

Bayesian Classifier Combination

Zoubin Ghahramani and Hyun-Chul Kim*

Gatsby Computational Neuroscience Unit

University College London

London WC1N 3AR, UK

<http://www.gatsby.ucl.ac.uk>

{zoubin,hckim}@gatsby.ucl.ac.uk

September 8, 2003

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 very 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 [1] and using MCMC to add more complex but important features to the model. Comparisons on several datasets 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 had led to the proliferation of classifier combination, a.k.a. ensemble learning, methods [3].

The Bayesian model averaging (BMA) framework appears to be ideally suited to combining the outputs of multiple classifiers. However, this is misleading. 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

*The work was done while H-C.K. was a visiting student from POSTECH, South Korea.

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

Assume there are K different classifiers. Bayesian model averaging starts with a prior over the classifiers, $p(k)$ for the k th 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:

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

where 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: 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. The human expert would not naturally define a likelihood function 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 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([3, 9]).², such as bagging, boosting, and dagging. These methods try to make individual classifiers different by training them with different training sets or weighting data points differently. This is because it is important to make the individual classifiers as independent as possible for ensemble methods to work well. In this work, we do not restrict how the individual classifiers are trained, but instead assume they are given and fixed.

Another powerful and general method, called stacked generalisation can be used to combine lower-level models [10]. Stacking methods for classifier combination use another classifier which has as inputs both the original inputs and the output of the individual classifiers. Stacking can be combined with bagging and dagging [9]. In

²Note 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.

this work, we do not use the input vectors and we explicitly model the errors and correlations between individual classifiers. Therefore, our work deals with a different problem from those which are usually handled using ensemble and stacking methods. It should be possible to extend our method to encompass a fully-Bayesian generalisation of stacking, but we leave this for future work.

The method we propose for Bayesian classifier combination in a machine learning context is directly derived from the method proposed in [5] for modelling disagreement between human assessors, which in turn is an extension of [2]. 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 [4] and [6].

2 Independent Models for Bayesian Classifier Combination

2.1 Probabilistic Model for Classifier Combination

We describe the method proposed in [2] with the view of applying it to classifier combination. For the i th data point, we assume the true label t_i is generated by a multinomial distribution with parameters \mathbf{p} : $p(t_i = j|\mathbf{p}) = p_j$. Then, we assume that the output $c_i^{(k)}$ of classifier k is generated by a multinomial distribution with parameters $\pi_j^{(k)}$: $p(c_i^{(k)}|t_i = j) = \pi_{j, c_i^{(k)}}^{(k)}$. For simplicity we assume that the classifiers have *discrete* outputs, i.e. $c_i^{(k)} \in \{1, \dots, J\}$ 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 $p(\mathbf{c}, t_i|\mathbf{p}, \boldsymbol{\pi}) = p_{t_i} \prod_{k=1}^K \pi_{t_i, c_i^{(k)}}^{(k)}$ 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

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

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 [2] and we also provide further details in a longer version of this paper [7]. It should be noted that not only does this perform classifier combination, 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 recently proposed in [5] for combining multiple human raters. They also considered multiple ratings (i.e. $c_{i1}^{(k)} \dots c_{iM}^{(k)}$) 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.

The Bayesian model needs priors on the parameters; we used hierarchical conjugate priors. A row of the confusion matrix $\pi_j^{(k)} = [\pi_{j,1}^{(k)}, \pi_{j,2}^{(k)}, \dots, \pi_{j,J}^{(k)}]$, is modeled to have a Dirichlet distribution with hyperparameters $\alpha_j^{(k)} = [\alpha_{j,1}^{(k)}, \alpha_{j,2}^{(k)}, \dots, \alpha_{j,J}^{(k)}]$. The prior distribution of $\alpha_{j,l}^{(k)}$ is modeled by an exponential distribution with parameters $\lambda_{j,l}$. All rows are assumed independent within and across classifiers; even so it is easy to bias the prior to prefer diagonal confusion matrices. (Detailed expressions are provided in the longer version of the paper [7].) The prior for the class proportions \mathbf{p} is also set to be Dirichlet, with hyperparameters ν .

