To appear in: Proceedings of SPIE vol 3457. Presented at SPIE's 43rd Annual Meeting, San Diego, CA, July 19-24 1998.

Training Bayesian networks for image segmentation

Xiaojuan Feng Christopher K. I. Williams
Neural Computing Research Group
Dept. of Electronic Engineering and Computer Science
Aston University, Birmingham B4 7ET

ABSTRACT

We are concerned with the problem of image segmentation in which each pixel is assigned to one of a predefined finite number of classes. In Bayesian image analysis, this requires fusing together local predictions for the class labels with a prior model of segmentations. Markov Random Fields (MRFs) have been used to incorporate some of this prior knowledge, but this not entirely satisfactory as inference in MRFs is NP-hard. The multiscale quadtree model of Bouman and Shapiro (1994) is an attractive alternative, as this is a tree-structured belief network in which inference can be carried out in linear time (Pearl 1988). It is an hierarchical model where the bottom-level nodes are pixels, and higher levels correspond to downsampled versions of the image.

The conditional-probability tables (CPTs) in the belief network encode the knowledge of how the levels interact. In this paper we discuss two methods of learning the CPTs given training data, using (a) maximum likelihood and the EM algorithm and (b) *conditional* maximum likelihood (CML). Segmentations obtained using networks trained by CML show a statistically-significant improvement in performance on synthetic images. We also demonstrate the methods on a real-world outdoor-scene segmentation task.

Keywords: belief networks, segmentation, EM, CML, learning

1. INTRODUCTION

We are concerned with the problem of image segmentation in which each pixel is assigned to one of a predefined finite number of classes. In Bayesian image analysis, this requires fusing together local predictions for the class labels with a prior model of segmentations.

Recently, much work has been directed toward stochastic model-based techniques. In such techniques, the image classes are modelled as random fields and the segmentation problem is posed as a statistical optimization problem. A Markov Random Field (MRF) model, where at each pixel the random variable would be the choice of label, is the most influential statistical model of this kind (see, e.g. Mardia *et al.*[1]). 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 a small neighbourhood size are used.

One alternative to MRFs is to use tree-structured belief network (TSBN) models of images[2,3] . In recent work[4,5] we have used TSBNs as prior models in image segmentation. For TSBNs inference can be carried out in time linear in the number of pixels using Pearl's message-passing scheme[6] . Williams and Feng[4] showed on a particular problem that classification accuracy was improved on 9 out of 11 classes by using the trained TSBN in image segmentation. Williams and Feng[5] also showed that a learned TSBN model is a better model of test images (as judged by average log likelihood) than models based on independent blocks of varying sizes.

An important disadvantage of TSBN models is that pixels that are spatially adjacent may not have common parents. Therefore, the models do not enforce continuity of regions, and this leads to the well-known "blocky" artifacts in segmentation results. New models have been studied to avoid the drawbacks. For example, Bouman and Shapiro[2] used a cross-connected pyramidal graphical model in which the number of parents (coarse scale neighbour for each pixel) has been increased. The disadvantage of the cross-connected graph structure is that it contains cycles, leading to inference computations that are exponential in the size of the network.

Rather than using more complex models, our aim in this paper has been to improve the performance of TSBNs by training them explicitly for the purpose of image segmentation. This has led to us using conditional maximum likelihood (CML) estimation of the parameters in the TSBN, rather than maximum likelihood (ML) estimation.

Send correspondence to Dr. Christopher K. I. Williams. Address after July 1: Department of Artificial Intelligence, University of Edinburgh, 5 Forrest Hill, Edinburgh EH1 2QL. E-mail ckiw@dai.ed.ac.uk, fengx@aston.ac.uk

The method of parameter estimation that has been used with TSBNs in our previous papers[4,5] is maximum likelihood estimation (MLE). The MLE is by far and away the most common method of parameter estimation in pattern recognition. There are many very important properties of the MLE, but most of them based on an implicit assumption of model correctness. The objective in MLE is to do as good as a job as possible of deriving the true model parameters. However, the observed distribution of visual images is complex and the training data is indeed limited. Currently, we do not know of a correct model, but we can be almost certain that TSBNs are not totally correct. Thus, the justification for MLEs is based on premises which are simply not valid in our case. In certain statistical problems, such as estimation of hidden Markov model (HMM) parameters for speech recognition, it was found empirically that estimation of parameters via some other criteria that used conditional likelihood (see, e.g. Krogh's paper[7]) and/or mutual information (see e.g. Bahl *et al.*[8] and Brown[9]) can give better results than estimation via maximum likelihood. We will describe and compare the maximum likelihood and the conditional maximum likelihood approaches in TSBN training for image segmentation.

The data we have available consists of both colour (rgb pixel) images and also label images, where the label indicates classes such as "sky", "road" etc. In ML training, we simply adjust the parameters in the TSBN to give high likelihood to the correct label images in the training set. On the other hand, for CML estimation, on each image we again try to increase the likelihood to the correct label image, but also try to decrease the likelihood for every incorrect label image by using the information from the rgb image.

The remainder of this paper is organised as follows: In Section 2 we describe the basic TSBN architecture, and how the inference can be carried out. In Section 3 we described in details of the training of TSBNs by using ML and CML methods. In section 4 we give experimental details and results of applying the trained TSBNs to image segmentation.

2. MODELLING

2.1. Generative model

The basis of our segmentation approach is a hierarchical model as illustrated in Figure 1a. A 1-D model illustrating a small (four level) TSBN is shown in Figure 1b.

