

Dynamic Bayesian Combination of Multiple Imperfect Classifiers

Edwin Simpson¹, Stephen Roberts¹, Ioannis Psorakis¹, Arfon Smith²

1. Department of Engineering Science, University of Oxford, UK.

Emails: {edwin, sjrob, psorakis}@robots.ox.ac.uk

2. Zooniverse. Email: arfon@zooniverse.org

June 8, 2012

Abstract

Classifier combination methods need to make best use of the outputs of multiple, imperfect classifiers to enable higher accuracy classifications. In many situations, such as when human decisions need to be combined, the base decisions can vary enormously in reliability. A Bayesian approach to such uncertain combination allows us to infer the differences in performance between individuals and to incorporate any available prior knowledge about their abilities when training data is sparse. In this paper we explore Bayesian classifier combination, using the computationally efficient framework of variational Bayesian inference. We apply the approach to real data from a large citizen science project, Galaxy Zoo Supernovae, and show that our method far outperforms other established approaches to imperfect decision combination. We go on to analyse the putative community structure of the decision makers, based on their inferred decision making strategies, and show that natural groupings are formed. Finally we present a dynamic Bayesian classifier combination approach and investigate the changes in base classifier performance over time.

1 Introduction

In many real-world scenarios we are faced with the need to aggregate information from cohorts of imperfect decision making agents (*base classifiers*), be they computational or human. Particularly in the case of human agents, we rarely have available to us an indication of how decisions were arrived at or a realistic measure of agent confidence in the various decisions. Fusing multiple sources of information in the presence of uncertainty is optimally achieved using Bayesian inference, which elegantly provides a principled mathematical framework for such knowledge aggregation. In this paper we provide a Bayesian framework for imperfect decision combination, where the base classifications we receive are greedy preferences (i.e. labels with no indication of confidence or uncertainty). The classifier combination method we develop aggregates the decisions of multiple agents, improving overall performance. We present a principled framework in which the use of weak decision makers can be mitigated and in which multiple agents, with very different observations, knowledge or training sets, can be combined to provide complementary information.

The preliminary application we focus on in this paper is a distributed *citizen science* project, in which human agents carry out classification tasks, in this case identifying transient objects from images as corresponding to potential supernovae or not. This application, *Galaxy Zoo Supernovae* [1], is part of the highly successful *Zooniverse* family of citizen science projects. In this application the ability of our base classifiers can be very varied and there is no guarantee over any individual's performance, as each user can have radically different levels of domain experience and have different background knowledge. As individual users are not overloaded with decision requests by the system, we often have little performance data for individual users (base classifiers). The methodology we advocate provides a scalable, computationally efficient, Bayesian approach to learning base classifier performance

thus enabling optimal decision combinations. The approach is robust in the presence of uncertainties at all levels and naturally handles missing observations, i.e. in cases where agents do not provide any base classifications. We develop extensions to allow for dynamic, sequential inference, through which we track information regarding the base classifiers. Through the application of social network analysis we can also observe behavioural patterns in the cohort of base classifiers.

The rest of this paper is organised as follows. In the remainder of the Introduction we briefly describe related work. In Section 2 we present a probabilistic model for independent Bayesian classifier combination, IBCC. Section 3 introduces the approximate inference method, variational Bayes, and details its application to IBCC. Section 4 shows an example application for classifier combination, Galaxy Zoo Supernovae, and compares results using different classifier combination methods, including IBCC. In Sections 5 and 6 we investigate how communities of decision makers with similar characteristics can be found using data inferred from Bayesian classifier combination. Section 7 presents an extension to independent Bayesian classifier combination that models the changing performance of individual decision makers. Using this extension, Section 8 examines the dynamics of individuals from our example application, while Sections 9 and 10 show how communities of decision makers change over time. Finally, Section 11 discusses future directions for this work.

1.1 Related Work

Previous work has often focused on aggregating expert decisions in fields such as medical diagnosis [2]. In contrast, *crowdsourcing* uses novice human agents to perform tasks that would be too difficult or expensive to process computationally or using experts [3, 4]. The underlying problem of fusing labels from multiple classifications has been dealt with in various ways and a review of the common methods is given by [5]. The choice of method typically depends on the type of labels we can obtain from agents (e.g. binary, continuous), whether we can manipulate agent performance, and whether we can also access input features. Weighted majority and weighted sum algorithms are popular methods that account for differing reliability in the base classifiers; an analysis of their performance is given by [6]. Bayesian model combination [7] provides a theoretical basis for soft-selecting from a space of combination functions. In most cases it outperforms Bayesian model averaging, which relies on one base classifier matching the data generating model. A well-founded model that learns the combination function directly was defined by [8], giving a Bayesian treatment to a model first presented in [2]. A similar model was also investigated by [9] with extensions to learn a classifier from expert labels rather than known ground truth labels. Both papers assume that base classifiers have constant performance, a problem that we address later in this paper.

2 Independent Bayesian Classifier Combination

Here we present a variant of Independent Bayesian Classifier Combination (IBCC), originally defined in [8]. The model assumes conditional independence between base classifiers, but performed as well as more computationally intense dependency modelling methods also given by [8].

We are given a set of data points indexed from 1 to N , where the i th data point has a true label t_i that we wish to infer. We assume t_i is generated from a multinomial distribution with the probabilities of each class denoted by $\kappa : p(t_i = j | \kappa) = \kappa_j$. True labels may take values $t_i = 1 \dots J$, where J is the number of true classes. We assume there are K base classifiers, which produce a set of discrete outputs \vec{c} with values $l = 1 \dots L$, where L is the number of possible outputs. The output $c_i^{(k)}$ from classifier k for data point i is assumed to be generated from a multinomial distribution dependent on the true label, with parameters $\pi_j^{(k)} : p(c_i^{(k)} = l | t_i = j, \pi_j^{(k)}) = \pi_{jl}^{(k)}$. This model places minimal requirements on the type of classifier output, which need not be probabilistic and could be selected from an arbitrary number of discrete values, indicating, for example, greedy preference over a set of class labels. Parameters $\pi_j^{(k)}$ and κ have Dirichlet prior distributions with hyperparameters $\alpha_{0,j}^{(k)} = [\alpha_{0,j1}^{(k)}, \dots, \alpha_{0,jL}^{(k)}]$ and $\nu = [\nu_{01}, \dots, \nu_{0J}]$ respectively. We refer to the set of $\pi_j^{(k)}$ for all base classifiers and all classes as $\Pi = \{ \pi_j^{(k)} | j = 1 \dots J, k = 1 \dots K \}$. Similarly, for the hyperparameters we use $\mathbf{A}_0 = \{ \alpha_{0,j}^{(k)} | j = 1 \dots J, k = 1 \dots K \}$

The joint distribution over all variables for the IBCC model is

$$p(\boldsymbol{\kappa}, \Pi, \vec{t}, \vec{c} | \mathbf{A}_0, \boldsymbol{\nu}) = \prod_{i=1}^N \left\{ \kappa_{t_i} \prod_{k=1}^K \pi_{t_i, c_i}^{(k)} \right\} p(\boldsymbol{\kappa} | \boldsymbol{\nu}) p(\Pi | \mathbf{A}), \quad (1)$$

The graphical model for IBCC is shown in Figure 1. A key feature of IBCC is that $\pi^{(k)}$ represents a *confusion*

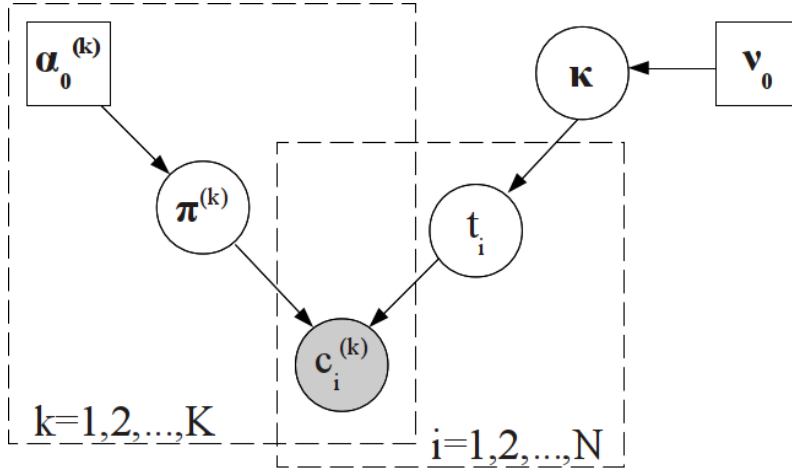


Figure 1: Graphical Model for IBCC. The shaded node represents observed values, circular nodes are variables with a distribution and square nodes are variables instantiated with point values.

matrix that quantifies the decision-making abilities of *each* base classifier. This potentially allows us to ignore or retrain poorer classifiers and assign expert decision makers to data points that are highly uncertain. Such efficient selection of base classifiers is vitally important when there is a cost to obtaining an output from a base classifier, for example, a financial payment to a decision maker per decision, or when the processing bandwidth of base classifiers is limited. The confusion matrices Π in IBCC also allow us to predict any missing classifier outputs in \vec{c} , so that we can naturally handle cases where only *partially observed* agents make decisions.

The IBCC model assumes independence between the rows in $\pi^{(k)}$, i.e. the probability of each classifier’s outputs is dependent on the *true label class*. In some cases it may be reasonable to assume that performance over one label class may be correlated with performance in another; indeed methods such as *weighted majority* [6] make this tacit assumption. However, we would argue that this is not universally the case, and IBCC makes no such strong assumptions.

The model here represents a simplification of that proposed in [8], which places exponential hyperpriors over \mathbf{A}_0 . The exponential distributions are not conjugate to the Dirichlet distributions over Π , so inference using *Gibbs Sampling* [10] requires an expensive adaptive rejection sampling step [11] for \mathbf{A} and the variational Bayesian solution is intractable. The conjugate prior to the Dirichlet is non-standard and its normalisation constant is not in closed form [12], so cannot be used. We therefore alter the model, to use point values for \mathbf{A}_0 , as in other similar models [13, 14, 15]. The hyperparameter values of \mathbf{A}_0 can hence be chosen to represent any prior level of uncertainty in the values of the agent-by-agent confusion matrices, Π , and can be regarded as pseudo-counts of prior observations, offering a natural method to include any prior knowledge and a methodology to extend the method to sequential, on-line environments.

3 Variational Bayesian IBCC

The goal of the combination model is to perform inference for the unknown variables \vec{t} , Π , and $\boldsymbol{\kappa}$. The inference technique proposed in [9] is maximum a posteriori (MAP) estimation, while [8] suggests a full Bayesian treatment

using Gibbs Sampling [10]. While the latter provides some theoretical guarantee of accuracy given the proposed model, it is often very slow to converge and convergence is difficult to ascertain. In this paper we consider the use of principled approximate Bayesian methods, namely *variational Bayes* (VB) [16] as this allows us to replace non-analytic marginal integrals in the original model with analytic updates in the *sufficient statistics* of the variational approximation. This produces a model that iterates rapidly to a solution in a computational framework which can be seen as a Bayesian generalisation of the *Expectation- Maximization* (EM) algorithm [17].

3.1 Variational Bayes

Given a set of observed data \mathbf{X} and a set of latent variables and parameters \mathbf{Z} , the goal of variational Bayes (VB) is to find a tractable approximation $q(\mathbf{Z})$ to the posterior distribution $p(\mathbf{Z}|\mathbf{X})$ by minimising the KL-divergence [18] between the approximate distribution and the true distribution [16, 19]. We can write the log of the model evidence $p(\mathbf{X})$ as

$$\begin{aligned} \ln p(\mathbf{X}) &= \int q(\mathbf{Z}) \ln \frac{p(\mathbf{X}, \mathbf{Z})}{q(\mathbf{Z})} d\mathbf{Z} - \int q(\mathbf{Z}) \ln \frac{p(\mathbf{Z}|\mathbf{X})}{q(\mathbf{Z})} d\mathbf{Z} \\ &= L(q) - \text{KL}(q||p). \end{aligned} \quad (2)$$

As $q(\mathbf{Z})$ approaches $p(\mathbf{Z}|\mathbf{X})$, the KL-divergence disappears and the lower bound $L(q)$ on $\ln p(\mathbf{X})$ is maximised. Variational Bayes selects a restricted form of $q(\mathbf{Z})$ that is tractable to work with, then seeks the distribution within this restricted form that minimises the KL-divergence. A common restriction is to partition \mathbf{Z} into groups of variables, then assume $q(\mathbf{Z})$ factorises into functions of single groups:

$$q(\mathbf{Z}) = \prod_{i=1}^M q_i(\mathbf{Z}_i). \quad (3)$$

For each factor $q_i(\mathbf{Z}_i)$ we then seek the optimal solution $q_i^*(\mathbf{Z}_i)$ that minimises the KL-divergence. Consider partitions of variables \mathbf{Z}_i and $\bar{\mathbf{Z}}_i$, where $\bar{\mathbf{Z}}_i = \{\mathbf{Z}_j | j \neq i, j = 1 \dots M\}$. *Mean field theory* [20] shows that we can find an optimal factor $q_i^*(\mathbf{Z}_i)$ from the conditional distribution $p(\mathbf{Z}_i|\mathbf{X}, \bar{\mathbf{Z}}_i)$ by taking the expectation over all the other factors $j | j \neq i, j = 1 \dots M$. We can therefore write the log of the optimal factor $\ln q_i^*(\mathbf{Z}_i)$ as the expectation with respect to all other factors of the log of the joint distribution over all variables plus a normalisation constant:

$$\ln q_i^*(\mathbf{Z}_i) = \mathbb{E}_{q(\bar{\mathbf{Z}}_i)}[\ln p(\mathbf{X}, \mathbf{Z})] + \text{const}. \quad (4)$$

In our notation, we take the expectation with respect to the variables in the subscript. In this case, $\mathbb{E}_{q(\bar{\mathbf{Z}}_i)}[\dots]$ indicates that we take an expectation with respect to all factors except $q(\mathbf{Z}_i)$. This expectation is implicitly conditioned on the observed data, \mathbf{X} , which we omit from the notation for brevity.

