Optimizing Automatic Classification of Neural Cells

Jung-Wook Bang

Department of Computing, Imperial College
London, United Kingdom
jbang@doc.ic.ac.uk

Abstract. Effective automatic classification of neural cell could be done by using Bayesian decision trees on features extracted from data. Data is normally taken from studies in which the cultures were photographed using a Photonic Science microscope camera. However Bayesian networks are based on a formal assumption that the unconnected nodes are conditionally independent given the states of their parent nodes. This assumption does not necessarily hold in practice and may lead to loss of accuracy. We propose a methodology whereby naïve Bayesian networks are adapted by the addition of hidden nodes to model the data dependencies more accurately. We examined the methodology in a computer vision application to classify and count the neural cell automatically. Our results show that a modified network with two hidden nodes achieved significantly better performance with an average prediction accuracy of 83.9% compared to 59.31% achieved by the original network. In this paper we also justify the improvement of performance by examining the changes in network accuracy using four network accuracy measurements; the Euclidean accuracy, the Cosine accuracy, the Jensen-Shannon accuracy and the MDL score. So that this approach utilized to optimise the automatic classification of neural cell morphology.

1 Introduction

Developmental biologists are frequently interested in classifying the development of cells in culture. In this way they can determine the effects of pollutants (or other reagents) on growth. Oligodendrocytes are a class of cell that is frequently studied. They provide the myelin sheath needed for nervous impulse conduction. Failure of these cells to develop leads to the disease multiple sclerosis. In studies, biologists view culture dishes under a microscope and attempt to count the cells using a small number of classes. This is, however, a difficult, inaccurate and subjective method that could be greatly improved by using computer vision.

Bayesian networks employ both probabilistic reasoning and graphical modelling can be adapted to computer vision. This approach, however, represents the relationships between variables in a given domain based on the assumption of conditional independence [1]. However, in practice the variables may contain a certain degree of dependence and as a result the validity of a network can be questioned. Pearl proposed a star-structure methodology to overcome the dependency problem by introducing a hidden node when any two nodes have strong condi-

I. Rudomín, J. Vázquez-Salceda, J. Díaz de León. (Eds.). e-Health: Application of Computing Science in medicine and Health Care. © IPN, México 2003.

tional dependency given a common parent [1,2]. Pearl's idea was to simulate the common cause between two nodes by introducing a hidden node, though he did not provide a mechanism for determining the parameters of a discrete node. In some cases, hidden nodes can be introduced subjectively through expert knowledge. However, it is rare to have information about common causes that result in variables being partially correlated. It is therefore necessary, in many cases, to use an objective method to introduce a hidden node into a network and estimate statistically the number of states and the link matrices. In neural networks, hidden layers have been widely used to discover symmetries or replicated structures; in particular, Boltzmann machine learning and backward propagation training have been proposed to determine hidden nodes [3].
Friedman proposed a technique called the Model Selection Expectation-Maximization (MS-EM) to update a network by discovering a hidden node. This approach, however, required defining the size of the hidden node prior to its process being carried out [4].

Bang and Gillies extended Kwoh and Gillies' idea [5] by proposing a diagonal propagation method to form a symmetric propagation scheme that compensated for the weakness of forward propagation in the gradient descent process [6]. This method utilized gradient descent to update the conditional probabilities of the matrices linking a hidden node to its parent's and children. Experiments in neural cell morphology showed significant improvement in performance [7]. The results showed that a modified network with two hidden nodes achieved 41.4% improvement in performance.

In this paper, we examine Bayesian networks with the hidden node methodology in terms of the improvement of classification accuracy and network accuracy that can be directly applied to improve the classification accuracy of neural cell morphology.

2 Hidden Node Methodology

2.1 General Concepts

Hidden nodes are introduced to a network (BN_H) by first identifying a triple (A, B, C in Figure 2.1) where the children nodes have a high conditional dependency given some states of the parent node in the original network (BN_O). Once the hidden node is introduced into the network, its states and link matrices are set to make B and C conditionally independent given A (BN_H). This requires the use of a representative data set with values for A, B and C.

