Bayesian Classifier Combination

Hyun-Chul Kim
Korea Institute of Science and Technology

Zoubin Ghahramani
University of Cambridge

Abstract

Bayesian model averaging linearly mixes the probabilistic predictions of multiple models, each weighted by its posterior probability. This is the coherent Bayesian way of combining multiple models *only* under certain restrictive assumptions, which we outline. We explore a general framework for Bayesian model combination (which differs from model *averaging*) in the context of classification. This framework explicitly models the relationship between each model's output and the unknown true label. The framework does not require that the models be probabilistic (they can even be human assessors), that they share prior information or receive the same training data, or that they be independent in their errors. Finally, the Bayesian combiner does not need to believe any of the models is in fact correct. We test several variants of this classifier combination procedure starting from a classic statistical model proposed by Dawid and Skene (1979) and using MCMC to add more complex but important features to the model. Comparisons on several data sets to simpler methods like majority voting show that the Bayesian methods not only perform well but result in interpretable diagnostics on the data points and the models.

1 Introduction

There are many methods available for classification. When faced with a new problem, where one has little prior knowledge, it is tempting to try many different classifiers in the hope that combining their predictions would give good performance. This has led to the proliferation of classifier combination, a.k.a. ensemble learning, methods (Dietterich, 2000; Tulyakov et al., 2008). Recently, the Netflix Grand Prize, a contest to develop methods for predicting how much people will enjoy a movie according to their movie preferences, was awarded to a team which combined many predictors (Toscher et al., 2009). During the contest, it was widely reported that model combination improves the prediction accuracy (Salakhutdinov et al., 2007; Takács et al., 2007; Bell et al., 2007)

The Bayesian model averaging (BMA) framework appears to be ideally suited to combining the outputs of multiple classifiers. However, this is misleading (Minka, 2002). Before we discuss Bayesian classifier combination (BCC), the topic of this paper, let us review BMA and outline why it is not the right framework for combining classifiers.[1]

Assume there are K different classifiers. Bayesian model averaging starts with a prior over the classifiers, $p(k)$ for the kth classifier. This is meant to capture the (prior) belief in each classifier. Then we observe some data D, and we compute the marginal likelihood or model evidence $p(D|k)$ for each k (which can involve integrating out the parameters of the classifier). Using Bayes rule we compute the posterior $p(k|D) = p(k)p(D|k)/p(D)$ and we use these posteriors to weight the classifiers predictions:

$$p(t_i|\mathbf{x}_i, D) = \sum_{k=1}^{K} p(t_i, k|\mathbf{x}_i, D) = \sum_{k=1}^{K} p(t_i|\mathbf{x}_i, k, D)p(k|D) \tag{1}$$

where $\mathbf{x}_i$ denotes a new input data point and t_i the predicted class label associated with data point i. The key element of this well-known procedure is that the predictive distribution of each classifier is linearly weighted by its posterior probability.

While this approach is appealing and well-motivated from a Bayesian framework, it suffers from three important limitations when misused as a combination

Appearing in Proceedings of the 15^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2012, La Palma, Canary Islands, Spain. Volume 11 of JMLR: W&CP 11.

[1] We have focused on classification, although many of the ideas carry forth to other modelling problems; we return to this in the discussion.

method:

1. It is only valid if we believe that the K classifiers capture mutually exclusive and exhaustive possibilities about how the data was generated. In fact, we might not believe at all that *any* of the K classifiers reflects the true data generation. However, we may still want to be able to combine them to form a prediction.

2. For many classification methods available in the machine learning community, it is not possible to compute, or even define, the marginal likelihood (for example, C4.5, kNN, etc.). Moreover, one should in principle be able to include human experts into any classifier combination framework. One cannot easily define a likelihood function for the human expert from which marginal likelihoods can be computed.

3. Not all classifiers may have observed the same data or started with the same prior assumptions. The Bayesian model averaging framework described above would have difficulties dealing with such cases, since the posterior is computed by conditioning on the same data set.

Here we propose an approach to Bayesian classifier combination which does not assume that any of the classifiers is the true one. Moreover, it does not require that the classifiers be probabilistic; they can even be human experts. Finally, the classifiers can embody widely different prior assumptions about the data, and have observed different data sets.

There are well-known techniques for classifier combination, so called ensemble methods[2], such as bagging, boosting, dagging, random forests (Dietterich, 2000; Breiman, 1996; Freund and Schapire, 1996; Breiman, 2001). These methods train individual classifiers and combine them with their own schemes such as training them with randomly sampled training sets, training them sequentially with training sets sampled in proportional to previously trained classifiers' errors for each data point, or training them with different input features. In this work, we do not restrict how the individual classifiers are trained, but instead assume they are given and fixed. Therefore, our work deals with a different problem from those which are usually handled using ensemble methods.

Another powerful and general method, called stacked generalization can be used to combine lower-level models (Wolpert, 1992). Stacking methods for classifier combination use another classifier which has as inputs the outputs of the individual classifiers. Stacking can be combined with bagging and dagging (Ting and Witten, 1997). Our method can be seen as similar to stacked generalization but using graphical models as higher-level models. It should be possible to extend our method to encompass a fully-Bayesian generalization of stacking, but we leave this for future work.

The method we propose for Bayesian classifier combination in a machine learning context is inspired by the method proposed in Haitovsky et al. (2002) for modelling disagreement between human assessors which in turn is an extension of Dawid and Skene (1979). This method assumes individual classifiers are independent, which is often unrealistic and results in limited performance. We therefore start with these models and propose three extensions for modelling the correlations between individual classifiers. The literature of combining probability distributions is quite extensive, and reviews of other methods including linear, logarithmic and multivariate normal opinion pools, can be found in Genest and Zidek (1986) and Jacobs (1995).

