabilistic classification algorithm will be able to recognize significant differences in target's intensity profiles, velocity profiles, etc. – but when two target types are difficult to classify, other contextual information available to a human operator would allow them to assist. Since this type of information is hard to account for and calibrate a priori, the use of Dirichlet priors allows us to capture the uncertainty in the associated human observation likelihood models. This also has the added benefit of allowing Gibbs sampling to converge much more quickly for simultaneous Bayesian data fusion and online operator parameter identification, since now a much lower dimensional parameter space needs to be explored in conjunction with all possible target types. This in turn makes online implementation of Gibbs sampling even easier in practice.

Chapter 5

Value of Information

To this point, the only problems that have been considered are approximation of the joint posterior $P(X, \theta_1, \theta_2 | O_{1:N^t})$ for a given target track t and bootstrapping of the human sensor model parameter priors $p(\theta_1)$ and $p(\theta_2)$ for next target track t+1 using approximate posterior information derived from processing of the previous target track t. These problems were addressed assuming that the automation has already decided to 'hand off' the target track to the operator for assistance, until a certain confidence threshold in the target type has been reached. This chapter now considers the problem of how the automation can optimally decide which tracks (out of a given sequence of target tracks to be processed) to hand off to the operator by querying them for input, and at what point any of these tracks should be handed off (i.e. after how many y_k sensor data frames per track), while also accounting for uncertainty in operator sensing characteristics.

To address this problem, a Value of Information (VOI) approach is considered. As discussed in Chapter 2, the use of VOI in querying human operators when the benefit of observation exceeds a certain cost has proven to be successful in other information gathering applications centered on human-machine teams [17, 21], where machine agents can encounter difficult decision-making scenarios that cannot be confidently resolved using only data that is immediately available. In such applications, machine agents can actively request targeted observations from human users/agents to resolve uncertainties and improve decision-making confidence. VOI-based querying requires assessment of both the state of the automation and the value of individual human observations. These assessments in turn determine the extent to which the automation 'knows' that it has encountered a difficult decision-making scenario (i.e. a target track that is difficult to characterize) and that the human operator will provide reliable and accurate inputs (to be fused with the probabilistic characterize results generated by the HMM classifier).

5.1 General VOI formulation for decision-making under uncertainty

Beginning with the simplest and most general statement of the VOI querying problem, suppose that a machine agent must make a decision by selecting one of a finite number of actions $a \in \{a_1, ..., a_{n_A}\}$ when the current state of the world is given by s and the future state (following the action) is given by s'. A rational agent will choose a to maximize a given utility function $U(s', a) : (s', a) \to \mathbb{R}$, which coherently reflects desired outcomes for different configurations of s' and a. When s' is uncertain, the agent will additionally act in accordance with their beliefs over p(s') via the principle of maximum expected utility [18], thus will select a to maximize $E[U(s', a)]_{p(s'|a)} = \sum_{s'} p(s'|a) \cdot U(s', a)$. Note that if s is known (uncertain decision making with observable world states), then p(s'|a) = p(s'|s, a); otherwise, if s is also uncertain (uncertain decision making with partially observable or unknown world states), then $p(s'|a) = \sum_s p(s'|s, a)p(s)$. Since the utility function U typically represents the 'long-term benefit' for the agent to occupy configuration (s', a), the same formulation can be applied to dynamic multi-step decision making problems, as well as static single-step actions (e.g. where in the former, utility measures such as discounted cumulative rewards could be used).

Next, consider the possibility that the agent could consult an information source I (e.g. a human sensor, or other some other source of data), which can provide (imperfect noisy) information about s'. In this case, the agent can decide to first query I with cost c before selecting a to maximize the expected utility in light of new information. If the agent decides to consult I and, say, receives data o distributed according to the observation likelihood p(o|s'), then the expected utility becomes $E[U(s',a)]_{p(s'|o,a)} = \sum_{s'} p(s'|o,a) \cdot U(s',a)$, where p(s'|o,a) can be computed from knowledge of p(s') and p(o|s') via Bayes' rule (the same logic is readily extended to partially observable cases where the 'current' state s is also unknown). Since the agent does not know

in advance which o will be returned by O, the agent must take an additional expectation to account for this additional uncertainty, so that the expected utility to maximize now becomes $E[U(s', a)]_{p(s', o|a)} = \sum_{o} p(o) \sum_{s'} p(s'|o, a) \cdot U(s', a)$, where p(o) can be obtained as the normalizing constant for Bayes' rule in computing p(s'|o, a) for each term of the inner sum for a given o outcome.

Since it costs c to consult I, it is not necessarily the case that the agent should always consult I, since the agent could in principle have made the same decision a^* without consulting I and thus obtained on average a utility of $E[U(s', a^*)]_{p(s'|a)}$, rather than $E[U(s', a^*)]_{p(s',o|a)} - c$. The resulting difference in maximum expected utility after consulting I vs. no consultation of I whatsoever is defined as the VOI for source I,

$$VOI(I) = \max_{a} E[U(s', a)]_{p(s', o|a)} - \max_{a} E[U(s', a)]_{p(s'|a)}$$
(5.1)

That is, the VOI indicates the marginal value of consulting I for information before making a decision. If VOI(I) > c, then the agent should consult O at cost c before selecting a, since p(s'|o, a) will lead to a more desirable expected utility with respect to s' regardless of the actual outcome of o. Otherwise, if VOI(I) < c, the agent does not gain any benefit by consulting I before selecting a. VOI reasoning via eq. (5.1) thus provides a principled decision-theoretic framework for determining when a machine agent should consult an external information source, as a function of its current knowledge/beliefs about the world state (i.e. the information it has gathered so far) and the accuracy/reliability of the external information source (i.e. the information it could receive).

5.2 VOI querying for human-assisted target track characterization

Now consider how the general VOI concept can be mapped onto the problem of requesting operator assistance for target track characterization. As before, assume that some fixed queue of target tracks t = 1, 2, ... is to be sequentially processed by an automatic HMM probabilistic classifier using a set of Z^t noisy target signal values $\{Y_j\}_{j=1}^{Z^t}$ as described in Chapter 3.4. The unknown 'world state' variable s' in this case is the outcome of correctly or incorrect classifying the unknown true target type, X^t , for each track t. In the course of processing each track t, the automation will determine, at each sensing frame $j = 1, ..., Z^t$, whether to query the human operator for input (i.e. to perform a hand-off of track t starting with the current $P(X^t)$ distribution), or to continue in its own automated assessment (and thereby continue updating $P(X^t)$ on its own). As such, there are two decision-making actions $a_{j'}^t$ to be considered by the automation at sensor frame $j', 1 < j' \leq Z^t$, where hand-off has not already occurred: $a_{j'}^t = 0$: 'let the automation continue on its own'; or $a_{j'}^t = 1$: 'query and hand-off to the human operator'. The utility function U in this case (defined more precisely below) is given by the classification outcome function for each track t across the T possible target types, which penalizes classification errors and rewards correct classifications.

A few immediate remarks are in order of this particular problem framing. Firstly, note that the act of querying the operator is formulated here as part of the decision $a_{j'}^t$ itself. That is, the act of consulting the external human operator information source I is now intimately tied to the machine agent's action a from the general case (which is allowed in the VOI framework). Secondly, if and when an operator hand-off point j' is selected by the automation for track t, then it is assumed that the HMM classifier stops processing Y_j signal data for j > j' (and thus stops modifying P(X)for track t) while the human operator proceeds to provide soft human sensor data $O_{1:N^t}$ for track t. This simplifies the reasoning process, but also permits additional sequential signal data Y_k for k > j' to be processed after the human has finished providing observations and has updated $P(X^t)$, if needed.