The observed data $\boldsymbol{Y}$ (e.g. the rgb values of the pixels) 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 Bouman and Shapiro[2] . Each X-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). The critical property of TSBNs is the conditional independencies which makes the computation more efficient.

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.

2.2. 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 $\boldsymbol{x}^* = \mathrm{argmax}_{\boldsymbol{x}} \mathrm{P}(\boldsymbol{X} = \boldsymbol{x}|\boldsymbol{Y} = \boldsymbol{y})$. These can be achieved by Pearl's message passing schemes[6] . These computations require a generative model for $P(Y_i|X_i^L)$.

An alternative to using a model for $P(Y_i|X_i^L)$ is to make use of predictions giving $\hat{P}(X_i^L|Y_i)$, as may be obtained from a neural network or some other classifier. As $P(Y_i|X_i^L) = P(X_i^L|Y_i)P(Y_i)/P(X_i^L)$ and $P(Y_i)$ is fixed when performing inference, we can define the *scaled likelihood* for location i as $L(Y_i) = P(X_i^L|Y_i)/P(X_i^L)$. To make use of our predictor we replace $L(Y_i)$ with $\hat{L}(Y_i) = \hat{P}(X_i^L|Y_i)/\hat{P}(X_i^L)$, where $\hat{P}$ denotes an estimated probability. $\hat{P}(X_i^L)$ is obtained from the overall frequency of each class in a set of training images. This method of combining neural networks with belief networks has been suggested (for HMMs) in Smyth[10] and Morgan and Bourlard[11] . A potential

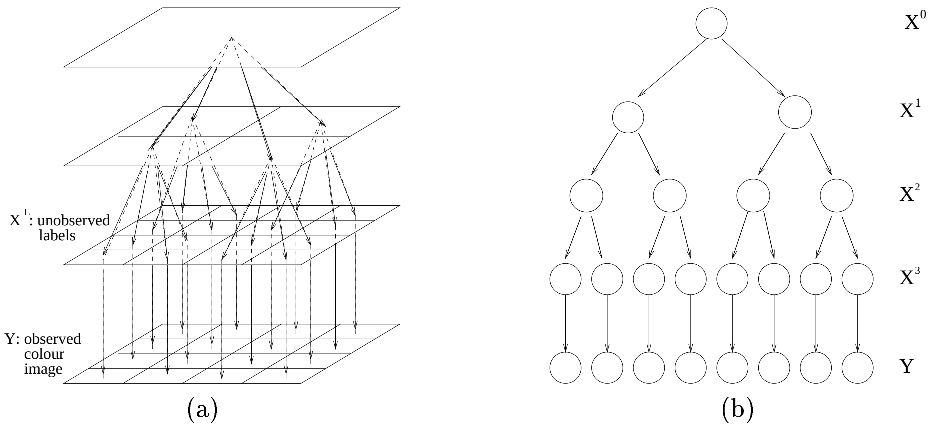


Figure 1. (a) Illustration of the three-level quadtree model. (b)A 1-D graphical model illustrating a small tree-structured belief network. The network nodes are partitioned into three categories: $\boldsymbol{Y}$ denotes the raw image information; $\boldsymbol{X}^L$ denotes the leaf nodes of $\boldsymbol{X}$ which are observed during training. The nodes in the layers above labelled $X^0, \ldots, X^2$ are always unobserved.

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, 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 $\boldsymbol{X}$.

3. TRAINING A TREE-STRUCTURED 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. Let x_{il}, $l = 1, \ldots, C$ denote the the possible values of X_i, and let pa_{ik}, $k = 1, \ldots, C$ denote the set of possible values taken on by Pa_i, the parent of X_i. The parameter θ_{ikl} denotes the CPT entry $P(X_i = x_{il}|Pa_i = pa_{ik})$. For simplicity the symbols X_i and Pa_i are dropped, and the probability is written as $P(x_{il}|pa_{ik})$.

For training the prior model it is assumed that a number of observation images $\boldsymbol{y}^m$ and associated labelled images $\boldsymbol{x}^{Lm}$ are available, where $m = 1, \cdots, M$ is the index to the images in the training set. We discuss in turn maximum likelihood training (§3.1) and conditional maximum likelihood training (§3.2).

3.1. Maximum likelihood

In maximum likelihood a parameter vector, $\boldsymbol{\theta}$, is estimated so that

$$
\begin{aligned}
\hat{\boldsymbol{\theta}}^{ML} &= \arg\max_{\boldsymbol{\theta}} \prod_{m=1}^{M} P(\boldsymbol{x}^{Lm}, \boldsymbol{y}^m|\boldsymbol{\theta}) \\
&= \arg\max_{\boldsymbol{\theta}} \prod_{m=1}^{M} P(\boldsymbol{y}^m|\boldsymbol{x}^{Lm}, \boldsymbol{\theta}) P(\boldsymbol{x}^{Lm}|\boldsymbol{\theta}).
\end{aligned}
$$

We can see that the likelihood model parameters and the prior model parameters can be estimated separately by choosing the likelihood model parameters to maximise the $P(\boldsymbol{y}^m|\boldsymbol{x}^{Lm}, \boldsymbol{\theta})$ and the prior model parameters to maximise $P(\boldsymbol{x}^{Lm}|\boldsymbol{\theta})$. Assuming that the likelihood model is fixed, we obtain

$$
\hat{\boldsymbol{\theta}}^{ML} = \arg\max_{\boldsymbol{\theta}} \prod_{m=1}^{M} P(\boldsymbol{x}^{Lm}|\boldsymbol{\theta}).
$$

