

BAYESIAN NEURAL NETWORKS FOR PROSTATE CANCER STUDY

M. GHOSH

University of Florida

Joint Work with S. CHAKRABORTY, T. MAITI,
D. KIM AND A. TEWARI

OUTLINE

1. A MOTIVATING EXAMPLE
2. NEURAL NETWORKS
3. HIERARCHICAL BAYESIAN NEURAL NETWORK MODELS
4. BIVARIATE LOGIT MODEL
5. EXAMPLE 1: A PROSTATE CANCER STUDY WITH CLINICAL COVARIATES
6. EXAMPLE 2: A PROSTATE CANCER STUDY WITH GENE EXPRESSION MICROARRAY DATA
7. SUMMARY AND CONCLUSION

1 A MOTIVATING EXAMPLE

- Prostate cancer is one of the most common cancers in men comprising approximately 33% of all cancers.
- It is the second leading cause of cancer death in men, exceeded only by lung cancer.
- It is the most common type of cancers found in American men, other than skin cancer.
- The American Cancer Society estimates that there will be about 230,900 new cases of prostate cancer in the United States in the year 2004. About 29,900 men will die of this disease.
- It is empirically found that 1 man in 6 will get prostate cancer during his lifetime, and 1 man in 32 will die of this disease.

- Typically 50% diagnosed with prostate cancer undergo radical prostatectomy.
- Management of prostate cancer:
 - > 1.5 billion dollars: direct medical expenses;
 - > 2.5 billion dollars: indirect costs.
- The cancer is either organ-confined, or it could spread outside the organ.
- When organ-confined, there are several options for treating and curing this disease. Otherwise, surgery is the only option.
- In extreme cases of outside spread, the cancer could very easily recur within a short time even after surgery and subsequent radiation therapy.

- It is important to know, based on pre-surgery biopsy results how likely the cancer is organ-confined or not.
- Some indicators of non-organ confined prostate cancer:
(i) Margin Positivity (MP), (ii) Seminal Vesicle (SV) Positivity, and (iii) Lymph Nodal (LN) Disease.
- Long term studies of radical prostatectomy show that recurrence of cancer within 10 years is seen in as many as 58% of patients with positive margins; 57% with seminal vesicle (SV) positivity.

- A variety of combinations of variables are proposed to construct relatively uncomplicated nomograms (or charts) for clinicians managing prostate cancer.
- The intent of these nomograms is to provide clinicians with tables that are easy to understand and utilize in day to day practice.
- These nomograms provide probabilities of having features indicative of non-organ confined prostate cancer - currently Partin's nomogram is the most well-used nomogram.
- We will use bivariate neural network methods for predicting jointly the probabilities of presence of margin positivity and seminal vesicle positivity in a patient, and compare it with several other methods.
- Based on these probabilities, the doctors can make decision for whether or not the cancer is organ-confined, and in marginal cases, can prescribe further diagnostic tests.

2 NEURAL NETWORKS

- A neural network is a set of simple computational units that are highly interconnected.
- These units are also called “nodes” and loosely represent the biological neuron.
- Neural networks most commonly used in engineering applications are *multilayer perceptron* networks, also known as *backpropagation* or *feedforward* networks.
- The networks take on a set of inputs x_i , and compute from them some outputs O_i using possibly certain number of layers.

- In a typical neural network with one hidden layer, the outputs might be computed as

$$O_i = b_0 + \sum_{j=1}^M b_j \psi \left(c_{j0} + \sum_{s=1}^p x_{is} c_{js} \right), i = 1, \dots, n.$$

- In the above c_{js} is the weight from the input x_{is} to the hidden unit j . Similarly, b_j is the weight attached to the hidden unit j . The c_{j0} and b_0 are the biases for the hidden nodes and the output units. However, if necessary, they can be absorbed in the c_{js} and the b_j .
- The function ψ is referred to as the *activation function*. Typically, ψ is nonlinear. Some of the most common choices of ψ are the logistic and the hyperbolic tangent functions.

