To appear in: Proc. 1998 IEEE Signal Processing Society Workshop on Neural Networks for Signal Processing (Cambridge, UK)

Combining neural networks and belief networks for image segmentation

Christopher K. I. Williams*, Xiaojuan Feng*
Neural Computing Research Group
Division of Electronic Engineering and Computer Science
Aston University, Birmingham B4 7ET, UK
{c.k.i.williams,fengx}@aston.ac.uk

Abstract

In this paper we are concerned with segmenting an image into a number of predefined classes. We show how to fuse together local predictions for the class labels with a prior model of segmentations using the scaled-likelihood method. The prior model is based on a tree-structured belief network. Both the neural network and belief network were trained on a set of training images, and then the combined system was used to make predictions on a set of test images. We show that the combined neural network/belief network classifier gives improved prediction accuracy on 9 out of the 11 classes.

1 Introduction

Neural networks have been used very successfully in a wide variety of domains for performing classification or regression tasks. A characteristic of most currently successful applications is that the input patterns are either independent (as in static pattern classification) or related over time, rather than being spatially distributed. To extend the use of neural networks to spatially distributed tasks, such as the prediction of image segmentations, it is necessary to combine local bottom-up predictions (e.g. classifications on a pixel-by-pixel basis) with global prior knowledge (about what segmentations are likely). The use of this

* Address after July 1: Department of Artificial Intelligence, University of Edinburgh, 5 Forrest Hill, Edinburgh EH1 2QL. Email ckiw@dai.ed.ac.uk.

kind of prior knowledge would help, for example, when pixel-wise predictions are locally ambiguous, but the correct answer can be inferred using contextual information.

To carry out this project, we need to provide a principled method for the combination of the bottom-up predictions with the prior model. To do this we can use the scaled-likelihood method, which has been applied in [4] and [6] in the context of Hidden Markov models, to more general belief networks, as described in section 2.2.

We also need a prior model over segmentations. One popular model would be a Markov Random Field (MRF) model, where at each pixel the random variable would be the choice of label. However, there are some problems with MRF models, particularly that inference procedures are NP-hard. Also, it can be difficult to incorporate longer-range information into the prior if small neighbourhood sizes are used. For these reasons we will use a tree-structured belief network as our prior over images; in this case inference can be achieved in time linear in the number of pixels using Pearl's message-passing scheme [5]. By adapting the conditional probability tables in the belief network using a set of training images, we can tune our prior model to the kinds of images encountered, and thus improve prediction performance.

The structure of the rest of the paper is as follows: In section 2 we describe how the scaled-likelihood method can be used to combine spatially distributed predictions from neural networks, and how inference and learning can be carried out in tree-structured belief networks. In section 3 we give experimental details of training the neural networks and belief networks, and give results of applying the method to the Sowerby Image Database.

2 Theory

2.1 Generative model

Our model for the data is illustrated in Figure 1. The observed data $\boldsymbol{Y}$ is assumed to have been generated from an underlying process $\boldsymbol{X}$. $\boldsymbol{X}$ is a tree-structured belief network. At the highest level (level 0) there is one node X^0, which has children in level 1. Typically in our experiments each parent node has four children, giving rise to a quadtree-type architecture (as used by [2]). Each node is a multinomial variable, taking on one of C class labels. These labels are those used for the segmentation, e.g. road, sky, vehicle etc. The links between the nodes are defined by conditional probability tables (CPTs).

At the lowest level L of the tree, we find the leaf nodes denoted $\boldsymbol{X}^L$. The ith leaf node is denoted X_i^L. The leaf nodes correspond to small regions of the image (in our case 4×4 pixel regions). The model for the observation Y_i in each region is that it is generated according to $P(Y_i|\boldsymbol{X}) = P(Y_i|X_i^L)$, i.e. that Y_i depends only on the corresponding leaf node X_i^L. In addition we assume that the distribution $P(Y_i|X_i^L)$ is independent of i. The overall model for $P(\boldsymbol{X}, \boldsymbol{Y})$ is

$$P(\boldsymbol{X}, \boldsymbol{Y}) = P(\boldsymbol{X}) \prod_{i=1}^{K} P(Y_i|X_i^L).$$