More precisely, let s' be a one-hot $T \times T$ matrix representing where a single non-zero diagonal entry indicates correct target type classification (i.e. type 1 is classified as type 1, type 2 as type 2, etc.), and a single non-zero off-diagonal entry indicates a classification error (e.g. an entry in row 2 column 4 indicating that a true type 2 target was classified as type 4). Further, let U(s') be the utility of correctly/incorrectly classifying track t according to some classification rule. In this case, a classification label x can be assigned to a target track if the probability $P(X^t = x)$ exceeds some pre-determined threshold. Here, U(s') will be given by a static reward function which assigns a fixed cost (e.g. -1) when s' represents any misclassification event, and a fixed gain (e.g. +1) when s' is any correct classification event. Note that the utility here is expressed here only as a function of s', since the resulting classification cost/reward will not depend on whether or not the human is consulted to achieve a particular final classification state. As such, the cost of asking the human operator is accounted for separately from the classification outcome. In the graphical models of Figure 2 (a) and (b), the combined net 'utility minus cost' relationship for each t is compactly expressed via the generic 'reward' (green diamond) node R^t .

With this framing, the VOI for querying and handing off to the human sensor at processing frame j' can now be determined. First, the expected utility at sensor frame j' prior to consulting the human is

$$EU_{a_{j'}^t=0} = \sum_{s'} \sum_{X} P(s'|X) P(X) U(s')$$
(5.2)

where P(X) gives the current belief over the possible true target states according to the HMM probabilistic classifier, and P(s'|X) reflects the expected classification accuracy for the classifier. Note that, in general, P(X) is not the same as P(s'|X), as the probabilities for P(X) generated by most machine learning algorithms are not necessarily consistent with their underlying precision, recall, and classification error rates [13]. However, it is relatively simple to estimate and obtain P(s'|X) from standard offline training and cross-validation/holdout validation procedures that can be applied to an HMM probabilistic classifier (or for that matter any other probabilistic machine learning algorithm). In particular, as illustrated in Chapter 4, the HMM classifier can be provided with many ground-truth labeled test examples to see where it requires the most 'help' in discriminating among the T different target types. Then, at any point in the data stream $\{Y_j\}_{j=1}^{Z^t}$ for track t, based on the current estimated target type probabilities P(X) and the knowledge of the HMM's accuracy P(s'|X), eq. (5.2) can be used calculate the expected value of letting the automation continue without requesting operator intervention.

Next, the expected utility of requesting human operator assistance is considered, while also accounting for the lack of knowledge in human operator sensor CPT observation likelihood parameters θ_1 and θ_2 that govern the expected response reliability and accuracy for the soft data $O_{1:N^t}$ being requested. In this case, the unknown θ_1 and θ_2 likelihood parameters must be considered also part of the unknown world state, and thus must also be marginalized out in the expected utility calculation. Following the graphical model in Figure 2 (b), this implies that

$$EU_{a_{j'}^t=1} = \int_{\theta_1} \int_{\theta_2} \sum_{O_{1:N^t}} \sum_{s'} \sum_X p(s'|O_{1:N}, X) p(O_{1:N}|X, \theta_1, \theta_2) P(X) p(\theta_1) p(\theta_2) U(s') d\theta_1 d\theta_2$$
(5.3)

where $p(O_{1:N}|X, \theta_1, \theta_2) = p(O_1|\theta_1, X) \cdot \prod_{k=1} p(O_{k+1}|O_k, \theta_2^{k+1|k}, X)p(\theta_2^{k+1|k})$, assuming N observations are required to classify a target to a chosen level of confidence. The left-hand side of expression requires computing the expectation with respect to 5 unknown random variables in an online manner, which precludes an exact answer. However, the left-hand side of eq. (5.3) can be approximated via Monte Carlo sampling with respect to the prior distributions P(X), $p(\theta_1)$ and $p(\theta_2)$, as well as by simulating $O_{1:N}$ by sampling with respect to the induced CPTs which make up $p(O_{1:N}|X, \theta_1, \theta_2)$ once X, θ_1 and θ_2 are instantiated. Note that such sampling with respect to θ_1 and θ_2 already occurs via Dirichlet distributions for the initial phase of Gibbs sampling for operator data fusion (once hand-off has occurred and actual human data $O_{1:N^t}$ is available). However, Gibbs sampling is not used here since actual human operator data has, at this point, not yet been been observed for Bayesian posterior data fusion; rather, direct Monte Carlo can be used instead.

The only remaining term to estimate on the left-hand side of eq. (5.3) is $P(s'|O_{1:N}, X)$, which represents the accuracy of classifying the target after the human has been asked to intervene and has given N (simulated possible) soft data observations $O_{1:N}$. Akin to the non-equivalence of $P(X^t)$ and P(s'|X) in eq. (5.2), the values for $P(s'|O_{1:N}, X)$ are not immediately obvious from knowing the true/false alarm and missed/true detection rate parameters from the (parameter-tied and sampled) values of θ_1 and θ_2 for the human operator (especially as it is assumed that the $O_{1:N}$ observations are non-independent). Hence, to estimate $P(s'|O_{1:N}, X)$, simulations of the observations $O_{1:N}$ expected from the operator (based on samples of θ_1 and θ_2 taken from the current $p(\theta_1)$ and $p(\theta_2)$ prior pdfs) must be generated and used in combination with samples of 'ground truth' X values (taken from the current P(X) target type prior) to produce an estimated new posterior $\hat{P}(X|O_{1:N})$ for each direct Monte Carlo sample instance, using eq. (4.4) in particular to produce this estimated/hypothetical posterior over X. For each possible target type x, $\hat{P}(X = x|O_{1:N})$ can then be compared to the desired classification threshold to determine a hypothetical classification result s' for each direct Monte Carlo sample instance given various X and $O_{1:N}$ configurations, thus allowing the classification utility U(s') to be computed for that sample instance.

Given eqs.(5.2) and (5.3), the VOI for querying the human operator at time j' can finally be computed as

$$VOI(I_{j'}) = EU_{a_{i'}^t = 1} - EU_{a_{i'}^t = 0}.$$
(5.4)

This VOI is then compared to a stated cost $c_{j'}$ of interrupting the operator at time j', such that a query is made if $VOI(I_{j'}) > c_{j'}$. The cost $c_{j'}$ may also factor in other considerations such as cognitive over-/under-loading [10]. The VOI assessment can be performed for every sensor signal frame Y'_{j} or less often, e.g. based on how busy human operators tend to be.

In this work, the utility function U(s') is set to demonstrate the capabilities of the system. In a deployable scenario, U(s') should be specified and tuned according to problem requirements. For example, the target types for the simulation results shown in Chapter 6 are considered equally likely and of equal value, i.e. the penalty for misclassifying any target type is identical, as is the reward for correct classification. In a real world scenario, if there were a high cost for misclassifying or confusing specific kinds of targets, the corresponding penalties in U(s') should be correspondingly larger, which in turn could have a much larger influence on the automation's decision to query the operator. If the operator's accuracy for certain 'difficult' targets is believed to be even slightly better than the automation, then a large misclassification penalty for such cases would lead to more requests for assistance. Conversely, if the operator is believed to be worse than the automation, the system will be less likely to risk gathering bad information which could endanger the classification. In the latter case, the operator could still voluntarily supply information about the target, although they would much less likely be prompted to do so by the automation.

Finally, the cost of querying operators can also be specified by the designer and is situationally dependent. For the implementations shown in Chapter 6, $c_{j'}$ is set to a constant and tuned to produce proof-of-concept scenarios that often resulted in the automation requesting the operator's

help, without relinquishing classification immediately. This tuning helps illustrate overall trends for VOI-querying behavior. For example, when moving from the T = 5 target to the T = 10target case, c'_j was increased to accommodate the increased uncertainty in the automation's own assessment via the HMM classifier, which otherwise would have led to earlier and more frequent operator intervention requests. In more general settings, problem-specific uncertainties for different surveillance regions and target types will influence how often queries will be made. Although beyond the scope of this work, $c_{j'}$ could vary based on the available number of operators at any given time. In this way, the cost and utility functions can set the workflow for groups of operators as well as assessing which particular operator the automation would like to query, based on expected utility. Such a distributed system would then be able to maximize the total utility for target characterization a team of operators, rather than just a single operator.