We can evaluate these optimal factors iteratively by first initialising all factors, then updating each in turn using the expectations with respect to the current values of the other factors. Unlike Gibbs sampling, each iteration is guaranteed to increase the lower bound on the log-likelihood, $L(q)$, converging to a (local) maximum in a similar fashion to standard EM algorithms. If the factors $q_i^*(\mathbf{Z}_i)$ are exponential family distributions [14], as is the case for the IBCC method we present in the next section, the lower bound is convex with respect to each factor $q_i^*(\mathbf{Z}_i)$ and $L(q)$ will converge to a *global* maximum of our approximate, factorised distribution. In practice, once the optimal factors $q_i^*(\mathbf{Z}_i)$ have converged to within a given tolerance, we can approximate the distribution of the unknown variables and calculate their expected values.

3.2 Variational Equations for IBCC

To provide a variational Bayesian treatment of IBCC, VB-IBCC, we first propose the form for our variational distribution, $q(\mathbf{Z})$, that factorises between the parameters and latent variables.

$$q(\boldsymbol{\kappa}, \vec{t}, \Pi) = q(\vec{t})q(\boldsymbol{\kappa}, \Pi) \quad (5)$$

This is the only assumption we must make to perform VB on this model; the forms of the factors arise from our model of IBCC. We can use the joint distribution in Equation (1) to find the optimal factors $q^*(\vec{t})$ and $q^*(\boldsymbol{\kappa}, \Pi)$ in the form given by Equation (4). For the true labels we have

$$\ln q^*(\vec{t}) = \mathbb{E}_{\boldsymbol{\kappa}, \Pi}[\ln p(\boldsymbol{\kappa}, \vec{t}, \Pi, \vec{c})] + \text{const.} \quad (6)$$

We can rewrite this into factors corresponding to independent data points, with any terms not involving t_i being absorbed into the normalisation constant. To do this we define ρ_{ij} as

$$\begin{aligned} \ln \rho_{ij} &= \mathbb{E}_{\boldsymbol{\kappa}_j, \boldsymbol{\pi}_j}[\ln p(\boldsymbol{\kappa}_j, t_i, \boldsymbol{\pi}_j, \vec{c})] \\ &= \mathbb{E}_{\boldsymbol{\kappa}}[\ln \kappa_j] + \sum_{k=1}^K \mathbb{E}_{\boldsymbol{\pi}_j}[\ln \pi_{j, c_i^{(k)}}^{(k)}] \end{aligned} \quad (7)$$

then we can estimate the probability of a true label, which also gives its expected value:

$$q^*(t_i = j) = \mathbb{E}_{\vec{t}}[t_i = j] = \frac{\rho_{ij}}{\sum_{l=1}^J \rho_{il}}. \quad (8)$$

To simplify the optimal factors in subsequent equations, we define expectations with respect to \vec{t} of the number of occurrences of each true class, given by

$$N_j = \sum_{i=1}^N \mathbb{E}_{\vec{t}}[t_i = j], \quad (9)$$

and the counts of each classifier decision $c_i^{(k)} = l$ given the true label $t_i = j$, by

$$N_{jl}^{(k)} = \sum_{i=1}^N \delta_{c_i^{(k)} l} \mathbb{E}_{\vec{t}}[t_i = j] \quad (10)$$

where $\delta_{c_i^{(k)} l}$ is unity if $c_i^{(k)} = l$ and zero otherwise.

For the parameters of the model we have the optimal factors given by:

$$\ln q^*(\boldsymbol{\kappa}, \Pi) = \mathbb{E}_{\vec{t}}[\ln p(\boldsymbol{\kappa}, \vec{t}, \Pi, \vec{c})] + \text{const} \quad (11)$$

$$\begin{aligned} &= \mathbb{E}_{\vec{t}} \left[\sum_{i=1}^N \ln \kappa_{t_i} + \sum_{i=1}^N \sum_{k=1}^K \ln \pi_{t_i, c_i^{(k)}}^{(k)} \right] + \ln p(\boldsymbol{\kappa} | \boldsymbol{\nu}_0) \\ &\quad + \ln p(\Pi | \mathbf{A}_0) + \text{const.} \end{aligned} \quad (12)$$

In Equation (11) terms involving $\boldsymbol{\kappa}$ and terms involving each confusion matrix in Π are separate, so we can factorise $q^*(\boldsymbol{\kappa}, \Pi)$ further into

$$q^*(\boldsymbol{\kappa}, \Pi) = q^*(\boldsymbol{\kappa}) \prod_{k=1}^K \prod_{j=1}^J q^*(\boldsymbol{\pi}_j^{(k)}). \quad (13)$$

In the IBCC model (Section 2) we assumed a Dirichlet prior for $\boldsymbol{\kappa}$, which gives us the optimal factor

$$\ln q^*(\boldsymbol{\kappa}) = \mathbb{E}_{\vec{t}} \left[\sum_{i=1}^N \ln \kappa_{t_i} \right] + \ln p(\boldsymbol{\kappa} | \boldsymbol{\nu}_0) + \text{const} \quad (14)$$

$$= \sum_{j=1}^J N_j \ln \kappa_j + \sum_{j=1}^J (\nu_{0,j} - 1) \ln \kappa_j + \text{const.} \quad (15)$$

Taking the exponential of both sides, we obtain a posterior Dirichlet distribution of the form

$$q^*(\boldsymbol{\kappa}) \propto \text{Dir}(\boldsymbol{\kappa} | \nu_1, \dots, \nu_J) \quad (16)$$

where ν is updated in the standard manner by adding the data counts to the prior counts ν_0 :

$$\nu_j = \nu_{0,j} + N_j. \quad (17)$$

The expectation of $\ln \kappa$ required to update Equation (8) is therefore:

$$\mathbb{E}[\ln \kappa_j] = \Psi(\nu_j) - \Psi\left(\sum_{\iota=1}^J \nu_\iota\right) \quad (18)$$

where Ψ is the standard digamma function [21].

For the confusion matrices $\pi_j^{(k)}$ the priors are also Dirichlet distributions giving us the factor

$$\ln q^*\left(\pi_j^{(k)}\right) = \sum_{i=1}^N \mathbb{E}_{t_i}[t_i = j] \ln \pi_{j,c_i^{(k)}}^{(k)} + \ln p\left(\pi_j^{(k)} | \alpha_{0,j}^{(k)}\right) + \text{const} \quad (19)$$

$$= \sum_{l=1}^L N_{jl}^{(k)} \ln \pi_{jl}^{(k)} + \sum_{l=1}^L \left(\alpha_{0,jl}^{(k)} - 1\right) \ln \pi_{jl}^{(k)} + \text{const}. \quad (20)$$

Again, taking the exponential gives a posterior Dirichlet distribution of the form

$$q^*\left(\pi_j^{(k)}\right) = \text{Dir}\left(\pi_j^{(k)} | \alpha_{j1}^{(k)}, \dots, \alpha_{jL}^{(k)}\right) \quad (21)$$

where $\alpha_j^{(k)}$ is updated by adding data counts to prior counts $\alpha_{0,j}^{(k)}$:

$$\alpha_{jl}^{(k)} = \alpha_{0,jl}^{(k)} + N_{jl}^{(k)}. \quad (22)$$

The expectation required for Equation (8) is given by

$$\mathbb{E}\left[\ln \pi_{jl}^{(k)}\right] = \Psi\left(\alpha_{jl}^{(k)}\right) - \Psi\left(\sum_{m=1}^L \alpha_{jm}^{(k)}\right). \quad (23)$$

To apply the VB algorithm to IBCC, we first choose an initial value for all variables $\mathbb{E}[\ln \pi_{jl}^{(k)}]$ and $\mathbb{E}[\ln \kappa_j]$ either randomly or by taking the expectations of the variables over their prior distributions (if we have enough domain knowledge to set informative priors). We then iterate over a two-stage procedure similar to the *Expectation-Maximization* (EM) algorithm. In the variational equivalent of the *E-step* we use the current expected parameters, $\mathbb{E}[\ln \pi_{jl}^{(k)}]$ and $\mathbb{E}[\ln \kappa_j]$, to update the variational distribution in Equation (5). First we evaluate Equation (8), then use the result to update the counts N_j and $N_{jl}^{(k)}$ according to Equations (9) and (10). In the variational *M-step*, we update $\mathbb{E}[\ln \pi_{jl}^{(k)}]$ and $\mathbb{E}[\ln \kappa_j]$ using Equations (18) and (23).

3.3 Variational Lower Bound

To check for convergence we can also calculate the lower bound $L(q)$ (see Equation (3)), which should always increase after a pair of *E-step* and *M-step* updates. While we could alternatively detect convergence of the expectation over the latent variables, the variational lower bound is a useful sanity check for our derivation of VB-IBCC and for its implementations.

$$\begin{aligned} L(q) &= \iiint q(\vec{t}, \Pi, \kappa) \ln \frac{p(\vec{c}, \vec{t}, \Pi, \kappa | \mathbf{A}_0, \nu_0)}{q(\vec{t}, \Pi, \kappa)} d\vec{t} d\Pi d\kappa \\ &= \mathbb{E}_{\vec{t}, \Pi, \kappa} [\ln p(\vec{c}, \vec{t}, \Pi, \kappa | \mathbf{A}_0, \nu_0)] - \mathbb{E}_{\vec{t}, \Pi, \kappa} [\ln q(\vec{t}, \Pi, \kappa)] \\ &= \mathbb{E}_{\vec{t}, \Pi} [\ln p(\vec{c} | \vec{t}, \Pi)] + \mathbb{E}_{\vec{t}, \kappa} [\ln p(\vec{t} | \kappa)] + \mathbb{E}_{\Pi} [\ln p(\Pi | \alpha_0)] + \mathbb{E}_{\kappa} [\ln p(\kappa | \nu_0)] \\ &\quad - \mathbb{E}_{\vec{t}, \Pi, \kappa} [\ln q(\vec{t})] - \mathbb{E}_{\vec{t}, \Pi} [\ln q(\Pi)] - \mathbb{E}_{\vec{t}, \kappa} [\ln q(\kappa)] \end{aligned} \quad (24)$$

The expectation terms relating to the joint probability of the latent variables, observed variables and the parameters are

$$\begin{aligned}\mathbb{E}_{\vec{t}, \Pi}[\ln p(\vec{c}|\vec{t}, \Pi)] &= \sum_{i=1}^N \sum_{k=1}^K \sum_{j=1}^J \mathbb{E}[t_i = j] \mathbb{E} \left[\ln \pi_{j c_i^{(k)}}^{(k)} \right] \\ &= \sum_{k=1}^K \sum_{j=1}^J \sum_{l=1}^L N_{jl}^{(k)} \mathbb{E} \left[\ln \pi_{j c_i^{(k)}}^{(k)} \right]\end{aligned}\quad (25)$$

$$\begin{aligned}\mathbb{E}_{\vec{t}, \kappa}[\ln p(\vec{t}|\kappa)] &= \sum_{i=1}^N \sum_{j=1}^J \mathbb{E}[t_i = j] \mathbb{E}[\ln \kappa_j] \\ &= \sum_{j=1}^J N_j \mathbb{E}[\ln \kappa_j]\end{aligned}\quad (26)$$

$$\mathbb{E}_{\Pi}[\ln p(\Pi|\mathbf{A}_0)] = \sum_{k=1}^K \sum_{j=1}^J \left\{ -\ln \mathbf{B}(\boldsymbol{\alpha}_{0,j}^{(k)}) + \sum_{l=1}^L (\alpha_{0,jl}^{(k)} - 1) \mathbb{E}[\ln \pi_{jl}^{(k)}] \right\}\quad (27)$$

$$\mathbb{E}_{\kappa}[\ln p(\kappa|\nu_0)] = -\ln \mathbf{B}(\nu_0) + \sum_{j=1}^J (\nu_{0,j} - 1) \mathbb{E}[\ln \kappa_j]\quad (28)$$

where $\mathbf{B}(\mathbf{a}) = \frac{\prod_{l=1}^L \Gamma(a_l)}{\Gamma(\sum_{l=1}^L a_l)}$ is the Beta function and $\Gamma(a)$ is the Gamma function [21]. Terms in the lower bound relating to the expectation of the variational distributions \mathbf{q} are

$$\mathbb{E}_{\vec{t}, \Pi, \kappa}[\ln q(\vec{t})] = \sum_{i=1}^N \sum_{j=1}^J \mathbb{E}[t_i = j] \ln \mathbb{E}[t_i = j]\quad (29)$$

$$\mathbb{E}_{\vec{t}, \Pi}[\ln q(\Pi)] = \sum_{k=1}^K \sum_{j=1}^J \left\{ -\ln \mathbf{B}(\boldsymbol{\alpha}_j^{(k)}) + \sum_{l=1}^L (\alpha_{j,l}^{(k)} - 1) \mathbb{E}[\ln \pi_{j,l}^{(k)}] \right\}\quad (30)$$

$$\mathbb{E}_{\vec{t}, \kappa}[\ln q(\kappa)] = -\ln \mathbf{B}(\mathbf{N} + \nu_0) + \sum_{j=1}^J (N_j + \nu_{0,j} - 1) \mathbb{E}[\ln \kappa_j]\quad (31)$$

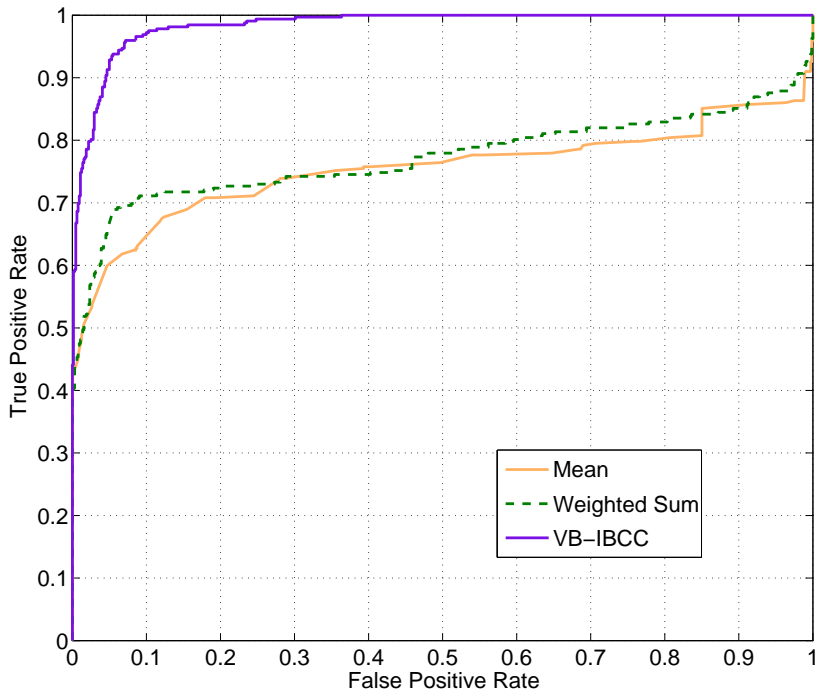
where $\mathbf{N} = [N_1, \dots, N_J]$ is a vector of counts for each true class.

