ADAPTIVE BAYESIAN NETWORKS FOR VIDEO PROCESSING

Benny P.L. Lo, Surapa Thiemjarus and Guang-Zhong Yang

Visual Information Processing Group,
Department of Computing,
Imperial College of Science, Technology and Medicine,
180 Queen's Gate, London SW7 2BZ, United Kingdom.
{benlo, st01, gzy}@doc.ic.ac.uk

ABSTRACT

Due to its static nature, the inference capability of Bayesian Networks (BNs) often deteriorates when the basis of input data varies, especially in video processing applications where the environment often changes constantly. This paper presents an adaptive BN where the network parameters are adjusted in accordance to input variations. An efficient re-training method is introduced for updating the parameters and the proposed network is applied to shadow removal in video sequence processing with quantitative results demonstrating the significance of adapting the network with environmental changes.

1. INTRODUCTION

Bayesian Networks (BNs) have been widely used as the inference mechanism in computer vision applications. The static network structure, however, has hindered its evolution and affected its inference ability for processing visual data under constantly varying environments. For video sequence processing, Dynamic Bayesian Networks (DBNs) are often used to incorporate temporal information into the network, as has been demonstrated by Pavlović *et al* for classifying object trajectories [1]. However, the structure and parameters of the DBNs are remained constant and unable to respond to environmental changes. Choudhury *et al* recently introduced a "boosting" algorithm for dynamically adapting the BN structure for improving its performance and ability to capture underlying dynamics of the data [2]. However, constant alteration of the network structure may result in a loss of the network causality and lead to network instability.

In our previous study, the Neuro-Fuzzy classifier demonstrates the value of adapting to changes in the environment with contextual information [3]. However, there are intrinsic difficulties in practical applications of the technique in that it involves an empirical process of choosing the fuzzy membership functions, and the framework is difficult to be extended for incorporating subjective decision rules. As such, an adaptive BN is proposed, where parameters in the network are continuously adjusted with respect to the domain specific rules applied. In this paper, we present an adaptive BN framework where the network parameters are adjusted in accordance to input variations. The strength of the proposed method is demonstrated by applying the result framework for shadow removal in video sequences processing, and detailed quantitative validations are provided.

2. ADAPTIVE BAYESIAN NETWORK

Bayesian network (BN) is often constructed by learning its structure and parameters from a training data set. However, it is unlikely in practice the training data could represent all the variations encountered, especially in video processing. As such, the performance of BN is often deteriorated when processing video sequences with constantly changing environments. In order to cope with these variations, the BN has to be re-trained in accordance to these changes. Among the two major components of BN, only the conditional probabilities between variables need to be re-trained, since the network structure, which corresponds to the dependency between variables, often remains constant in different environments.

2.1 Re-training the Bayesian Network

To maintain an efficient performance, the size of the re-training data has to be minimised. In addition, the re-training data should only reflect the significant misclassifications in order to adjust and improve the BN. If updating the conditional probabilities by drawing statistics only from the re-training data, the network will become unstable and erratic. Therefore, a revised backward propagation method, originally proposed by Kwoh *et al* for learning hidden node parameters, is introduced [4]. Unlike determining parameters for hidden nodes, the conditional probabilities between every concerned node in the network have to be updated during

0-7803-7750-8/03/$17.00 ©2003 IEEE

the re-training process. As such, the proposed method propagates the differences or errors backward from layer to layer in the BN, similar to the backward propagation in training multi-layered preceptrons. Upon receiving the propagated errors, a gradient descent method is applied to update the conditional probabilities of each node. To provide a detailed illustration of the proposed method, a multi-layered BN is used and shown in Figure 1.