Recently, methods for corroboration which can learn truth values of knowledge and its trust have been proposed. Galland et al. (2010) proposed three algorithms to aggregate disagreeing views and estimate both their truth values and the trust in them. Their models assume that views and facts are all probabilistically independent. Raykar et al. (2010) proposed a supervised learning approach for the case that we have multiple assessors but no gold standard. They estimate the classifier and the ground truth jointly, but they assume that assessors are independent. Kasneci et al. (2011) has proposed a Bayesian framework called CoBayes with an assessment model, a logical model, and an expertise model. With a logical model, they model the dependency between truth values of knowledge, and with an expertise model they model the expertise of an assessor for knowledge. In this model they still assume that assessors are independent. In classifier combination problems, classifiers (or assessors) are usually not independent (since classifiers are trained with correlated training sets, or with correlated features). The models proposed in our paper consider and explicitly attempt to model the correlation between classifiers.

Xu et al. (1992) developed a Bayesian approach based on confusion matrix modelling that could be seen as similar to the first one among our proposed approaches. However, they regarded the confusion matrix as the prior knowledge, rather than something learned from data. Distinguished from their work, we use both prior class proportions and confusion matrices as separate hidden variables, and do Bayesian learning on them. Furthermore, we develop another

[2]Note that the term "ensemble learning" has also been used in the Bayesian literature in a different context to refer to approximate Bayesian model averaging using variational methods.

enhanced model which models the correlation among individual classifiers, and two more models that use undirected graphical models.

The paper is organized as follows. In Section 2, we briefly review the methods proposed in Dawid and Skene (1979) and Haitovsky et al. (2002), which we start with. In Section 3, we propose three extensions of the starting models to deal with the correlation between classifiers. In Section 4, we show the experiment results of our proposed methods and compare with other methods. In Section 5, we conclude the paper with discussions.

2 Independent Models for Bayesian Classifier Combination

2.1 Probabilistic Model for Classifier Combination

We describe the method for observer modelling proposed in Dawid and Skene (1979) with the view of adapting it to classifier combination. For the ith data point, we assume the true label t_i is generated by a multinomial distribution with parameters $\mathbf{p}$: $p(t_i = j|\mathbf{p}) = p_j$, which models the class proportions. Then, we assume that the output $c_i^{(k)}$ of classifier k is generated by a multinomial distribution with parameters $\pi_j^{(k)}$: $p(c_i^{(k)}|t_i = j) = \pi^{(k)}_{j,c_i^{(k)}}$. For simplicity we assume that the classifiers have *discrete* outputs, i.e. $c_i^{(k)} \in \{1, \ldots, J\}$ where J is the number of classes. The extension to individual classifiers which output probability distributions is obviously important and will be explored in the future. The matrix $\pi^{(k)}$ captures the *confusion matrix* for classifier k.

If we assume that the classifier outputs are independent given the true label t_i, we get $p(\mathbf{c}_i, t_i|\mathbf{p}, \boldsymbol{\pi}) = p_{t_i} \prod_{k=1}^{K} \pi^{(k)}_{t_i,c_i^{(k)}}$ where $\mathbf{c}$ denotes the vector of class labels over all classifiers. If we further assume that labels across data points are independent and identically distributed, we obtain the likelihood

$$p(\mathbf{c}, \mathbf{t}|\mathbf{p}, \boldsymbol{\pi}) = \prod_{i=1}^{I} \left\{ p_{t_i} \prod_{k=1}^{K} \pi^{(k)}_{t_i,c_i^{(k)}} \right\}. \tag{2}$$

Usually, $c_i^{(k)}$ is known and the other variables and parameters are unknown. By considering t_i as hidden variables, we can apply the EM algorithm to find ML estimates for $\mathbf{p}$ and $\boldsymbol{\pi}$. This is the approach taken in Dawid and Skene (1979). It should be noted that not only does this perform classifier combination, by inferring the posterior $p(t_i|\mathbf{c}, \mathbf{p}, \boldsymbol{\pi})$, but it provides estimates of interpretable quantities such as the confusion matrices.

2.2 Independent BCC Model

A Bayesian treatment of the probabilistic model in Section 2.1 was proposed in Haitovsky et al. (2002) for combining multiple human raters.[3] They also considered multiple ratings (i.e. $c_{i1}^{(k)} \ldots c_{iM}^{(k)}$) for the same input vector by the same raters. Since artificial classifiers are not usually variable in how they respond to the same input, we do not consider replicates in the ratings. In this section we develop a model for BCC inspired by this work.

The Bayesian model needs priors on the parameters; we used hierarchical conjugate priors. A row of the confusion matrix $\boldsymbol{\pi}_j^{(k)} = [\pi_{j,1}^{(k)}, \pi_{j,2}^{(k)}, \cdots, \pi_{j,J}^{(k)}]$, is modeled to have a Dirichlet distribution

$$p(\boldsymbol{\pi}_j^{(k)}|\boldsymbol{\alpha}_j^{(k)}) = \frac{\Gamma(\sum_{l=1}^{J} \alpha_{j,l}^{(k)})}{\prod_{l=1}^{J} \Gamma(\alpha_{j,l}^{(k)})} \prod_{l=1}^{J} (\pi_{j,l}^{(k)})^{\alpha_{j,l}^{(k)}-1}, \tag{3}$$

where $\boldsymbol{\alpha}_j^{(k)} = [\alpha_{j,1}^{(k)}, \alpha_{j,2}^{(k)}, \cdots, \alpha_{j,J}^{(k)}]$ and $\sum_{l=1}^{J} \pi_{j,l}^{(k)} = 1$. The prior distribution of $\alpha_{j,l}^{(k)}$ is modeled by an exponential distribution with parameters $\lambda_{j,l}$.