Hence the maximum likelihood estimator $\hat{\boldsymbol{\theta}}^{ML}$ can be obtained by maximising the likelihood of $\prod_m P(\boldsymbol{x}^{Lm}|\boldsymbol{\theta})$ only. The Baum-Welch algorithm is well known in maximising the likelihood function in an HMM. The generalisation of the Baum-Welch algorithm for HMM to the TSBN was used to maximise $P(\boldsymbol{x}^{Lm}|\boldsymbol{\theta})$. This algorithm is a special case of EM that uses the bottom-up and top-down message passing to infer the posterior probabilities of the hidden nodes in the E-steps and uses the expected counts of the transitions to reestimate the CPT[12] . The re-estimation formulas can be derived directly by maximising Baum's auxiliary function,

$$Q(\bar{\boldsymbol{\theta}}, \boldsymbol{\theta}) = \sum_{m=1}^{M} \sum_{\boldsymbol{x}_h} P(\boldsymbol{x}_h|\boldsymbol{x}^{Lm}, \boldsymbol{\theta}) \ \log [\ P(\boldsymbol{x}_h, \boldsymbol{x}^{Lm}|\bar{\boldsymbol{\theta}})\ ], \tag{1}$$

over $\bar{\boldsymbol{\theta}}$ (the new estimated parameter vector). The maximisation problem in (1) is a constrained optimisation problem because our solutions must be legal assignment of the CPT entries in the network. Then the update for each entry in CPTs is given by

$$\bar{\theta}_{ikl} = \frac{\sum_{m=1}^{M} P(x_{il}, pa_{ik}|\boldsymbol{x}^{Lm}, \boldsymbol{\theta})}{\sum_{m=1}^{M} \sum_{k'} P(x_{il}, pa_{ik'}|\boldsymbol{x}^{Lm}, \boldsymbol{\theta})}.$$

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

$$P(x_{il}, pa_{ik}|\boldsymbol{x}^{Lm}, \boldsymbol{\theta}) = \frac{1}{\sum_{k'} \pi(pa_{ik'})\lambda(pa_{ik'})} \ \lambda(x_{il}|\boldsymbol{\theta}) \ \theta_{ikl} \ \pi(pa_{ik}|\boldsymbol{\theta}) \prod_{y \in s(X_i)} \lambda_y(pa_{ik}|\boldsymbol{\theta}),$$

where $s(X_i)$ is the set of nodes that are siblings of node X_i and $\lambda_y(.)$ is the λ-message sent to node Pa_i by node y.

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 the EM parameter update is given by

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

3.2. Conditional maximum likelihood

In the CML procedure, the objective is to predict correctly the labels $\boldsymbol{x}^L$ associated with "virtual" evidence $\boldsymbol{y}$. The parameters are then estimated by maximising the probability of the correct labelling given the evidence $\boldsymbol{y}$.

$$\begin{aligned} \hat{\boldsymbol{\theta}}^{CML} &= \arg\max_{\boldsymbol{\theta}} \prod_{m=1}^{M} P(\boldsymbol{x}^{Lm}|\boldsymbol{y}^m, \boldsymbol{\theta}) \\ &= \arg\max_{\boldsymbol{\theta}} \prod_{m=1}^{M} \frac{P(\boldsymbol{x}^{Lm}, \boldsymbol{y}^m|\boldsymbol{\theta})}{P(\boldsymbol{y}^m|\boldsymbol{\theta})}. \end{aligned} \tag{2}$$

By analogy to the Boltzmann machine, we observe that computing the conditional probability requires computation of (1) the probability $P(\boldsymbol{x}^{Lm}, \boldsymbol{y}^m|\boldsymbol{\theta})$ in the *clamped phase* (i.e. with $\boldsymbol{x}^{Lm}$ and $\boldsymbol{y}^m$ fixed), and (2) the probability $p(\boldsymbol{y}^m|\boldsymbol{\theta})$ in the *free-running* phase (with only $\boldsymbol{y}^m$ fixed).

To carry out the optimisation in (2) we switch to logarithms and define

$$\begin{aligned} L_f(\boldsymbol{\theta}) &= \sum_{m=1}^{M} \log \ P(\boldsymbol{y}^m|\boldsymbol{\theta}), \\ L_c(\boldsymbol{\theta}) &= \sum_{m=1}^{M} \log \ P(\boldsymbol{x}^{Lm}, \boldsymbol{y}^m|\boldsymbol{\theta}). \end{aligned}$$

Since we have assumed the likelihood model is fixed, $L_c(\boldsymbol{\theta})$ can be further simplified as

$$L_c(\boldsymbol{\theta}) = \sum_{m=1}^{M} \log[\ P(\boldsymbol{y}^m|\boldsymbol{x}^{Lm})P(\boldsymbol{x}^{Lm}|\boldsymbol{\theta})\ ] = \sum_{m=1}^{M} \log P(\boldsymbol{y}^m|\boldsymbol{x}^{Lm}) + \sum_{m=1}^{M} \log P(\boldsymbol{x}^{Lm}|\boldsymbol{\theta}).$$

Here we have used the subscripts c and f to mean "clamped" and "free". Then, to find $\hat{\boldsymbol{\theta}}^{CML}$ in (2) we need to maximise

$$L(\boldsymbol{\theta}) = \sum_{m=1}^{M} \log \frac{P(\boldsymbol{x}^{Lm}, \boldsymbol{y}^m|\boldsymbol{\theta})}{P(\boldsymbol{y}^m|\boldsymbol{\theta})} = L_c(\boldsymbol{\theta}) - L_f(\boldsymbol{\theta}). \tag{3}$$

Unfortunately the EM algorithm is not applicable to the CML estimation, because the CML criterion is expressed as a rational function[13] . However, maximisation of equation (3) can be carried out in various ways based on the gradient of L. In speech analysis[7,14] , methods based on gradient ascent have been used. The scaled conjugate gradient optimisation algorithm[15,16] was used in our work. Firstly, we need to calculate the gradient of L w.r.t. $\boldsymbol{\theta}$.