Chapter 6

Simulation Results & Discussion

This chapter presents several sets of results that demonstrate and assess major features of our proposed human-machine data fusion framework for online robust dynamic target characterization. First, in Chapter 6.1, results of the Gibbs sampling procedure developed in Chapter 4 for the non-independent human sensor model in Figure 2 (b) are shown. These results validate that the samples used for inference in subsequent simulation results are mixing well and do not include biasing from the Markov dependencies. Chapter 6.2 provides an example application of the momentmatching procedure for bootstrapping $p(\theta_1)$ and $p(\theta_2)$ to demonstrate how it works in practice and to validate that the amount of statistical information required to describe the human parameters can be compressed without sacrificing modeling accuracy.

In Chapter 6.3 The accuracy of the automated probabilistic HMM target classifier is then established for typical sets of target tracks to provide a baseline level of target characterization for the VOI-based human-aided system. In Chapter 6.4 inference of the human sensor parameters is demonstrated using a similar set of tracks. Truth model simulations are used to generate human operator observation sequences, and it is shown that the proposed hierarchical online Bayesian modeling and inference approach can adapt over time to human response rates from operators with different accuracy/reliability levels.

Chapter 6.5 finally presents truth model simulations where VOI querying is used to prompt observation sequences different types of human operators. The results here demonstrate the proposed framework's adaptability to different operators as well as the ability to recognize targets the automation has trouble classifying. To show the feasibility of implementation in a real-time scenario, the computing costs for three different variations of the human operator data fusion model (the 'naive' independent observation model from Figure 2(a), and the parameter-tied and fully parameterized Markov dependent observation model from Figure 2(b)) are also compared, along with their target type classification performance measures.

For the truth model simulations presented in each subsection, ground truth human operator observation parameters were randomly sampled from a set of fixed Dirichlet priors $p(\theta_1)$ and $p(\theta_2)$, whose hyperparameters are set such that the expected values of the resulting observation likelihood CPTs conform to specified baseline levels of operator accuracy and reliability (e.g. examples for typical 'expert' and 'novice' operators are provided in Chapter 6.4). All simulations use T = 5target types until Chapter 6.5, where scenarios involving T = 10 target types are also considered to examine the scalability of the proposed techniques. As mentioned in Chapter 3, an HMM probabilistic classifier was trained using 500 labeled synthetic target track data sets, and used to automatically generate simulated P(X) values online for a queued set of 100 new synthetic target tracks with simulated $\{Y_j\}_{j=1}^{Z^t}$ intensity data from notional space-based sensors, which included Gaussian noise for atmospheric disturbances as well as weather attenuation. For each track $t, Z^t =$ 100 sensor frames, and the probabilistic confidence threshold for declaring a particular confidence type for the track was set to 90%. In Sections 6.3-5, the ground truth simulations were each run with all 100 test target tracks and repeated 30 times with different randomized ground truth values for the unknown X target labels and human observation parameters θ_1 and θ_2 .

6.1 Gibbs sampling inference for simulated operator

Figure 6.1 shows the results of one sample run of the Gibbs sampler using simulated observations from a nominal human operator. For brevity, the plots here show only the results from a parameter-tied model simulation and for the θ_2 CPT parameters that describe successive pairs of 'true positive' observations (left plots), and a 'true positive' observation followed by a 'false positive' observation (right plots). In this parameter tied simulation, 14 other parameters are also



Figure 6.1: Signal, autocorrelation, and histogram of parameter-tied samples of θ_2 .

sampled for every observation. The figures show the posterior θ_2 parameter Gibbs samples over time (top left subfigures), the histogram of the samples (top right subfigures), and the autocorrelation of the Markov chain sample sequence (bottom subfigures). These plots demonstrate the typical behavior of the Gibbs sampler for a single inference/fusion instance, in particular showing that the θ_2 parameters can in practice be quickly identified under the parameter-tying assumption, even with relatively sparse amounts of operator observation data. The autocorrelation function plots confirm proper mixing of the Gibbs samples, i.e. such that successive Gibbs stored samples can be considered i.i.d samples from the target posterior over θ_2 . In this example, 5000 total samples were drawn for each human observation provided via Gibbs sampling; 1000 burn-in samples were used to ensure proper mixing, and every 5th sample afterwards was stored, giving 800 samples total per human observation. The hyperparmeter values were set to $\boldsymbol{\alpha} = [5, 2, 0.5, 8]$ for parameter tied observations [TP, FP, FN, TN] keeping in mind for a 5 target scenario there are four different false positive and true negative observations. For θ_2 each repeated type of observation (TP, TP) was doubled. The graphs displayed are a product of sampling this prior distribution and the first observations of the human operators.



Figure 6.2: Moment matching example for: the parameter-tied true positive followed by a second true positive observation; (b) the parameter-tied false positive followed by a true positive observation.

6.2 Moment Matching

Figure 6.2 demonstrates results for operator parameter bootstrapping using the Dirichlet pdf moment-matching methods described in Chapter 3.3. The θ_1 and θ_2 Gibbs sample data used for these plots is taken from the same simulation shown in Figure 6.1. The histogram of samples are shown in blue and the matched distribution is shown in yellow. There are 10 sets of these samples per Dirichlet and each of these distributions describe one of those categories. In order to graph a single pdf of the Dirichlet, the other dimensions were marginalized out, leaving a single Beta function described by the Dirichlet α parameters [5]. As stated previously, the Dirichlet distributions can be completely described by the α hyperparameters. This reduces the number of parameters required to describe any individual operator. The lines shown in yellow are constructed using only these parameters and correctly match the distributions sampled.



Figure 6.3: HMM Target classification confusion matrices for (a) 5 targets and (b) 10 targets.

6.3 Hidden Markov Model

$$p(s_k|y_{1:k}) = \alpha(s_k) = p(y_k|s_k) \sum_{s_{k-1}} p(s_k|s_{k-1})\alpha_{k-1}(s_{k-1})$$
(6.1)

Figure 6.3 (a) shows the confusion matrix for 3000 generated targets and the accuracy of the automation without human operator intervention. Overall, the HMM is able to achieve 81% correct classification. It is important to note that the simulation presented is by design. The application of this system is in scenarios where the automation struggles and requires assistance from human operators. While it is possible to create a dataset with smaller overlap or a dataset where only a few targets overlap, we aim to show a scenario where the automation is able to classify certain targets, but has room for improvement and has specific targets that require more attention than others i.e. targets 1 & 3.

Further inspection of Figure 6.3 (a) shows that targets 1 & 3 are correctly classified the least often at 62.8% and 58.1% respectively. This will prove to be important later, as one would expect those targets to be asked about by the automation more frequently than targets 0, 2, & 4. The HMM of course does not know what target type it is looking at while attempting to classify, but based on P(X), the HMM can predict the likelihood of correctly classifying a target. The graphs shown here directly represents the confidence in each target type classification that can be used by Value-Of-Information to access when to ask for assistance. In a real classification scenario, a training set can be used to create the HMM parameters while a test set can be used to estimate the reliability/accuracy statistics needed to specify p(s'|X) in eq. (5.2). As mentioned before, the proposed framework is not confined to only the use of an HMM, as any machine learning algorithm can be substituted to classify targets along any number of feature vectors.

Figure 6.3 (b) shows the confusion matrix for the 10 target scenario. As stated before, the first 5 targets are exactly the same, but 5 more targets all similar to the first 5 are now included. The variance is exactly the same as are the number of training scenarios and test scenarios presented. Due to the unchanged variance, the overall accuracy of the HMM classification drops to 69%. This is once again by design. In a real world scenario, adding more targets with the same variance and similar intensity data would result in a drop in accuracy for the automation. As with the 5 target case, this would be uncommon to encounter, but is demonstrative of the adaptability of this methodology. With the 10 target case, one would now expect to see the automation ask for assistance for often on targets 0, 8, & 9.

6.4 Inferring Human Sensor Parameters

In this subsection, computational comparisons between the 'naive' independent model of human observations in Figure 2(a) and the proposed Markov dependent model of human observations in Figure 2(b) are considered, using parameter-tying in both cases. For all simulations presented below, the prior Dirichlet distributions over θ_1 and θ_2 are exactly the same and identical to the α presented above ($\alpha = [5, 2, 0.5, 8]$ for [TP, FP, FN, TN] and doubled for repeated observation types).