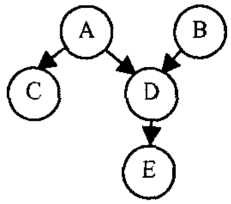

Figure 1 A sample multi-layered Bayesian Network

Least square estimation is applied to measure the difference between the posterior probabilities [$P'_n(a_x)$ and $P'_n(b_y)$] and the desired outputs [$d_n(a_x)$ and $d_n(b_y)$] as follows:

$$\begin{aligned}\xi(n) &= \sum_x^{|A|}\left(d_n(a_x) - P'_n(a_x)\right) + \sum_y^{|B|}\left(d_n(b_y) - P'_n(b_y)\right) \\ &= \sum_x^{|A|} e_n^2(a_x) + \sum_y^{|B|} e_n^2(b_y)\end{aligned} \tag{1}$$

where n indicates the n[th] data in the re-training set, and |A| and |B| represent the number of states of node A and B respectively. The average error of the re-training is formulated as follows:

$$\xi_{av} = \frac{1}{N}\sum_{n=1}^{N}\xi(n) = \frac{1}{N}\sum_{n=1}^{N}\left(\sum_x^{|A|} e_n^2(a_x) + \sum_y^{|B|} e_n^2(b_y)\right) \tag{2}$$

where N represents the size of the re-training set. To propagate the errors from the BN's output layer to other layers, two variables, delta(a_x,a_i) and delta(b_y,b_j), are defined as follows:

$$\begin{aligned}delta(a_x, a_i) &= \frac{\partial \xi(n)}{\partial e_n(a_x)}\frac{\partial e_n(a_x)}{\partial P'_n(a_x)}\frac{\partial P'_n(a_x)}{\partial \lambda(a_i)} \\ &= -2e_n(a_x)\alpha\pi(a_i)(\delta(x,i) - P'_n(a_x)) \\ delta(b_y, b_j) &= \frac{\partial \xi(n)}{\partial e_n(b_y)}\frac{\partial e_n(b_y)}{\partial P'_n(b_y)}\frac{\partial P'_n(b_y)}{\partial \lambda(b_y)} \\ &= -2e_n(b_y)\alpha\pi(b_j)(\delta(y,j) - P'_n(b_y))\end{aligned} \tag{3}$$

where α is the normalising constant of the posterior probabilities. $\lambda(a_x)$ and $\lambda(a_y)$ are the likelihood evidences and $\pi(a_x)$ and $\pi(a_y)$ are the prior evidences for state a_x and b_y respectively. In addition, another two variables are defined for propagating the errors from node D to its child node, as follows:

$$\begin{aligned}delta_D^a(a_i, b_j) &= \pi_D(b_j)\lambda_C(a_i) \\ delta_D^b(a_i, b_j) &= \pi_D(a_i)\end{aligned} \tag{4}$$

With respect to the error, the gradient for updating the conditional probabilities of node D is calculated as follows:

$$\begin{aligned}\frac{\partial \xi(n)}{\partial P(d_q \mid a_i \& b_j)} &= \frac{\partial}{\partial P(d_m \mid a_i \& b_j)}\left(\sum_x^{|A|} e_n^2(a_x) + \sum_y^{|B|} e_n^2(b_y)\right) \\ &= \sum_x^{|A|}\frac{\partial \xi(n)}{\partial e_n(a_x)}\frac{\partial e_n(a_x)}{\partial P'_n(a_x)}\frac{\partial P'_n(a_x)}{\partial \lambda(a_i)}\frac{\partial \lambda(a_i)}{\partial \lambda_D(a_i)}\frac{\partial \lambda_D(a_i)}{\partial P(d_q \mid a_i \& b_j)} + \\ &\quad \sum_y^{|B|}\frac{\partial \xi(n)}{\partial e_n(b_y)}\frac{\partial e_n(b_y)}{\partial P'_n(b_y)}\frac{\partial P'_n(b_y)}{\partial \lambda(b_j)}\frac{\partial \lambda(b_j)}{\partial \lambda_D(b_j)}\frac{\partial \lambda_D(b_j)}{\partial P(d_q \mid a_i \& b_j)} \\ &= \left(\sum_x^{|A|}\sum_y^{|B|}\begin{pmatrix} delta_n^a(a_x, b_y)delta(a_x, a_i) + \\ delta_n^b(a_x, b_y)delta(b_y, b_j)\end{pmatrix}\right)\lambda(d_q)\end{aligned} \tag{5}$$

whereas the gradient for updating the conditional probabilities for node C is

$$\begin{aligned}\frac{\partial \xi(n)}{\partial P(c_r \mid a_i)} &= \sum_x^{|A|} delta_C(a_i)\lambda(c_r)delta(a_x, a_i) \\ delta_C(a_x, a_i) &= delta(a_x, a_i)\lambda_D(a_i)\end{aligned} \tag{6}$$