- Cybenko (1989) showed that if the number of hidden nodes tends to infinity, these can be used as universal approximators of any continuous function in a compact range.
- In a classical approach, the weights and biases in neural networks are learnt based on a set of training cases (\mathbf{x}_i, z_i) with inputs \mathbf{x}_i and targets z_i .
- Standard neural network training procedures adjust the weights and the biases in the network so as to minimize some error measure, most commonly the sum of squared deviations between the network outputs and the targets, that is
$$\sum_1^n (O_i - z_i)^2.$$
- Finding these weights and biases that minimize the chosen error function is usually done by a method referred to as the *backpropagation algorithm*.
- Cheng and Titterington (Statistical Science, 1994) and Warner and Misra (American Statistician, 1996) are excellent reviews describing the neural network methodology in the language of statisticians.

- In Bayesian approach to neural network learning, the objective is to find the predictive distributions for the target values in new test cases given the inputs for those cases as well as inputs and targets for the training cases.
- Buntine and Weigand (1991, Complex Systems, 5) and Mackay (1992, Neural Computation 4) implemented Bayesian procedures via Gaussian approximations.
- Neal (Bayesian Learning for Neural Networks, 1996) applied Hybrid Markov Chain Monte Carlo (MCMC).
- Muller and Rios Insua (1998, Neural Computation, 10) introduced efficient MCMC schemes based on fixed and variable number of nodes.
- Rios Insua and Muller (1998) - nonparametric models with unconstrained number of hidden nodes.

3 HIERARCHICAL BAYESIAN NEURAL NETWORK MODELS

- $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n$ independent binary random vectors;
 $\mathbf{y}_i = (y_{i1}, y_{i2})^T, i = 1, \dots, n.$
- Conditional on p_{i1} and p_{i2} , y_{i1} and y_{i2} are independent with
 $y_{ik} | p_{ik} \sim \text{Bin}(1, p_{ik});$
- $\theta_{ik} = \text{logit}(p_{ik}), k = 1, 2, i = 1, \dots, n;$
- $\theta_{ik} = \sum_{j=1}^M \beta_{jk} \psi(\mathbf{x}_i^T \boldsymbol{\gamma}_{jk}) + e_{ik};$
- $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})^T$ is the input vector or covariates corresponding to \mathbf{y}_i , M is the number of nodes in the hidden layer of the neural network, and ψ is the activation function.
- We consider M as fixed and take ψ as the logistic function.

- $\mathbf{e}_i = (e_{i1}, e_{i2})^T \stackrel{iid}{\sim} N_2(\mathbf{0}, \boldsymbol{\Sigma}_e)$.
- Rewrite $\boldsymbol{\theta}_i = \boldsymbol{\beta}^T \boldsymbol{\eta}_i(\boldsymbol{\gamma}) + \mathbf{e}_i$, where $\boldsymbol{\theta}_i = (\theta_{i1}, \theta_{i2})^T$,
 $\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \boldsymbol{\beta}_2)$, $\boldsymbol{\beta}_k = (\beta_{1k}, \dots, \beta_{Mk})^T$,
 $\boldsymbol{\eta}_i = (\eta_{i1}(\boldsymbol{\gamma}), \eta_{i2}(\boldsymbol{\gamma}))$,
 $\eta_{ik} = (\psi(\mathbf{x}_i^T \boldsymbol{\gamma}_{1k}), \dots, \psi(\mathbf{x}_i^T \boldsymbol{\gamma}_{Mk}))^T$.
- First Stage Priors:

$$\begin{aligned} \boldsymbol{\beta}_k &\stackrel{iid}{\sim} N_M(\mu_{\beta_k} \mathbf{1}_M, \sigma_{\beta_k}^2 \mathbf{I}_M); \\ \boldsymbol{\gamma}_{jk} &\stackrel{iid}{\sim} N_p(\boldsymbol{\mu}_{\gamma_k}, \mathbf{S}_{\gamma_k}); \\ \boldsymbol{\Sigma}_e &\sim \text{IW}(c_e, c_e^{-1} \mathbf{C}_e^{-1}), \end{aligned}$$

where $j = 1, \dots, M$ and $k = 1, 2$.