6.4.1 Naive independent observations model

Shown in Figure 6.4 are the starting Dirichlet distributions, the ending distributions (after 100 target events) and the 'true' θ_1 likelihood parameters for the human operator in the parameter



Figure 6.4: Initial and final distributions for parameter-tied θ_1 values with simulated ground truth values ("novice" operator).

tied simulation. These estimates use the same data the generated the 3000 samples for the HMM above, but unlike the HMM, this is an algorithm designed to adapt over time. The graphs shown are averages taken over 30 simulations of 100 target events each. All distributions are part of a Dirichlet distribution, but in order to shown them graphically, each has been reduced to the underlying Beta distribution by marginalizing out the other dimensions of the Dirichlet. The blue curves represent the prior distributions that the simulation is initialized with. The orange lines are the final distributions constructed from the sampler and moment matching. The blue dots represent the 'true' values. These are chosen at the beginning of the simulation and never change. The sampler is unaware of these values, but are shown here to validate the system's ability to characterize the human sensor. Figure 6.4 also shows that for the θ_1 parameters, the priors are not considerably different from the final results. The human operator's likelihoods are chosen prior to simulation as are the prior distributions, but this still demonstrates an important point. Even after 30 simulations of 100 targets each, the differences are small. Intuitively, this is expected. Under the assumption there are only 4 types of observations and that a human operator answers truthfully more often than not, the relative frequency differences are going to be small.

It can be seen that, at least on the first observation given per target, that the prior on the human's true positive rate is a little high. Conversely it can also be seen that the prior on true negative observations was a little low. The θ_1 parameters could be estimated using only maximum likelihood given the narrow scope of options, but we get these samples for free when running Gibbs



Figure 6.5: Initial and final distributions for parameter-tied θ_2 values with simulated ground truth values ("expert" operator).

Sampling because of the compounding nature of observations. As the figure shows, the Dirichlet distributions are able to accurately characterize the human operator and increases its certainty over time.

6.4.2 Markov dependency observation model

While the Dirichlet distributions are certainly capable of characterizing the independent likelihood vectors, they are primarily designed to handle dependent likelihood vectors. Using our parameter tied approach, we have 16 combinations of observations. Each of the 16 combinations are shown in Figure 6.5 from true positive followed by another true positive and true negative followed by another true negative. Each row represents a single Dirichlet and each column represents a single category.

As detailed in the data generation section, the θ_2 likelihood vectors for the human operator are assumed to be constant. These parameters are what drive the observations seen by the model. In a real scenario, these values would never be known. Here, they are used to show a proof of concept of the methodology. As mentioned previously, the prior Dirichlet distributions were set to a lower level of accuracy than the 'true' human in the expert scenario. This can clearly be seen in the true positive-true positive case at the top left of the figure. A true positive observation is the most informative an operator can give. To start, the distribution shows a higher probability of giving a true negative following a true positive and a lower true positive rate. After seeing target events and observations from the human operator, the Dirichlet is more accurately distributed to represent the individual operator. This same effect can be seen clearly in the false positive and true negative rows as well.

The one case were the distributions appear to lag behind the 'true' values is the false negative row. As shown in Figure 6.6 (a), a false negative observation is extremely rare for the expert operator. As a result, this is a very rarely sampled case. Using Dirichlet distributions does allow for updating the parameters of observation cases that are otherwise rarely/never observed for certain targets. As long the generic type of observation type was seen (i.e. True Positive followed by False Positive, etc.), the entire row of the CPT can be updated in the Gibbs sampling cycles. This is why the true positive-false negative distribution is can still be modeled despite the availability of so few observations. However, rarely seeing a false negative observation means the Dirichlet describing the observation type and the following observations has little information to learn from. The 'true' values are still accounted for by the uncertainty in the false-negative distributions, but as opposed to other groupings, the uncertainty isn't reduced over time. the false-negative distributions are still attempting to center on the correct likelihood values before it can reduce its uncertainty.

Also of note is the apparent bi-modal grouping of θ_2 parameters in the true negative-true negative case. The generated operators have a higher likelihood of giving the same observation twice in a row rather than providing new information. This is to model an assumed 'stubbornness' in human operators. While the Dirichlet distributions do account for this repetitive nature, the tying of parameters means there is no distinguishing from the observation 0⁻ and 1⁻ (not of type 0, not of type 1). To the parameter tied simulation, this is simply a repeated true negative observation. This same effect can be seen to a smaller extent in the false positive-false positive case. Once again, the human operator is more likely to repeat the exact same observation. For the double false positive case, the Dirichlet is able to model the likelihoods regardless. For the repeated true negative case, the Dirichlet is not able to accurately account for all types of observations, but as seen in Figure 6.20, this does not affect the overall accuracy compared to the full parameter simulation.

6.4.3 Sensitivity to operator reliability/accuracy levels

Figures 6.4 and 6.5 demonstrate the ability to characterize the likelihood vectors when using an 'expert' operator where the correct classification of the target is both easier and more informative to the human parameters as a result, however we desire the ability to characterize 'novice' operators or characterize operators even when the classifications of targets is more unsure i.e. double the number of target possibilities. A feature of our Bayesian inference approach is that it can automatically calibrate itself to fuse information from operators with a variety of skill and experience levels. Truth model simulations can be used to demonstrate this capability.

First we will look at a 'novice' operator. Figures 6.6 (a-d) show confusion matrices for different simulated operators with different levels of experience. Shown are the true positive (top left), false negative (top right), false positive (bottom left), and true negative (bottom right) rates



Figure 6.6: Simulated individual operator confusion matrices: (a) "expert" operator; (b) 2nd operator; (c) 3rd operator; (d) "novice" operator.

for each operator. On the top-left is an 'expert' or veteran operator, who almost never provides the worst type of observation (false negative) and rarely provide false positive observations. On the bottom-right is a 'novice' operator, who performs better than the automation can on its own, but reports false negatives 12% more often and false positives 18% more often than their veteran counterpart. Also included are two operators that sit between our 'expert' and 'novice' and will be used to demonstrate the adaptability of the system not just to the extreme cases, but any operator. Applying the same 'generic' operator likelihood model to each of these operators would result in discounting the expert's observations. It could also present a major liability in the novice, as trusting their observations too much would result in fast, but inaccurate classification. It should be noted that these operator profiles are used only for generating the observations, and have no bearing on which type of methodology is used to infer the likelihoods. Figure 6.7 shows the same



Figure 6.7: Initial and final distributions for parameter-tied θ_2 values with simulated ground truth values ("novice" operator).

16 parameter graph for the 'novice' operator. Looking only down the first column, we see the rate of all observations followed by an true-positive observation. While the prior distributions are under confident for the expert operator, they are overconfident for the novice operator. In contrast to Figure 6.5, all true-positive likelihoods are lower and therefore less likely to happen. This clearly demonstrates that the priors were far too high for this operator as shown in blue. However, the



Figure 6.8: Initial and final distributions for parameter-tied θ_2 values with simulated ground truth values for 10 target simulation ("expert" operator).

model was able to learn this and adjusted accordingly. All likelihoods of a true positive observation were dropped to more accurately describe the 'real' values.

Next, we want to show the system's ability to scale and still account for the differences in human operators. Figure 6.8 shows the same four values as the 'novice' operator (first column rotated). This time, the human is the 'expert' operator but instead of only discriminating between T = 5 target types, they must discriminate between T = 10. The classification results of this simulation are shown below. For the characterization, it can be seen that the Dirichlet distributions display the same means and variances as the 5 target case. This shows two things: 1) The likelihood vectors remain the same even when there are twice as many targets and 2) the system is able to characterize the human likelihoods even when the number of targets are not only increased, but are difficult to distinguish for the automation as shown in Figure 6.3 (b). This shows that Dirichlet moment matching is a useful approximation for inferring human parameters in an online multitarget classification setting.

6.5 Value of Information



Figure 6.9: Cumulative average frame where classification is passed from the HMM to the human operator ("expert" operator).

After showing the ability to characterize individual human sensors, that information can now be used to inform how the automation fuses information provided by the human operator. Ideally, if the automation can handle the classification on its own, then there is no need to ask the human

Target Type	Avg Pass Off Frame	STD
0	54.64	41.79
1	32.09	42.21
2	52.87	42.21
3	49.85	42.78
4	52.38	40.96

Table 6.1: Average pass off frame for individual target types ("expert" operator).