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 [5]), the posterior density is

$$p(\mathbf{p}, \boldsymbol{\pi}, \mathbf{t}, \boldsymbol{\alpha} | \mathbf{c}) \propto \prod_{i=1}^I \left\{ p_{t_i} \prod_{k=1}^K \pi_{t_i, c_i^{(k)}} \right\} p(\mathbf{p} | \boldsymbol{\nu}) p(\boldsymbol{\pi} | \boldsymbol{\alpha}) p(\boldsymbol{\alpha} | \boldsymbol{\lambda}). \quad (3)$$

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

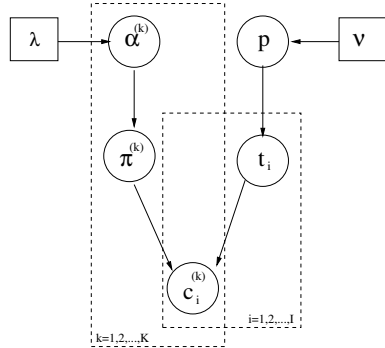


Figure 1: The directed graphical model for IBCC, with plates over classifiers K and data points I .

Inference for the unknown random variables \mathbf{p} , $\boldsymbol{\pi}$, \mathbf{t} , and $\boldsymbol{\alpha}$ can be done via Gibbs sampling. Since the conditional densities on \mathbf{p} and $\boldsymbol{\pi}^{(k)}$ 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.

Hyperparameters $\boldsymbol{\nu}$ are set so that class 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.

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—marginalising 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. 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.

$$p(\mathbf{c}, \mathbf{t} | \mathbf{p}, \boldsymbol{\pi}, \mathbf{s}) = \prod_{i=1}^I \left\{ p_{t_i} \left(\prod_{k=1}^K \pi_{t_i, c_i^{(k)}}^{(k)} \right)^{s_i} \left(\prod_{k=1}^K E_{t_i, c_i^{(k)}} \right)^{(1-s_i)} \right\} \quad (4)$$

We call this model the Enhanced Bayesian Classifier Combination (EBCC) model. The graphical model for the EBCC model is shown in Fig 2. Inference is again performed using Gibbs and rejection sampling.

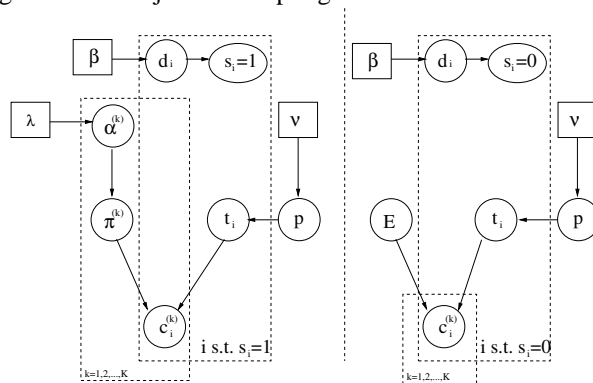


Figure 2: The graphical model for the EBCC model. Note that we have a *different* graphical model conditional on the setting of s_i for each point; the left graph is for “hard” data and the right graph is for “easy” data. (The usual DAG formalism does not represent such dependence of structure on variable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

3.2 Dependent BCC Model

To model correlations between classifiers more directly, we extend the IBCC model with a Markov network. The part related to confusion matrices is replaced with the following Markov network.

$$p(\mathbf{c}_i | \mathbf{V}, \mathbf{W}, t_i) = \frac{1}{Z(\mathbf{V}, \mathbf{W}, t_i)} \exp\left\{ \sum_{j < k} W_{j,k} \delta(c_i^{(j)}, c_i^{(k)}) + \sum_k V_{t_i, c_i^{(k)}} \right\} \quad (5)$$

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). 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 . Sampling for most of the parameters of this model is again straightforward. However, sampling from \mathbf{V} , \mathbf{W} is more subtle due to the partition function, so we implemented it using a Metropolis sampling method. 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).

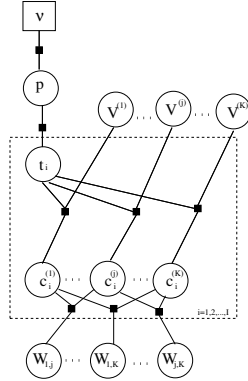


Figure 3: The factor graph for the DBCC model. Each dot represents a factor in the joint probability and connects variables involved in that factor.