- Second Stage Priors:

$$\begin{aligned} \mu_{\beta_k} &\sim N(a_{\beta_k}, A_{\beta_k}) \\ \sigma_{\beta_k}^2 &\sim \text{IG}(c_{\beta_k}/2, c_{\beta_k} \mathbf{C}_{\beta_k}/2) \\ \boldsymbol{\mu}_{\gamma_k} &\sim N_p(\mathbf{a}_{\gamma_k}, \mathbf{A}_{\gamma_k}) \\ \mathbf{S}_{\gamma_k} &\sim \text{IW}(c_{\gamma_k}, c_{\gamma_k}^{-1} \mathbf{C}_{\gamma_k}^{-1}) \end{aligned}$$

$$\begin{aligned}
& f(\boldsymbol{\theta}, \boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \boldsymbol{\beta}, \mu_{\beta_1}, \mu_{\beta_2}, \sigma_{\beta_1}^2, \sigma_{\beta_2}^2, \boldsymbol{\Sigma}_e, \mathbf{S}_{\gamma_1}, \mathbf{S}_{\gamma_2} \mid \mathbf{y}) \\
& \propto \prod_{i=1}^n [p_{i1}^{y_{i1}} (1 - p_{i1})^{1-y_{i1}} p_{i2}^{y_{i2}} (1 - p_{i2})^{1-y_{i2}}] \\
& \times |\boldsymbol{\Sigma}_e|^{-\frac{n}{2}} \exp\left[-\frac{1}{2} \sum_{i=1}^n \{\boldsymbol{\theta}_i - \boldsymbol{\beta}^T \boldsymbol{\eta}_i(\boldsymbol{\gamma})\}^T\right. \\
& \times \boldsymbol{\Sigma}_e^{-1} \{\boldsymbol{\theta}_i - \boldsymbol{\beta}^T \boldsymbol{\eta}_i(\boldsymbol{\gamma})\}] \\
& \times \prod_{k=1}^2 \left[(\sigma_{\beta_k}^2)^{-\frac{1}{2}M} \exp\left\{-\|\boldsymbol{\beta}_k - \mu_{\beta_k} \mathbf{1}_M\|^2 / (2\sigma_{\beta_k}^2)\right\} \right] \\
& \times \prod_{k=1}^2 \left[|\mathbf{S}_{\gamma_k}|^{-\frac{1}{2}M} \exp\left\{-\frac{1}{2} \sum_{j=1}^M (\gamma_{jk} - \boldsymbol{\mu}_{\gamma_k})^T\right. \right. \\
& \times \mathbf{S}_{\gamma_k}^{-1} (\gamma_{jk} - \boldsymbol{\mu}_{\gamma_k}) \left. \left. \right\} \right] \\
& \times \prod_{k=1}^2 \left[\exp\left\{-(2A_{\beta_k})^{-1} (\mu_{\beta_k} - a_{\beta_k})^2\right\} \right] \\
& \times \prod_{k=1}^2 \left[\exp\left\{-\frac{1}{2} (\boldsymbol{\mu}_{\gamma_k} - \mathbf{a}_{\gamma_k})^T \mathbf{A}_{\gamma_k}^{-1} (\boldsymbol{\mu}_{\gamma_k} - \mathbf{a}_{\gamma_k})\right\} \right] \\
& \times \prod_{k=1}^2 \left[\exp\left\{-(c_{\beta_k} C_{\beta_k}) / (2\sigma_{\beta_k}^2)\right\} (\sigma_{\beta_k}^2)^{-\frac{c_{\beta_k}}{2}-1} \right]
\end{aligned}$$