$$p(\alpha_{j,l}^{(k)}|\lambda_{j,l}) = \frac{1}{\lambda_{j,l}} \exp(-\frac{\alpha_{j,l}^{(k)}}{\lambda_{j,l}}). \tag{4}$$

All rows of the confusion matrix are assumed independent within and across classifiers. $\boldsymbol{\pi}^{(k)}$ is a collection of $\{\boldsymbol{\pi}_j^{(k)}\}$ and $\boldsymbol{\pi}$ is a collection of $\boldsymbol{\pi}^{(k)}$. Also, $\boldsymbol{\alpha}^{(k)}$ is a collection of $\boldsymbol{\alpha}_j^{(k)}$ and $\boldsymbol{\alpha}$ is a collection of $\boldsymbol{\alpha}^{(k)}$. $\boldsymbol{\lambda}_j = [\boldsymbol{\lambda}_{j,1}, \boldsymbol{\lambda}_{j,2}, \cdots, \boldsymbol{\lambda}_{j,J}]$ and $\boldsymbol{\lambda}$ is a collection of $\boldsymbol{\lambda}_j$. Then, we get

$$p(\boldsymbol{\pi}^{(k)}|\boldsymbol{\alpha}^{(k)}) = \prod_{j=1}^{J} p(\boldsymbol{\pi}_j^{(k)}|\boldsymbol{\alpha}_j^{(k)}), \tag{5}$$

$$p(\boldsymbol{\pi}|\boldsymbol{\alpha}) = \prod_{k=1}^{K} p(\boldsymbol{\pi}^{(k)}|\boldsymbol{\alpha}^{(k)}), \tag{6}$$

$$p(\boldsymbol{\alpha}^{(k)}|\boldsymbol{\lambda}) = \prod_{j=1}^{J} p(\boldsymbol{\alpha}_j^{(k)}|\boldsymbol{\lambda}_j), \tag{7}$$

$$p(\boldsymbol{\alpha}|\boldsymbol{\lambda}) = \prod_{k=1}^{K} p(\boldsymbol{\alpha}^{(k)}|\boldsymbol{\lambda}). \tag{8}$$

The prior for the class proportions $\mathbf{p}$ is also set to be Dirichlet, with hyperparameters $\boldsymbol{\nu}$.

$$p(\mathbf{p}|\boldsymbol{\nu}) = \frac{\Gamma(\sum_{j=1}^{J} \nu_j)}{\prod_{j=1}^{J} \Gamma(\nu_j)} \prod_{j=1}^{J} p_j^{\nu_j - 1}, \tag{9}$$

[3] Unfortunately, this interesting paper presented at the 2002 Valencia Bayesian statistics meeting does not appear to have been published.

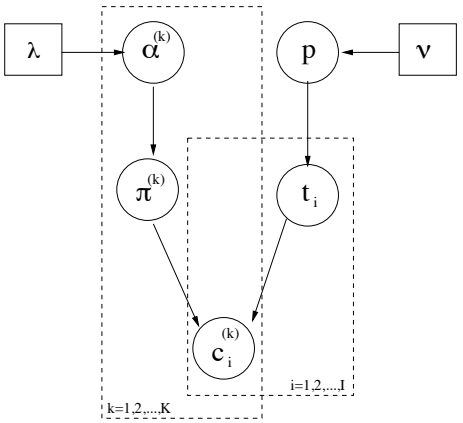

Figure 1: The directed graphical model for IBCC, with plates over classifiers K and data points I.

where $\boldsymbol{\nu} = [\nu_1, \nu_2, \cdots, \nu_J]$.

Based on the above prior, we can get the posterior for all random variables given the observed class labels. Since we assumed independence among classifiers (as in Haitovsky et al. (2002)), the posterior density is

$$p(\mathbf{p}, \boldsymbol{\pi}, \mathbf{t}, \boldsymbol{\alpha}|\mathbf{c}) \propto \prod_{i=1}^{I} \left\{ p_{t_i} \prod_{k=1}^{K} \pi_{t_i, c_i^{(k)}} \right\} p(\mathbf{p}|\boldsymbol{\nu}) p(\boldsymbol{\pi}|\boldsymbol{\alpha}) p(\boldsymbol{\alpha}|\boldsymbol{\lambda}). \quad (10)$$

We call this model the Independent Bayesian Classifier Combination (IBCC) model. The graphical model for IBCC is shown in Fig 1.

Inference for the unknown random variables $\mathbf{p}$, $\boldsymbol{\pi}$, $\mathbf{t}$, and $\boldsymbol{\alpha}$ can be done via Gibbs sampling. From the posterior density function Eq (10), we can get the conditional density functions for sampling as follow.

$$P(\mathbf{p}|rest) \propto \prod_{j=1}^{J} p_j^{|\{i|t_i=j, i=1,2,\cdots,I\}|+\nu_j-1} \quad (11)$$

$$P(\boldsymbol{\pi}_j^{(k)}|rest) \propto \prod_{l=1}^{J} (\pi_{j,l}^{(k)})^{|\{i|t_i=j \wedge c_i^{(k)}=l\}|+\alpha_{j,l}^{(k)}-1} \quad (12)$$

$$P(t_i = j|rest) \propto p_j \prod_{k=1}^{K} \pi_{j, c_i^{(k)}} \quad (13)$$

$$P(\alpha_{j,l}^{(k)}|rest) \propto \frac{\Gamma(\sum_{m=1}^{J} \alpha_{j,m}^{(k)})}{\Gamma(\alpha_{j,l}^{(k)})} (\pi_{j,l}^{(k)})^{\alpha_{j,l}^{(k)}} \exp(-\frac{\alpha_{j,l}^{(k)}}{\lambda_{j,l}^{(k)}})$$

Since the conditional densities on $\mathbf{p}$ and $\boldsymbol{\pi}_j^{(k)}$ are both Dirichlet, they can be sampled easily; also, t_i can be sampled since it is a multinomial distribution. However, the exact conditionals for $\alpha_{j,l}^{(k)}$ are not easily obtained, so we use rejection sampling.