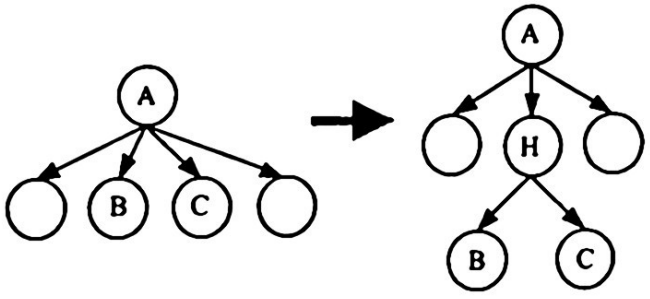


Fig. 2.1. Adding hidden nodes

Having inserted the hidden node H, the three conditional probability matrices (CPTs) linked to the hidden node are initialised. Empirical results showed that the optimal number of states of a hidden node lies between the largest numbers of states among the other nodes (A, B and C) and two times the largest states [6].

To obtain the CPTs, we compute the derivative of the error cost function E with respect to each component of the vector $\vec{p}$ containing all the conditional probabilities. The vector derivative, $\nabla E(\vec{p})$, is called the gradient of E with respect to $\vec{p}$ and denoted as

$$\nabla E(\vec{p}) \equiv \left[\frac{\partial E}{\partial p_1}, \frac{\partial E}{\partial p_2}, \ldots \frac{\partial E}{\partial p_n}\right]. \tag{1}$$

The training rule of gradient descent is given as

$$\vec{p}_i \leftarrow \vec{p}_i + \Delta\vec{p}_i. \tag{2}$$

where $\Delta\vec{p}_i$ is $-\mu\nabla E$, and μ is a positive constant called the step size (or a learning rate) that determines how fast the process converges. For individual probabilities the rule is further expanded to

$$p_i \leftarrow p_i - \mu\left[\frac{\partial E}{\partial p_i}\right]. \tag{3}$$

The objective of gradient descent is to determine iteratively the minimum error:

$$E(\vec{p}) = E_{\min}. \tag{4}$$

or equivalently

$$E'(\vec{p}) = 0. \tag{5}$$

In our case, using backward propagation the error function can be written as

$$E(\vec{p}) = \sum_{data}\sum_{x=1}^{|A|}\left[D(a_x) - P'(a_x)\right]^2. \tag{6}$$

where |A| is the number of values of A, a_x is the x^{th} value, and the vector $\vec{p}$ contains, as its elements, all the unknown conditional probabilities in the link matrices. $P'(a_x)$ is the posterior probability of the parent node A and is calculated by instantiating the children and propagating these values through the hidden node. $D(a_x)$ is the desired value of the parent node originally from the data.

An exact gradient solution is only available in the linear cases. We, therefore, need to expand the equations to derive discrete operating equations.

2.2 Operating Equations for Gradient Descent in Bayesian Networks

The operating equations for gradient descent are derived using the chain rule to differentiate the error function. The equations for diagonal propagation are summarized.

In right-to-left propagation we instantiate root node A and child node C simultaneously. The information from the instantiated nodes propagates through hidden node H until it reaches node B. We need to determine the derivative of the error cost function $E(p)$ with respect to the three link matrix elements. For example consider $\partial E(p)/\partial P(b_j \mid h_t)$. The derivative is expanded using a chain rule as

$$\frac{\partial E(p)}{\partial P(b_j \mid h_t)} = \sum_{y=1}^{|B|}\left[\frac{\partial E(p)}{\partial P'(b_y)}\frac{\partial P'(b_y)}{\partial \pi(b_j)}\frac{\partial \pi(b_j)}{\partial P(b_j \mid h_t)}\right]. \tag{7}$$

The first term on the right side of the above equation is the derivative of the sum of square error cost function $E(p)$ with respect to $P'(b_y)$. Differentiating $E(p)$ with respected to $P'(b_y)$ yields

$$\frac{\partial E(p)}{\partial P'(b_y)} = \sum_{y=1}^{|B|} -2\left[D(b_y) - P'(b_y)\right]. \tag{8}$$

The second term of the equation is the derivative of the posterior probabilities of a target node $P'(b_y)$ with respect to $\pi(b_j)$. Initially the posterior probabilities are denoted as the product of the evidence of the hidden node H and the prior probability distribution of target node B, respectively.