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:

$$p^{easy}(c_i^{(\cdot)} | \mathbf{U}, t_i) = \frac{1}{Z^e(\mathbf{U}, t_i)} \exp\left\{ \sum_k U_{t_i, c_i^{(k)}} \right\} \quad (6)$$

\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 factor graph for the EDBCC model is shown in (Fig 4).

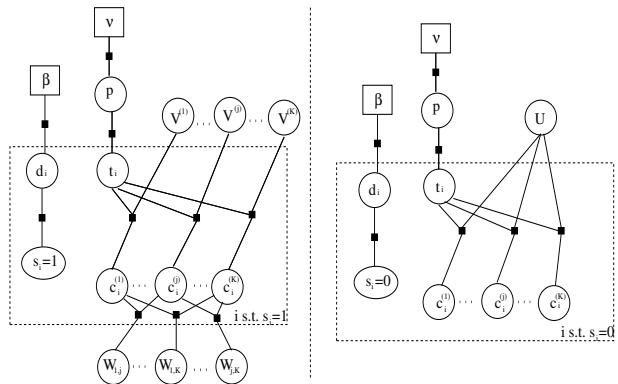


Figure 4: The factor graph for the EDBCC model. Again we have a different graph conditional on the setting of s_i . The left half shows the factor graph for hard data points ($s_i = 1$) and the right half for easy data points.

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([8]) and the UCI digit data set ([1])³. 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 data⁴ and majority voting. 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 did not seem to work 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.⁵ In Experiment 2, we trained several different classifiers on the (same) whole training set.⁶ For all BCC models ran

³The DNA data set has a training set of 2000, a test set of 1186 with 3 classes and 50 variables. Satellite has a training set of 4435, a test set of 2000 with 6 classes and 36 variables. UCI digit data set has a training set of 3823, a test set of 1797, 10 classes and 64 variables.

⁴500, 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.

⁵For 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 $C = 100.0$.

⁶For 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),

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

Table 1: 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)

the MCMC sampler for at least 50000 samples, averaging every 100th and discarding the first 10000. The dependent models (DBCC and EDBCC) were generally slower to converge. Details of the sampling and hyperparameter settings are provided in the longer version of the paper.

Table 1 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. 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.

We examined the \mathbf{V} and \mathbf{W} matrices inferred by the dependent methods and the difficulty assigned to each point by the enhanced methods. These have intuitive interpretations and may provide useful diagnostics, one of the strengths of the BCC approach. Due to space limitations we do not display these matrices or discuss them in this paper; see [7].

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 $C = 100.0$.

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 some of them were always better than or as good as 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 emphasised that Bayesian classifier combination is not the same as Bayesian model averaging. Our approach is closely related to *supra-Bayesian* methods for aggregating opinions [4, 6]. Other models and extensions are certainly possible; we outline some here.

Clearly the model presented here needs to 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 regressions is another important problem which could be handled by an appropriate choice of the density of regressor outputs given true target.

A Bayesian generalisation 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.

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. Such approximate inference could also be a tractable replacements for all the MCMC computations.

References

- [1] C. L. Blake and C. J. Merz. *UCI Repository of machine learning databases*. Irvine, CA: University of California, Department of Information and Computer Science., 1998.
- [2] A. Dawid and A. Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied Statistics*, 28:20–28, 1979.
- [3] T. G. Dietterich. Ensemble methods in machine learning. *First International Workshop on Multiple Classifier Systems, LNCS*, pages 1–15, 2000.
- [4] C. Genest and J. V. Zidek. Combining probability distributions: A critique and an annotated bibliography. *Statistical Science*, 1:114–118, 1986.
- [5] Y. Haitovsky, A. Smith, and Y. Liu. Modelling disagreements among and within raters’ assessments from the bayesian point of view. *In Draft. Presented at the Valencia meeting 2002*, 2002.
- [6] R. Jacobs. Methods for combining experts’ probability assessments. *Neural Computation*, 7:867–888, 1995.
- [7] H. Kim and Z. Ghahramani. Graphical models for Bayesian classifier combination. *GCNU Technical Report (in preparation)*, 2003.

- [8] D. Michie, D. Spiegelhalter, and C. Taylor. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood Limited, 1994.
- [9] K. Ting and I. H. Witten. Stacking bagged and dagged models. In *Proc. of ICML'97*. San Francisco, CA, 1997.
- [10] D. H. Wolpert. Stacked generalization. *Neural Networks*, 5:241–259, 1992.