The hyperparameter $\boldsymbol{\nu}$ is set so that classes are roughly balanced a priori; $\boldsymbol{\lambda}$ is set to have bigger values on the diagonal than the off-diagonals. This encodes the prior that classifier outputs are better than random.

The whole process for Gibbs sampling is performed like this. First, we initialize $\mathbf{p}$, $\boldsymbol{\pi}$, $\boldsymbol{\alpha}$, and then sampled $\mathbf{t}$, $\mathbf{p}$, $\boldsymbol{\pi}$, and $\boldsymbol{\alpha}$ in order from the above equations, iteratively. Sampling $\boldsymbol{\pi}$ means sampling $\{\boldsymbol{\pi}_j^{(k)}\}$ for $j = 1, 2, \cdots, J$ and $k = 1, 2, \cdots, K$. Sampling $\boldsymbol{\alpha}$ also means sampling $\{\boldsymbol{\alpha}_{j,l}^{(k)}\}$ for $j = 1, 2, \cdots, J$ and $k = 1, 2, \cdots, K$.

3 Dependent Models for Bayesian Classifier Combination

One of the problems with the above model is the assumption that classifiers are independent, which is often not true in a real situation. Consider several poor classifiers that make highly correlated mistakes and one good classifier. Assuming independence results in performance biased toward majority voting, whereas accounting for the dependence would discount the poor classifiers by an amount related to their correlation. Modelling dependence therefore appears to be an essential element of Bayesian classifier combination.

We propose three models to deal with correlation among classifier outputs. First, we insert a new hidden variable representing the difficulty of each data point—marginalizing this out results in a weakly dependent model. Second, we explicitly model pairwise dependence between classifiers using a Markov Network. Third, we combine the above two ideas.

3.1 Enhanced BCC Model

We enhance the IBCC model by using different confusion matrices according to difficulty of each data point for classification. This captures the idea that qualitatively different combination may be needed for hard points (near the class boundaries) than for easy points which all component classifiers will generally label correctly. Easy data points are classified using a confusion matrix E which is fixed to have diagonal elements $1-\epsilon$ and off-diagonal elements $\epsilon/(J-1)$ (we've also tried extensions where E is learned). For hard data points, each classifier uses its own confusion matrix, $\pi^{(k)}$, as before. Whether a data point is "easy" or "hard" is controlled by independent Bernoulli latent variables s_i (=1, if hard) with mean d_i, which is given a Beta prior. The likelihood term is as follows.

$$p(\mathbf{c}, \mathbf{t}|\mathbf{p}, \boldsymbol{\pi}, \mathbf{s}) = \prod_{i=1}^{I} \left\{ p_{t_i} (\prod_{k=1}^{K} \pi_{t_i, c_i^{(k)}}^{(k)})^{s_i} (\prod_{k=1}^{K} E_{t_i, c_i^{(k)}})^{(1-s_i)} \right\} \quad (14)$$

The distributions for the other random variables are the same as in the IBCC model. The indicator variables s_i are assumed to have a prior Bernoulli distribution with parameter d_i, and d_i is assumed to have a Beta distribution as follows.

$$p(s_i|d_i) = d_i^{s_i}(1-d_i)^{1-s_i} \tag{15}$$

$$p(d_i|\boldsymbol{\beta}) = \frac{\Gamma(\beta_1+\beta_2)}{\Gamma(\beta_1)\Gamma(\beta_2)} d_i^{\beta_1-1}(1-d_i)^{\beta_2-1} \tag{16}$$

We call this model the Enhanced Bayesian Classifier Combination (EBCC) model. The graphical model for the EBCC model in shown in Fig 2.

Inference is again performed using Gibbs and rejection sampling. The conditional density functions to sample s_i and d_i are as follows.

$$p(s_i|rest) \propto (\prod_{k=1}^{K} \pi^{(k)}_{t_i,c_i^{(k)}} d_i)^{s_i} (\prod_{k=1}^{K} E_{t_i,c_i^{(k)}}(1-d_i))^{1-s_i} \tag{17}$$

$$p(d_i|rest) \propto d_i^{(s_i+\beta_1-1)}(1-d_i)^{(1-s_i+\beta_2-1)} \tag{18}$$

The conditional density functions to sample t_i and $\boldsymbol{\pi}$ is changed as follows from the IBCC:

$$P(\boldsymbol{\pi}_j^{(k)}|rest) \propto \prod_{l=1}^{J}((\pi^{(k)}_{j,l})^{s_i})^{|\{i|t_i=j\wedge c_i^{(k)}=l\}|+\alpha^{(k)}_{j,l}-1} \tag{19}$$

$$P(t_i=j|rest) \propto p_j \prod_{k=1}^{K}(\pi_{j,c_i^{(k)}})^{s_i}(E_{t_i,c_i^{(k)}})^{(1-s_i)} \tag{20}$$

The conditional density function to sample $\mathbf{p}$ is not changed.

3.2 Dependent BCC Model

To model correlations between classifiers more directly, we extend the IBCC model with a Markov network (i.e. undirected graphical model). Markov networks are a natural way of explicitly modeling dependence.[4] The part of the model related to confusion matrices is replaced with the following Markov network.

$$p(\mathbf{c}_i|\mathbf{V},\mathbf{W},t_i) = \frac{1}{Z(\mathbf{V},\mathbf{W},t_i)} \exp\{\sum_{j<k} W_{j,k}\delta(c_i^{(j)},c_i^{(k)}) + \sum_k V^{(k)}_{t_i,c_i^{(k)}}\} \tag{21}$$

In this Markov network, $\mathbf{V}$ relates t_i with $c_i^{(k)}$, and $\mathbf{W}$ relates $c_i^{(j)}$ with $c_i^{(k)}$, which models correlations between classifiers; Z is a partition function (normaliser).