In Equation (6), $\lambda_D(a_i)$ is the likelihood message sent by node D to node A. For node E, the gradient for updating the conditional probabilities is calculated by

$$\begin{aligned}\frac{\partial \xi(n)}{\partial P(E_s \mid d_q)} &= \sum_x^{|A|} U(x) + \sum_y^{|B|} V(x) \\ &= \left(\sum_x^{|A|}\sum_y^{|B|}\begin{pmatrix} delta_D^a(a_x, b_y)P(d_q \mid a_x \& b_y)delta(a_x, a_x) + \\ delta_D^b(a_x, b_y)P(d_q \mid a_x \& b_y)delta(b_y, b_y)\end{pmatrix}\right)\lambda(E_s)\end{aligned} \tag{7}$$

where

$$U(x) = \frac{\partial \xi(n)}{\partial e_n(a_x)}\frac{\partial e_n(a_x)}{\partial P'_n(a_x)}\frac{\partial P'_n(a_x)}{\partial \lambda(a_x)}\frac{\partial \lambda(a_x)}{\partial \lambda_D(a_x)}\frac{\partial \lambda_D(a_x)}{\partial \lambda(d_q)}\frac{\partial \lambda(d_q)}{\partial \lambda_E(d_q)}\frac{\partial \lambda_E(d_q)}{\partial P(E_s \mid d_q)}$$

$$V(x) = \frac{\partial \xi(n)}{\partial e_n(b_y)}\frac{\partial e_n(b_y)}{\partial P'_n(b_y)}\frac{\partial P'_n(b_y)}{\partial \lambda(b_y)}\frac{\partial \lambda(b_y)}{\partial \lambda_D(b_y)}\frac{\partial \lambda(b_y)}{\partial \lambda(d_q)}\frac{\partial \lambda(d_q)}{\partial \lambda_E(d_q)}\frac{\partial \lambda_E(d_q)}{\partial P(E_s \mid d_q)}$$

For these gradients, the conditional probabilities are updated by Equation (8).

$$\begin{aligned}P^{t+1}(d_q \mid a_i \& b_j) &= P^t(d_q \mid a_i \& b_j) + \Delta P(d_q \mid a_i \& b_j) \\ P^{t+1}(c_r \mid a_i) &= P^t(c_r \mid a_i) + \Delta P(c_r \mid a_i) \\ P^{t+1}(E_s \mid d_q) &= P^t(E_s \mid d_q) + \Delta P(E_s \mid d_q)\end{aligned} \tag{8}$$

where t represents the t[th] iteration in the re-training process,

$$\begin{aligned}\Delta P(d_q \mid a_i \& b_j) &= -\eta\frac{1}{N}\sum_n^N\left(\frac{\partial \xi(n)}{\partial P(d_q \mid a_i \& b_j)}\right) \\ \Delta P(c_r \mid a_i) &= -\eta\frac{1}{N}\sum_n^N\left(\frac{\partial \xi(n)}{\partial P(c_r \mid a_i)}\right) \\ \Delta P(E_s \mid d_q) &= -\eta\frac{1}{N}\sum_n^N\left(\frac{\partial \xi(n)}{\partial P(E_s \mid d_q)}\right)\end{aligned} \tag{9}$$

and η is the learning rate.

3. SHADOW FILTER

To evaluate the concept of adapting the BN to environmental changes, the proposed network is applied to a shadow removal application in which multiple low level visual cues are used to classify shadows and objects. Figure 2 illustrates the processes involved for shadow removal with the proposed adaptive BN.