The probability $P(\boldsymbol{y}^m|\boldsymbol{\theta})$ can be written as a sum over all nodes in a TSBN, $P(\boldsymbol{y}^m|\boldsymbol{\theta}) = \sum_{\boldsymbol{x}} P(\boldsymbol{x}|\boldsymbol{\theta})P(\boldsymbol{y}^m|\boldsymbol{x}, \boldsymbol{\theta})$, where the sum is over all possible values of $\boldsymbol{x}$. Using the conditional independence relations, $P(\boldsymbol{x}|\boldsymbol{\theta})$ is easily decomposed into a product of the transition probabilities on all links.

Following the ideas in Krogh[7] for HMMs, the derivative of $L_f(\boldsymbol{\theta})$ w.r.t θ_{ikl} is

$$\begin{aligned}
\frac{\partial L_f(\boldsymbol{\theta})}{\partial \theta_{ikl}} &= \sum_{m=1}^{M} \frac{1}{P(\boldsymbol{y}^m|\boldsymbol{\theta})} \frac{\partial P(\boldsymbol{y}^m|\boldsymbol{\theta})}{\partial \theta_{ikl}} \\
&= \sum_{m=1}^{M} \sum_{\boldsymbol{x}} \frac{1}{P(\boldsymbol{y}^m|\boldsymbol{\theta})} \frac{\partial P(\boldsymbol{y}^m, \boldsymbol{x}|\boldsymbol{\theta})}{\partial \theta_{ikl}} \\
&= \sum_{m=1}^{M} \sum_{\boldsymbol{x}} \frac{1}{P(\boldsymbol{y}^m|\boldsymbol{\theta})} \frac{P(\boldsymbol{y}^m, \boldsymbol{x}|\boldsymbol{\theta})}{\theta_{ikl}} \delta(x_i, l)\delta(pa_i, k) \\
&= \sum_{m=1}^{M} \sum_{\boldsymbol{x}} \frac{P(\boldsymbol{x}|\boldsymbol{y}^m, \boldsymbol{\theta})}{\theta_{ikl}} \delta(x_i, l)\delta(pa_i, k) \\
&= \sum_{m=1}^{M} \frac{P(x_{il}^m, pa_{ik}^m|\boldsymbol{y}^m, \boldsymbol{\theta})}{\theta_{ikl}}.
\end{aligned}$$

Let $n_{ikl}^m = P(x_{il}^m, pa_{ik}^m|\boldsymbol{y}^m, \boldsymbol{\theta})$, then

$$\frac{\partial L_f(\boldsymbol{\theta})}{\partial \theta_{ikl}} = \frac{\sum_{m=1}^{M} n_{ikl}^m}{\theta_{ikl}}. \tag{4}$$

The derivative of the other term, $L_c(\boldsymbol{\theta})$, can be calculated in a similar manner. We have,

$$\begin{aligned}
\frac{\partial L_c(\boldsymbol{\theta})}{\partial \theta_{ikl}} &= \sum_{m=1}^{M} \frac{1}{P(\boldsymbol{x}^{Lm}|\boldsymbol{\theta})} \frac{\partial P(\boldsymbol{x}^{Lm}|\boldsymbol{\theta})}{\partial \theta_{ikl}} \\
&= \sum_{m=1}^{M} \sum_{\boldsymbol{x}_h} \frac{1}{P(\boldsymbol{x}^{Lm}|\boldsymbol{\theta})} \frac{\partial P(\boldsymbol{x}_h, \boldsymbol{x}^{Lm}|\boldsymbol{\theta})}{\partial \theta_{ikl}} \\
&= \sum_{m=1}^{M} \sum_{\boldsymbol{x}_h} \frac{1}{P(\boldsymbol{x}^{Lm}|\boldsymbol{\theta})} \frac{P(\boldsymbol{x}_h, \boldsymbol{x}^{Lm}|\boldsymbol{\theta})}{\theta_{ikl}} \delta(x_i, l)\delta(pa_i, k) \\
&= \sum_{m=1}^{M} \sum_{\boldsymbol{x}} \frac{P(\boldsymbol{x}_h|\boldsymbol{x}^{Lm}, \boldsymbol{\theta})}{\theta_{ikl}} \delta(x_i, l)\delta(pa_i, k) \\
&= \sum_{m=1}^{M} \frac{P(x_{il}^m, pa_{ik}^m|\boldsymbol{x}^{Lm}, \boldsymbol{\theta})}{\theta_{ikl}},
\end{aligned}$$

where we have let $\boldsymbol{x}_h = \boldsymbol{x} \backslash \boldsymbol{x}^L$ denote the "hidden" $\boldsymbol{x}$ variables. Let $m_{ikl}^m = P(x_{il}^m, pa_{ik}^m|\boldsymbol{x}^{Lm}, \boldsymbol{\theta})$, then

$$\frac{\partial L_c(\boldsymbol{\theta})}{\partial \theta_{ikl}} = \frac{\sum_{m=1}^{M} m_{ikl}^m}{\theta_{ikl}}. \tag{5}$$

Finally the derivative of the total log likelihood $L(\boldsymbol{\theta})$ is obtained by using equations (4) and (5),

$$\frac{\partial L}{\partial \theta_{ikl}} = \frac{\sum_{m=1}^{M}(m_{ikl}^{m} - n_{ikl}^{m})}{\theta_{ikl}},$$

where m_{ikl} and n_{ikl} can be obtained by propagating $\boldsymbol{y}^m$ and $\boldsymbol{x}^{Lm}$ respectively.