Average Pass Off Frame Over Time

Figure 6.10: Cumulative average frame where classification is passed from the HMM to the human operator (2nd operator).

for assistance. In a real scenario, this evaluation changes based on how much a specific human operator can help. Rather than treating all human sensors as the same, the online adaptation can further inform us on the best way to utilize the human operator.

Using equation 5.3, one can substitute the newly found θ_2 values to infer the expected value of a specific operator's contributions relative to the automation. Once again taking the 'expert' and 'novice' operators, one can see how the Value-Of-Information algorithm adapts over time. Based on the the HMM's current estimate and the estimate of the human operator, each time step (new data) the automation determines if the classification should be passed off to the human operator.

Target Type	Avg Pass Off Frame	STD
0	79.15	34.47
1	54.88	46.75
2	76.62	36.31
3	71.77	40.6
4	78.54	34.92

Table 6.2: Average pass off frame for individual target types (2nd operator).



Figure 6.11: Cumulative average frame where classification is passed from the HMM to the human operator (3rd operator).

In practice, this means the automation would function in the background until it needs help, then prompts the human operator with the required target information if/when needed.

One expects with an 'expert' operator, the automation would ask early and often. If the human is better at classification than the automation by a significant margin, one would expect better accuracy by asking that operator on more target events. This does not mean we ask the human operator for every single target immediately. For all data presented in this paper, the reward structure was to award 1 for a correct classification, -1 for an incorrect classification, and to ask the human operator if they do 45% better than the automation on average. These averages

Target Type	Avg Pass Off Frame	STD
0	89.24	26.93
1	67.45	44.83
2	87.87	28.53
3	84.59	33.54
4	90.02	25.82

Table 6.3: Average pass off frame for individual target types (3rd operator).



Average Pass Off Frame Over Time

Figure 6.12: Cumulative average frame where classification is passed from the HMM to the human operator ("novice" operator).

were assessed by taking the current information and running a simulation where the estimated human classifies the target 100 times, taking the HMM self-confidence and also classifying 100 times, and comparing the difference in expected value. While the expert operator is better than the automation in this scenario, the automation only asks if it believes the human will do 45% better. This number was chosen to provide a healthy balance of asking for assistance while also allowing the automation to function on its own. This allows for validation of the system, however, a real scenario could adjust these values to put more preference on the human's time or on the accuracy of classification. But as the results show, placing higher utility on accuracy can backfire.

Target Type	Avg Pass Off Frame	STD
0	95.64	17.48
1	72.71	43.28
2	93.30	22.21
3	88.54	29.25
4	94.88	18.62

Table 6.4: Average pass off frame for individual target types ("novice" operator).



Average Pass Off Frame Over Time

Figure 6.13: Cumulative average frame where classification is passed from the HMM to the human operator (10 targets "expert" operator).

From Figure 6.9, it can be seen that over 30 runs, the cumulative average frame where classification is passed off to the expert human operator goes from approximately 40 to 45 with a high variance. Looking more closely at these results, the accompanying table shows for a given target type, what the average pass of frame is. Looking back at the graphs for the HMM confidence (Figure 6.3 (a)), one would expect the automation to ask about target types 1 and 3 more often as they are the hardest for the HMM to distinguish. This is exactly what is seen in the results. The automation asks for help earliest on target 1 with an average pass off frame of 32.09. The next earliest it asks for assistance in with target 3. It does ask about the target earlier than the

Target Type	Avg Pass Off Frame	STD
0	59.33	42.72
1	56.69	41.44
2	63.26	33.89
3	76.08	28.91
4	58.16	34.65
5	57.22	47.01
6	52.44	34.34
7	61.48	44.70
8	43.41	34.86
9	48.04	30.7

Table 6.5: Average pass off frame for individual target types (10 targets "expert" operator).

ones it gets correct, but it is a small difference. Referencing back to Figure 3.4, it can been seen that while target type 1 is difficult to distinguish at all times, target type 3 is actually separated from other target types until frame 40, where it begins to look like every other target. As a result, one would expect it to be asked about earlier than targets the HMM gets correct, but not by a significant margin because it won't be until frame 40 that the automation begins to lose confidence in its assessment.

Moving to the 'novice' operator, there is drastically different behavior. Figure 6.12 shows how the average pass off frame moves later and later with a very tight distribution. While it is not and apples to apples comparison, the 'novice' operator is worse than the automation on its own. The HMM has 81% accuracy while the novice operator only gives a true observation 71.6% of the time. As a result the automation sees little opportunity for improvement by asking this operator to help. The only reason it asks at all is due to the prior belief. At least to start, the system believes the human operator is worth asking. In only a few targets, the automation learns that is should stop asking and simply classify on its own. In this case, putting an emphasis on more accuracy, and thus lowering the threshold to ask the operator, would be detrimental. Not only would the human be asked to classify more targets, but if there was any doubt in the automation that it would be able to correctly classify a target, it might perform worse by asking a human that is worse than the automation alone.
Figures 6.10 and 6.11 show a gradual increase in the average pass off frame. This is exactly the behavior that is expected. Each operator has different levels of accuracy and the automation must decide how useful that operator is. The worse the operator, the less often and the later the automation should ask. Indeed this is what we see in both the cumulative averages and the accompanying tables. Additionally, we see that target types 1 & 3 are still asked about earlier on average than the other target types despite all targets being passed off later.

The accompanying table does show how even this human may be helpful. Overall, the average pass off frame is much higher (many reach the full 100 frames and the human is not needed at all), but on targets where the automation struggles, it still asks the human more often. This demonstrates two things. First, the automation is able to learn when it should ask an individual operator to maximize overall classification. Second, the automation can distinguish between different target types and supplement its knowledge with the human operator when dealing with targets that are difficult to classify, squeezing as much useful information out of a bad operator.

Figure 6.13 shows the cumulative average pass of frame for the 10 target simulation. Referencing back to Figure 6.3 (b), we would expect the targets asked about earlier to be 0, 8, and 9. Looking at the accompanying table, we do see that targets 8 and 9 are asked about earlier than other targets with average pass off frames of 43.41 and 48.04 respectively. The graph shows a much tighter distribution on when the automation decides to ask and all are later on average than the 5 target case with the same operator. For the 10 target type simulation, the threshold was raised to ask the human for assistance. This was done because the uncertainty increases as more targets are added. As a result, the automation began asking about targets almost immediately. This is not desirable. While it does mean the accuracy of the classification should go up, the goal is to ask for assistance only when necessary, not every time a target appears on the screen. When increasing the number of possible target types, the threshold for passing off the classification must be altered to reach the desired amount. In a target classifying scenario, this could be dependent on how many targets per hour there are, how many operators there are, and how much you desire to utilize your resources. In this paper, the threshold was chosen in order to represent overall trends in the fusion



Figure 6.14: Average computation time for independent, parameter tied, and full parameter observation models for 5000 gibbs samples and average computation time for moment matching done after target classification.

pipeline.

6.6 Timing

Now that the ability to characterize both dependent and independent observations has been demonstrated from human operators, it is also important to examine how much the associated approximate inference process will cost computationally, since such inference must be performed online and since both time and computational resources are both valuable in many applications. To examine this question, the computation times for three different methodologies will be compared: the full parameter simulation with dependencies between each target type, the parameter-tied simulation where we only account for dependencies between the four types of observations (TP, FN, FP, TN), and a simulation with only independent observations. Each simulation is given the exact same data for each target and averaged over 30 runs of 100 target events each. It is assumed that they human observations are instantaneous. This is not to assess how fast individual humans make decisions, but to assess the relative computation time for different models.

Shown in Figure 6.14 are the average times to complete the Gibbs sampling step for each methodology, as well as the average time to complete the moment matching after the target is



Figure 6.15: (a) Average computation time for moment matching when increasing to 10 target types; (b) Average total computation times incorporating the average number of observations required to classify a target and moment matching ("expert" operator).

classified. Dues to the increase in number of samples, the parameter tied and full parameter versions both take more time to complete all Gibbs samples. The full simulation, having larger Dirichlet distributions to sample, take the most time overall. Conversely, the simulation assuming independent observations takes less than half the time on average.