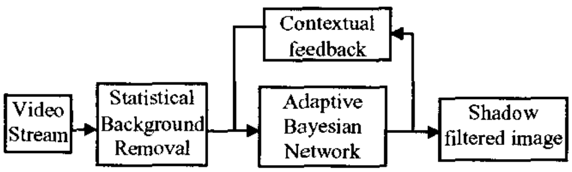

Figure 2. Schematic diagram of the adaptive BN shadow filter

As shown in the diagram above, background objects are first subtracted for highlighting the shadow and foreground objects. Similar to the Neuro-fuzzy shadow filter proposed in our previous study, the BN fuses four different shadow likelihood measurements (intensity difference, intensity attenuation, RGB angular difference and colour invariant model) to classify shadow and foreground objects [3]. By using the background image as a reference, the measurements are calculated through measuring the intensity and colour differences between a pixel and the background. In addition, the four contextual rules used for training the Neuro-fuzzy filter are employed to re-train the BN, and the rules are defined as follows:

1) *If the shadow likelihood measurements are all "high", the corresponding pixel is a shadow pixel.*
2) *If the shadow likelihood measurements are all "low", the corresponding pixel is an object pixel.*
3) *If a shadow pixel is surrounded mainly by object pixels and the likelihood measurements are not "high", the corresponding pixel should be re-classified as an object pixel instead.*
4) *If an object pixel is surrounded mainly by shadow pixels and the likelihood measurements are not "low", the corresponding pixel should be re-classified as a shadow pixel instead.*

The initial network structure and parameters of the BN are derived through supervised training with a pre-define training set. Once the structure is constructed, the BN is re-trained constantly according to the results from the four context rules described above.

4. RESULT

A video sequence recorded in an operating theatre was used for evaluating the performance of the proposed adaptive BN shadow filter. A sample image was arbitrarily picked from the sequence and manually classified as the initial training data. By learning the structure from the training data and inserting hidden nodes to group correlated variables, the BN is constructed as shown in Figure 3.

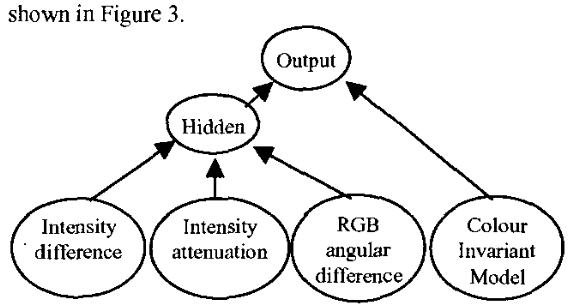

Figure 3. The resulting Bayesian network structure learned from the training set

Figure 4 shows the images before and after applying the shadow filter with the static (disabled the re-training) version and the adaptive version of the BN. To highlight the shadows, the resulting images are overlaid onto the original images.

As shown in the images, the shadow filter with a static BN only filtered out a small amount of shadow pixels. In contrast, with adaptive BN, most shadow pixels were effectively removed.

To quantitatively assess the performance of the adaptive BN, a randomly selected fifty image samples are used. Since the number of shadow pixels in an image frame is normally less than the number of non-shadow or object pixels, the percentage accuracy is not a good indicator of performance. As the output of the shadow filter is binary, true positive rate (TP) and false positive rate (FP) can be determined, the ROC Euclidian distance comparison (AC_d) is employed to measure and compare the performance of the static and adaptive BN [5].

As the resulting Euclidian distance ranges from 0 (perfect classification) to $\sqrt{2}$, to ease the illustration, the AC_d values are normalized as follows:

$$\textit{Normalised}\quad AC_d = 1 - \frac{1}{\sqrt{2}} + \sqrt{\frac{W \times (1-TP)^2 + (1-W) \times FP^2}{2}} \quad (10)$$

where W is a factor. The resulting distances, with W=1, is shown in Figure 5, where the larger the value of normalized AC_d, the better the performance of the classifier