[4] An alternative approach would be to develop a directed probabilistic graphical model with latent variables to model correlated classifiers.

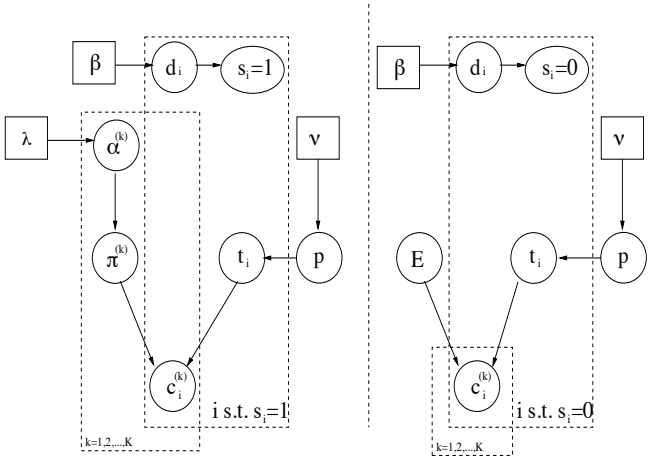

Figure 2: The graphical model for the EBCC model. Note that we have a *different* graphical model conditional on the setting of s_i for each point; the left graph is for "hard" data and the right graph is for "easy" data. (The usual DAG formalism does not represent such dependence of structure on variable setting elegantly.)

The whole likelihood term is as follows.

$$p(\mathbf{c},\mathbf{V}|\mathbf{W},\mathbf{t}) = \prod_{i=1}^{I} p_{t_i} p(\mathbf{c}_i|\mathbf{V},\mathbf{W},t_i) \tag{22}$$

$$= \prod_{i=1}^{I} p_{t_i} \frac{1}{Z(\mathbf{V},\mathbf{W},t_i)} \exp\{\sum_{j<k} W_{j,k}\delta(c_i^{(j)},c_i^{(k)}) + \sum_k V^{(k)}_{t_i,c_i^{(k)}}\} \tag{23}$$

The same priors $p(\mathbf{t}|\mathbf{p})p(\mathbf{p}|\nu)$ as in IBCC are used. As priors for elements of $\mathbf{V}$ and $\mathbf{W}$, we use zero-mean independent Gaussians with variance σ_v^2 and σ_w^2.

$$p(V^k_{lm}) = \mathcal{N}(0,\sigma_v^2) \tag{24}$$

$$p(W^{(j,k)}_{lm}) = \mathcal{N}(0,\sigma_w^2) \tag{25}$$

The priors for $\mathbf{V}$ and $\mathbf{W}$ is set like

$$p(\mathbf{V}) = \prod_{k=1}^{K}\prod_{l=1}^{J}\prod_{m=1}^{J} p(V^k_{lm}), \tag{26}$$

$$p(\mathbf{W}) = \prod_{j<k}\prod_{l=1}^{J}\prod_{m=1}^{J} p(W^{(j,k)}_{lm}). \tag{27}$$

We call this model the Dependent Bayesian Classifier Combination (DBCC) model. Since it's a mix of directed and undirected conditional independence relations it is most simply depicted as a factor graph (Fig 3).

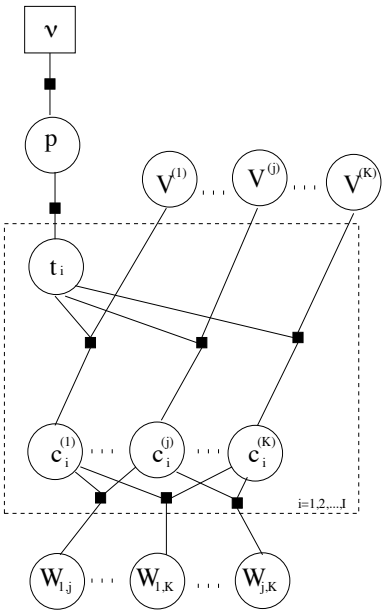

Figure 3: The factor graph for the DBCC model. Each dot represents a factor in the joint probability and connects variables involved in that factor.

Sampling for most of the parameters of this model is again straightforward. The conditional density function to sample t_i is as follows.

$$P(t_i = j|rest) \propto p_j \, p(\mathbf{c}_i|\mathbf{V}, \mathbf{W}, t_i) \tag{28}$$

The conditional density function to sample $\mathbf{p}$ is not changed. However, sampling from $\mathbf{V}$, $\mathbf{W}$ is more subtle due to the partition function, so we implemented it using a Metropolis sampling method.

3.3 Enhanced Dependent BCC model

The Enhanced Dependence BCC model (EDBCC) combines the easy/hard latent variable for the EBCC with the explicit model of correlation between classifiers of the DBCC. For easy data, the conditional probability of each class is given by:

$$p^{easy}(\mathbf{c}_i|\mathbf{U}, t_i) = \frac{1}{Z^e(\mathbf{U}, t_i)} \exp\{\sum_k U_{t_i, c_i^{(k)}}\} \tag{29}$$

$\mathbf{U}$ relates t_i with $c_i^{(k)}$ (playing a role analogous to the E matrix in EBCC). For easy data points, it is assumed that classifiers are independent, for hard data it is assumed to be as in DBCC. The whole likelihood term is as follows.