$$\times \prod_{k=1}^2 \left[|\mathbf{S}_{\gamma_k}|^{-\frac{1}{2}(c_{\gamma_k} + p + 1)} \exp \left\{ -\frac{1}{2} \text{tr}(\mathbf{S}_{\gamma_k}^{-1} c_{\gamma_k} \mathbf{C}_{\gamma_k}) \right\} \right] \cdot$$

$$\times |\boldsymbol{\Sigma}_e|^{-\frac{1}{2}(c_e + 3)} \exp \left\{ -\frac{1}{2} \text{tr}(\boldsymbol{\Sigma}_e^{-1} c_e \mathbf{C}_e) \right\}.$$

Our objective is to find the predictive distribution for the target values $\mathbf{y}^{(n+1)}$ in a new “test” case given the inputs in that case, and the inputs and the targets in the training cases.

$$P(\mathbf{y}^{(n+1)} | \text{Training Set}) = \int P(\mathbf{y}^{(n+1)} | \boldsymbol{\theta}) \Pi(\boldsymbol{\theta} | \text{Training Set}) d\boldsymbol{\theta}.$$

4 BIVARIATE LOGIT MODEL

- We write $\mathbf{y} = (y_1, y_2)^T$, where y_1 and y_2 each takes only the values 0 and 1.
- Let $p_{rs} = P(y_1 = r, y_2 = s)$, $r, s = 0, 1$, be the joint probabilities, and $p_j = P(y_j = 1)$, $j = 1, 2$, be the marginal probabilities.
- The bivariate logistic model (BLM) is specified by modelling the marginal distributions of each y_j , and the odds ratio, $\psi = (p_{00}p_{11}) / (p_{10}p_{01})$.
- The model is

$$\text{logit}(p_j) = \eta_j(\mathbf{x}), \quad j = 1, 2; \quad \log(\psi(\mathbf{x})) = \eta_3(\mathbf{x}),$$

where $\eta_j = \boldsymbol{\beta}_j^T \mathbf{x}$, $j = 1, 2, 3$, and \mathbf{x} is the vector of covariates.

- The probability p_{11} can be obtained from p_1 , p_2 and ψ as

$$p_{11} = \begin{cases} \frac{1}{2}(\psi - 1)^{-1} \{a - \sqrt{a^2 + b}\} & \text{if } \psi \neq 1 \\ p_1 p_2 & \text{if } \psi = 1 \end{cases}$$

where $a = 1 + (p_1 + p_2)(\psi - 1)$ and
 $b = -4\psi(\psi - 1)p_1 p_2$.

- The other three joint probabilities p_{rs} can then be obtained easily from the marginals and p_{11} .
- The above bivariate logistic model is fitted using the iteratively reweighted least squares technique.
- The BLM is similar to the bivariate probit model but has several advantages: it is computationally simpler, and odds ratios are preferred to correlation coefficients when describing the association between two binary variables.
- However in theory, there is no reason why other link functions could not be used for marginal probabilities.

5 EXAMPLE 1: A PROSTATE CANCER STUDY WITH CLINICAL COVARIATES

- Cohort of 834 patients; 600 were selected by simple random sample as training cases, and remaining 234 as test cases.

Outcome y_i for the i th patient

- (a) presence or absence of margin positivity (coded as 1 or 0)
 - (b) presence or absence of SV positivity (coded as 1 or 0)
- Input factors
 - (i) gleason score (measured in an ordinal scale of 2-10).
 - (ii) $\log[\text{prostate specific antigen (PSA)}+1]$
 - (iii) unilateral or bilateral tumor (coded as 0 or 1)
 - We fit a single layer feedforward neural network with fixed number of hidden nodes M . We tried several values of M and observed that the prediction error stabilizes at $M = 8$.
 - So in our final model we use a neural network with 8 hidden nodes.