2.2 Scaled-likelihood method

Above we have described a fully generative model for $\boldsymbol{Y}$, defined by CPTs from the root downwards. However, the database we are using provides both the raw data $\boldsymbol{Y}$ and labelled segmentations $\boldsymbol{X}^L$. Given this information it is natural to train classifiers (e.g. neural networks) to predict X_i^L given Y_i, and it is well known that trained neural networks approximate the posterior probabilities $P(X_i^L|Y_i)$. Fusion of these predictions into the belief network for $\boldsymbol{X}$ cannot then be achieved immediately, as this requires $P(Y_i|\boldsymbol{X}_i^L)$ and not $P(\boldsymbol{X}_i^L|Y_i)$ terms. However, by using Bayes' theorem we obtain

$$P(Y_i|X_i^L) = \frac{P(X_i^L|Y_i)P(Y_i)}{P(X_i^L)}$$

For inference on $\boldsymbol{X}$, the data $\boldsymbol{Y}$ is fixed and so the factors $P(Y_i)$ need not be considered further. Defining the *scaled likelihood* for a particular location i and denoting it as $L(Y_i) = P(X_i^L|Y_i)/P(X_i^L)$, we obtain

$$P(\boldsymbol{X}|\boldsymbol{Y}) \propto P(\boldsymbol{X}) \prod_{i=1}^{K} L(Y_i).$$

By replacing $L(Y_i)$ with $\hat{L}(Y_i) = \hat{P}(X_i^L|Y_i)/\hat{P}(X_i^L)$ (where $\hat{P}$ denotes an estimated probability)[1] we obtain a principled method for the fusion of the local predictions $\hat{P}(X_i^L|Y_i)$, $i = 1, \ldots, K$ with the global prior model $P(\boldsymbol{X})$.

One potential advantage of using the scaled-likelihood method is that the generative model for $P(Y_i|X_i)$ may be quite complex, although the predictive distribution $P(X_i|Y_i)$ is actually quite simple,

[1]Here we take $\hat{P}(X_i)$ to be the overall frequency of each class in a set of images, averaged over all pixel locations.

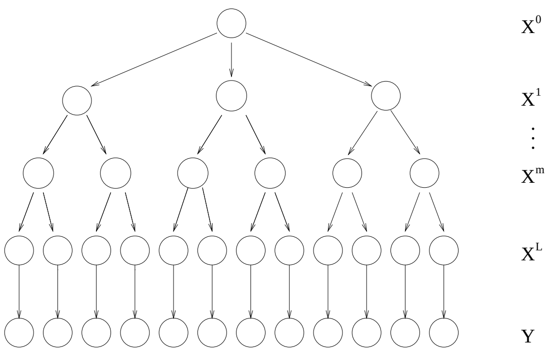

Figure 1: A 1-D graphical model illustrating a small tree-structured belief network. The network nodes are partitioned into three categories: $\boldsymbol{Y}$ denotes the raw image information; $\boldsymbol{X}^L$ denotes the leaf nodes of $\boldsymbol{X}$ which are observed during training. The nodes in the layers above labelled $X^0, \ldots, X^m$ are always unobserved.

i.e. the generative approach may spend a lot of resources on modelling $P(Y_i|X_i)$ which are not particularly relevant to the task of inferring $\boldsymbol{X}$.

2.3 Inference (Segmentation)

Given a new image $\boldsymbol{Y} = \boldsymbol{y}$ we wish to carry out inference on $\boldsymbol{X}^L$, given the probabilistic model. Computing the posterior $P(\boldsymbol{X}^L = \boldsymbol{x}^L|\boldsymbol{Y} = \boldsymbol{y})$ would be highly expensive, as it would require enumerating all possible C^K states in $\boldsymbol{X}^L$. There are two alternatives that are computationally feasible, (i) the computation of the posterior marginals $P(X_i^L = x_i^L|\boldsymbol{Y} = \boldsymbol{y})$ and (ii) the overall most likely interpretation of the data $\boldsymbol{x}^* = \text{argmax}_{\boldsymbol{x}} P(\boldsymbol{X} = \boldsymbol{x}|\boldsymbol{Y} = \boldsymbol{y})$. These can be achieved by Pearl's λ-π message passing schemes, as described in [5].