$$\begin{aligned} p(\mathbf{c}, \mathbf{t}|\mathbf{V}, \mathbf{W}, \mathbf{U}, \mathbf{s}) = &\prod_{i=1}^{I} p_{t_i} [p^{hard}(\mathbf{c}_i|\mathbf{V}, \mathbf{W}, t_i)]^{s_i} \\ &[p^{easy}(\mathbf{c}_i|\mathbf{U}, t_i)]^{(1-s_i)} \qquad (30) \\ = &\prod_{i=1}^{I} p_{t_i} [\frac{1}{Z^h(\mathbf{V}, \mathbf{W}, t_i)} \\ &\exp\{\sum_{j<k} W_{j,k} \delta(c_i^{(j)}, c_i^{(k)}) \\ &+ \sum_k V^{(k)}_{t_i, c_i^{(k)}}\}]^{s_i} \\ &[\frac{1}{Z^e(\mathbf{U}, t_i)} \exp\{\sum_k U_{t_i, c_i^{(k)}}\}]^{(1-s_i)} \qquad (31) \end{aligned}$$

The priors for elements $V_{lm}^{(k)}$ and U_{lm} of $\mathbf{V}$ and $\mathbf{U}$ are Gaussians with mean $\mu_v = (\log(p_v) - \log((1-p_v)/(J-1))\delta(l,m)$ and $\mu_u = (\log(p_u) - \log((1-p_u)/(J-1))\delta(l,m)$[5] and variance σ_v^2 and σ_w^2.

$$p(V_{lm}) = \mathcal{N}(\mu_v, \sigma_v^2) \tag{32}$$

$$p(U_{lm}) = \mathcal{N}(\mu_u, \sigma_u^2) \tag{33}$$

The prior for $\mathbf{U}$ is set as

$$p(\mathbf{V}) = \prod_{l=1}^{J} \prod_{m=1}^{J} p(V_{lm}), \tag{34}$$

$$p(\mathbf{U}) = \prod_{l=1}^{J} \prod_{m=1}^{J} p(U_{lm}). \tag{35}$$

The same prior for $\mathbf{W}$ as used in the DBCC model is used here. The factor graph for the EDBCC model is shown in (Fig 4).

The conditional density functions to sample t_i, s_i,d_i are as follows.

$$\begin{aligned} P(t_i = j|rest) \propto &\, p_j (p^{hard}(\mathbf{c}_i|\mathbf{V}, \mathbf{W}, t_i))^{s_i} \\ &(p^{easy}(\mathbf{c}_i|\mathbf{U}, t_i))^{(1-s_i)} \qquad (36) \\ p(s_i|rest) \propto &\, (p^{hard}(\mathbf{c}_i|\mathbf{V}, \mathbf{W}, t_i) d_i)^{s_i} \\ &(p^{easy}(\mathbf{c}_i|\mathbf{U}, t_i)(1-d_i))^{(1-s_i)} \qquad (37) \\ p(d_i|rest) \propto &\, d_i^{(s_i+\beta_1-1)} (1-d_i)^{(1-s_i+\beta_2-1)} \qquad (38) \end{aligned}$$

The conditional density function to sample $\mathbf{p}$ is not changed. $\mathbf{V}$, $\mathbf{W}$, $\mathbf{U}$ can be sampled by Metroplis sampling method.

[5] This means that means of the priors for $\mathbf{V}^{(k)}$ and $\mathbf{U}$ have the high diagonal elements so that the probability of correct classification for each class is p_v and p_u.

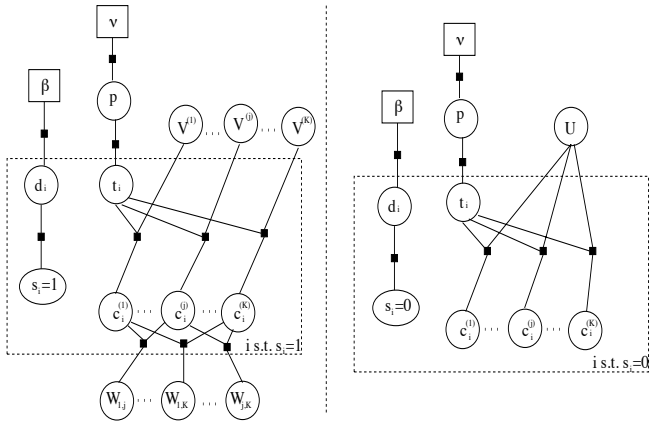

Figure 4: The factor graph for the EDBCC model. Again we have a different graph conditional on the setting of s_i. The left half shows the factor graph for hard data points ($s_i = 1$) and the right half for easy data points.

Data set	DNA	Satellite	UCI digit
# of training set	2000	4435	3823
# of test set	1186	2000	1797
# of classes	3	6	10
# of variables	60	36	64

Table 1: Data set

4 Experimental Results

We compared the Bayesian classifier combination methods on several data sets and using different component classifiers. We used Satellite and DNA data sets from the Statlog project(Michie et al. (1994)) and the UCI digit data set (Blake and Merz (1998)). The detailed information about the data sets is in Table 1.

Our goal was not to obtain the best classifier performance—for this we would have paid very careful attention to the component classifiers and chosen sophisticated models suited to the properties of each data set—rather our goal was to compare the usefulness of different BCC methods even when component classifiers are poor, correlated or trained on partial data. We compared the four variants of the BCC idea outlined above to two other methods: selecting the best classifier using validation data[6] and majority voting[7].

[6]500, 1000, 797 data points were selected from the original test set as a validation set for DNA data set, Satellite data set, UCI digit data set, respectively. The rest of the original test set was used to evaluate the performance.

[7]The performance of majority voting is obtained in an average sense considering the case of ties.

In all BCC models the validation data was used as known t_i to "ground" the estimates of model parameters. In theory this grounding is not necessary: we can treat the labels in the observed data set as simply another classifier's outputs (perhaps the human who hand-labelled the data) and assume that *no* true labels t_i are ever observed. This variant will only do well if the prior enforces some notion that the true label should agree with the majority class predictions; in practice it did not seem to perform as well in initial experiments but needs to be explored further. BCC results are based on comparing the posterior mode of t_i for data points in the test set to the true observed label.