$$P'(b_y) = \beta\lambda(b_y)\pi(b_y) = \beta\pi(b_y). \tag{9}$$

where the normalization factor β is $1\Big/\sum_{y=1}^{|B|}\pi(b_y)$ and $\lambda(b_j)$ has unit values. In the denominator of β the sum is taken over the states of target node B. The derivation of the second term yields

$$\frac{\partial P'(b_y)}{\partial \pi(b_j)} = \beta\frac{\partial \pi(b_y)}{\partial \pi(b_j)} + \pi(b_y)\frac{\partial \beta}{\partial \pi(b_j)}. \tag{10}$$

where $\dfrac{\partial \beta}{\partial \pi(b_j)} = \dfrac{1}{\left[\sum_{y=1}^{|B|}\pi(b_y)\right]^2} = -\beta^2$

The second term is, furthermore, extended with respect to $\pi(b_j)$ for two cases; $j = y$ and $j \neq y$.

$$\beta\delta(j,y) - \pi(b_y)\beta^2 = \beta\left[\delta(j,y) - \beta\pi(b_y)\right]. \tag{11}$$

where $\delta(j,y) = 1$ for $j = y$, 0 otherwise.
The last term is the derivative of $\pi(b_j)$ with respect to $P(b_j \mid h_t)$. Initially we have

$$\pi(b_y) = \sum_{s=1}^{|H|} P(b_y \mid h_s)\pi_b(h_s). \tag{12}$$

$$= \sum_{s=1}^{|H|} P(b_y \mid h_s)\lambda(h_s)\pi(h_s). \tag{13}$$

Then the derivation yields

$$= \sum_{s=1}^{|H|} P(b_y \mid h_s)\lambda(h_s)\pi(h_s). \tag{14}$$

After combing the three terms, we have

$$\frac{\partial E(p)}{\partial P(b_j \mid h_t)} = \sum_{y=1}^{|B|} \left(\sum_{data} -2\left[D(b_y) - P'(b_y)\right] \cdot \sum_{data} \beta\left[\delta(j,y) - \beta\pi(b_y)\right] \cdot \lambda(h_t)\pi(h_t) \right). \tag{15}$$

Other elements are derived similarly as follows

$$\frac{\partial E(p)}{\partial P(c_k \mid h_t)} = \sum_{y=1}^{|B|} \left[\frac{\partial E(p)}{\partial P'(b_y)} \frac{\partial P'(b_y)}{\partial \varepsilon(h_t)} \frac{\partial \varepsilon(h_t)}{\partial \lambda(h_t)} \frac{\partial \lambda(h_t)}{\partial P(c_k \mid h_t)} \right]. \tag{16}$$

$$= \sum_{y=1}^{|B|} \left(\sum_{data} -2\left[D(b_y) - P'(b_y)\right] \cdot \beta P(b_y \mid h_t)^2 \cdot \pi(h_t) \cdot P'(c_k) \right). \tag{17}$$

where ε is a posterior probability of hidden node H.

$$\frac{\partial E(p)}{\partial P(h_t \mid a_i)} = \sum_{y=1}^{|B|} \left[\frac{\partial E(p)}{\partial P'(b_y)} \frac{\partial P'(b_y)}{\partial \varepsilon(h_t)} \frac{\partial \varepsilon(h_t)}{\partial \pi(h_t)} \frac{\partial \pi(h_t)}{\partial P(h_t \mid a_i)} \right]. \tag{18}$$

$$= \sum_{y=1}^{|B|} \left(\sum_{data} -2\left[D(b_y) - P'(b_y)\right] \cdot \beta P(b_y \mid h_t)^2 \cdot \pi(h_t) \cdot P'(c_k) \right). \tag{19}$$

The operating equations for right-to-left propagation are found simply by swapping b and c in the above equations.

3 Case Study: Neural Cell Morphology

Our data was taken from studies in which the cultures were photographed using a Photonic Science microscope camera. Biologists classified the cells in the pictures into four developmental classes. One data set had 12 progenitor cells, 24 immature type 1, 15 immature type 2 and 9 fully differentiated cells. The images were then

processed to extract several features, of which five proved to have good discriminant properties [11]. These were called the Scholl coefficient [12], the fractal dimension [13], the 2nd moment [14], the total length and the profile count.

We conducted a series of tests using the cell class as a hypothesis node, and the five measured features as variables. In particular, we were interested in the possibility of improving the prediction accuracy of the networks with the help of hidden nodes.

3.1 Naïve Bayesian Networks

A naïve Bayesian network was constructed using a randomly selected training data set and then evaluated with a randomly selected test data set. The process was repeated 1000 times for each test. Fig. 3.1.1 shows the naïve Bayesian network with five variables connected to a root node, *neuron type*.