When maximising $L(\boldsymbol{\theta})$ it must be ensured that the probability parameters remain positive and properly normalised. The softmax function is used to meet these constraints. We define

$$\theta_{ikl} = \frac{e^{z_{ikl}}}{\sum_{l'} e^{z_{ikl'}}},$$

where z_{ikl} are the new unconstrained auxiliary variables and θ_{ikl} always sum to one by construction. The gradients w.r.t. z_{ikl} can be expressed entirely in terms of θ_{ikl} and m_{ikl}^{n} and n_{ikl}^{n},

$$\frac{\partial L(\boldsymbol{\theta})}{\partial z_{ikl}} = -\sum_{n=1}^{N}\left[m_{ikl}^{n} - n_{ikl}^{n} - \theta_{ikl}\sum_{l'}(m_{ikl'}^{n} - n_{ik'}^{n})\right].$$

On iteration τ $\boldsymbol{z}$ is updated by $\boldsymbol{z}^{(\tau+1)} = \boldsymbol{z}^{(\tau)} + \Delta\boldsymbol{z}^{(\tau)}$. This yields a change in θ_{ikl} given by

$$\theta_{ikl}^{(\tau+1)} = \frac{\theta_{ikl}^{(\tau)}\exp(\Delta z_{ikl}^{(\tau)})}{\sum_{l'}\theta_{ikl'}^{(\tau)}\exp(\Delta z_{ikl'}^{(\tau)})}.$$

To understand the differences between ML and CML estimation, we consider equation (3) in more detail. The first term on the right is equivalent to finding the MLE of $\boldsymbol{\theta}$; the difference between MLE and CMLE is the second term. To have an insight into the effect of this term, let us first sum over all possible label images $\tilde{\boldsymbol{x}}$ and then factorize the joint probability. This gives

$$P(\boldsymbol{y}^m|\boldsymbol{\theta}) = \sum_{\tilde{\boldsymbol{x}}^{Lm}} P(\tilde{\boldsymbol{x}}^{Lm}, \boldsymbol{y}^m|\boldsymbol{\theta}) = \sum_{\tilde{\boldsymbol{x}}^{Lm}} P(\boldsymbol{y}^m|\tilde{\boldsymbol{x}}^{Lm})P(\tilde{\boldsymbol{x}}^{Lm}|\boldsymbol{\theta}).$$

Let $\phi(\boldsymbol{x}^{Lm}|\boldsymbol{\theta}) = \partial P(\boldsymbol{x}^{Lm}|\boldsymbol{\theta})/\partial\theta_{ikl}$. Following Brown[9] , we consider the derivative of $L(\boldsymbol{\theta})$ w.r.t. θ_{ikl},

$$\begin{aligned}
\frac{\partial L(\boldsymbol{\theta})}{\partial \theta_{ikl}} &= \sum_{m=1}^{M}\left[\frac{\phi(\boldsymbol{x}^{Lm}|\boldsymbol{\theta})}{P(\boldsymbol{x}^{Lm}|\boldsymbol{\theta})} - \sum_{\tilde{\boldsymbol{x}}^{Lm}}\frac{\phi(\tilde{\boldsymbol{x}}^{Lm}|\boldsymbol{\theta})P(\boldsymbol{y}^m|\tilde{\boldsymbol{x}}^{Lm})}{P(\boldsymbol{y}^m|\boldsymbol{\theta})}\right] \\
&= \sum_{m=1}^{M}\phi(\boldsymbol{x}^{Lm}|\boldsymbol{\theta})\left[\frac{1}{P(\boldsymbol{x}^{Lm})} - \frac{P(\boldsymbol{y}^m|\boldsymbol{x}^{Lm})}{P(\boldsymbol{y}^m|\boldsymbol{\theta})}\right] - \sum_{m=1}^{M}\sum_{\tilde{\boldsymbol{x}}^{Lm}\neq\boldsymbol{x}^{Lm}}\frac{\phi(\tilde{\boldsymbol{x}}^{Lm}|\boldsymbol{\theta})P(\boldsymbol{y}^m|\tilde{\boldsymbol{x}}^{Lm})}{P(\boldsymbol{y}^m|\boldsymbol{\theta})}.
\end{aligned} \tag{6}$$

$\phi(\boldsymbol{x}^{Lm}|\boldsymbol{\theta})$ is the the derivative of the objective function used in ML estimation. From $P(\boldsymbol{x}^{Lm}, \boldsymbol{y}^m|\boldsymbol{\theta}) < P(\boldsymbol{y}^m|\boldsymbol{\theta})$, it is easy to show that $1/P(\boldsymbol{x}^{Lm}) > P(\boldsymbol{y}^m|\boldsymbol{x}^{Lm})/P(\boldsymbol{y}^m|\boldsymbol{\theta})$. Thus, the first term in (6) is in the same direction as the MLE derivative. The second term subtracts a component in the direction of $\phi(\tilde{\boldsymbol{x}}^{Lm}|\boldsymbol{\theta})$ for each incorrect label image $\tilde{\boldsymbol{x}}^{Lm} \neq \boldsymbol{x}^{Lm}$.