Using these equations, the lower bound can be calculated after each pair of *E-step* and *M-step* steps. Once the value of the lower bound stops increasing the algorithm has converged to the optimal approximate solution.

4 Galaxy Zoo Supernovae

We tested the model using a dataset obtained from the Galaxy Zoo Supernovae citizen science project [1]. The aim of the project is to classify candidate supernova images as either “supernova” or “not supernova”. The dataset contains scores given by individual volunteer citizen scientists (base classifiers) to candidates after answering a series of questions. A set of three linked questions are answered by the users, which are hard-coded in the project repository to scores of -1, 1 or 3, corresponding respectively to decisions that the data point is very unlikely to be a supernova, possibly a supernova and very likely a supernova. These scores are our base classifier outputs \vec{c} .

In order to verify the efficacy of our approach and competing methods, we use “true” target classifications obtained from full spectroscopic analysis, undertaken as part of the Palomar Transient Factory collaboration [22]. We note that this information, is not available to the base classifiers (the users), being obtained retrospectively. We compare IBCC using both variational Bayes (VB-IBCC) and Gibbs sampling (Gibbs-IBCC), using as output the expected values of t_i . We also tested simple majority voting, weighted majority voting & weighted sum [6] and



(a) Receiver operating characteristic (ROC) curves.

Method	AUC
Mean of Scores	0.7543
Weighted Sum	0.7722
Simple Majority	0.7809
Weighted Majority	0.7378
Gibbs-IBCC	0.9127
VB-IBCC	0.9840

(b) Area under the ROC curves (AUCs).

Figure 2: Galaxy Zoo Supernovae: ROC curves and AUCs with 5-fold cross validation.

mean user scores, which the Galaxy Zoo Supernovae currently uses to filter results. For majority voting methods we treat both 1 and 3 as a vote for the supernova class.

The complete dataset contains many volunteers that have provided very few classifications, particularly for positive examples, as there are 322 classifications of positive data points compared to 43941 “not supernova” examples. Candidate images vary greatly in how difficult they are to classify, so volunteers who have classified small numbers of positive examples may have seen only easy or difficult examples, leading us to infer biased confusion matrices. Including large numbers of volunteers with little data will also affect our inference over true labels and confusion matrices of other decision makers. Therefore, we perform inference over a subsample of the data. Inferred parameters can be used to update the hyperparameters before running the algorithm again over other data point. To infer confusion matrices accurately, we require sufficient numbers of examples for both positive and negative classes. We therefore first select all volunteers that have classified at least 50 examples of each class, then select all data points that have been classified by at least 10 such volunteers; we then include other volunteers that have classified the expected examples. This process produced a data set of 963 examples with decisions produced from 1705 users. We tested the imperfect decision combination methods using five-fold cross validation. The dataset is divided randomly into five partitions, then the algorithm is run five times, each with a different partition designated as the test data and the other partitions used as training data. In the test partition the true labels are withheld from our algorithms and are used only to measure performance.

Figure 2a shows the average *Receiver-Operating Characteristic* (ROC) curves [23] taken across all cross-validation datasets for the mean score, weighted sum and VB-IBCC. Each point on the ROC curve corresponds to a different threshold value; classifier output values above a given threshold are taken as positive classifications and those below as negative. At each threshold value we calculate a true positive rate – the fraction of positive candidate images correctly identified – and a false positive rate – the fraction of negative candidates incorrectly

classified as positive.

The ROC curve for VB-IBCC clearly outperforms the mean of scores by a large margin. Weighted sum achieves a slight improvement on the mean by learning to discount base classifiers each time they make a mistake. The performance of the majority voting methods and IBCC using Gibbs sampling is summarised by the area under the ROC curve (AUC) in Table 2b. The AUC gives the probability that a randomly chosen positive instance is ranked higher than a randomly chosen negative instance. Majority voting methods only produce one point on the ROC curve between 0 and 1 as they convert the scores to votes (-1 becomes a negative vote, 1 and 3 become positive) and produce binary outputs. These methods have similar results to the mean score approach, with the weighted version performing slightly worse, perhaps because too much information is lost when converting scores to votes to be able to learn base classifier weights correctly.

With Gibbs-sampling IBCC we collected samples until the mean of the sample label values converged. Convergence was assumed when the total absolute difference between mean sample labels of successive iterations did not exceed 0.01 for 20 iterations. The mean time taken to run VB-IBCC to convergence was 13 seconds, while for Gibbs sampling IBCC it was 349 seconds. As well as executing significantly faster, VB produces a better AUC than Gibbs sampling with this dataset. Gibbs sampling was run to thousands of iterations with no change in performance observed. Hence it is likely that the better performance of the approximate variational Bayes results from the nature of this dataset; Gibbs sampling may provide better results with other applications but suffers from higher computational costs.

5 Communities of Decision Makers Based on Confusion Matrices (π Communities)

In this section we apply a recent community detection methodology to the problem of determining most likely groupings of base classifiers, the imperfect decision makers. Grouping decision makers allows us to observe the behaviours present in our pool of base classifiers and could influence how we allocate classification tasks or train base classifiers. Community detection is the process of clustering a “similarity” or “interaction” network, so that classifiers within a given group are more strongly connected to each other than the rest of the graph. Identifying overlapping communities in networks is a challenging task. In recent work [24] we presented a novel approach to community detection that infers such latent groups in the network by treating communities as explanatory latent variables for the observed connections between nodes, so that the stronger the similarity between two decision makers, the more likely it is that they belong to the same community. Such latent grouping is extracted by an appropriate factorisation of the connectivity matrix, where the effective inner rank (number of communities) is inferred by placing shrinkage priors [25] on the elements of the factor matrices. The scheme has the advantage of soft-partitioning solutions, assignment of node participation scores to communities, an intuitive foundation and computational efficiency.

We apply the approach described in [24] to a similarity matrix calculated over all the citizen scientists in our study, based upon the expected values of each users’ confusion matrix. Expectations are taken over the distributions of the confusion matrices inferred using the variational Bayesian method in Section 3.2 and characterise the behaviour of the base classifiers. Denoting $\mathbb{E}[\pi^{(i)}]$ as the (3×2) confusion matrix inferred for user i we may define a simple similarity measure between agents m and n as

$$V_{m,n} = \exp \left(- \sum_{j=1}^J \mathcal{H} \mathcal{D} \left(\mathbb{E}[\pi_j^{(m)}], \mathbb{E}[\pi_j^{(n)}] \right) \right), \quad (32)$$

where $\mathcal{H} \mathcal{D}()$ is the *Hellinger distance* between two distributions, meaning that two agents who have very similar confusion matrices will have high similarity. Since the confusion matrices are multinomial distributions, Hellinger distance is calculated as:

$$\mathcal{H} \mathcal{D} \left(\mathbb{E}[\pi_j^{(m)}], \mathbb{E}[\pi_j^{(n)}] \right) = 1 - \sum_{l=1}^L \sqrt{\mathbb{E}[\pi_{jl}^{(m)}] \mathbb{E}[\pi_{jl}^{(n)}]} \quad (33)$$

As confusion matrices represent probability distributions, so Hellinger distance is chosen as an established, symmetrical measure of similarity between two probability distributions [14]. Taking the exponential of the negative

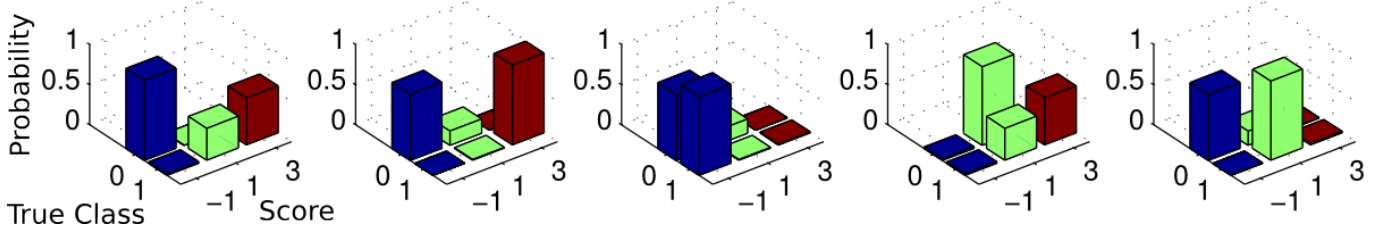


Figure 3: Prototypical confusion matrices for each of the five communities inferred using Bayesian social network analysis. Each graph corresponds to the most central individual in a community, with bar height indicating probability of producing a particular score for a candidate of the given true class.

Hellinger distance converts the distance measure to a similarity measure with a maximum of 1, emphasising cases of high similarity.

Application of Bayesian community detection to the matrix \mathbf{V} robustly gave rise to *five* distinct groupings of users. In Figure 3 we show the centroid confusion matrices associated with each of these groups of citizen scientists. The centroids are the expected confusion matrices of the individuals with the highest node participation scores for each community. The labels indicate the “true” class (0 for “not supernova” or 1 for “supernova”) and the preference for the three scores offered to each user by the Zooniverse questions (-1, 1 & 3). Group 1, for example, indicates users who are clear in their categorisation of “not supernova” (a score of -1) but who are less certain regarding the “possible supernova” and “likely supernova” categories (scores 1 & 3). Group 2 are “extremists” who use little of the middle score, but who confidently (and correctly) use scores of -1 and 3. By contrast group 3 are users who almost always use score -1 (“not supernova”) whatever objects they are presented with. Group 4 almost never declare an object as “not supernova” (incorrectly) and, finally, group 5 consists of “non-committal” users who rarely assign a score of 3 to supernova objects, preferring the middle score (“possible supernova”). It is interesting to note that all five groups have similar numbers of members (several hundred) but clearly each group indicates a very different approach to decision making.

6 Common Task Communities

In this section we examine groups of decision makers that have completed classification tasks for similar sets of objects, which we label *common task communities*. Below we outline how these communities and the corresponding confusion matrices could inform the way we allocate tasks and train decision makers. Intelligent task assignment could improve our knowledge of confusion matrices, increase the independence of base classifiers selected for a task, and satisfy human agents who prefer to work on certain types of task. We apply the overlapping community detection method [24] to a co-occurrence network for the Galaxy Zoo Supernovae data. Edges connect citizen scientists that have completed a common task, where edge weights w_{mn} reflect the proportion of tasks common to both individuals, such that

$$w_{mn} = \frac{\text{number_of_common_tasks}(m, n)}{0.5(N^{(m)} + N^{(n)})}, \quad (34)$$

where $N^{(k)}$ is the total number of observations seen by base classifier k . The normalisation term reduces the weight of edges from decision makers that have completed large numbers of tasks, as these would otherwise have very strong links to many others that they proportionally have little similarity to. The edge weights capture the correlation between the tasks that individuals have completed and give the expectation that for a classifier label $c_i^{(m)}$ chosen randomly from our sample, the classifier n will also have produced a label $c_i^{(n)}$. It is possible to place a prior distribution over these weights to provide a fully Bayesian estimate of the probability of classifiers completing the same task. However, this would not affect the results of our community analysis method, which uses single similarity values for each pair of nodes. For decision makers that have made few classifications, edge weights may

be poor estimates of similarity and thus introduce noise into the network. We therefore filter out decision makers that have made fewer than 10 classifications.

The algorithm found 32 communities for 2131 citizen scientists and produced a strong community structure with modularity of 0.75. Modularity is a measure between -1 and 1 that assumes a strong community structure has more intra-community edges (edges that connect nodes in the same community) than inter-community edges. It is the fraction of intra-community edges minus the expected fraction of intra-community edges for a random graph with the same node degree distribution [26]. In Galaxy Zoo Supernovae, this very modular community structure may arise through users with similar preferences or times of availability being assigned to the same objects. Galaxy Zoo Supernovae currently prioritises the oldest objects that currently lack a sufficient number of classifications and assigns these to the next available citizen scientists. It also allows participants to reject tasks if desired. Possible reasons for rejecting a task are that the decision maker finds the task too difficult or uninteresting. Common task communities may therefore form where decision makers have similar abilities, preferences for particular tasks (e.g. due to interesting features in an image) or are available to work at similar times. When considering the choice of decision makers for a task, these communities could therefore inform who is likely to be available and who will complete a particular task.