2.4 Training the belief network

Above it was assumed that the CPTs (denoted $\boldsymbol{\theta}$) used to define $P(\boldsymbol{X})$ are known. In fact we estimated these from training data, using a maximum likelihood procedure. Let x_{ik}, $k = 1, \ldots, C$ denote the the possible values of X_i, and let pa_{ij}, $j = 1, \ldots, C$ denote the set of possible values taken on by Pa_i, the parent of X_i. The parameter θ_{ijk} denotes the CPT entry $P(X_i = x_{ik}|Pa_i = pa_{ij})$.

We can obtain a maximum likelihood estimate of $\boldsymbol{\theta}$ by using the EM algorithm (this is a generalization of the EM algorithm for Hidden Markov Models to tree-structured networks). Let the segmented training images be denoted by $\boldsymbol{x}^{Lm}$, $m = 1, \ldots, M$. Then the update for each CPT is given by

$$\tilde{\theta}_{ijk} = \frac{\sum_{m=1}^{M} P(X_i = x_{ik}, Pa_i = pa_{ij} | \boldsymbol{x}^{Lm}, \boldsymbol{\theta})}{\sum_{m=1}^{M} \sum_{k'} P(X_i = x_i^{k'}, Pa_i = pa_{ij} | \boldsymbol{x}^{Lm}, \boldsymbol{\theta})},$$

where the joint probability $P(X_i = x_{ik}, Pa_i = pa_{ij} | \boldsymbol{x}^{Lm}, \boldsymbol{\theta})$ can be obtained locally using the λ-value of node X_i, the π-value of the parent node Pa_i and the λ-message from the siblings of node X_i.

This update gives a separate update for each link in the tree. Given limited training data this is undesirable. If the set of variables sharing a CPT is denoted as $\boldsymbol{X}_I$, then

$$\tilde{\theta}_{Ijk} = \frac{\sum_{m=1}^{M} \sum_{X_i \in \boldsymbol{X}_I} P(X_i = x_{ik}, Pa_i = pa_{ij} | \boldsymbol{x}^{Lm}, \boldsymbol{\theta})}{\sum_{m=1}^{M} \sum_{X_i \in \boldsymbol{X}_I} \sum_{k'} P(X_i = x_{ik'}, Pa_i = pa_{ij} | \boldsymbol{x}^{Lm}, \boldsymbol{\theta})}.$$

In our experiments all of the CPTs in each level were constrained to be equal, except for the transition from layer 0 to layer 1, where each table was separate. This allows knowledge about the broad nature of scenes (e.g. sky occurs near to the top of images) to be learned by the network.

Below we will also need to evaluate the *marginal likelihood* of an image given parameters $\boldsymbol{\theta}$. This can be calculated efficiently at the root node of the tree, where $P(\boldsymbol{x}^L) = \sum_{k=1}^{C} \lambda(x_k^0)\pi(x_k^0)$.

3 Experimental results

The performance of our models have been tested by using the colour images of out-door scenes from the Sowerby Image Database of British Aerospace. There are eleven categories to be classified, see Table 1. The full-resolution colour images of size 512 by 768 pixels were subsampled into 128 by 192 regions of size 4 by 4 pixels. (From now on we will often refer to these 4 by 4 regions as pixels; they are pixels in a subsampled image.)

Eighteen summary statistics including the grey-level intensity and colour hue, location and texture features, e.g. entropy, contrast and local homogeneity of the gray-level different vectors (GLDV), were calculated for each region. These features were fed to a Multi-Layer Perceptron (MLP) with 18 input nodes, 11 output nodes and one hidden layer which was used to classify each region into one of the 11 classes. The activation functions of output nodes and hidden nodes were a softmax function and the tanh sigmoid function respectively. The error function used in training process was cross-entropy for multiple classes. A scaled conjugate gradient algorithm was used to minimize the error function. The original 104 images were divided randomly into independent training, validation and testing sets of size 61, 20 and 23 respectively. About 100 regions for each class were chosen randomly from each image to form training and validation datasets. The validation dataset was used in order to choose the optimal number of hidden nodes in the MLP; eventually a MLP with 20 hidden nodes was selected.

In the training set of the MLP, we tried to use equal numbers of regions from each class. Thus these frequencies (denoted $\tilde{p}(\omega_k)$, $k = 1, \ldots, C$) are not equal to the overall prior probability of each class. This was done deliberately, to aid the detection of low-frequency classes. We must *compensate* for this bias by multiplying the network outputs by $p(\omega_k)/\tilde{p}(\omega_k)$ and renormalizing (see [1], §6.5).