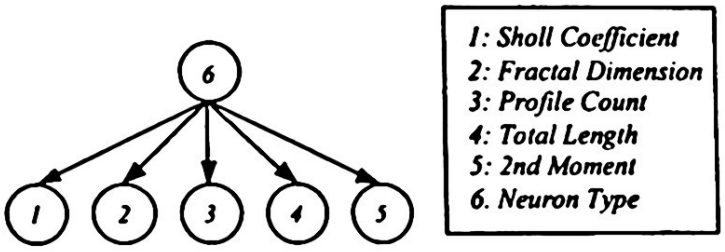


Fig. 3.1.1. A naïve Bayesian network in the morphometric analysis of neural cells

The prediction accuracy of the network was measured in terms of the success ratio (%) of finding the correct answer by comparing the calculated posterior probability of the network with the desired posterior probability in the data. We conducted initial study to decide the ratio of the training data to the test set data. Even though 90/10 performed well, we used 70/30 and 80/20 through out the experiment since 90/10 could yield a biased outcome due to the small number in the test set. After we conducted our series of experiments based on these two ratios, we averaged them to generate the final results. The initial naïve network produced an average prediction accuracy of 59.31%.

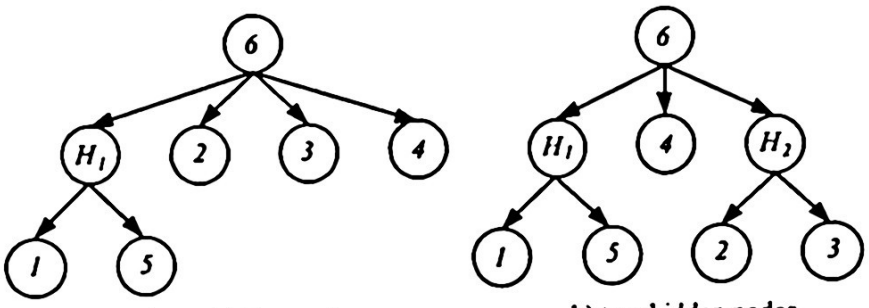
3.2 Training Hidden Node(s)

Based on the results of the conditional dependency measure derived from the mutual information formulae proposed by Chow and Liu [15], we found that the *Sholl Coefficient* and *2nd Moment* showed the strongest conditional dependency (0.731).

We investigated the effect on performance of adding hidden nodes between the different pairs of variables in the network. The places where each hidden node was added are indicated by the node numbers of Fig. 3.1.1. In our experiments we used two different propagation methods for the gradient descent (backwards and forwards (BF), and backwards and diagonals (BLR)). In all cases the performance was found

to improve, and though there was a trend to finding better improvement when placing hidden nodes between the higher correlated variables.

After investigating the single hidden node cases, we tried using a number of structures using two hidden nodes. These were placed at sites where the conditional dependency was high. Examples of the modified network structures are shown in Fig. 3.2.2. The best performance could be achieved by the introduction of two hidden nodes. The addition of two hidden nodes improved the performance to above 83.9% in contrast to the original 59.31%.



a) one hidden node b) two hidden nodes

Fig. 3.2.2. Examples of the structure variations of a naïve Bayesian network with up to two hidden node(s)

4 Network Accuracy

In addition to the prediction accuracy, the Euclidean, the Cosine and the Jensen-Shannon inaccuracy, along with the MDL score are determined for each of the Bayesian networks employed in the experiments. Of particular interest is the improvement in the network accuracy achieved due to the introduction of a hidden node.

Figure 4.1 illustrates the improvement in prediction accuracy (far left of each case) and the improvement in network accuracy, for five single hidden node cases. For example, case *126* represents the case when a hidden node is introduced between node index 1 and 2 given root node 6.

The experimental results demonstrate that the introduction of a hidden node consistently improves the network accuracy. This is due to the proper training of the hidden node, which results in a modified Bayesian network that does not violate the independence assumptions to such an extreme degree as the original Bayesian network.

Furthermore, the experimental results indicate the existence of a correlation between the improvement in network accuracy and the improvement in prediction accuracy. This seems to indicate that indeed the improvement in network accuracy contributes to the improvement in prediction accuracy.