4. EXPERIMENTS

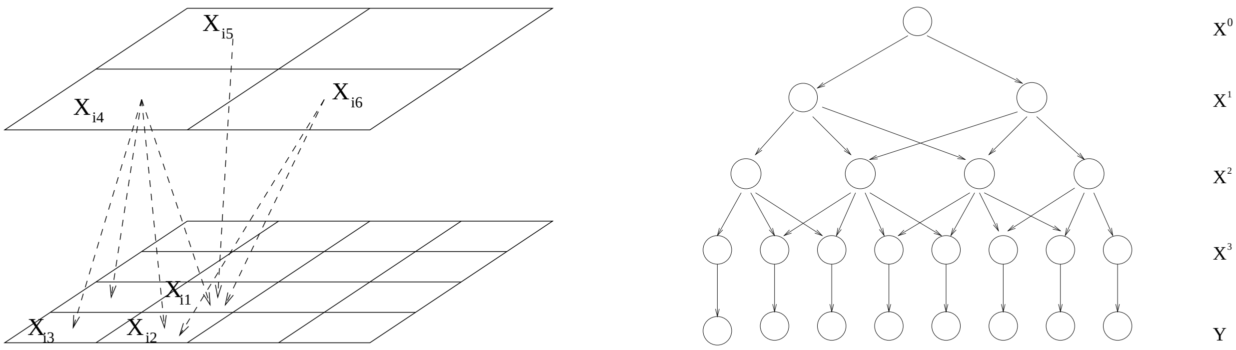
In this section, we describe the performance of TSBNs trained by using the ML and CML algorithms on synthetic images and real-world outdoor images.

4.1. Synthetic images

The synthetic label images are generated from a cross-connected pyramidal graphical model[2] . The cross connection between two levels in a pyramidal graphical model is shown in Figure 2a. Figure 2b illustrates a 1-D analogue of a four-level pyramidal graphical model. At each level, each inner node has three parents, for example, node X_{i1} has three parents, natural parent X_{i4}, column-parent X_{i5} and row-parent X_{i6}; each edge node has two parents, for example, node X_{i2} has parents X_{i4} and X_{i6}; and each corner node has only one parent, for example, node X_{i3} has only its natural parent X_{i4}. Corresponding to the difference in the number of parents, three different conditional probability functions are designed. Let $P(m|i,j,k)$ be the conditional probability for child node to be in class m given that its natural parent is in class i and the other two parents are in classes j and k respectively; $P(m|i,j)$ be the conditional probability for the child node to be in class m given that its natural parent is in class i and the other parent is in class j; $P(m|i)$ be the conditional probability for the child node to take on class m given its only parent is in class i. We define

$$\begin{aligned}
P(m|i,j,k) &= \frac{\theta_l}{7}(3\delta_{m,i} + 2\delta_{m,j} + 2\delta_{m,k}) + \frac{1-\theta_l}{C}, \\
P(m|i,j) &= \frac{\theta_l}{7}(4.5\delta_{m,i} + 2.5\delta_{m,j}) + \frac{1-\theta_l}{C}, \\
P(m|i) &= \theta_l\delta_{m,i} + \frac{1-\theta_l}{C},
\end{aligned}$$

where the parameter $\theta_l = \theta^{(L-l+1)}$ for $1 \le l \le L$ (with the root node at level 0) determines the probability that the label of the child node will be the same as one of its parents. Note that the natural parent has a stronger influence than the other parents. In our experiments, θ is set to be 0.85. As the size of the generated images is 16×16 pixels, there are 5 levels in the each belief network so $L = 4$. Parameter C denotes the number of classes. We have used three classes, denoted the Red class, Green class and Blue class. The prior over the classes at the root was $(0.7, 0.2, 0.1)$ for classes R, G, B respectively.



(a) (b)

Figure 2. (a) Illustration of cross connection between two levels in graphical model. (b)A 1-D analogue of three-level pyramidal graph-structured model. 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.

Colour images were generated from the label images by adding Gaussian noise to the mean rgb value for each class. The means were $\boldsymbol{\mu}_R = (150, 0, 0)^T$, $\boldsymbol{\mu}_G = (0, 150, 0)^T$ and $\boldsymbol{\mu}_B = (0, 0, 150)^T$ for Red class, Green class and Blue class respectively. All three classes had the same covariance matrix, a diagonal matrix with standard deviation 75 on each dimension.

To investigate the effect of training set size, training sets of size 10, 10^2, 10^3 and 10^4 were generated randomly from the cross-connected model. To help investigate the effects of training-set variability, three training sets were generated for each size (except for size 10^4). A test set of size 10^3 was also generated. Three label images and associated colour noise images are shown in Figure 3.

As we know the generative model for producing pixel values from labels, it is possible to invert this to make predictions using just the local rgb values. The local prediction for each pixel was obtained from the values of likelihood function, and the class with largest value was chosen. The local prediction accuracies for the test set were 86.57%, 86.62%, and 86.51% for class R, G, and B respectively and the overall local prediction accuracy was 86.57%.