The belief network structure that was used was basically a quadtree, except that there were six children of the root node (reflecting the aspect ratio of the images). In the learning phase, we initialized the network parameters $\boldsymbol{\theta}$ in a number of different ways. It was found that the highest marginal likelihood on the training data was obtained when the initial values of $\boldsymbol{\theta}$ were computed using probabilities derived from downsampled versions of the images. The plot of marginal likelihood against iteration number levelled off after ten iterations. In the data some pixels are unlabelled; assuming these values are "missing at random", we treated them as uninstantiated nodes, which can easily be handled in a belief network framework.

The learned quadtree model is a much better model for the images than the independent-pixel model. The coding cost an image $\boldsymbol{x}^{Lm}$ under a model $P_M(\cdot)$ is $\frac{1}{K} \log_2 P_M(\boldsymbol{x}^{Lm})$ bits per pixel (bpp)[2]. The average of this cost over the images in the validation and testing sets (which were not used to train the belief network) is 2.15 bpp under an independent-pixel model, compared to 0.33 bpp for the quadtree model. Using an independent-pixel model with coding probabilities derived from the marginals of the belief network gives 1.74 bpp, showing that the correlations between pixels modelled by the tree are of considerable importance.

We now turn to the segmentation of the validation and testing

[2]The minimum attainable coding cost occurs when $P_M(\cdot)$ is the true generator of the images. Hence lower coding costs indicate a better model.

images. The "raw" segmented image obtained by simply choosing the most likely class at each pixel will typically have a salt-and-pepper appearance due to local misclassifications. Our belief network method is one way to smooth the raw classifications. Majority filtering [3] is a commonly used method with the same objective, in which one simply chooses the most common class within a window. For comparison with the belief net method we used majority filtering with a window size 7×7 on the subsampled image; a size of 7 was found to give the highest overall classification accuracy. The performance of the methods on a test image is shown in Figure 2[3].

The overall classification accuracies for the validation and testing images were 80.27%, 81.73% and 81.56% for the MLP, majority filter and belief network methods, respectively. However, these raw figures do not give the whole story; further information can be derived by studying the confusion matrices (one for each method). Individual class accuracies obtained from the diagonals of the normalised confusion matrices are shown in in Table 1.

Class	Overall fraction	MLP (compensated)	Majority filter	Bayesian network
Clouds	0.1208	0.9467	0.9408	0.9676
Vegetation	0.4001	0.8459	0.8760	0.8599
Road Marking	0.0017	0.6192	0.3719	0.7734
Road Surface	0.3475	0.9537	0.9707	0.9352
Road Border	0.0473	0.0586	0.0407	0.0492
Building	0.0371	0.3648	0.3432	0.4633
Bounding object	0.0317	0.0078	0.0013	0.1563
Road sign	0.0021	0.0014	0.0005	0.0582
Telegraph Pole	0.0033	0.0508	0.0201	0.1641
Illumination	0.0029	0.0257	0.0118	0.1272
Shadow				
Mobile Object	0.0055	0.1271	0.0897	0.5402

Table 1: Column 2 shows the overall fraction of each class in the validation and testing images. Columns 3, 4 and 5 show the class accuracies for the MLP, majority filter and belief net methods respectively.

[3]These images are better seen in colour; to do this, obtain a copy of this paper from `ftp://cs.aston.ac.uk/neural/willicki/ieee98.ps.gz`.