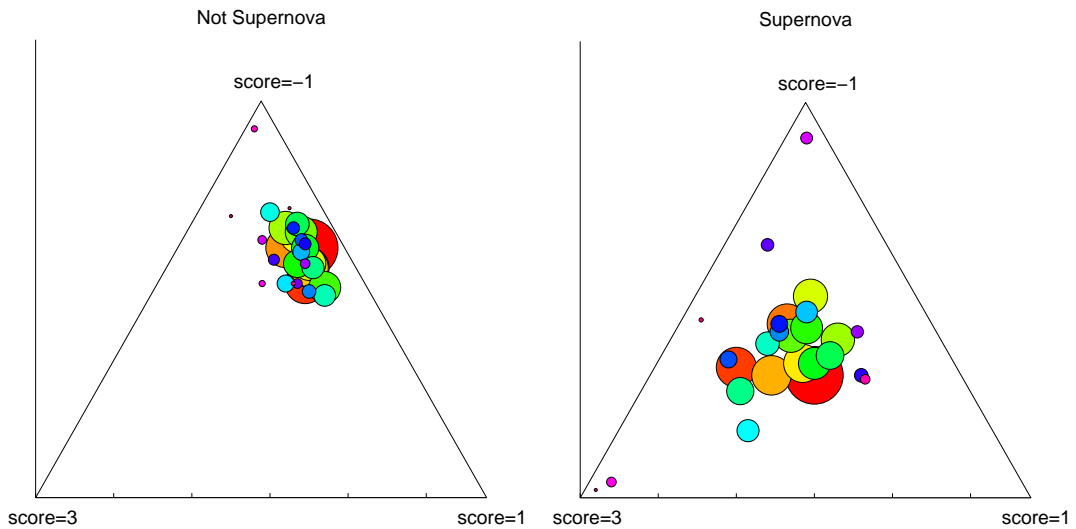


Figure 4: Distribution of means of community members’ confusion matrices for all common task communities. Proximity to a vertex indicates the probability of a score given an object with the stated true label class, e.g. in the graph labelled “Supernova”, a point near the vertex labelled “score=-1” indicates a very high probability of giving a decision of -1 when presented with images of a genuine supernova. The left-hand plot shows the mean confusion matrices for $t = 0$, i.e. the class “not a supernova”; the right-hand plot shows the confusion matrices for $t = 1$ or “supernova”. The size of the nodes indicates the number of members of the cluster.

In Figure 4 we plot the distribution of the means of the community members’ confusion matrices for each of the true classes. Differences between communities for $t = 0$ (not supernova class) are less pronounced than for the $t = 1$ (the supernova class). For the latter class we have 5134 observations as opposed to 48791 for the former, so that decision makers see fewer tasks with true label “supernova”. This means that individual tasks with features that are more likely to elicit a certain base classifier response can have a greater effect on the confusion matrices learned. For instance, some tasks may be easier to classify than others or help a decision maker learn through the experience of completing the task, thus affecting the confusion matrices we infer. As we would expect, some smaller communities have more unusual means as they are more easily influenced by a single community member. The effect of this is demonstrated by the difference between community means in Figure 4 for groups of decision makers that have completed common sets of tasks.

7 Dynamic Bayesian Classifier Combination

In real-world applications such as Galaxy Zoo Supernovae, confusion matrices can change over time as the imperfect decision makers learn and modify their behaviour. We propose a dynamic variant of IBCC, *DynIBCC*, that models the change each time a decision maker performs a classification task. Using these dynamic confusion matrices we can also observe the effect of each observation on our distribution over a confusion matrix.

In DynIBCC, we replace the simple update step for α_j given by Equation (22) with an update for every sample classified at time-steps denoted by τ , giving time-dependent parameters $\alpha_{\tau,j}$. Figure 5 shows the graphical model for DynIBCC. As we detail in this section, the values of $\alpha_{\tau,j}$ are determined directly (rather than generated from a distribution) from the values of Π for the previous and subsequent samples seen by each base classifier k . We use a *dynamic generalised linear model* [27], which enables us to iterate through the data updating α_τ depending on the previous value $\alpha_{\tau-1}$. This is the forward pass which operates according to Kalman filter update equations. We then use a Modified Bryson-Frazier smoother [28] to scroll backward through the data, updating α_τ based on the subsequent value $\alpha_{\tau+1}$. The backward pass is an extension to the work in [29], where updates are dependent only on earlier values of α . DynIBCC hence enables us to exploit a fully Bayesian model for dynamic classifier combination, placing distributions over $\pi_{\tau,j}$, while retaining computational tractability by using an approximate method to update the hyperparameters at each step.

The base classifier k may not classify the samples in the order given by their global indexes $i = 1, \dots, N$, so we map global indexes to time-steps $\tau = 1, \dots, T^{(k)}$ using

$$\tau_i^{(k)} = \mathbf{f}^{(k)}(i) \quad (35)$$

The mapping $\mathbf{f}^{(k)}$ records the order that k classified items, with time-step τ being the time-step that sample i was classified. For an object i_{unseen} not classified by k , $\mathbf{f}^{(k)}(i_{unseen}) = \emptyset$. The inverse of $\mathbf{f}^{(k)}(\tau)$ is $i_\tau^{(k)} = \mathbf{f}^{-1(k)}(\tau)$, a mapping from the time-step τ to the object i that was classified at that time-step.

The dynamic generalised linear model allows us to estimate the probability of a classifier producing output l given true label t_{i_τ} :

$$\tilde{\pi}_{\tau,l} = \pi_{\tau,t_{i_\tau}l} = p(c_{i_\tau} = l | t_{i_\tau}) \quad (36)$$

in which we omit the superscripts (k) for clarity. We first specify our *generalised linear model* [29] by defining a basis function model with the form

$$\tilde{\pi}_{\tau,l} = g(\mathbf{h}_\tau^T \mathbf{w}_{\tau,l}) \quad (37)$$

where \mathbf{h}_τ is a binary input vector of size J , with $h_{\tau,t_{i_\tau}} = 1$ and all other values equal to zero, i.e. a binary vector representation of $t_{i_\tau} = j$. The function $g(\cdot)$ is an activation function that maps the linear predictor $\eta_{\tau,l} = \mathbf{h}_\tau^T \mathbf{w}_{\tau,l}$ to the probability $\tilde{\pi}_{\tau,l}$ of base classifier response l . If we consider each possible classifier output l separately, the value $\tilde{\pi}_{\tau,l}$ is the probability of producing output l and can be seen as the parameter to a binomial distribution. Therefore $g(\cdot)$ is the logistic function, hence

$$\tilde{\pi}_{\tau,l} = \frac{\exp(\eta_{\tau,l})}{1 + \exp(\eta_{\tau,l})} \quad (38)$$

and its inverse, the canonical link function, is the logit function:

$$\eta_{\tau,l} = \text{logit}(\tilde{\pi}_{\tau,l}) = \log\left(\frac{\tilde{\pi}_{\tau,l}}{1 - \tilde{\pi}_{\tau,l}}\right) \quad (39)$$

In the *dynamic* generalised linear model [27, 29], we track changes to the distribution over $\tilde{\pi}_{\tau,l}$ over time by treating $\mathbf{w}_{\tau,l}$ as a state variable that evolves according to a random walk

$$\mathbf{w}_{\tau,l} = \mathbf{w}_{\tau-1,l} + \mathbf{v}_{\tau,l} \quad (40)$$

where $\mathbf{v}_{\tau,l}$ is the state noise vector that corresponds to the drift in the state variable over time. We assume that state noise has a distribution where only the first two moments are known, $\mathbf{v}_{\tau,l} \sim (\mathbf{0}, q_{\tau,l} \mathbf{I})$, where \mathbf{I} is the identity matrix. The state noise variance $q_{\tau,l}$ will be estimated from our distributions over $\tilde{\pi}_{\tau,l}$, as explained below.

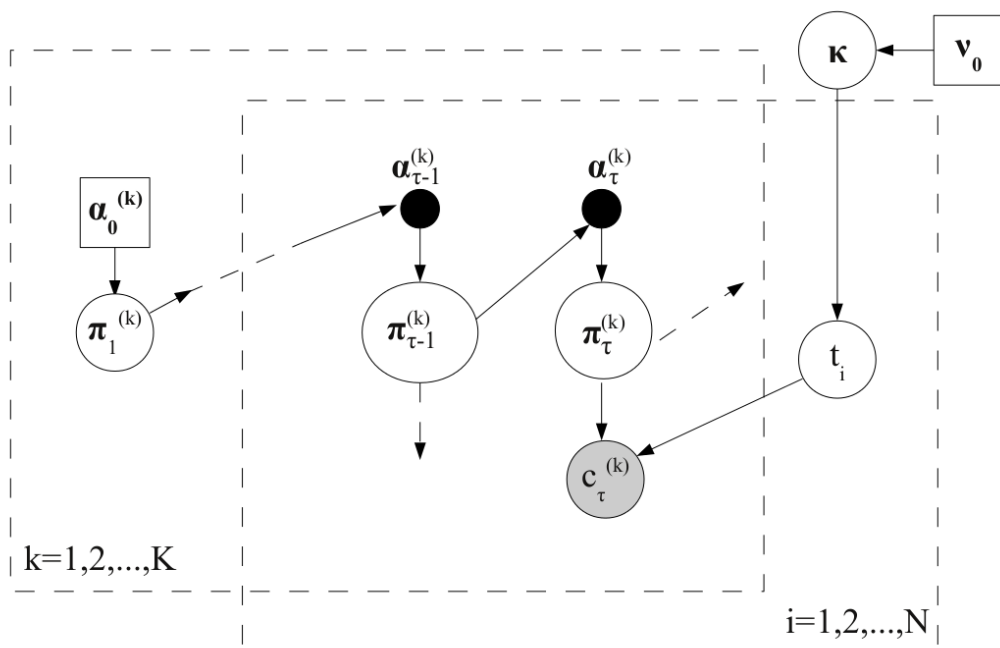


Figure 5: Graphical model for DynIBCC. The dashed arrows indicate dependencies to nodes at previous or subsequent time-steps. Solid black circular nodes are variables calculated deterministically from their predecessors. The shaded node represents observed values, circular nodes are variables with a distribution and square nodes are variables instantiated with point values.

7.1 Prior Distributions over the State Variables

Here we consider Bayesian inference over the state variable $\mathbf{w}_{\tau,l}$. As we place a distribution over $\tilde{\pi}_{\tau,l}$ in our model, we also have a distribution over $\mathbf{w}_{\tau,l}$. Our sequential inference method is approximate since we only estimate the mean and covariance over $\mathbf{w}_{\tau,l}$ rather than the full form of the distribution. At time τ , given observations up to time-step $\tau - 1$, the prior state mean at time τ is $\hat{\mathbf{w}}_{\tau|\tau-1,l}$ and its prior covariance is $\mathbf{P}_{\tau|\tau-1,l}$. These are related to the posterior mean and covariance from the previous time-step $\tau - 1$ by

$$\hat{\mathbf{w}}_{\tau|\tau-1,l} = \hat{\mathbf{w}}_{\tau-1|\tau-1,l} \quad (41)$$

$$\mathbf{P}_{\tau|\tau-1,l} = \mathbf{P}_{\tau-1|\tau-1,l} + q_{\tau,l}\mathbf{I}. \quad (42)$$

We estimate the state noise variance $q_{\tau,l}$ as

$$q_{\tau+1,l} = \max[u_{\tau|\tau,l} - u_{\tau|\tau-1,l}, 0] + z_{\tau,l} \quad (43)$$

where $u_{\tau|\tau,l}$ and $u_{\tau|\tau-1,l}$ are the variances in the distribution over the classifier outputs \vec{c} after observing data up to time τ and $\tau - 1$ respectively, and $z_{\tau,l}$ is the uncertainty in the classifier outputs. For observations up to time-step v , we define $u_{\tau|v,l}$ as:

$$\begin{aligned} u_{\tau|v,l} &= \mathbb{V}[\delta_{c_{i_\tau}l} | c_{i_1}, \dots, c_{i_v}, \vec{t}, \boldsymbol{\alpha}_0] \\ &= \hat{\pi}_{\tau|v,l} (1 - \hat{\pi}_{\tau|v,l}) \end{aligned} \quad (44)$$

where

$$\hat{\pi}_{\tau|v,l} = \mathbb{E}[\tilde{\pi}_{\tau,l} | c_{i_1}, \dots, c_{i_v}, \vec{t}, \boldsymbol{\alpha}_0] = g(\mathbf{h}_\tau^T \hat{\mathbf{w}}_{\tau|v,l}). \quad (45)$$

When the classifier outputs are observed, $z_{\tau,l}$ is zero; when they are not observed, we use $\hat{\pi}_{\tau|\tau-1,l}$ as an estimate of the missing output, so $z_{\tau,l} = u_{\tau|\tau-1,l}$.