- The joint prediction probability of margin positivity and SV positivity is given by

$$P(y_{u1} = r, y_{u2} = s | \mathbf{y}) = E \left[\frac{\exp(r\theta_1 + s\theta_2)}{(1 + \exp(\theta_1))(1 + \exp(\theta_2))} \mid \mathbf{y} \right],$$

- The above posterior expectation is approximated by

$$\frac{1}{B} \sum_{i=1}^B \frac{\exp(r\theta_1^{(i)} + s\theta_2^{(i)})}{(1 + \exp(\theta_1^{(i)}))(1 + \exp(\theta_2^{(i)}))},$$

where $\boldsymbol{\theta}^{(i)}$ are generated from $N_2 \left((\boldsymbol{\beta}^{(i)})^T \boldsymbol{\eta}(\boldsymbol{\gamma}^{(i)}), \boldsymbol{\Sigma}_e^{(i)} \right)$, $\boldsymbol{\beta}^{(i)}$, $\boldsymbol{\gamma}^{(i)}$, $\boldsymbol{\Sigma}_e^{(i)}$ are obtained from the i th MCMC sample, and B is the total number of such samples used.

- For our problem we used 5 independent chains each of length 50,000. To reduce the autocorrelation, every fifth sample is kept. We discard the first half as the burn in and used the last half of each chain and finally pooled all the samples from the 5 different chains to produce the final estimate.

- There are only 4 different values that \mathbf{y}_u can take; $(0, 0)^T$, $(1, 0)^T$, $(0, 1)^T$, $(1, 1)^T$.
- Corresponding to a future observation \mathbf{x}_u we compute the value of the predictive probability for all combinations of r and s and predict a \mathbf{y}_u to take that particular combination for which the computed predictive probability is maximum.
- In order to examine sensitivity in the choice of priors, we have considered several different choices of near-diffuse but proper priors. The misclassification rate remains stable with all such choices.
- We report the results for two such choices.
 - (i) $c_e = 5$, $\mathbf{C}_e = 5\mathbf{I}_2$;
 $c_{\beta_1} = c_{\beta_2} = 0.5$, $\mathbf{C}_{\beta_1} = \mathbf{C}_{\beta_2} = 15$;
 $c_{\gamma_1} = c_{\gamma_2} = 15$, $\mathbf{C}_{\gamma_1} = \mathbf{C}_{\gamma_2} = 16\mathbf{I}_4$;
 $a_{\beta_1} = a_{\beta_2} = 0$, $A_{\beta_1} = A_{\beta_2} = 1000$;
 $\mathbf{a}_{\gamma_1} = \mathbf{a}_{\gamma_2} = \mathbf{0}$, $\mathbf{A}_{\gamma_1} = \mathbf{A}_{\gamma_2} = 100\mathbf{I}_4$;
 - (ii) $c_e = 10$, $\mathbf{C}_e = \mathbf{I}_2$;
 $c_{\beta_1} = c_{\beta_2} = 0.05$, $\mathbf{C}_{\beta_1} = \mathbf{C}_{\beta_2} = 1$;
 $c_{\gamma_1} = c_{\gamma_2} = 10$, $\mathbf{C}_{\gamma_1} = \mathbf{C}_{\gamma_2} = \mathbf{I}_4$;
 $a_{\beta_1} = a_{\beta_2} = 0$, $A_{\beta_1} = A_{\beta_2} = 1000$;
 $\mathbf{a}_{\gamma_1} = \mathbf{a}_{\gamma_2} = \mathbf{0}$, $\mathbf{A}_{\gamma_1} = \mathbf{A}_{\gamma_2} = 1000\mathbf{I}_4$.

- Table 1 reports the percentage of correct joint prediction of MP and SV in the 234 test set samples with the hyperparameter values given in (i) and (ii) after we build our model on the 600 training samples.
- We say a prediction is correct if both SV positivity and MP are correctly predicted.
- For the choice of hidden nodes, we started with $M = 4$ hidden nodes, and after $M = 8$, we noticed that increasing the number of hidden nodes does not improve our prediction sufficiently relative to the increased computing time and complexity.
- Table 2 compares our method with some other standard methods such as Ripley's neural network, Neal's Bayesian neural network and bivariate logistic models.