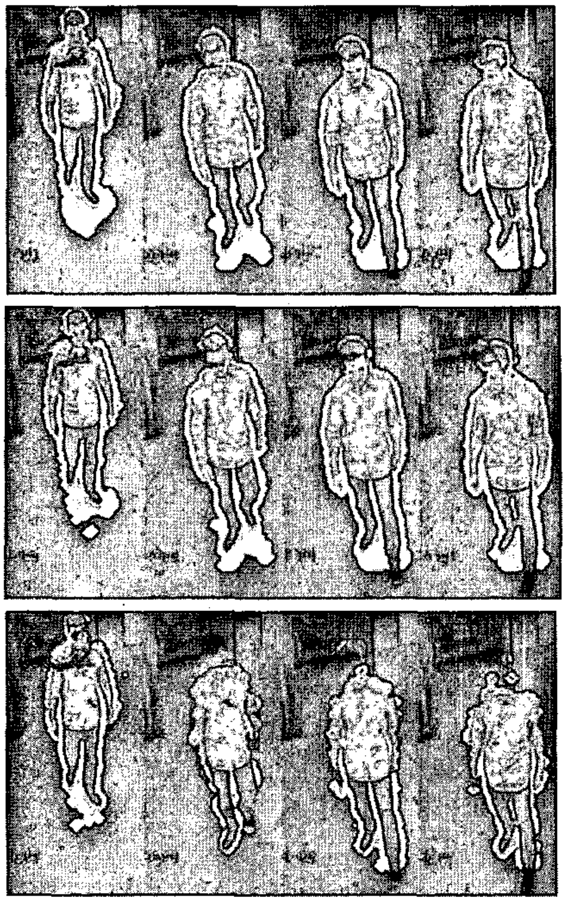

Figure 4. Images before applying the filter (top), after applying the filter with static BN (middle), and after applying the filter with adaptive BN (bottom)

Similar to the visual results shown in Figure 4, Figure 5 demonstrates that the adaptive BN outperform the static BN, which suggests the improvement in performance of the BN with adaptation to environment variations.

5. DISCUSSIONS AND CONCLUSIONS

In this paper, we have presented an adaptive BN framework which is able to evolve and adapt to environmental changes in video processing. Based on the backward propagation method, an effective technique has been introduced to propagate the errors or changes for updating the parameters in a multi-layered BN. The ability of the proposed framework for adapting to changes has been assessed with the shadow removal problem in video sequence processing. From the quantitative analysis and visual evaluation of the processing results, the proposed adaptive BN is shown to be able to remove shadows effectively, and its performance is significantly better that of static BNs.

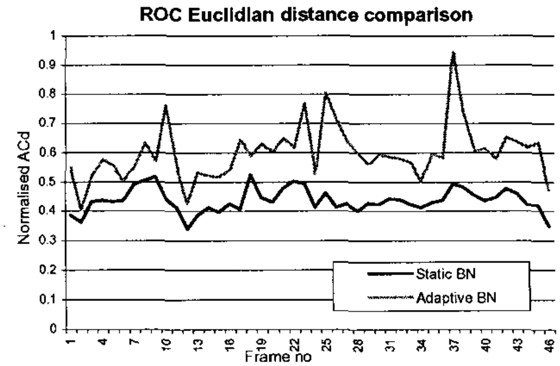

Figure 5. ROC Euclidian distance comparison where frame-by-frame normalized AC_d values are measured from the outputs of a static and an adaptive BN after applying to the same video sequence

6. REFERENCES

[1] V. Pavlović, B. Frey and T. Huang, "Time-Series Classification Using Mixed-State Dynamic Bayesian Networks", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 609-615, 1999.

[2] T. Choudhury, J. Rehg, V.Pavlovic and A.Pentland, "The Role of Boosting and Structure Leaning in Dynamic Bayesian Networks for Multi-modal Speaker Detection", *Proceedings of the ICPR*, 2002.

[3] Benny P.L. Lo and Guang-Zhong Yang, "Neuro-Fuzzy Shadow Filter", *European Conference on Computer Vision 2002*, Part III p. 381-392, May 2002.

[4] C.K. Kwoh, and D.F. Gillies, "Using Hidden Nodes in Bayesian Networks", *Artificial Intelligence*, vol. 88, pp. 1-38, 1996.

[5] http://www.cs.uregina.ca/~hamilton/courses/831/notes/ROC/ROC.html