From Equations (41) and (42) we can specify the mean and variance of the prior distribution of $\eta_{\tau,l}$:

$$\hat{\eta}_{\tau|\tau-1,l} = \mathbf{h}_\tau^T \hat{\mathbf{w}}_{\tau|\tau-1,l} \quad (46)$$

$$r_{\tau|\tau-1,l} = \mathbf{h}_\tau^T \mathbf{P}_{\tau|\tau-1,l} \mathbf{h}_\tau \quad (47)$$

We can now use $\hat{\eta}_{\tau|\tau-1,l}$ and $r_{\tau|\tau-1,l}$ to estimate the parameters of the prior distribution over $\tilde{\pi}_{\tau,l}$ as follows. The dynamic generalised linear model allows the distribution of the output variable c_{i_τ} to have any exponential family distribution [14]. In DynIBCC, the discrete outputs have a multinomial distribution, which is a member of the exponential family, with the Dirichlet distribution as the conjugate prior. Therefore, DynIBCC places a Dirichlet prior over $\tilde{\pi}_\tau$ with hyperparameters $\tilde{\boldsymbol{\alpha}}_\tau$ that are dependent on the true label t_{i_τ} . If we consider a single classifier output $c_{i_\tau} = l$, then

$$\tilde{\pi}_{\tau,l} \sim \text{Beta}(\tilde{\alpha}_{\tau,l}, \beta_{\tau,l}), \quad (48)$$

where $\beta_{\tau,l} = \sum_{m=1, l \neq m}^L \tilde{\alpha}_{\tau,m}$ and L is the number of possible base classifier output values. Since $\tilde{\pi}_\tau$ is related to $\eta_{\tau,l}$ by the logistic function (Equation (38)), we can write the full prior distribution over $\eta_{\tau,l}$ in terms of the same hyperparameters:

$$p(\eta_{\tau,l} | c_{i_1}, \dots, c_{i_{\tau-1}}, \vec{t}) = \frac{1}{\mathbf{B}(\tilde{\alpha}_{\tau,l}, \beta_{\tau,l})} \frac{\exp(\eta_{\tau,l})^{\tilde{\alpha}_{\tau,l}-1}}{(1 + \exp(\eta_{\tau,l}))^{\tilde{\alpha}_{\tau,l} + \beta_{\tau,l}}} \quad (49)$$

where $\mathbf{B}(a, b)$ is the beta function. This distribution is recognised as a beta distribution of the second kind [30]. We can approximate the moments of this prior distribution as follows:

$$\hat{\eta}_{\tau|\tau-1,l} = \mathbb{E}[\eta_{\tau,l} | c_1, \dots, c_{\tau-1}, t_{i_1}, \dots, t_{i_\tau}] = \Psi(\tilde{\alpha}_{\tau,l}) - \Psi(\beta_{\tau,l}) \quad (50)$$

$$\approx \log\left(\frac{\tilde{\alpha}_{\tau,l}}{\beta_{\tau,l}}\right) \quad (51)$$

$$r_{\tau|\tau-1,l} = \mathbb{V}[\eta_{\tau,l} | c_1, \dots, c_{\tau-1}, t_{i_1}, \dots, t_{i_\tau}] = \Psi'(\tilde{\alpha}_{\tau,l}) + \Psi'(\beta_{\tau,l}) \quad (52)$$

$$\approx \frac{1}{\tilde{\alpha}_{\tau,l}} + \frac{1}{\beta_{\tau,l}} \quad (53)$$

From these approximations we can calculate $\tilde{\alpha}_\tau$ and β_τ in terms of $\hat{\eta}_{\tau|\tau-1}$ and $r_{\tau|\tau-1}$:

$$\tilde{\alpha}_{\tau,l} = \frac{1 + \exp(\hat{\eta}_{\tau|\tau-1,l})}{r_{\tau|\tau-1,l}} \quad (54)$$

$$\beta_{\tau,l} = \frac{1 + \exp(-\hat{\eta}_{\tau|\tau-1,l})}{r_{\tau|\tau-1,l}}. \quad (55)$$

This gives us approximate hyperparameters for the prior distribution over $\tilde{\pi}_\tau$.

7.2 Forward Pass Filtering Steps

The forward pass filtering steps update the distribution over $\tilde{\pi}_\tau$ given an observation of the base classifier output c_{i_τ} at time τ . We calculate the posterior hyperparameters $\tilde{\alpha}_{\tau|\tau,l}$ and $\beta_{\tau|\tau,l}$ by adding to the prior parameters:

$$\tilde{\alpha}_{\tau|\tau,l} = \tilde{\alpha}_{\tau,l} + \delta_{c_{i_\tau}l} \quad (56)$$

$$\beta_{\tau|\tau,l} = \beta_{\tau,l} + (1 - \delta_{c_{i_\tau}l}). \quad (57)$$

In this way we update the pseudo-counts of base classifier output values, as we did in the static IBCC model in Equation (22). The posterior mean and variance are then approximated by

$$\hat{\eta}_{\tau|\tau,l} \approx \log\left(\frac{\tilde{\alpha}_{\tau|\tau}}{\beta_{\tau|\tau}}\right) \quad (58)$$

$$r_{\tau|\tau} \approx \frac{1}{\alpha_{\tau|\tau}} + \frac{1}{\beta_{\tau|\tau}}. \quad (59)$$

Then, we can apply an update to the mean and covariance of the state variable using linear Bayesian estimation, described in [31]:

$$\hat{\mathbf{w}}_{\tau|\tau,l} = \hat{\mathbf{w}}_{\tau|\tau-1,l} + \mathbf{K}_{\tau,l} (\hat{\eta}_{\tau|\tau,l} - \hat{\eta}_{\tau|\tau-1,l}) \quad (60)$$

$$\mathbf{P}_{\tau|\tau,l} = (\mathbf{I} - \mathbf{K}_{\tau,l} \mathbf{h}_\tau^\top) \mathbf{P}_{\tau|\tau-1,l} \left(1 - \frac{r_{\tau|\tau,l}}{r_{\tau|\tau-1,l}}\right) \quad (61)$$

where $\mathbf{K}_{\tau,l}$, the equivalent of the optimal Kalman gain is

$$\mathbf{K}_{\tau,l} = \frac{\mathbf{P}_{\tau|\tau-1,l} \mathbf{h}_\tau}{r_{\tau|\tau-1,l}} \quad (62)$$

and \mathbf{I} is the identity matrix. The term $\frac{r_{\tau|\tau,l}}{r_{\tau|\tau-1,l}}$ in the covariance update corresponds to our uncertainty over $\eta_{\tau,l}$, which we do not observe directly. Linear Bayes estimation gives an optimal estimate when the full distribution over the state variable $\mathbf{w}_{\tau,l}$ is unknown, and therefore differs from a Kalman filter in not specifying a Gaussian distribution over $v_{\tau,l}$ in Equation (40).

To perform the forward pass we iterate through the data: for each time-step we calculate the prior state moments using Equations (41) and (42), then update these to the posterior state moments $\hat{\mathbf{w}}_{\tau|\tau}$ and $\mathbf{P}_{\tau|\tau}$. The forward pass filtering operates in a sequential manner as posterior state moments from time-step $\tau - 1$ are used to calculate the prior moments for the subsequent time-step τ .

7.3 Backward Pass Smoothing Steps

After filtering through the data calculating $\hat{\mathbf{w}}_{\tau|\tau,l}$ and $\mathbf{P}_{\tau|\tau,l}$ we then run a backward pass to find the approximate posterior moments given all subsequent data points, $\hat{\mathbf{w}}_{\tau|N,l}$ and $\mathbf{P}_{\tau|N,l}$, and from these the posterior hyperparameters given all data, $\tilde{\alpha}_{\tau|N}$. The backward pass is a Modified Bryson-Frazier smoother [28], which updates the distribution using the adjoint state vector $\hat{\lambda}_{\tau,l}$ and adjoint covariance matrix $\hat{\Lambda}_{\tau,l}$ as follows:

$$\hat{\mathbf{w}}_{\tau|N,l} = \hat{\mathbf{w}}_{\tau|\tau,l} - \mathbf{P}_{\tau|\tau,l} \hat{\boldsymbol{\lambda}}_{\tau,l} \quad (63)$$

$$\mathbf{P}_{\tau|N,l} = \mathbf{P}_{\tau|\tau,l} - \mathbf{P}_{\tau|\tau,l} \hat{\boldsymbol{\Lambda}}_{\tau,l} \mathbf{P}_{\tau|\tau,l}. \quad (64)$$

In our dynamical system the state \mathbf{w}_τ evolves according to Equation (40), so $\hat{\boldsymbol{\lambda}}_\tau$ and $\hat{\boldsymbol{\Lambda}}_\tau$ are defined recursively as the posterior updates from the subsequent step $\tau + 1$ given data from $\tau + 1$ to N .

$$\tilde{\boldsymbol{\lambda}}_{\tau,l} = -\frac{\mathbf{h}_\tau}{r_{\tau|\tau-1,l}} (\hat{\eta}_{\tau|\tau,l} - \hat{\eta}_{\tau|\tau-1,l}) + (\mathbf{I} - \mathbf{K}_{\tau,l} \mathbf{h}_\tau^\top) \hat{\boldsymbol{\lambda}}_{\tau,l} \quad (65)$$

$$\hat{\boldsymbol{\lambda}}_\tau = \tilde{\boldsymbol{\lambda}}_{\tau+1} \quad (66)$$

$$\hat{\boldsymbol{\lambda}}_N = \mathbf{0} \quad (67)$$

$$\tilde{\boldsymbol{\Lambda}}_{\tau,l} = \frac{\mathbf{h}_\tau \mathbf{h}_\tau^\top}{r_{\tau|\tau-1,l}} \left(1 - \frac{r_{\tau|\tau,l}}{r_{\tau|\tau-1,l}} \right) + (\mathbf{I} - \mathbf{K}_{\tau,l} \mathbf{h}_\tau^\top) \hat{\boldsymbol{\Lambda}}_{\tau,l} (\mathbf{I} - \mathbf{K}_{\tau,l} \mathbf{h}_\tau^\top)^\top \quad (68)$$

$$\hat{\boldsymbol{\Lambda}}_\tau = \tilde{\boldsymbol{\Lambda}}_{\tau+1} \quad (69)$$

$$\hat{\boldsymbol{\Lambda}}_N = \mathbf{0} \quad (70)$$

Estimates for final posterior hyperparameters are therefore given by

$$\hat{\eta}_{\tau|N,l} = \mathbf{h}_\tau^\top \hat{\mathbf{w}}_{\tau|N,l} \quad (71)$$

$$r_{\tau|N,l} = \mathbf{h}_\tau^\top \mathbf{P}_{\tau|N,l} \mathbf{h}_\tau \quad (72)$$

$$\tilde{\alpha}_{\tau|N,l} = \frac{1 + \exp(\hat{\eta}_{\tau|N,l})}{r_{\tau|N,l}} \quad (73)$$

$$\beta_{\tau|N,l} = \frac{1 + \exp(-\hat{\eta}_{\tau|N,l})}{r_{\tau|N,l}}. \quad (74)$$

7.4 Variational Update Equations

We can now replace the variational distribution for $q^*(\boldsymbol{\pi}_j)$ given by Equation (21). We continue to omit the (k) notation for clarity. The dynamic model instead uses a variational distribution for each time-step, $q^*(\boldsymbol{\pi}_{\tau,j}^{(k)})$ given by

$$q^*(\boldsymbol{\pi}_{\tau,j}) = \frac{1}{B(\boldsymbol{\alpha}_{\tau|N,j})} \prod_{l=1}^L (\pi_{\tau,j,l})^{\alpha_{\tau|N,jl}-1} \quad (75)$$

$$= \text{Dir}(\boldsymbol{\pi}_{\tau,j} | \alpha_{\tau|N,j1}, \dots, \alpha_{\tau|N,jL}) \quad (76)$$

where $\text{Dir}()$ is the Dirichlet distribution with parameters $\boldsymbol{\alpha}_{\tau|N,j}$ calculated according to

$$\alpha_{\tau|N,jl} = \frac{1 + \exp(\hat{w}_{\tau|N,jl})}{P_{\tau|N,jl}} \quad (77)$$

We calculate $\hat{\mathbf{w}}_{\tau|N}$ and $\mathbf{P}_{\tau|N}$ using the above filtering and smoothing passes, taking the expectation over \tilde{t} so we replace $h_{\tau,j}$ with $\tilde{h}_{\tau,j} = \mathbb{E}_{\tilde{t}}[t_{i_\tau} = j]$. Equation (77) is used to derive the hyperparameters for each row of the confusion matrix and thus each possible value of t_{i_τ} ; thus it is equivalent to Equation (74) with $h_{\tau,j} = 1$. This update equation replaces Equation (22) in the static model. The expectation given by Equation (23) becomes

$$\mathbb{E}[\ln \pi_{\tau,jl}] = \Psi(\alpha_{\tau|N,jl}) - \Psi \left(\sum_{m=1}^L \alpha_{\tau|N,jm} \right) \quad (78)$$

This can then be used in the variational distribution over t_i as follows, replacing Equation (8):

$$\ln q^*(t_i) = \mathbb{E}[\ln \kappa_{t_i}] + \sum_{k=1}^K \mathbb{E} \left[\ln \pi_{\tau_i^{(k)}, t_i c_i^{(k)}}^{(k)} \right] + \text{const.} \quad (79)$$

7.5 DynIBCC Joint and Posterior Distributions

In DynIBCC, we altered the IBCC model to use the time-dependent confusion matrices, giving the joint distribution over all latent variables and parameters in the model as follows. Here we are considering distributions for all base classifiers and therefore must re-introduce the k superscript notation.

$$p(\boldsymbol{\kappa}, \Pi, \vec{t}, \vec{c} | \boldsymbol{\alpha}_0, \boldsymbol{\nu}_0) = \prod_{i=1}^N \left\{ \kappa_{t_i} \prod_{k \in \mathbf{C}_i} \pi_{\tau_i^{(k)}, t_i c_i^{(k)}}^{(k)} \right. \\ \left. p \left(\boldsymbol{\pi}_{\tau_i^{(k)}}^{(k)} | t_i, \boldsymbol{\pi}_{\tau_i^{(k)}-1}^{(k)}, \boldsymbol{\alpha}_{0,j}^{(k)} \right) \right\} p(\boldsymbol{\kappa} | \boldsymbol{\nu}_0) \quad (80)$$

where \mathbf{C}_i is the set of base classifiers that have completed classification task i . There is a change of notation from Equation (1) for the static IBCC model, which iterates over all K classifiers. Here we iterate over the set \mathbf{C}_i because $\tau_i^{(k)}$ is undefined for objects that have not been classified by k . The static model (Equation 1) does not have this issue as the confusion matrix is the same for all tasks, and thus Equation (1) defines a joint probability over all observed and unobserved base classifier outputs. In DynIBCC, if we wish to determine a distribution over an unobserved base classifier output $c_{i_{unseen}}^{(k)}$ we must also determine a suitable confusion matrix by determining which time-step the unseen classifier output occurred at.