- Both in Neal's neural network and classical neural network we used 8 hidden nodes similar to our Bayesian neural network.
- Ripley's R *nnet* routine is used with the entropy fit option.
- For Neal's Bayesian neural network, we used his software with 3 inputs and 8 hidden nodes.
- As a neural network model may have several local minima, we used 10 networks with different random starting points and averaged the prediction over all 10 networks as the final prediction.
- We provide percentages of correct joint prediction.
- We brought in 300 new cases for model validation from a somewhat different population. All the methods discussed above applied on this new set of data also. The outcome or prediction probability in the validation set is given in the second row of Table 1 and Table 2 under the name "Validation Set".

Table 1. Percentage of correct prediction in the 234 test cases and 300 validation cases using our bivariate Hierarchical Bayesian Neural Network (HBNN) in Example 1.

	(i)		(ii)	
	$M = 8$	$M = 12$	$M = 8$	$M = 12$
Test Set	86.99	86.75	87.00	87.00
Validation Set	88.00	88.00	87.89	88.00

Table 2. Percentage of correct prediction in the 234 test cases and 300 validation cases using our bivariate Hierarchical Bayesian Neural Network (HBNN) as well as Ripley's Classical Neural Network (Ripley), Neal's Bayesian Neural Network (Neal), Bivariate Logistic Regression Models (BLM), and Univariate Hierarchical Bayesian Neural Network (UHBNN) in Example 1.

	HBNN	Ripley	Neal	BLM	UHBNN
Test Set	86.99	75.64	76.12	72.65	82.35
Validation Set	88.00	79.33	78.91	74.00	84.09

6 EXAMPLE 2: A PROSTATE CANCER STUDY WITH GENE EXPRESSION MICROARRAY DATA

- The recent advent of DNA microarray technique has made simultaneous monitoring of thousands of gene expressions possible.
- Staging and diagnosis based on gene expression profiles may provide more information than standard morphology, and thereby more accurately predict MP and SV positivity.
- We consider a dataset coming from a study of gene expression in benign and malignant prostate tissue.
- Gene expression levels were measured using a cDNA microarray containing 9,984 human genes.
- We have $n = 35$ observations.
- We excluded all those genes from our analysis which are missing in more than ten percent of the prostate cancer samples and have low standard deviation across samples. After this initial gene filtering we have $p = 6,546$ genes.

- Among these 6,546 genes, some are still missing for some of the observations. We impute the missing gene expression by the sample mean of that class.
- The gene expression data is summarized by an $n \times p$ matrix; so x_{ij} is the measurement of the expression level of the j th gene for the i th sample ($i = 1, \dots, n; j = 1, \dots, p$).
- Then we take a log base 2 transformation on these expression levels x_{ij} , normalize and scale the log transformed measurements.
- In the response we have \mathbf{y} a bivariate binary vector which indicates the presence or absence of MP and SV in a patient.

- Out of these 6,546 genes, many genes do not contain information that is useful for determining the differences between the samples. These genes should not be used for prediction; also sometimes they may even contain noise that can lead to incorrect classification.
- Beside this, speed of training a neural network can be improved considerably by reducing the input information. It is impossible to train a neural network with so many input variables or covariates.
- Hence a simple criteria, proposed by Dudoit *et al.* (2002) is used to rank the marginal relevance of each gene in class separation based on the ratio of between group and within group sums of squares.
- According to the values of \mathbf{y} i.e., $(0, 0)^T$, $(1, 0)^T$, $(0, 1)^T$, $(1, 1)^T$ we have 4 groups, 1,2,3, and 4 respectively. Let t_i denote the group of the i th patient. Let t_i take the values 1, 2, 3, or 4. For a gene k , the ratio is

$$BW(k) = \frac{\sum_i \sum_h I(t_i = h) (\bar{x}_{hk} - \bar{x}_{.k})^2}{\sum_i \sum_h I(t_i = h) (x_{ik} - \bar{x}_{hk})^2}$$

where $\bar{x}_{.k}$ and \bar{x}_{hk} denote the average expression level of gene k across all tumor samples and across samples belonging to class h only.