Looking at the moment matching times, we see that again the full simulation takes the most time, but the parameter tied method takes the least amount of time by a significant margin with almost no variance. This is because the parameter tied sim will have four Dirichlet distributions each with four parameters and never changes. The full simulation has 50 Dirichlet distributions with 10 parameters and not all Dirichlet distributions will necessarily be sampled. Even the independent simulation has 5 distributions each with 10 parameters since we are not using parameter tying in this model. This means that as more targets are added, the moment matching for both the independent and full parameter simulations will increase exponentially, while the parameter tied simulation will always remain the same. When we increase the number of targets to 10, this is exactly what we see in Figure 6.15 (a).

These results indicate that the timing behavior is as expected. The independent simulation takes the least amount of time because it is the least computationally complex. However, these



Figure 6.16: Average total computation time for independent, parameter tied, and full parameter observation models for 5000 gibbs samples and moment matching done after target classification (a) 2nd operator and (b) 3rd operator.

are averages for single observations, not the full set of observations. By including the information gained by coupling observations, one ideally would need fewer observations to communicate the same information. Taking a look at Figure 6.15 (b), this is exactly what is showed. Shown are the average total times for each model, broken up by the average number of observations required to classify a target. Additionally, the moment matching times which are performed at the end of a target event, are also added. The moment matching times for the independent and parameter tied simulations are fast enough that they are difficult to see, but the time spent on the full parameter simulation is non-negligible.

The main takeaway here is that while the parameter tied simulation takes more time per sample, it requires fewer observations in total. As a result, on average, even while having more complexity in the Gibbs sampler, the parameter tied simulation is faster than the independent simulation. One can also see that the full parameter simulation, which requires far more observations to accurately sample each parameter, takes both more time and more observations to classify a given target event. Taken over a full simulation these time savings begin to add up. Table 6.6 displays the average total computation time over all 100 target events for each method.

Table 6.6 displays the average total computation time over all 100 target events for each



Figure 6.17: Average total computation time for independent, parameter tied, and full parameter observation models for 5000 gibbs samples and moment matching done after target classification (a) "novice" operator and (b) 10 targets "expert" operator.

method. For our 'expert' operator, we see a significant gap between the independent and parameter tied simulations amounting to a minute more on average. Once again we are assuming the observations are instant. Majority of the divide comes due to a larger number of required observations in the independent simulation which would add even more time. Comparing to the full parameter simulation, parameter tying saves 11 minutes on average. These times will of course change depending on the speed of individual systems, but the relative difference between computation times will still persist. Also worth noting, the moment matching process can be parallelized, but due to the dependencies of Gibbs samples, all sampling must be done on a single thread.

At first glance, these advantages seem to disappear when moving down our operators to our novice. The full parameter simulation still takes the longest amount of time, but the independent simulation is now faster than the parameter tied, and all take less time than their expert counterparts. As we've shown previously, once the automation learns that our novice operator is not to be trusted, it stops asking for help altogether. This results in less total time spent computing, as the automation doesn't need to fuse human observations as often when it can classify the target alone.

Looking at Figures 6.16-6.17, we start to see this trend in more detail. For each simulation, the Gibbs sampling step takes the same amount of time, but fewer observations are taken. At the

Sim Type	'Expert'	Human 2	Human 3	'Novice'	'Expert' 10 Targets
Independent	$249.79 \ s$	145.75	$80.95~\mathrm{s}$	$51.54~\mathrm{s}$	283.53 s
Full Parameter	881.81 s	525.37	282.24 s	$180.98~{\rm s}$	1443.16 s
Parameter Tied	$187.50 { m s}$	203.73	$172.34 {\rm \ s}$	$135.00~\mathrm{s}$	219.37 s

Table 6.6: Average total computation time in seconds for each simulation method over 100 target events. 30 runs

point the automation asks for help, it doesn't take many observations to meet the criteria to classify a target. The automation has seen a majority of data and doesn't need as much information from the human. In the case of our worst operators, the system knows the information gained from the human isn't as informative or valuable.

When scaling to double the number of targets, we see a slight increase in computation time for both the independent and parameter tied sims with a significant increase in the full simulation. By adding 5 more targets, the full simulation increase from 500 unique parameters to 4000, requiring even more information to characterize the human operator. The increase in uncertainty for any given target type means more information is required from the human operator in both the independent and parameter tied cases, but the parameter increase is far more manageable for the independent simulation going from 50 to 200 and in the case of the parameter tied sim it doesn't change at all from 16 parameters.

This can be seen in Figure 6.17 (b) which looks very similar to our original expert operator with 5 targets. The full parameter model takes significantly longer, but a majority of that time is spent on moment matching and the longer time for Gibbs Sampling. As the number of parameters exponential increases, the time to sample also increases. In particular the moment matching starts to take a non-negligible amount of time. While the data streams for target tracking scenarios are slow to update, as the number of targets grow the more time is spent on the moment matching step. Eventually, the computation required would begin eating into valuable time. This is one of the most important improvements that the parameter tying provides.



Figure 6.18: Confusion matricies for independent, parameter tied, and full parameter observation models. 30 runs of 100 target events each. "Expert" operator

6.7 Classification

6.7.1 Difference Between Models

So far it has been established that we are able to characterize human sensors, use this information to intelligently ask the operator for assistance, and that we are able to do all of this online for dependent observation sequences while still reducing computation time. However, the time and computational savings are not worth anything if the system cannot correctly classify targets. Hence, in this subsection, the focus is turned to assessing metrics of online target classification for the combined (simulated) human-machine system.

Referring back to the confusion matrix generated by the HMM (Figure 6.3 (a)), we can show an improvement by the expert operator for both the full and parameter tied simulations (Figures 6.18 and 6.20 (a)). For all simulations, the same data is used, the same HMM is used, the automation is passed off at the same frame, and the exact same human responses are used. Only the assumptions about model dependence and parameter-tying are different between the models considered here.

Both dependent observation models are able to improve classification, particularly for targets



Figure 6.19: Precision-Recall curves for independent, parameter tied, and full parameter observation models "expert" operator.

1 and 3 where the HMM struggles. Conversely, the independent model does not improve accuracy. The independent model is also able to improve the accuracy for targets 1 and 3, but does considerably worse on all other targets. Overall, the methodology actually does worse than the HMM on its own seen in Figure 6.20 (a). Looking at Figure 6.19 one can further see the difference between simulations. These are precision-recall graphs where precision is the number of true positives over true positives plus false positives $\frac{T_p}{T_p+F_p}$. Recall is the number of true positives over true positives plus false negatives $\frac{T_p}{T_p+F_p}$. A perfect classifier would be a straight line across the top, meaning every single target was classified correctly. As we move to the left of the graph, we expect higher rates of precision over a smaller number of targets the system is more confident in. This is what is seen with the parameter tied and full simulations. For the independent model, the opposite behavior is observed: the classifier does worst when it is most confident in its answer.

For these simulations, the human operators are built upon our assumption that observations are dependent. We would expect the parameter tied and full simulations to characterize these sensors better, but without communicating anything other than observations, treating observations as independent from another can actually hurt when asking for outside assistance.



Figure 6.20: Cumulative averages for HMM prior to being passed off to operator (red), HMM without human intervention (purple), independent observations (green), parameter tied observation (blue), and full parameter observations (orange) for (a) "expert" operator; (b) Operator 2.

6.7.2 Effects of varying operator reliability/accuracy

The terms 'expert' and 'novice' operator have consistently been used to demonstrate the fusion engine's ability to adapt to different human sensors, but we also want to look at operators in-between. Shown in Figures 6.20 and 6.21 are a selection of operators each with different rates of correct responses. As before, this data is taken from 30 runs of 100 target events each where each simulation method is given the same data at the same point with the same observations from the human operator. Additionally, the same sequence of targets are given to each human operator i.e. the 43rd target in run 11 will always be target type 2 for all operators. All figures are cumulative averages, meaning the percent correct is over all previous targets to that point. The expert operator, giving a correct observation 92.2% of the time (a). Operator 2, who gives a correct observation at the same rate as the HMM at 80.2% (b). Operator 3, who gives a correct observation slightly less than the HMM at 75.8% (c). And the novice operator, who gives correct observations significantly less than the HMM at 71.6%. These operators were chosen to show both the system's ability to adapt to different operators, but to demonstrate at what point a human stops being helpful given automation with a known accuracy.