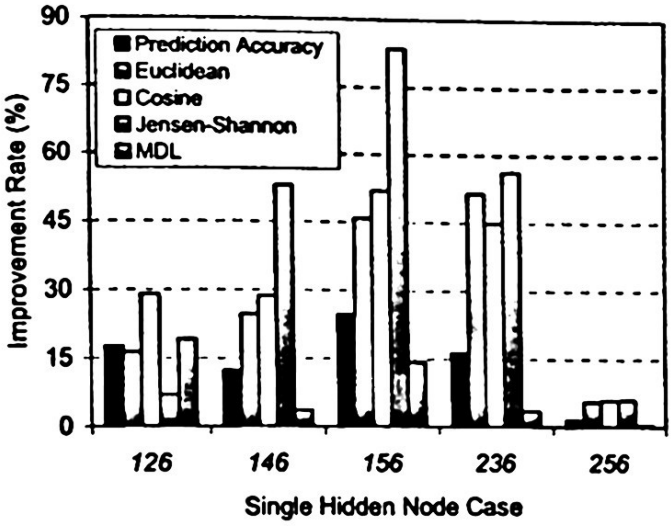


Figure 4.1 Comparison between improvement in prediction accuracy and improvement in network accuracy, in single hidden node cases.

5 Discussion and Conclusion

This study demonstrated that a computer vision application to successfully classify and count the neural cell automatically could be achieved with Bayesian networks with hidden nodes. The improvement in performance is due to the reduction of conditional dependence. Generally it was found that measuring the conditional dependence of two nodes given their parents provided an effective way of deciding where to place the hidden node. The data set that we used did contain a high degree of correlation between the variables allowing for potential improvement through the use of hidden nodes.

The methodology has the advantage of starting from a naïve structure where causal information is as simple as possible, and there is great potential for identifying variables that are related through a common cause or hidden variable. This allows great flexibility in identifying structural changes to the network. The methodology has two further advantages. Firstly the resulting classifier is always tree structured, and therefore fast and efficient to use in practice. Secondly, the performance is guaranteed to be equal or better than the original network, since the three new link matrices, associated with the hidden node, can encode all the information of the original link matrix joining the two children to the parent.

In addition, the experimental results demonstrate the improvement of network accuracy due to the introduction of a hidden node and its proper training. Furthermore, the experimental results indicate the existence of a correlation between the improvement in network accuracy and the improvement in prediction accuracy. Thus, we have provided an experimental justification to the empirically observed improvement in prediction accuracy when employing the hidden node methodology.

Acknowledgement

The authors would like to thank Duncan Gillies at Imperial College London and Peter Lucas at University of Nijmegen for their help and advice on this work.

References

1. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufman, San Mateo, California (1988)
2. Verma, T.S. and Pearl J.: Equivalence and Synthesis of Causal Models. Uncertainty in Artificial Intelligence, 6, Cambridge, MA, Elsevier Science Publishers, 220-22, 1991
3. Ackley, D.H., Hinton, G.E., and Sejnowski, T.J.: A learning algorithm for Boltzmann machines. Cognitive Science, 9;147-169, 1985
4. Friedman, N., Geiger, D., and Goldszmidt, M.: Bayesian network classifiers, Machine Learning, vol. 29, no. 2-3, 131-163, 1997
5. Kwoh C-K. and Gillies, D.: Using hidden nodes in Bayesian networks. Artificial Intelligence, 88:1-38, 1996
6. J-W Bang and D. Gillies. Estimating Hidden nodes in Bayesian Networks. *Proceeding of Int'l Conference on Machine Learning and Applications*, Las Vegas, USA, 2002.
7. J-W Bang and D. Gillies. Using Bayesian Networks with Hidden Nodes to Recognize Neural cell Morphology. *In Proceedings of the Seventh Pacific Rim Int'l Conference in Artificial Intelligence*, LNAI, Springer-Verlag, Tokyo, Japan, 2002.
8. Kim, J. and Gillies, D.: Automatic Morphometric analysis of neural cells. Machine Graph ics & Vision, Vol. 7, No. 4, 693-709, 1998
9. Sholl, D. A.: Dendritic Organization in the Neurons of the Visual and Motor cortices of the Cat. Journal of Anatomy, 87, 387-406, 1953
10. Flook, A. G.: The use of Dilation Logic on the Quantimet to Achieve Fractal Dimension Characterisation of Textured and Structured Profiles. Powder Technology, 21, 195-198, 1978
11. Wechsler, H.: Computational Vision. Academic Press Inc (1990)
12. Chow, C.K. and Liu, C.N.: Approximation discrete probability distributions with dependence trees. IEEE Trans. Inform. Theory 14:462-467, 1968