- After ranking the genes according to the criteria we select top 20 genes in our model. So in our final model we have $p = 21$ input nodes including the intercept.
- As we have only 35 samples, instead of splitting the data into test and training sets, we check our model accuracy by leave one out cross validation method.
- In this example after 10 hidden nodes the cross validation error remains the same, so finally we kept $M = 10$ in our model.
- Two choices of hyperparameters:
 - (i) $c_e = 5, \mathbf{C}_e = 5\mathbf{I}_2; c_{\beta_1} = c_{\beta_2} = 0.5, C_{\beta_1} = C_{\beta_2} = 15;$
 $c_{\gamma_1} = c_{\gamma_2} = 22, \mathbf{C}_{\gamma_1} = \mathbf{C}_{\gamma_2} = 25\mathbf{I}_{21};$
 $a_{\beta_1} = a_{\beta_2} = 0, A_{\beta_1} = A_{\beta_2} = 1000;$
 $\mathbf{a}_{\gamma_1} = \mathbf{a}_{\gamma_2} = \mathbf{0}, \mathbf{A}_{\gamma_1} = \mathbf{A}_{\gamma_2} = 100\mathbf{I}_{21};$
 - (ii) $c_e = 10, \mathbf{C}_e = \mathbf{I}_2;$
 $c_{\beta_1} = c_{\beta_2} = 0.05, C_{\beta_1} = C_{\beta_2} = 1;$
 $c_{\gamma_1} = c_{\gamma_2} = 30, \mathbf{C}_{\gamma_1} = \mathbf{C}_{\gamma_2} = \mathbf{I}_{21};$
 $a_{\beta_1} = a_{\beta_2} = 0, A_{\beta_1} = A_{\beta_2} = 1000;$
 $\mathbf{a}_{\gamma_1} = \mathbf{a}_{\gamma_2} = \mathbf{0}, \mathbf{A}_{\gamma_1} = \mathbf{A}_{\gamma_2} = 1000\mathbf{I}_{21}.$

Table 3. Leave one out cross validation error of prediction using bivariate Hierarchical Bayesian Neural Network (HBNN) in Example 2.

	$M = 5$	$M = 10$	$M = 20$
Hyperparameter Choice (i)	6	3	3
Hyperparameter Choice (ii)	5	3	4

Table 4. Leave one out cross validation error using our bivariate Hierarchical Bayesian Neural Network (HBNN) as well as Ripley's Classical Neural Network (Ripley), Neal's Bayesian Neural Network (Neal), Bivariate Logistic Regression Models (BLM), and Univariate Hierarchical Bayesian Neural Network (UHBNN).

	HBNN	Ripley	Neal	BLM	UHBNN
Cross validation error	3	7	6	8	6

7 SUMMARY AND CONCLUSION

- We have introduced a hierarchical Bayesian neural network model for the analysis of bivariate binary data. Rather than a deterministic search for minimization of some error measure, it uses a stochastic search method motivated by a Bayesian model, and uses the posterior predictive distribution to predict a future observation.
- Our method establishes a dependence among the components of the output variable, builds a strong learning algorithm, and finally has excellent prediction power. This is revealed in the two examples considered.
- Bayesian learning integrates over the posterior distribution for the network parameters, rather than picking a single “optimal” set of parameters.
- Using a hierarchical prior, we can automatically determine how relevant each input or covariate is in predicting the MP and SV.
- If a covariate is irrelevant in predicting the MP or SVP, the hyperparameters corresponding to its weight will tend to be small, thus forcing the weight to be near zero.