We did two sets of experiments. In Experiment 1, we combined the outputs of the same type of classifier trained on disjoint training sets.[8] In Experiment 2, we trained several different classifiers on the (same) whole training set.[9]

For IBCC and EBCC models, we sampled 50,000 samples and averaged every 100 samples except for the first 10,000 samples. For DBCC and EDBCC models, we sampled 100,000 or more samples and averaged every 100 samples except for the first 10,000 samples.

For IBCC and EBCC models, the hyperparameters were set as $\lambda_{ij} = 1 + 7\delta(i,j)$, $\nu_j = 1$, $\beta = 0.5$, $\epsilon = 0.01, \text{or } 0.003$. The initial values for random variables were set as follows. t_i was set to the result of the majority voting. $\mathbf{p}$ is initialized by the result of counting t_i. π was initialized by the result of counting t_i and $c_{lm}^{(k)}$. $\alpha_{lm}^{(k)}$ were all initialized with 2. Initial values for d_i were sampled from Beta distribution with $\beta = 0.5$ and initial values for s_i were sampled from multinomial distribution with parameters d_i. For DBCC and EDBCC models, the hyperparameters σ_v, σ_w were all set to 3. For EDBCC model, σ_u was set to 3 and p_v and p_u was set to 0.85 and 0.99, respectively. The initial values for $\mathbf{V}$, $\mathbf{W}$, $\mathbf{U}$ are all set to the means of the priors. In Metropolis sampling for $\mathbf{V}$, $\mathbf{W}$, $\mathbf{U}$, Gaussian proposal distribution with proposal width $\sigma = 0.1$ was

[8]For DNA data set, we had 5 disjoint training sets and trained C4.5 for each of them. For Satellite data set, we had 4 disjoint training sets and trained C4.5 for each of them. For UCI digit data set, we had 3 disjoint training sets and trained SVM with 2nd-order polynomial kernel and $C = 100.0$.

[9]For DNA data set, we trained 5 classifiers: C4.5 (C1), SVM with 2nd-order polynomial kernel and $C = 100.0$ (C2), 1-Nearest Neighbor (C3), logistic regression (C4), and Fisher discriminant (C5). For Satellite data set, we trained 4 classifiers: C4.5 (C1), SVM with 2nd-order polynomial kernel and $C = 100.0$ (C2), logistic regression (C3), and Fisher discriminant (C4). For UCI digits, we trained 3 classifiers: SVM with linear kernel (C1), SVM with 2nd-order polynomial kernel (C2), and SVM with Gaussian kernel ($\sigma = 0.01$) (C3), where all SVMs has $C = 100.0$.

	Experiment 1		
Data set	Satellite	UCI digit	DNA
C1	0.1920	0.0320	0.1210
C2	0.1820	0.0320	0.1458
C3	0.1910	0.0390	0.1283
C4	0.1860	N/A	0.1254
C5	N/A	N/A	0.1050
Val	0.1910	0.0390	0.1458
MV	0.1505	0.0263	0.0780
IBCC	0.1510	0.0260	**0.0758**
EBCC	0.1490	0.0260	**0.0758**
DBCC	0.1520	**0.0240**	0.0904
EDBCC	**0.1410**	0.0290	0.0889
	Experiment 2		
Data set	Satellite	UCI digit	DNA
C1	0.1420	0.0460	0.0714
C2	0.1450	0.0250	0.1137
C3	0.1760	0.0290	0.2551
C4	0.2560	N/A	0.1020
C5	N/A	N/A	0.0598
Val	0.1450	0.0250	0.0598
MV	0.1460	0.0250	0.0415
IBCC	**0.1240**	0.0250	**0.0408**
EBCC	0.1250	0.0250	**0.0408**
DBCC	0.1300	**0.0230**	0.0423
EDBCC	0.1280	**0.0230**	0.0466

Table 2: The performances of individual classifiers and various combination schemes in the case of using the same classifier with the disjoint training sets (Experiment 1) and different classifiers with the same whole training set (Experiment 2)

used.

Table 2 shows the performance of each classifier and BCC combination strategy for both experiments. "Val" and "MV" denote selecting the classifier with smallest validation set errors, and majority voting, respectively. "N/A" in C4 or C5 means that there is no result because only 3 or 4 classifiers are combined for the corresponding data sets (See the footnotes 8 and 9 below). IBCC and EBCC have similar performance and EBCC model is always better than or as good as majority voting. Model selection by validation set is quite bad especially in Experiment 1. BCC methods are always better than or as good as model selection by validation. The dependent factor graph models (DBCC and EDBCC) do not always work well. Especially on the DNA data set, they did not seem to learn reasonable parameters, perhaps because the DNA data set is relatively small and has biased class distribution. For Satellite and UCI digits, it learned reasonable parameters and showed comparable performance to other BCC methods.

5 Discussion

We have shown several approaches to classifier combination which explicitly model the relation between true labels and classifier outputs. They worked reasonably well and in our experiments the best method was always one of the BCC methods, rather than an individual classifier, majority voting or validation selection. The parameters in BCC models can be interpreted reasonably and give useful information such as confusion matrices, correlations between classifiers, and difficulty of data points.

We emphasized that Bayesian classifier combination is not the same as Bayesian model averaging. Our approach is closely related to *supra-Bayesian* methods for aggregating opinions (Genest and Zidek, 1986; Jacobs, 1995). Other models and extensions are certainly possible; we outline some here.