After generating the label and colour images from a cross-connected graphical model, we modeled the training label images by using a standard five-level quadtree belief network as shown in Figure 1. We used parameter sharing, with one distinct CPT per level in the TSBN. To reduce the effect of initial values for parameters $\boldsymbol{\theta}$ on training results, the TSBNs were initialized at ten randomly-chosen starting points. For each training set/initialization combination, a TSBN was trained by using both ML and CML algorithms. Thus 30 trained TSBNs were obtained for training sets of size 10, 10^2 and 10^3, and 10 for training sets of size 10^4. For ML training, the EM updates were terminated when the difference between the marginal log likelihood (averaged over the number of training samples) on successive steps was less than $\epsilon = 10^{-20}$. For CML training, a similar criterion was applied on the log conditional likelihood, and training was also terminated if the absolute difference between the values of the CPTs between two successive steps was less than $\epsilon = 10^{-20}$. The ML/EM algorithm took 52 iterations to convergence on average, compared to 182 for CML/SCG. Note also that each iteration of CML/SCG is more expensive, because (i) it requires both $P(\boldsymbol{x}^{Lm}|\boldsymbol{\theta})$ and $P(\boldsymbol{y}^m|\boldsymbol{\theta})$ and thus needs two bottom-up/top-down sweeps and (ii) the scaled conjugate gradient algorithm uses two function evaluations per iteration.

Each trained TSBN was used in the image segmentation task, by calculating the *maximum a posteriori* configuration for $\boldsymbol{x}$ given an image $\boldsymbol{y}$. Examples are shown of the segmentations obtained on three test images in Figure 3. The average percentage of each class label that was correctly classified and the overall average classification accuracies for the test set are shown in Table 1, along with the standard deviations of these figures due to the randomness induced by training set and initialization variability. Notice that for each training set size, the CML result for the overall average is always superior to the ML result. This also holds for the R and G results individually, although for training set sizes of 10 and 100, the performance on class B is slightly worse for the CML method. To test the statistical significance, a paired comparison between the overall classification accuracies was used. The differences between the EM and CML learning methods were computed on the same training set/initialization parings. Using a two-tailed t-test we found that the differences were statistically significant at better than the 0.01 level for all sizes of training set.

Measuring pixelwise classification accuracy is not the only way to evaluate the quality of the segmentation obtained from the belief network. For example this measure says nothing about over- or under-segmentation of the result obtained compared to the reference label image. Nor does it take into account any loss function in evaluating mis-classifications. Also, note that the TSBN does not simply produce a single segmentation, but a probability distribution $P(\boldsymbol{x}^{Lm}|\boldsymbol{y}^m)$ over labelling $\boldsymbol{x}^{Lm}$. Thus we can investigate distribution further; for example, calculating the entropy of the posterior marginal distribution for each pixel would indicate how uncertain the classification decision is at each site.

Table 1. Average percentage of each class that was correctly classified and the overall average percentage classification accuracy for the test set.

	ML				CML			
	Class label				Class label			
	R	G	B	Average	R	G	B	Average
10	92.71	88.71	85.89	89.82±0.056	94.55	90.18	85.53	90.99±0.186
10^2	92.66	88.50	86.70	89.94±0.021	94.51	90.81	86.36	91.37±0.037
10^3	92.79	88.39	86.66	89.95±0.007	94.64	90.72	86.66	91.48±0.004
10^4	92.82	88.39	86.65	89.98±0.001	94.65	90.98	86.68	91.60±0.003

4.2. Real images

Colour images of out-door scenes from the Sowerby Image Database of British Aerospace are used in our experiments. Both colour images and their corresponding label images are provided in this database. The original 104 images were divided randomly into independent training and test sets of size 61 and 43 respectively. There are 7 different labels in all, namely "sky", "vegetation", "road markings", "road surface", "building", "street furniture" and "mobile object".

The original label images of size 512 by 768 pixels were subsampled into 128 by 192 regions to form the reduced label images. The label of the reduced region was chosen by majority vote, with ties being resolved by an ordering on the label categories. From now on we will refer to the reduced label images as label images because the original label images will no longer be used. Twenty-one features including colour features[17] , location and texture features, e.g. entropy, contrast and local homogeneity of the gray-level different vectors (GLDV)[18,19] , were calculated for each region. These features were fed to a Multi-Layer Perceptron (MLP) with 21 input nodes, 7 output nodes and one hidden layer which was used to classify each region into one of the 7 classes. The activation functions of the output nodes and hidden nodes were the softmax function and tanh sigmoid functions respectively. The error function used in the training process was cross-entropy for multiple classes. A scaled conjugate gradient algorithm was used to minimise the error function. About 150 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 30 hidden nodes was selected. The neural network predictions were input as virtual evidence to the belief network using the scaled-likelihood method described in Section 2.2.

The belief network structure used was basically a quadtree, except that there were six children of the root node (reflecting the aspect ratio of the images). 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, as is indeed reflected in the learned CPTs. 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.

In the learning phase, we initialised 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 log marginal likelihood against iteration number levelled off after 30 iterations when the EM method was used in obtaining the MLE. CML training was run for 32 iterations using scaled conjugate gradient optimization.

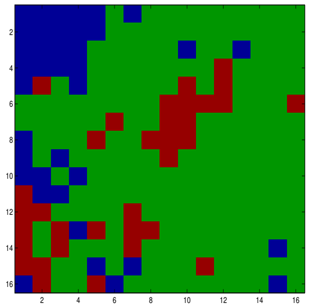
The overall classification accuracies for the testing images were 83.38%, 87.13% and 91.64% for the MLP, the TSBN trained by the ML algorithm and TSBN trained by the CML algorithm respectively.