In Equation (80) above, the prior over $\boldsymbol{\pi}_{\tau_i^{(k)}}^{(k)}$ can be estimated by finding the mean $\bar{\eta}_{\tau| \tau-1, jl}$ and variance $\bar{r}_{\tau| \tau-1, jl}$ of the linear predictor from its value at the previous time-step, given by:

$$\eta_{\tau-1, jl} = \ln \left(\frac{\pi_{\tau-1, jl}}{\sum_{m=1, \neq l}^L \pi_{\tau-1, jl}} \right) \quad (81)$$

The bar notation $\bar{\eta}$ indicates a variable that is calculated deterministically from the previous state given the value of $\boldsymbol{\pi}_{\tau_i^{(k)}}^{(k)}$. Considering the random walk Equation (40), where the change $\mathbf{v}_{\tau, l}$ from the previous state has mean $\mathbf{0}$ and covariance $q_{\tau, l} \mathbf{I}$, the moments of the linear predictor are

$$\bar{\eta}_{\tau| \tau-1, jl} = \eta_{\tau-1, jl} \quad (82)$$

$$\bar{r}_{\tau| \tau-1, jl} = q_{\tau, jl} \quad (83)$$

where $q_{\tau, l}$ is estimated as per Equations (43). From these values we can calculate the parameters $\bar{\alpha}_{\tau, jl}$ for a Dirichlet prior over $\boldsymbol{\pi}_{\tau_i^{(k)}}^{(k)}$:

$$p \left(\boldsymbol{\pi}_{\tau_i^{(k)}}^{(k)} | t_i, \boldsymbol{\pi}_{\tau_i^{(k)}-1}^{(k)}, \boldsymbol{\alpha}_{0,j}^{(k)} \right) = \text{Dir} \left(\boldsymbol{\pi}_{\tau_i^{(k)}}^{(k)} | \bar{\alpha}_{\tau_i^{(k)}, j1}^{(k)}, \dots, \bar{\alpha}_{\tau_i^{(k)}, jL}^{(k)} \right). \quad (84)$$

For $\tau_i^{(k)} = 0$ the parameter $\bar{\alpha}_{1, jl} = \alpha_{0, jl}$. For $\tau_i^{(k)} > 0$ it is given by:

$$\bar{\alpha}_{\tau, jl} = \frac{1 + \exp(\bar{\eta}_{\tau, jl})}{\bar{r}_{\tau| \tau-1, jl}}. \quad (85)$$

7.6 Duplicate Classifications

The original static model did not allow for duplicate classifications of the same object by the same base classifier. We assumed that even if a base classifier alters their decision when they see an object a second time, the two decisions are likely to be highly correlated and so cannot be treated as independent. However, the dynamic model reflects the possibility that the base classifier may change its own underlying model; therefore responses may be uncorrelated if they are separated by a sufficient number of time-steps or if the confusion matrix changes rapidly over a small number of time-steps. A model that handles dependencies between duplicate classifications at time

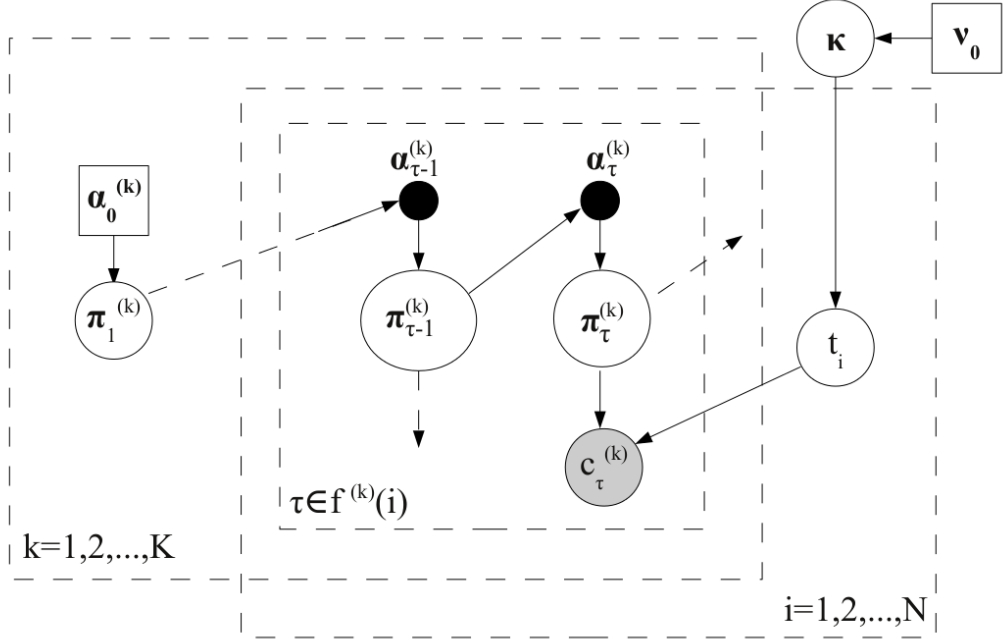


Figure 6: Graphical model for DynIBCC allowing multiple classifications of the same object by the same classifier (duplicate classifications). The dashed arrows indicate dependencies to nodes at previous or subsequent time-steps. Solid black circular nodes are variables calculated deterministically from their predecessors. The shaded node represents observed values, circular nodes are variables with a distribution and square nodes are variables instantiated with point values.

$\tau_{original}$ and time $\tau_{duplicate}$ may adjust $\pi_{\tau_{original}}^{(k)}$ and $\pi_{\tau_{duplicate}}^{(k)}$ to compensate for correlation. However, in applications where duplicates only occur if they are separated by a large number of time-steps it may be reasonable to treat them as independent observations. In cases where duplicates are allowed we index decisions by their time-step as $c_{\tau}^{(k)}$. For model variants that permit duplicates the joint distribution is hence:

$$\begin{aligned}
p(\boldsymbol{\kappa}, \Pi, \vec{t}, \vec{c} | \mathbf{A}_0, \nu_0) \\
= p(\boldsymbol{\kappa} | \nu_0) \prod_{i=1}^N \kappa_{t_i} \prod_{k \in \mathbf{C}_i} \prod_{\tau \in \mathbf{f}^{(k)}(i)} \pi_{\tau, t_i, c_{\tau}^{(k)}}^{(k)} p\left(\pi_{\tau, t_i, c_{\tau}^{(k)}}^{(k)} | t_i, \pi_{\tau-1}^{(k)}, \alpha_0^{(k)}\right)
\end{aligned} \tag{86}$$

where as before $\mathbf{f}^{(k)}(i)$ maps an object i to the time-step at which i was classified by base classifier k . For a sample i_{unseen} not classified by k , $\mathbf{f}^{(k)}(i_{unseen}) = \emptyset$.

We must also update Equation (79) as to allow duplicates as follows:

$$\ln q^*(t_i) = \mathbb{E}[\ln \kappa_{t_i}] + \sum_{k \in \mathbf{C}_i} \sum_{\tau \in \mathbf{f}^{(k)}(i)} \mathbb{E} \left[\ln \pi_{\tau, t_i, c_{\tau}^{(k)}}^{(k)} \right] + \text{const.} \tag{87}$$

The resulting graphical model is shown in Figure 6, with an additional plate to allow different time-steps τ that correspond to the same base classifier k and object i .

7.7 Variational Lower Bound

We now give the variational lower bound for Dynamic IBCC using the formulation that permits duplicates. We use $\Pi = \left\{ \pi_{\tau,j}^{(k)} \mid \tau = 1, \dots, T^{(k)}, j = 1, \dots, J, k = 1, \dots, K \right\}$ to refer to the set of confusion matrices for all classifiers and all time-steps.

$$\begin{aligned}
L(q) &= \iiint q(\vec{t}, \Pi, \kappa) \ln \frac{p(\vec{c}, \vec{t}, \Pi, \kappa \mid \mathbf{A}_0, \nu_0)}{q(\vec{t}, \Pi, \kappa)} d\vec{t} d\Pi d\kappa \\
&= \mathbb{E}_{\vec{t}, \Pi, \kappa} [\ln p(\vec{c}, \vec{t}, \Pi, \kappa \mid \mathbf{A}_0, \nu_0)] - \mathbb{E}_{\vec{t}, \Pi, \kappa} [\ln q(\vec{t}, \Pi, \kappa)] \\
&= \mathbb{E}_{\vec{t}, \Pi} [\ln p(\vec{c} \mid \vec{t}, \Pi)] + \mathbb{E}_{\Pi} [\ln p(\Pi \mid \vec{t}, \mathbf{A}_0)] + \mathbb{E}_{\vec{t}, \kappa} [\ln p(\vec{t} \mid \kappa)] + \mathbb{E}_{\kappa} [\ln p(\kappa \mid \nu_0)] \\
&\quad - \mathbb{E}_{\vec{t}, \Pi, \kappa} [\ln q(\vec{t})] - \mathbb{E}_{\vec{t}, \Pi} [\ln q(\Pi)] - \mathbb{E}_{\vec{t}, \kappa} [\ln q(\kappa)]
\end{aligned} \tag{88}$$

The expectation terms relating to the joint probability of the latent variables, observed variables and the parameters are as for the static model in Subsection 3.3, except

$$\mathbb{E}_{\vec{t}, \Pi} [\ln p(\vec{c} \mid \vec{t}, \Pi)] = \sum_{i=1}^N \sum_{k \in \mathcal{C}_i} \sum_{\tau \in \mathbf{f}^{(k)}(i)} \sum_{j=1}^J \mathbb{E}[t_i = j] \mathbb{E} \left[\ln \pi_{\tau, j c_\tau}^{(k)} \right] \tag{89}$$

$$\mathbb{E}_{\Pi} [\ln p(\Pi \mid \mathbf{A}_0)] = \sum_{k=1}^K \sum_{j=1}^J \sum_{\tau=1}^{T^{(k)}} \left\{ -\ln \mathbf{B} \left(\bar{\alpha}_{\tau, j}^{(k)} \right) \right. \tag{90}$$

$$\left. + \sum_{l=1}^L \left(\bar{\alpha}_{\tau, j l}^{(k)} - 1 \right) \mathbb{E} \left[\ln \pi_{\tau, j l}^{(k)} \right] \right\} \tag{91}$$

In DynIBCC, the expectation over the variational distribution $q^*(\pi)$ also differs from static IBCC:

$$\mathbb{E}_{\vec{t}, \Pi} [\ln q(\Pi)] = \sum_{k=1}^K \sum_{j=1}^J \sum_{\tau=1}^{T^{(k)}} \left\{ -\ln \mathbf{B} \left(\alpha_{\tau | N, j}^{(k)} \right) \right. \tag{92}$$

$$\left. + \sum_{l=1}^L \left(\alpha_{\tau | N, j l}^{(k)} - 1 \right) \mathbb{E} \left[\ln \pi_{\tau, j l}^{(k)} \right] \right\} \tag{93}$$

8 Dynamics of Galaxy Zoo Supernovae Contributors

We applied the variational Bayesian DynIBCC to the Galaxy Zoo Supernovae data from Section 4 to examine the changes to individual confusion matrices. There is a large variation in the dynamics of different decision makers but in many there are sustained drifts in a particular direction. We give examples of the different types of changes found for different base classifiers – the Galaxy Zoo Supernovae volunteers – in Figures 7 and (8). These are ternary plots of the expected confusion matrices at each time-step or observation of the decision maker. Each line corresponds to the confusion vector $\pi_j^{(k)}$ for true class j . To help the reader time-align the traces, certain time-steps have been labelled with a blue marker and edge between the two confusion vectors, with the label denoting the global number of observations for all base classifiers at this point. The example volunteers classified 29651, 21933, 23920 and 20869 candidates respectively. The first example is Figure 7a, which shows a confusion matrix with some drift in the earlier time-steps and some small fluctuations later on for the “not supernova” class. The decision maker shown in Figure 7b appears to have a more sustained drift away from scores of 3 for both true classes. The earliest changes in both these decision makers, such as for the “supernova” class in Figure 7b, appear to be a move away from the prior, which affects the first data points most. The last two examples show more sudden changes. In Figure 8a we see a very significant change in the later observations for the “not supernova” class, after which

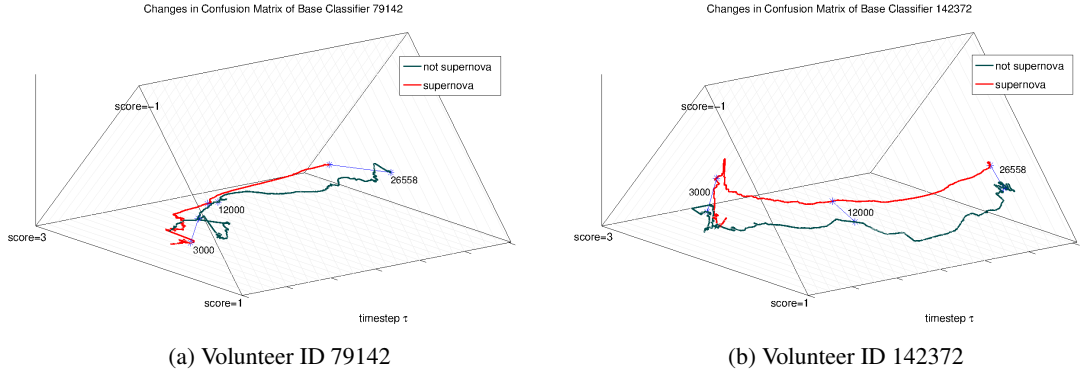


Figure 7: Ternary plot showing the dynamics for Galaxy Zoo Supernovae volunteers. Each line plots the evolution of a row of the confusion matrix corresponding to a particular true class. Proximity of the line to a vertex indicates the probability of generating a certain score for a candidate object with the given true class. Blue ‘*’ markers help the reader align points on the two lines, with a label indicating the global number of observations at that point.