The model presented here can be generalised to combine classifiers that output probability distributions. In this case, e.g. instead of a matrix $\pi^{(k)}$ we need a model that relates t_i to class probability distributions. Conditional Dirichlet distributions seem a natural choice for this. Similarly, there is no reason to restrict this approach to combining classifiers. Combining different regressors or rankings are another important problems which could be handled by an appropriate choice of the distribution of outputs given true target. Our models can be also extended by adapting an expertise model inside the CoBayes model (Kasneci et al., 2011), so that the expertise of each classifier can be estimated.

One practical limitation of the DBCC approach is that the computation time for the exact partition function of the Markov network grows exponentially with the number of classifiers. Efficient approximations to the partition function, many of which have been recently developed, could be used here (Murray and Ghahramani, 2004; Welling and Sutton, 2005; Welling and Parise, 2006; Qi, 2005). Such approximate inferences could also be tractable replacements for all the MCMC computations.

A Bayesian generalization of "stacking" methods is another important avenue for research. The combiner, in our setup, does not see the input data. If the combiner does see the input and the outputs of all the other classifiers, then it should model the full relation between true labels, inputs, and other classifier outputs. This can be done either as a joint generative model, or as a Bayesian discriminative model, both of which present interesting avenues for extending the work in this paper.

References

Bell, R. M., Koren, Y., Volinsky, C., 2007. The BellKor solution to the Netflix Prize. http://www2.research.att.com/~volinsky/netflix/ProgressPrize2007BellKorSolution.pdf.

Blake, C. L., Merz, C. J., 1998. UCI Repository of machine learning databases. Irvine, CA: University of California, Department of Information and Computer Science.

Breiman, L., August 1996. Bagging predictors. Machine Learning 24, 123–140.

Breiman, L., October 2001. Random forests. Machine Learning 45, 5–32.

Dawid, A., Skene, A., 1979. Maximum Likelihood Estimation of Observer Error-rates using the EM Algorithm. Applied Statistics 28, 20–28.

Dietterich, T. G., 2000. Ensemble methods in machine learning. In: Proceedings of the First International Workshop on Multiple Classifier Systems. Springer-Verlag, London, UK, pp. 1–15.

Freund, Y., Schapire, R. E., 1996. Experiments with a New Boosting Algorithm. In: International Conference on Machine Learning. pp. 148–156.

Galland, A., Abiteboul, S., Marian, A., Senellart, P., 2010. Corroborating information from disagreeing views. In: Proceedings of the third ACM international conference on Web search and data mining. WSDM '10. ACM, pp. 131–140.

Genest, C., Zidek, J. V., 1986. Combining probability distributions: A critique and an annotated bibliography. Statistical Science 1, 114–118.

Haitovsky, Y., Smith, A., Liu, Y., 2002. Modelling disagreements among and within raters' assessments from the bayesian point of view. In Draft. Presented at the Valencia meeting 2002.

Jacobs, R., 1995. Methods for combining experts' probability assessments. Neural Computation 7, 867–888.

Kasneci, G., Gael, J. V., Stern, D., Graepel, T., 2011. CoBayes: bayesian knowledge corroboration with assessors of unknown areas of expertise. In: Proceedings of the fourth ACM international conference on Web search and data mining. WSDM '11. ACM, pp. 465–474.

Michie, D., Spiegelhalter, D., Taylor, C., 1994. Machine Learning, Neural and Statistical Classification. Ellis Horwood Limited.

Minka, T. P., 2002. Bayesian model averaging is not model combination. MIT Media Lab note.

Murray, I., Ghahramani, Z., 2004. Bayesian learning in undirected graphical models: Approximate MCMC algorithms. In: Proceedings of the 20th Annual Conference on Uncertainty in Artificial Intelligence (UAI-04). AUAI Press, Arlington, Virginia, pp. 392–399.

Qi, Y., 2005. Extending expectation propagation for graphical models. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, USA.

Raykar, V. C., Yu, S., Zhao, L. H., Valadez, G. H., Florin, C., Bogoni, L., Moy, L., 2010. Learning From Crowds. Journal of Machine Learning Research, 1297–1322.

Salakhutdinov, R., Mnih, A., Hinton, G., 2007. Restricted Boltzmann machines for collaborative filtering. In: Proceedings of the International Conference on Machine Learning. Vol. 24. pp. 791–798.

Takács, G., Pilászy, I., Németh, B., Tikk, D., 2007. On the Gravity recommendation system. In: Proc. of KDD Cup Workshop at SIGKDD'07, 13th ACM Int. Conf. on Knowledge Discovery and Data Mining. San Jose, CA, USA, pp. 22–30.

Ting, K., Witten, I. H., 1997. Stacking bagged and dagged models. In: Proc. of ICML'97. San Francisco, CA.

Toscher, A., Jahrer, M., Bell, R., 2009. The BigChaos Solution to the Netflix Grand Prize. http://www.netflixprize.com/assets/GrandPrize2009_BPC_BigChaos.pdf.

Tulyakov, S., Jaeger, S., Govindaraju, V., Doermann, D., 2008. Review of Classifier Combination Methods. In: Simone Marinai, H. F. (Ed.), Studies in Computational Intelligence: Machine Learning in Document Analysis and Recognition. Springer, pp. 361–386.

Welling, M., Parise, S., 2006. Bayesian random fields: The bethe-laplace approximation. In: Proceedings of the Proceedings of the Twenty-Second Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-06). AUAI Press, Arlington, Virginia, pp. 512–519.

Welling, M., Sutton, C., 2005. Learning in Markov Random Fields with Contrastive Free Energies. In: Tenth International Workshop on Artificial Intelligence and Statistics (AISTATS).

Wolpert, D. H., 1992. Stacked generalization. Neural Networks 5, 241–259.

Xu, L., Krzyzak, A., Suen, C. Y., May 1992. Methods of combining multiple classifiers and their applications to handwriting recognition. IEEE Transactions on Systems, Man, and Cybernetics 22 (3), 418–435.