With aid from the expert operator (Operator 1), targets can be classified correctly 90% of the



Figure 6.21: Cumulative averages for HMM prior to being passed off to operator (red), HMM without human intervention (purple), independent observations (green), parameter tied observation (blue), and full parameter observations (orange) for (a) Opeartor 3; (b) "novice" operator.

time, compared to the 80% of the time if the automation did not request help. It has been shown so far that the parameter tied model is able to characterize different operators and that it has significant computation advantages, but it also is able to classify with the same level of accuracy as the full parameter simulation. Both have the same accuracy at all points with the same variance. As discussed above, the independent model even with the expert operator, still does worse overall than the HMM alone.

Moving to Operator 2, we find that even though they give correct observations at the same rate as the HMM, the human can still provide a small increase in accuracy up to 83%. Once again, this is the same result even using the parameter tied simulation and the independent model under performs the HMM alone. Looking at the HMM prior to being passed off, it can be seen that it improves over time. As shown in the VOI section above, as the engine learns the response rates of the operator, it starts to pass off the classification later and later. As a result, the HMM receives more data and becomes more accurate on its own. This behavior is more pronounced for the least accurate/reliable operator.

Operator 3, who is slightly worse than the HMM, has the same accuracy overall as the HMM, but doesn't drop below the 80% average. The automation is able assess the operator, and despite receiving larger rates of incorrect observations, it still able to achieve the same level of accuracy. A



Figure 6.22: Confusion matricies for independent, parameter tied, and full parameter observation models. 30 runs of 100 target events each. "Novice" operator.

correct response rate of 75% seems to be an inflection point of when a human operator stops being helpful.

The novice operator, shown in figure (d), is the only one that consistently starts off poorly. Each graph starts from 5 targets (to get an average), but given that the Dirichlet priors are far too confident for this operator, it immediately gets the first few targets wrong. At this point, one can see that all lines, the tied, full, independent, and HMM prior to being handed off, all have the same level of accuracy continuing on. They also all approach the 80% mark of the HMM alone. This demonstrates the point at which the automation stops asking the operator for help, and starts classifying nearly all targets on its own. The reason the models doesn't immediately approach the HMM is because we are showing the cumulative percent correct. By getting the first few targets wrong, unless they are able to get a target correct that the HMM doesn't, they will never catch it. With the automation recognizing they should almost never ask the human operator for help, the best each model can do is asymptotically approach the same accuracy as the HMM; which after the first few targets, each model is essentially indistinguishable from the HMM.

Looking at the confusion matrices and the precision-recall curves, we see this same trend. In every case, the full parameter model and the parameter tied model both perform the same on



Figure 6.23: Precision-Recall curves for independent, parameter tied, and full parameter observation models "novice" operator.

each respective target. This validates that even with less total information in the parameter tied simulation, the accuracy of the system is not compromised. However, this does demonstrate the need for a competent human operator.

As the accuracy of any operator goes down, so does the accuracy of the system. While it has been shown that the system compensates for it by relying more on the automation, if the human operator doesn't give correct information, the system cannot improve. Eventually, if the operator isn't any better than the automation, the overall classification will be the same as the HMM. This also highlights the importance of the automation itself. A better classifier doesn't need to rely on a human's assistance if it is indeed more accurate. In the case that the automation needs help, it can only be improved by a human that is more accurate.

In the precision-recall curves, an interesting trend can be seen as the operators get less accurate. Previously, the independent observation model had a low precision due to overconfidence. As the human operators provide less useful information, both the parameter tied and full parameter simulation begin trending in the same direction. In particular, Figure 6.23 shows the same curve for all three models. With the 'novice' operator, the precision of each curve is decreased because the information the human provides isn't informative. With the 'expert' operator, the independent model suffered by assuming each observation was unique with new information that hadn't been



Figure 6.24: Confusion matricies for independent, parameter tied, and full parameter observation models (10 targets). 30 runs of 100 target events each. 10 target "expert" operator.

fused previously. When dealing with a below average operator, the independent model has the same poor information as the dependent models and doesn't suffer for it. Also as mentioned, when the pipeline stops asking the human operator for help, all of the models converge to the same point. In this case it would model the HMM if given enough time, but the precision-recall curves shown also include the initial observations that were neither accurate or informative.

6.7.3 Sensitivity to number of target types

As it has now been established that the parameter tied model is able to match the full parameter model, and that it is scalable for computation, the scalability of classification is considered next. When adding double the number of possible target types with similar intensity data, it can be seen that the HMM has a drop in accuracy. Once again using the expert operator, it can be seen in Figure 6.24 that both the full parameter and parameter tied models are able to provide improvements. Looking closer, there are a few targets the HMM classifies better without human intervention. The differences are small (i.e. 98% down to 96%), and looking at Figure 3.4 (b), it can be seen for some targets like 5, larger differences in the intensity data might come later in the lifespan of the target after the automation would have already passed it off. Prior to the final stage, the automation has little confidence in its assessment even though it might get it right if



Figure 6.25: Cumulative averages for HMM prior to being passed off to operator (red), HMM without human intervention (purple), independent observations (green), parameter tied observation (blue), and full parameter observations (orange) for 10 targets.

it happens to see the correct data that distinguishes it from other targets. The human operator wouldn't see anything different as far as the intensity data is concerned, but as stated before it is assumed the human operator is able to fuse other information such a regional differences or intel not available to the HMM. One key difference between 5 and 10 targets is that the independent model, for the first time, improves the classification accuracy. The independent model still faces the same problem of being overconfident, but in the scenario where there is more uncertainty in target classification, the observations provided by the expert operator are still useful even if they aren't modeled accurately.

Figure 6.25 confirms the same findings from the confusion matrices in Figure 6.18. The full parameter and parameter tied models have the same level of classification accuracy and while the independent model doesn't perform as well, it does provide improvement over the HMM alone. I can also be seen that even using the same expert operator and priors, the rise in complexity still results in a loss of accuracy. The HMM is only 69% accurate compared to 80%, and the tied and full sims are only 85% compared to 90% in the 5 target scenario. Even with the increase in difficulty, the human operator is still able to provide significant improvements in classification over



Figure 6.26: Precision-Recall curves for independent, parameter tied, and full parameter observation models. 10 targets "expert" operator.

the automation alone.

Figure 6.26 shows the same trend as the 5 target precision-recall curve. Both the parameter tied and full parameter models perform at the same level while the independent model suffers from overconfidence. Interestingly, even though the overall accuracy of the classification is reduced when increasing the number of targets, the area under the precision-recall curves aren't reduced. The classifications and their probabilities are still accurate, but when having to chose between 10 discrete choices, the classifier loses a small amount of accuracy. The important takeaway is that the system is still correctly fusing the information and providing accurate values to the human operator by conveying uncertainty.

6.8 Discussion

Use of the fusion engine to combine both machine learning and 'soft' information from human operators was able to improve accuracy in simulation for various target classification tasks. Under the assumptions of first order Markov dependence between observations, the system was able to outperform simpler naive Bayes models and while still being computationally practical. These improvements are also based on the assumption that human operators are able to access info or background knowledge that is independent of the information available to the HMM/ML portion of the pipeline.

The system was also able to accurately characterize multiple individual human sensors in an online environment and use this information to further improve classification either by requiring more/fewer observations or placing more weight on the automation. Aiding the computation speed up was the use of parameter tying, which also allowed for easy scalability without compromising any accuracy. The use of parameter tying in enabled by the specific observation types presented in this paper (TP, FP, FN, TN). Other observation types would require different parameter tying models.

The system was also able to intelligently ask for assistance by utilizing the learned human parameters and knowledge of gaps in the automation. Even in cases where an operator was significantly worse than the automation, the system was able to quickly identify this and rely solely on the automation, losing little accuracy at the beginning and establish a floor level of accuracy.