- We have used both clinical and gene expression microarray covariates in our neural network model, and have found the outcomes to be superior to the other competing methods.
- It is particularly noticed that in a gene expression microarray study where the sample size is very small, our bivariate HBNN is the most effective and accurate compared to the other standard methods.
- Selecting differentially genes is very important for the performance of our bivariate HBNN predictor and utmost care must be taken to identify the active genes from the nonactive genes.
- If cancer is found in prostate, physicians try to determine the stage, or the extent, of the disease. This method can help the doctors to identify the stage of prostate cancer much more accurately by jointly predicting certain features like margin positivity and seminal vesicle positivity, which are strong indicators of non-organ confined cancer.

- The models considered so far are fully parametric. It is possible to enrich the proposed class of models by adopting a semiparametric hierarchical Bayesian approach.
- Moreover, as a general methodology, we can consider a variable number of nodes, and assign a Poisson or negative binomial prior to the number of nodes. A reversible jump MCMC algorithm can be devised to select the number of nodes.
- Apart from this a stochastic search variable selection method can be integrated with our bivariate HBNN model to select differentially expressed gene and predict the margin and SV positivity simultaneously.

LEMMA. (Besag,1974). Let X_1 and X_2 be two random variables with pdf $f(x_1, x_2)$. It is assumed that the joint support of X_1 and X_2 is Cartesian product of the supports of the conditional pdf's of X_1 given X_2 and of X_2 given X_1 . Then the joint pdf $f(x_1, x_2)$ is uniquely determined from the conditional pdf's $f(x_2|x_1)$ and $f(x_1|x_2)$.

Proof. We begin with

$$\begin{aligned} \frac{f(x_1, x_2)}{f(y_1, y_2)} &= \frac{f(x_1, x_2)}{f(y_1, x_2)} \cdot \frac{f(y_1, x_2)}{f(y_1, y_2)} \\ &= \frac{f(x_2)f(x_1|x_2)}{f(x_2)f(y_1|x_2)} \cdot \frac{f(y_1)f(x_2|y_1)}{f(y_1)f(y_2|y_1)} \\ &= \frac{f(x_1|x_2)}{f(y_1|x_2)} \cdot \frac{f(x_2|y_1)}{f(y_2|y_1)}. \end{aligned}$$

Now write

$$\begin{aligned} \frac{1}{f(y_1, y_2)} &= \frac{\int \int f(x_1, x_2) dx_1 dx_2}{f(y_1, y_2)} \\ &= \int \int \frac{f(x_1, x_2)}{f(y_1, y_2)} dx_1 dx_2. \end{aligned}$$

- The neural network model should be contrasted with a standard non-linear mixed effects model. In the latter, both the fixed effects and random effects are identified with specific covariates. In contrast, neither the β_j nor the γ_j are identified with any of the covariates x_i . Indeed, based on the given model, the β_j or the γ_j themselves are not identifiable.
- To see this in a simple case, consider a logistic activation function with $M = 2$ and $p = 2$. Then

$$\sum_{j=1}^2 \beta_j \psi(\mathbf{x}_i^T \boldsymbol{\gamma}_j) = \beta_1 [1 + \exp(-\gamma_{11}x_{i1} - \gamma_{12}x_{i2})]^{-1} + \beta_2 [1 + \exp(-\gamma_{21}x_{i1} - \gamma_{22}x_{i2})]^{-1}.$$

It is easy to see from the above that when $\beta_1 = \beta_2$, two different sets of values of $(\gamma_{11}, \gamma_{12}, \gamma_{21}, \gamma_{22})$, namely, $(1, -1, -1, 1)$ and $(-1, 1, 1, -1)$ yield the same value of

$$\sum_{j=1}^2 \beta_j \psi(\mathbf{x}_i^T \boldsymbol{\gamma}_j).$$

- However, as a special case of Lemma 1 of Ghosh *et al.* (2000), the joint posterior of the parameters is proper if the joint prior is proper, even in the case of a nonidentifiable likelihood. In this study we use only proper priors so that the nonidentifiability of parameters is not an issue.