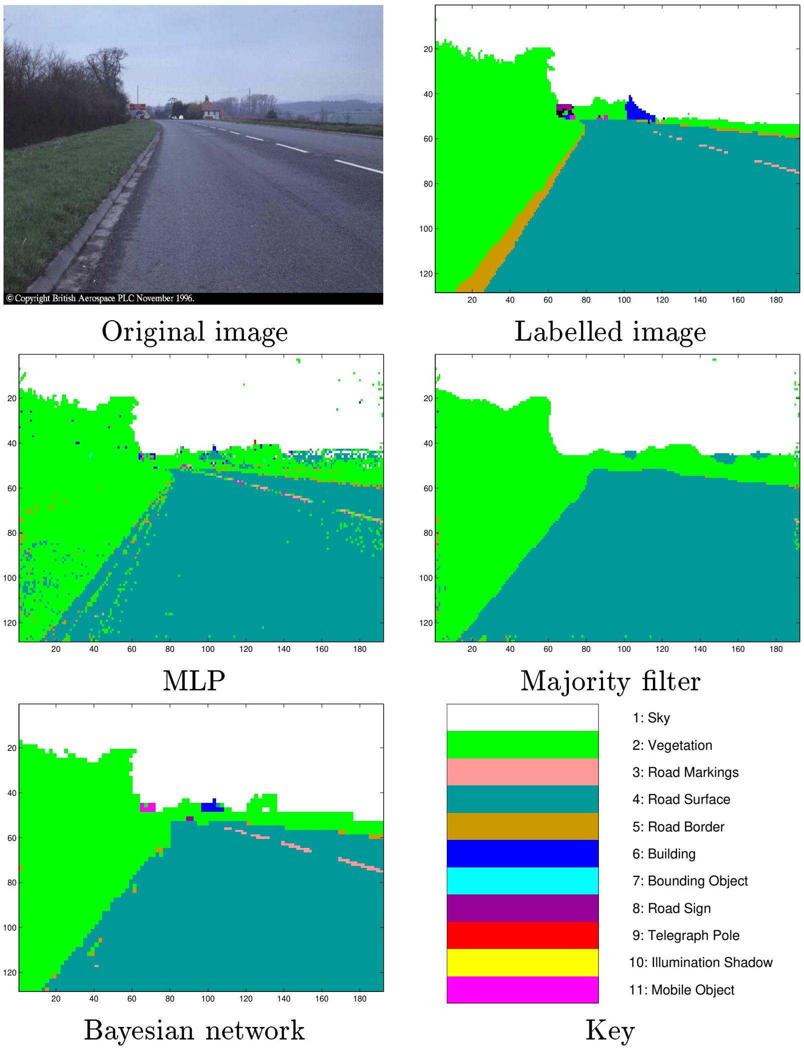

Figure 2: Demonstrating the various methods on a test image. Labelled image denotes the segmentation provided with the database.

Table 1 shows that the belief network method gives better accuracy than either the raw MLP or majority filter on nine out of the eleven classes, and is better than either of the alternatives on eight out of eleven classes. A comparison of confusion matrices is not the whole story; we really need to compare the quality of segmentations derived from the various methods. We note that this is a difficult task which may depend on the uses the segmentation will be put to.

3.1 Conclusions

In this paper we have shown how to combine a prior model for segmentations with neural network predictions for class labels using the "scaled-likelihood" method, and that this method yields improved prediction accuracy for nine out of the eleven classes in our images. We have also shown that for image coding the tree-structured belief network is considerably superior to the independent-pixel model.

There are many further directions to explore, including the effect that adding in information from neighbouring regions will have on the MLP classifier, and how this will interact with the belief network.

Acknowledgements

This work is funded by EPSRC grant GR/L03088, *Combining Spatially Distributed Predictions From Neural Networks.* The authors gratefully acknowledge the assistance of British Aerospace in the project and in making the Sowerby Image Database available to us. We also thank Dr. Andy Wright of BAe for helpful discussions, Dr. Ian Nabney (Aston) for comments on the paper and NETLAB routines for neural networks, and Dr. John Elgy for introducing us to the work of [2].

References

[1] C. M. Bishop. *Neural Networks for Pattern Recognition.* Clarendon Press, Oxford, 1995.

[2] C. A. Bouman and M. Shapiro. A Multiscale Random Field Model for Bayesian Image Segmentation. *IEEE Transactions on Image Processing*, 3(2):162–177, 1994.

[3] Lillesand, T. M. and Kiefer, R. W. *Remote Sensing and Image Interpretation.* Wiley, New York, third edition, 1994.

[4] N. Morgan and H. A. Bourlard. Neural Networks for Statistical Recognition of Continuous Speech. *Proceedings of the IEEE*, 83(5):742–770, 1995.

[5] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, San Mateo, CA, 1988.

[6] P. Smyth. Hidden Markov models for fault detection in dynamic systems. *Pattern Recogntion*, 27(1):149–164, 1994.