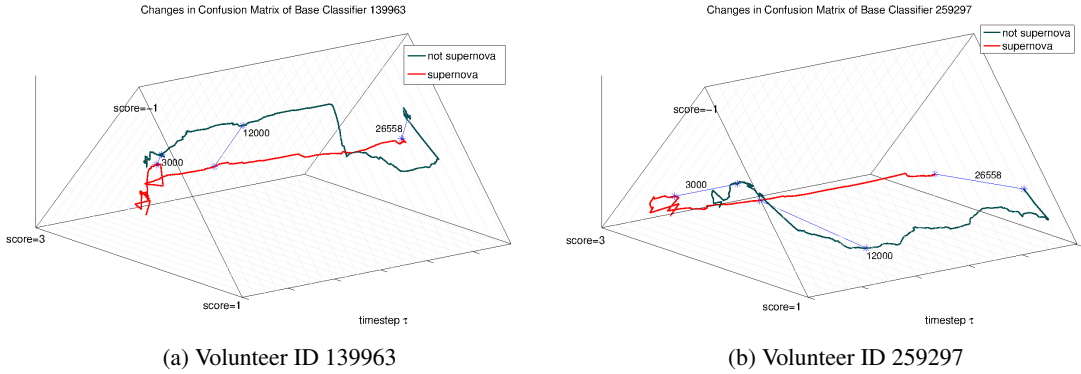


Figure 8: Ternary plot showing the dynamics for Galaxy Zoo Supernovae volunteers. Each line plots the evolution of a row of the confusion matrix corresponding to a particular true class. Proximity of the line to a vertex indicates the probability of generating a certain score for a candidate object with the given true class. Blue ‘*’ markers help the reader align points on the two lines, with a label indicating the global number of observations at that point.

the confusion matrix returns to a similar point to before. Figure 8b shows little initial change followed by a sudden change for the “not supernova” class, which then becomes fairly stable. The dynamics observed were inferred over a large number of data points, suggesting that the longer trends are due to genuine changes in performance of base classifiers over time. Smaller fluctuations may be due to bias in the observations (e.g. for a task that is very easy) or a change in behaviour of the citizen scientists, but sustained changes after the initial move away from the priors are more suggestive of a change in behaviour or in the information presented to agents when making classifications. For all four examples, there are more initial fluctuations, which could relate to the way that new volunteers adapt when they complete their first tasks.

9 Dynamics of π Communities

We now apply the community analysis method used in Section 5 to the dynamic confusion matrices to examine the development of the community structure over time. After different a number of observations s we run the community detection method [24] over an adjacency matrix, using equation (32) with the most recent confusion matrices for all base classifiers observed up to s .

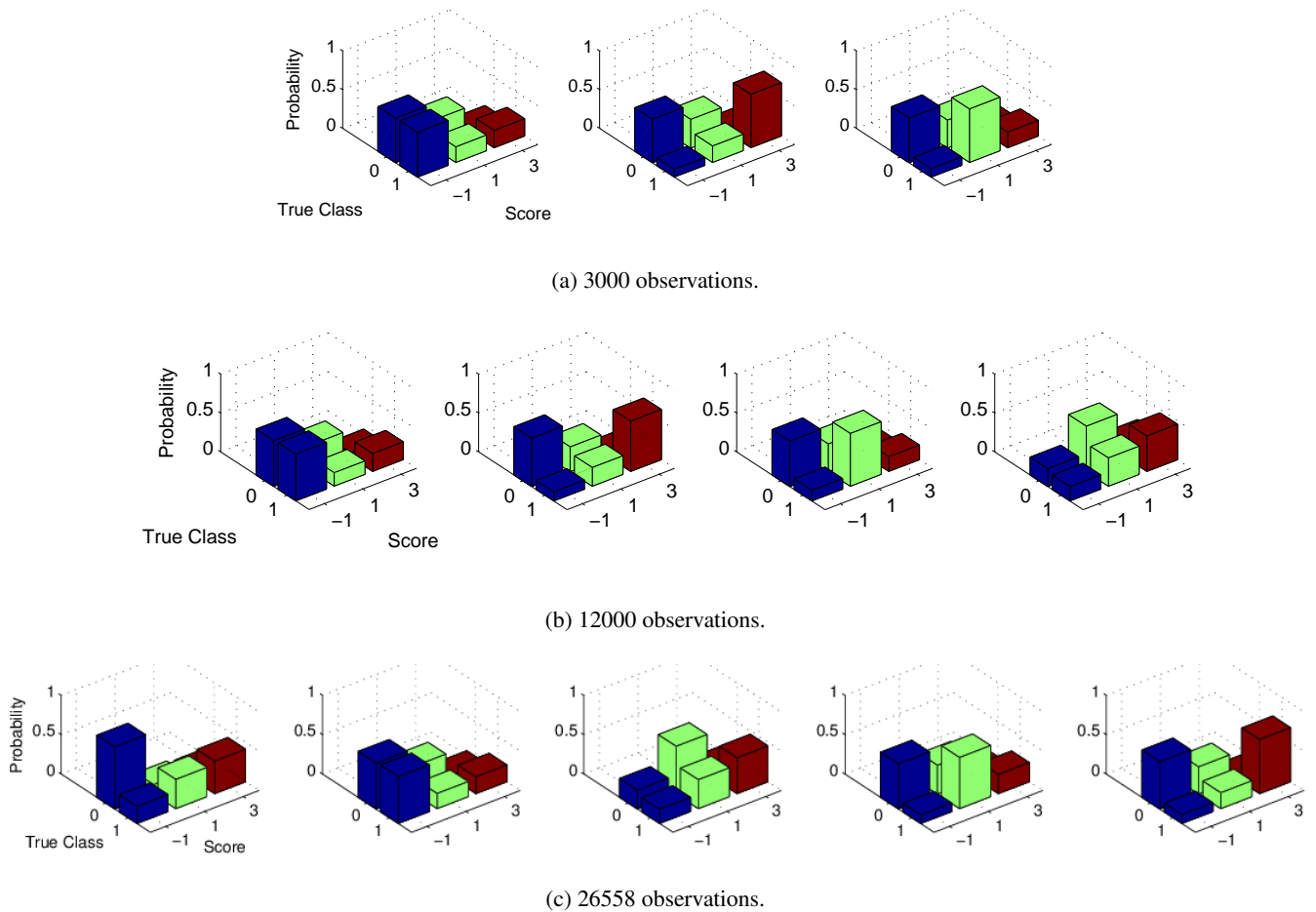


Figure 9: π communities: means over the expected confusion matrices of community members after different numbers of observations. At each time point we see a new community appear while previous communities persist with similar means.

In Figure 9 we see how the same communities emerge over time as we saw in Section 5 in Figure 3. Initially, only three communities are present, with those corresponding to groups 4 (“optimists”) and 1 (“reasonable”) in Figure 3 only appearing after 1200 and 26558 observations. The “reasonable” group is the last to emerge and most closely reflects the way the designers of the system intend good decision makers to behave. It may therefore appear as a result of participants learning, or of modifications to the user interface or instructions as the Galaxy Zoo Supernovae application was being developed to encourage this behaviour.

We also note that agents switch between communities. In Figure 10 we show the node participation scores at each number of observations s for the individuals we examined in Section 8. Community membership changes after significant changes to the individual’s confusion matrix. However, the communities persist despite the movement of members between them.

10 Dynamics of Common Task Communities

Finally, we look at the evolution of the common task communities to observe the effect of recent tasks on the community structure and confusion matrices. We wish to observe whether distinct communities are persistent as more tasks are completed. Changes to the structure inferred may be a result of observing more data about the base classifiers. Alternatively, individual behaviours may evolve as a result of learning new types of tasks or changes to

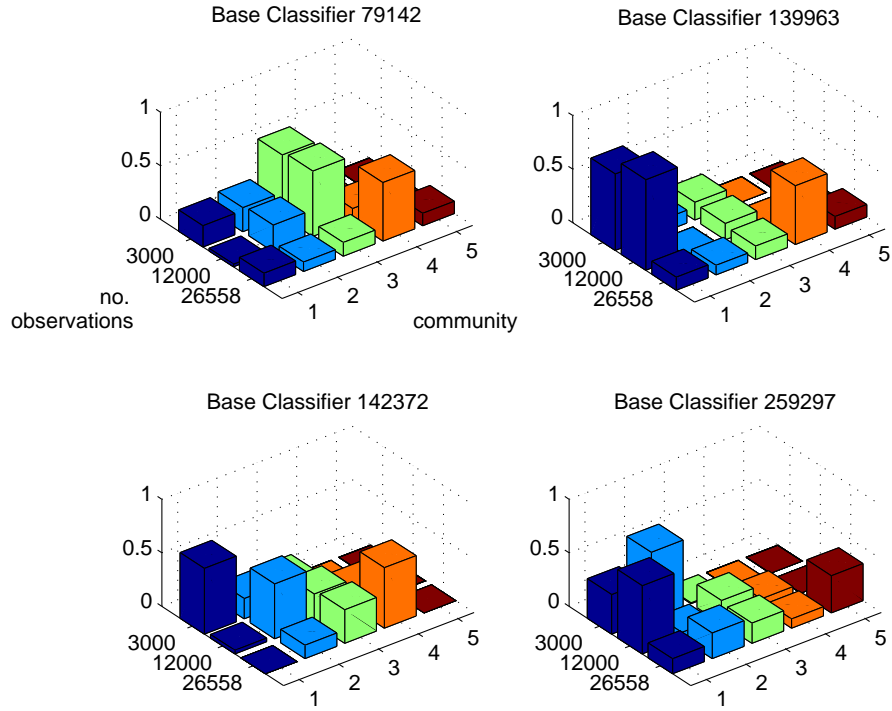


Figure 10: Node participation scores for the π communities for selected individuals shown in Section 8 after different numbers of observations. Each bar corresponds to the individual’s participation score in a particular community after running the community analysis over all observations up to that point. Participation scores close to one indicate very strong membership of a community. The node participation scores for one number of observations sum to one over the communities 1 to 5.

individual circumstances, such as when a volunteer is available to carry out tasks. Our choice of community analysis method, given in Section 5 has the advantage that only a maximum number of communities need be chosen by the programmer, with the algorithm itself finding the most likely number of communities from the network itself. Here we show the changes that occur in Galaxy Zoo Supernovae. We generated three co-occurrence networks from all tasks completed up to 50,000, 200,000 and 493,048 observations. As before, we remove base classifiers with fewer than 10 classifications to filter out edges that may constitute noise rather than significant similarity. The algorithm produced community structures with modularities of 0.67, 0.69 and 0.75 respectively, showing that good community structure is present for smaller periods of observations (see Section 5 for definition of modularity). Figures (11), (12) and (13) show the means of the community members at each of these time slices, weighted by node participation. Since DynIBCC models the dynamics of base classifier confusion matrices as a random walk, the observations closest to the current time have the strongest effect on the distribution over the confusion matrices. Therefore, the expected confusion matrices can readily be used to characterise a community at a given point in time. When calculating the means we use the expected confusion matrix from the most recent time-step for that network.

In all three networks there is a persistent core for both true classes, where the means for the large communities remain similar. Some communities within this group move a small amount, for example, the large red community in the “Supernova” class. In contrast, we see more scattered small communities appear after 200,000 observations and at 493,048 observations. It is possible that the increase in number of base classifiers as we see more data

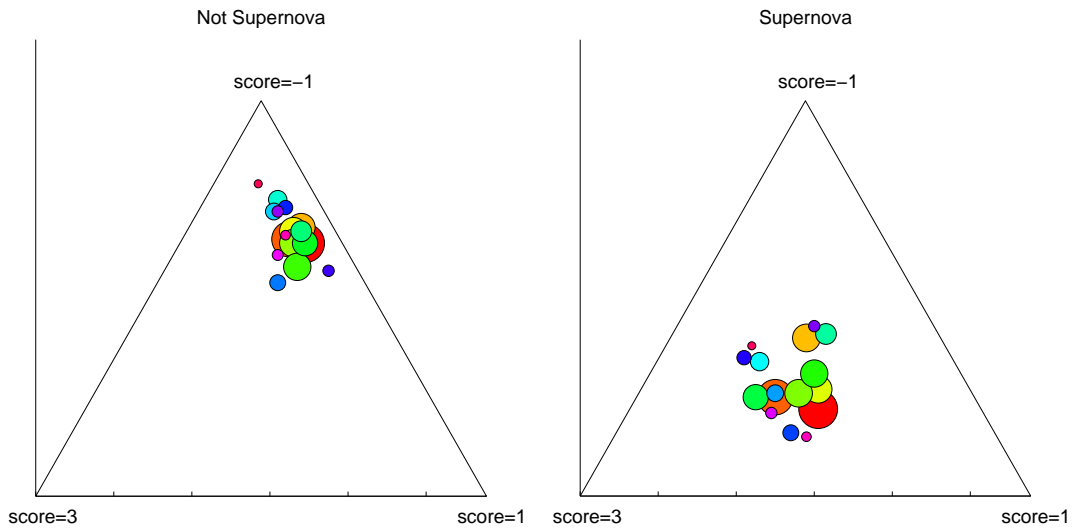


Figure 11: Common task communities: ternary plot of means of expected confusion matrices for community members after 50,000 observations. Each point corresponds to one common task community and represents the mean of $\mathbb{E}[\pi_j]$ for the community members. Proximity of a point to a vertex indicates the probability of outputting a particular score when presented with an object of true class “supernova” or “not supernova”.