Chapter 7

Conclusion

This paper considered the problem of fusing observations provided by human operators with probabilistic information in a multi-target tracking and surveillance scenario. A new hierarchical fully Bayesian probabilistic model that accounted for observations that were not conditionally independent was proposed that also accounted for uncertainties in 'human sensor' quality and was shown to improve upon a conditionally independent implementation even with identical information. Through the use of Gibbs sampling, this framework was able to simultaneously update the belief of the human sensor parameters and the target type classification. Human parameters were modeled using Dirichlet distributions in order to allow for fast, online sampling and inference. Moment matching was shown to provide closed form solutions to the human sensor parameters, reducing the amount of information required to fully describe the system. Parameter tying was used to drastically reduce the number of parameters to infer and hold constant for any multi-target tracking scenario regardless of size. The system was shown to accurately characterize operators of different skill levels online using the same prior information. Finally, the system was able to able to adapt to different operators and intelligently ask for assistance through the use of VOI.

It has been shown that the convenient modeling assumption of conditional independence between human sensor updates can be removed and handled by using a different framework to fuse probabilistic information, while acknowledging the dependent nature of human sensors. While this leads to increases in computation and parameter dimensionality, the use of Gibbs sampling in concert with Dirichlet distributions, moment matching, and parameter tying allows for a robust fusion system capable of running online with little to no loss in information. This also allows not only the ability to run online with this framework, but also for individual operator characterization without any prior training and minimal/no a priori calibration. This individual characterization enables adaptive querying techniques to further improve performance/ensure baseline accuracy as good as the chosen machine learning implementation, which can easily be swapped out.

Future work will be focused on real human operators to characterize the differences actually seen between different operators, as well as enable a comparison of methodologies (i.e. conditionally dependent observations) outside of a simulated environment.

Bibliography

- N. Ahmed, M. Campbell, D. Casbeer, Y. Cao, and D. Kingston. Fully Bayesian learning and spatial reasoning with flexible human sensor networks. In <u>Proceedings of the 2015 Int'l Conf.</u> on Cyberphysical Systems (ICCPS 2015), pages accepted, to appear. ACM/IEEE, 2015.
- [2] Nisar Ahmed, David Casbeer, Yongcan Cao, and Derek Kingston. Bayesian hidden markov models for uav-enabled target localization on road networks with soft-hard data. In <u>Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR VI</u>, volume 9464, page 94640Q. International Society for Optics and Photonics, 2015.
- [3] Nisar Ahmed, Eric Sample, and Mark Campbell. Bayesian multi-categorical soft data fusion for human-robot collaboration. IEEE Transactions on Robotics, 29(1):189–206, 2013.
- [4] Adrian N Bishop and Branko Ristic. Fusion of natural language propositions: Bayesian random set framework. In <u>Information Fusion (FUSION)</u>, 2011 Proceedings of the 14th International Conference on, pages 1–8. IEEE, 2011.
- [5] C.M. Bishop. Pattern Recognition and Machine Learning. Springer, New York, 2006.
- [6] Frederic Bourgault, A Chokshi, J Wang, D Shah, Jonathan Schoenberg, R Iyer, F Cedano, and Mark Campbell. Scalable Bayesian human-robot cooperation in mobile sensor networks. In International Conference on Intelligent Robots and Systems, pages 2342–2349, 2008.
- [7] Luke Burks and Nisar Ahmed. Collaborative semantic data fusion with dynamically observable decision processes. In <u>2019 22nd International Conference on Information Fusion (FUSION)</u>. IEEE, 2019.
- [8] Luke Burks, Ian Loefgren, Luke Barbier, Jeremy Muesing, Jamison McGinley, Sousheel Vunnam, and Nisar Ahmed. Closed-loop bayesian semantic data fusion for collaborative human-autonomy target search. In <u>2018 21st International Conference on Information Fusion</u> (FUSION), pages 2262–2269. IEEE, 2018.
- [9] George Casella and Christian P Robert. <u>Monte Carlo Statistical methods</u>. Springer-Verlag, New York, 1999.
- [10] M.L. Cummings, C. Mastracchio, K.M. Thornburg, and A Mkrtchyan. Boredom and distraction in multiple unmanned vehicle supervisory control. <u>Interacting with Computers</u>, 25(1):34– 47, 2013.

- [11] A. Dani, M. McCourt, J.W. Curtis, and S.S. Mehta. Information fusion in human-robot collaboration using neural network representation. In <u>2014 IEEE Int'l Conf. on Systems, Man</u> and Cybernetics, pages 2114–2120. IEEE, 2014.
- [12] Michael A Goodrich, Bryan S Morse, C Engh, J L Cooper, and Julie A Adams. Towards using Unmanned Aerial Vehicles (UAVs) in Wilderness Search and Rescue. <u>Interaction Studies</u>, 10(3):453–478, 2009.
- [13] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In Proceedings of the 34th International Conference on Machine Learning-Volume 70, pages 1321–1330. JMLR. org, 2017.
- [14] D L Hall and J M Jordan. Human-centered Information Fusion. Artech House, 2010.
- [15] Michael Irwin Jordan. <u>Learning in graphical models</u>, volume 89. Springer Science & Business Media, 1998.
- [16] T Kaupp, B Douillard, F Ramos, A Makarenko, and B Upcroft. Shared Environment Representation for a Human-Robot Team Performing Information Fusion. <u>Journal of Field Robotics</u>, 24(11):911–942, 2007.
- [17] Tobias Kaupp, Alexei Makarenko, and Hugh Durrant-Whyte. Human-robot communication for collaborative decision making: A probabilistic approach. <u>Robotics and Autonomous Systems</u>, 58(5):444–456, May 2010.
- [18] Mykel J Kochenderfer. <u>Decision making under uncertainty: theory and application</u>. MIT press, 2015.
- [19] Michael Lewis, H Wang, P Velgapudi, P Scerri, and K Sycara. Using humans as sensors in robotic search. In <u>12th International Conference on Information Fusion (FUSION 2009)</u>, pages 1249–1256, Seattle, WA, 2009.
- [20] J.S. Liu. Monte Carlo Strategies in Scientific Computing. Springer, New York, 2001.
- [21] Kin Gwn Lore, Nicholas Sweet, Kundan Kumar, Nisar Ahmed, and Soumik Sarkar. Deep value of information estimators for collaborative human-machine information gathering. In <u>Proceedings of the ACM/IEEE Seventh Int'l Conf. on Cyber-Physical Systems (ICCPS 2016)</u>, pages 80–89. ACM, 2015.
- [22] Tobias P. Mann. Numerically stable hidden markov model implementation. Ms. Feb, 2006.
- [23] Simon Maskell. A bayesian approach to fusing uncertain, imprecise and conflicting information. Information Fusion, 9(2):259–277, 2008.
- [24] SS Mehta, Michael McCourt, EA Doucette, and JW Curtis. A touch interface for soft data modeling in bayesian estimation. In <u>2014 IEEE Int'l Conf. on Systems, Man and Cybernetics</u>, pages 3732–3737. IEEE, 2014.
- [25] Thomas P. Minka. Estimating a dirichlet distribution. Technical report, M.I.T., 11 2000.
- [26] Maximilian Panzner and Philipp Cimiano. Comparing hidden markov models and long short term memory neural networks for learning action representations. In <u>International Workshop</u> on Machine Learning, Optimization, and Big Data, pages 94–105. Springer, 2016.

- [27] Judea Pearl. <u>Probabilistic reasoning in intelligent systems: networks of plausible inference</u>. Elsevier, 2014.
- [28] Lawrence Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE, 77(2):257–286, 1989.
- [29] Christian Ruf and Peter Stutz. Establishing a variable automation paradigm for uav-based flight path reconnaissance in a military transport helicopter cockpit: Experimental evaluation of cockpit automation. pages 1–15. EAAP, 2018.
- [30] D. Wang, L. Kaplan, H. Le, and T. Abdelzaher. On truth discovery in social sensing: A maximum likelihood estimation approach. In Proc. of the 11th Int'l Conf. on Information Proc. in Sensor Networks, pages 233–244. ACM, 2012.