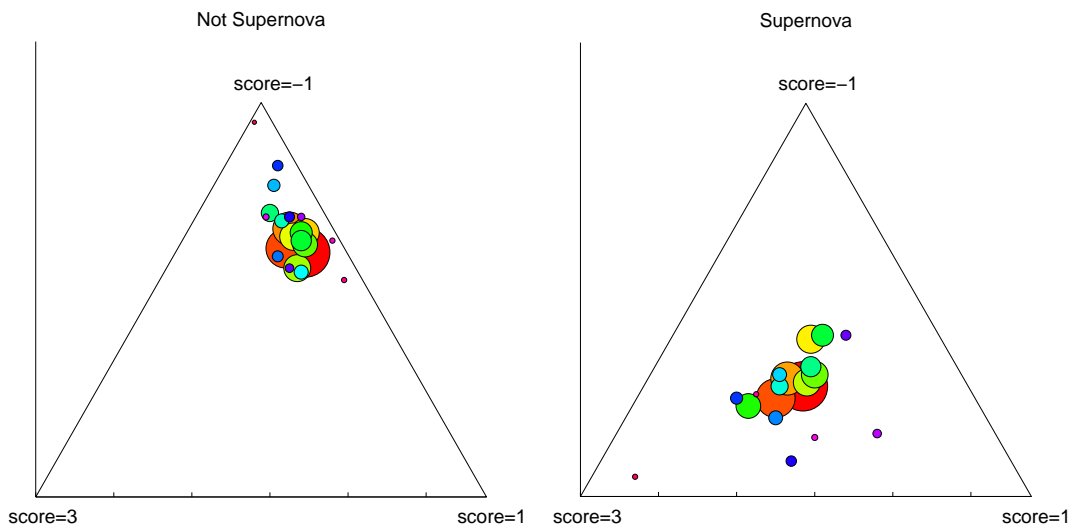


Figure 12: Common task communities: ternary plot of means of expected confusion matrices for community members after 50,000 observations. Each point corresponds to one common task community and represents the mean of $\mathbb{E}[\pi_j]$ for the community members. Proximity of a point to a vertex indicates the probability of outputting a particular score when presented with an object of true class “supernova” or “not supernova”.

means that previous individual outliers are now able to form communities with similar outliers. Therefore outlying communities could be hard to detect with smaller datasets. Many of these appear in the same place in only one of the figures, suggesting that they may contain new base classifiers that have made few classifications up to that

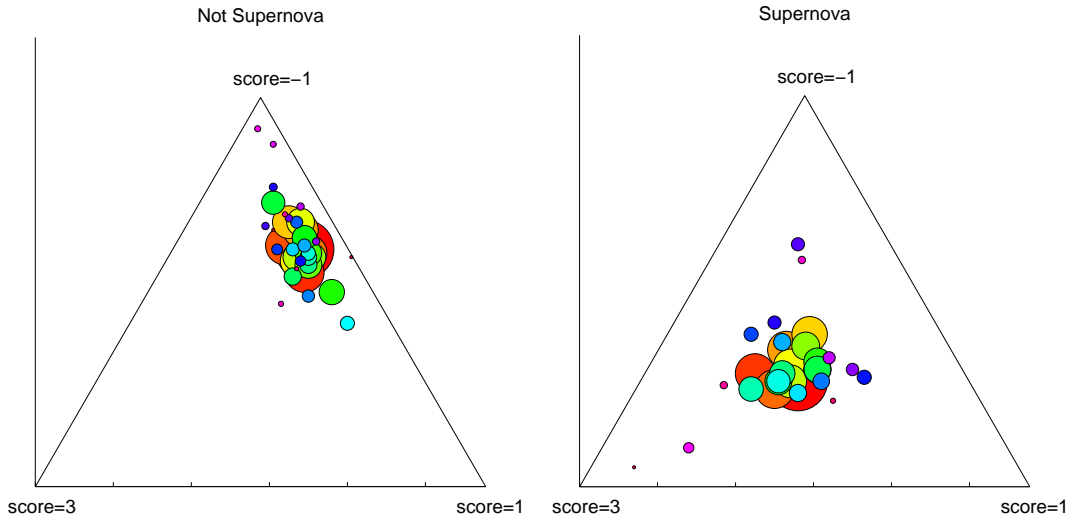


Figure 13: Common task communities: ternary plot of means of expected confusion matrices for community members after 50,000 observations. Each point corresponds to one common task community and represents the mean of $\mathbb{E}[\pi_j]$ for the community members. Proximity of a point to a vertex indicates the probability of outputting a particular score when presented with an object of true class “supernova” or “not supernova”.

point. Some are less transient however: the top-most community in the “not supernova” class in Figures (12) and (13) moves only a small amount. Similar sets of tasks may produce more extreme confusion matrices such as these for different agents at different times, implying that these tasks induce a particular bias in the confusion matrices. The changes we observe in Figures (11), (12) and (13) demonstrate how we can begin to identify the effect of different tasks on our view of the base classifiers by evaluating changes to the community structure after classifying certain objects. Future investigations may consider the need to modify the co-occurrence network to discount older task-based classifier associations.

11 Discussion

In this paper we present a very computationally efficient, variational Bayesian, approach to imperfect multiple classifier combination. We evaluated the method using real data from the Galaxy Zoo Supernovae citizen science project, with 963 classification tasks, 1705 base classifiers and 26,558 observations. In our experiments, our method far outperformed all other methods, including weighted sum and weighted majority, both of which are often advocated as they also learn weightings for the base classifiers. For our variational Bayes method the required computational overheads were far lower than those of Gibbs sampling approaches, giving much shorter compute time, which is particularly important for applications that need to make regular updates as new data is observed, such as our application here. Furthermore, on this data set at least, the accuracy of predictions was also better than the slower sampling-based method. We have shown that social network analysis can be used to extract sensible structure from the pool of decision makers using information inferred by Bayesian classifier combination or task co-occurrence networks. This structure provides a useful grouping of individuals and gives valuable information about their decision-making behaviour. We extended our model to allow for on-line dynamic analysis and showed how this enables us to track the changes in time associated with individual base classifiers. We also demonstrated how the community structures change over time, showing the use of the dynamic model to update information about group members.

Our current work considers how the rich information learned using our models can be exploited to improve the base classifiers, namely the human volunteer users. For example, we can use the confusion matrices, Π , and the

community structure to identify users who would benefit from more training. This could take place through interaction with user groups who perform more accurate decision making, for example via extensions of *apprenticeship learning* [32]. We also consider ways of producing user specialisation via selective object presentation such that the overall performance of the human-agent collective is maximised. We note that this latter concept bears the hallmarks of *computational mechanism design* [33] and the incorporation of incentive engineering and coordination mechanisms into the model is one of our present challenges. Future work will also investigate selecting individuals for a task to maximise both our knowledge of the true labels and of the confusion matrices, for example, by looking at the effects of previous tasks on the confusion matrices. To bring these different aspects together, we consider a global utility function for a set of classification and training tasks indexed $i = 1, \dots, N$ assigned to a set of base classifiers $k = 1, \dots, K$. Classifiers assigned to object i are part of coalition \mathbf{C}_i to maximise the total expected value of these assignments:

$$V(\mathbf{C}_1, \dots, \mathbf{C}_N) = \sum_{i=1}^N V_{object}(\mathbf{C}_i) + V_{dm}(\mathbf{C}_i) + V_{cost}(\mathbf{C}_i) \quad (94)$$

where $V_{object}(\mathbf{C}_i)$ is the expected information gain about the true class of object i from the classifiers in \mathbf{C}_i , $V_{dm}(\mathbf{C}_i)$ is the improvement to the decision makers through this assignment and $V_{cost}(k, i)$ captures other costs, such as payments to a decision maker. The value $V_{object}(\mathbf{C}_i)$ should be higher for classifiers in \mathbf{C}_i that are independent, so coalitions of decision makers from different communities may be favoured as different experience and confusion matrices may indicate correlation is less likely. $V_{object}(\mathbf{C}_i)$ should also account for specialisations, for example, by members of the same common task community. $V_{dm}(\mathbf{C}_i)$, captures expected changes to confusion matrices that result from the combiner learning more about base classifiers and from base classifiers improving through training or experience. In Galaxy Zoo Supernovae, for example, the contributors are attempting to identify objects visually from a textual description. The description may leave some ambiguity, e.g. “is the candidate roughly centred”. Seeing a range of images may alter how “roughly” the candidate can be centred before the contributor answers “yes”. Thus the value $V_{dm}(k, i)$ will depend on the objects previously classified by classifier k . A key direction for future work is defining these values so that systems such as Galaxy Zoo Supernovae can feed back information from confusion matrices and community structure to improve the overall performance and efficiency of the pool of decision makers. Common task communities and π communities may play a central role in estimating the effects of task assignments and training on related individuals. The could also be exploited to reduce the size of the task assignment problem to one of choosing classifiers from a small number of groups rather than evaluating each classifier individually.

12 Acknowledgements

The authors would like to thank Chris Lintott at Zooniverse. We gratefully acknowledge funding from the UK Research Council EPSRC for project ORCHID, grant EP/I011587/1. Ioannis Psorakis is funded by Microsoft Research European PhD Scholarship Programme.

References

- [1] A. M. Smith, S. Lynn, M. Sullivan, C. J. Lintott, P. E. Nugent, J. Botyanszki, M. Kasliwal, R. Quimby, S. P. Bamford, L. F. Fortson15, et al. Galaxy Zoo Supernovae. *Monthly Notices of the Royal Astronomical Society*, 2010.
- [2] A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer Error-Rates using the EM algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):20–28, January 1979.
- [3] M. Bloodgood and C. Callison-Burch. Using Mechanical Turk to build machine translation evaluation sets. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazons Mechanical Turk, Los Angeles*, page 208211, 2010.

- [4] A. J Quinn, B. B Bederson, T. Yeh, and J. Lin. CrowdfLOW: Integrating machine learning with Mechanical Turk for speed-cost-quality flexibility. Technical report, HCIL-2010-09, University of Maryland, College Park, September 2010.
- [5] S. Tulyakov, S. Jaeger, V. Govindaraju, and D. Doermann. Review of classifier combination methods. *Machine Learning in Document Analysis and Recognition*, page 361386, 2008.
- [6] N. Littlestone and M. K Warmuth. The weighted majority algorithm. In *Foundations of Computer Science, 1989., 30th Annual Symposium on*, page 256261, November 1989.
- [7] K. Monteith, J. L Carroll, K. Seppi, and T. Martinez. Turning Bayesian model averaging into Bayesian model combination. In *The 2011 International Joint Conference on Neural Networks (IJCNN)*, pages 2657–2663. IEEE, August 2011.
- [8] Z. Ghahramani and H. C. Kim. Bayesian classifier combination. *Gatsby Computational Neuroscience Unit Technical Report GCNU-T., London, UK:*, 2003.
- [9] Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. *J. Mach. Learn. Res.*, 11:12971322, August 2010.
- [10] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the Bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-6(6):721–741, November 1984.
- [11] W. R. Gilks and P. Wild. Adaptive rejection sampling for gibbs sampling. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 41(2):337–348, January 1992.
- [12] S. Lefkimmiatis, P. Maragos, and G. Papandreou. Bayesian inference on multiscale models for Poisson intensity estimation: Applications to Photon-Limited image denoising. *Image Processing, IEEE Transactions on*, 18(8):1724–1741, 2009.
- [13] R. Choudrey and S. Roberts. Variational mixture of Bayesian independent component analysers. *Neural Computation*, 15(1), 2003.
- [14] C. M. Bishop. *Pattern recognition and machine learning*. Information Science and Statistics. Springer Science+Business Media, LLC, 4 edition, 2006.
- [15] W. D Penny and S. J Roberts. Dynamic logistic regression. In *Neural Networks, 1999. IJCNN'99. International Joint Conference on*, volume 3, page 15621567, 2002.
- [16] H. Attias. *Advances in Neural Information Processing Systems 12*, chapter A Variational Bayesian Framework for Graphical Models, pages 209–215. 2000.
- [17] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, January 1977.
- [18] S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, March 1951.
- [19] Charles Fox and Stephen Roberts. A tutorial on variational Bayesian inference. *Artificial Intelligence Review*, pages 1–11, 2011.
- [20] G. Parisi and R. Shankar. Statistical field theory. *Physics Today*, 41:110, 1988.
- [21] Milton Abramowitz and Irene A. Stegun, editors. *Handbook of Mathematical Functions: with Formulas, Graphs, and Mathematical Tables*. Dover Publications, June 1965.
- [22] N. M. Law, S. R. Kulkarni, R. G. Dekany, E. O. Ofek, R. M. Quimby, P. E. Nugent, J. Surace, C. C. Grillmair, J. S. Bloom, M. M. Kasliwal, et al. The palomar transient factory: System overview, performance, and first results. *Publications of the Astronomical Society of the Pacific*, 121(886):1395–1408, 2009.
- [23] Tom Fawcett. An introduction to ROC analysis. *Pattern Recogn. Lett.*, 27(8):861874, June 2006.
- [24] I. Psorakis, S. J. Roberts, M. Ebden, and B. Sheldon. Overlapping community detection using Bayesian non-negative matrix factorization. *Physical Review E*, 83(6), June 2011.

- [25] Vincent Y. F. Tan and Cdric Fvotte. Automatic relevance determination in nonnegative matrix factorization. In *SPARS'09 - Signal Processing with Adaptive Sparse Structured Representations*. Inria Rennes-Bretagne Atlantique, Saint Malo, Royaume-Uni, 2009.
- [26] M. Girvan and M. E. J Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, November 2002.
- [27] M. West, P. J. Harrison, and H. S. Migon. Dynamic generalized linear models and bayesian forecasting. *Journal of the American Statistical Association*, 80(389):73–83, March 1985.
- [28] Gerald J. Bierman. Fixed interval smoothing with discrete measurements. *International Journal of Control*, 18(1):65–75, 1973.
- [29] Seung Min Lee and Stephen J Roberts. Sequential dynamic classification using latent variable models. *The Computer Journal*, 53(9):1415–1429, January 2010.
- [30] Charles Ernest Weatherburn. *A First Course in Mathematical Statistics*. CUP Archive, January 1949.
- [31] Mike West and Jeff Harrison. *Bayesian forecasting and dynamic models*. Springer, 1997.
- [32] Pieter Abbeel and Andrew Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning, ICML '04*, page 1, New York, NY, USA, 2004. ACM.
- [33] R. K Dash, N. R Jennings, and D. C Parkes. Computational-mechanism design: a call to arms. *IEEE Intelligent Systems*, 18(6):40– 47, December 2003.