5. CONCLUSIONS

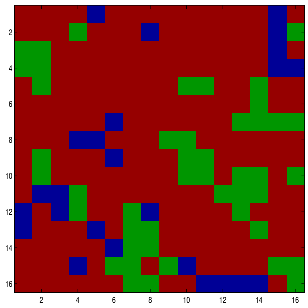
In this paper we have investigated the training of tree-structured belief networks for the image segmentation task. Our results show that superior classification performance can be obtained using conditional maximum likelihood training as compared to maximum likelihood training. However, we note that classification accuracy is just one measure of comparison between segmentations, and one strength of probabilistic formulations of the problem (including the belief network method) is that a posterior distribution over segmentations is returned, rather than just a single label image. One disadvantage of CML training is that it typically requires more training time as gradient-based search methods must be used instead of the EM algorithm.

Acknowledgements

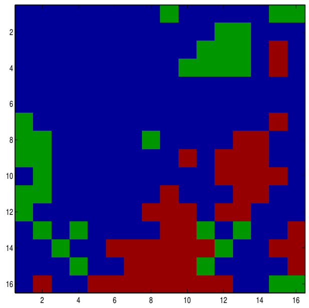
This work is funded by EPSRC grant GR/L03088, *Combining Spatially Distributed Predictions From Neural Networks*. The authors gratefully acknowledge the assistance of British Aerospace (and particularly Dr. Andy Wright) in the project and in making the Sowerby Image Database available to us. They also thank the Isaac Newton Institute (Cambridge, UK) for its hospitality and excellent working environment during the "Neural Networks and Machine Learning" programme, 1997.



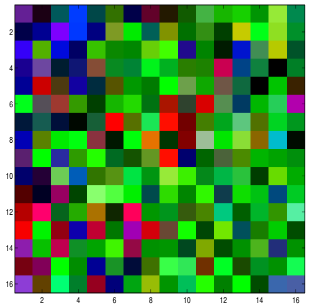
Label image 1



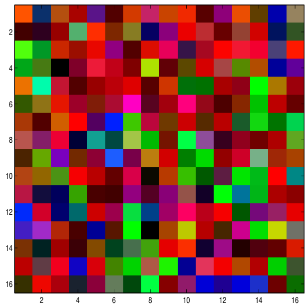
Label image 2



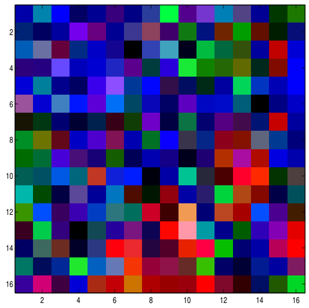
Label image 3



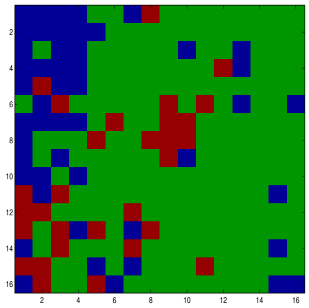
Colour image 1



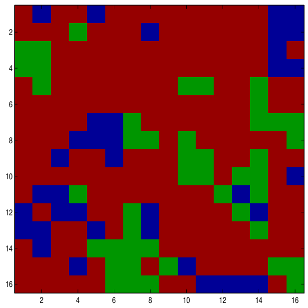
Colour image 2



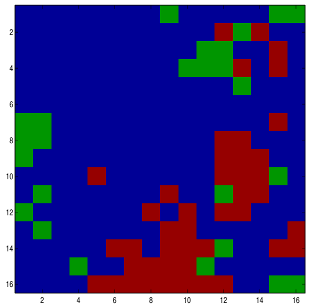
Colour image 3



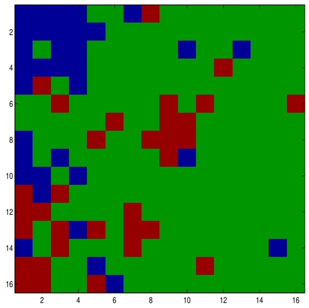
Classified label image 1
by using EM Trained TSBN



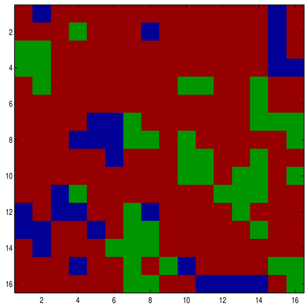
Classified label image 2
by using EM Trained TSBN



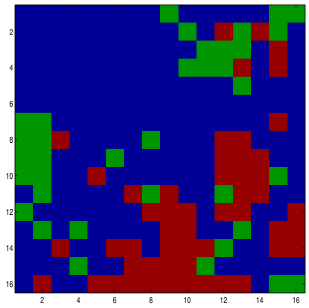
Classified label image 3
by using EM Trained TSBN



Classified label image 1
by using CML Trained TSBN



Classified label image 2
by using CML Trained TSBN



Classified label image 3
by using CML Trained TSBN

Figure 3. Three synthetic test label images, associated colour images and classification results by using EM and CML trained TSBNs..

REFERENCES

1. K. V. Mardia, A. J. Baczkowski, Xiaojuan Feng, and T. J. Hainsworth. "Statistical methods for automatic interpretation of digitally scanned finger prints". *Pattern Recognition Letters*, **18**:pp.1197–1203, 1997.
2. C. A. Bouman and M. Shapiro. "A multiscale random field model for bayesian image segmentation". *IEEE Transactions on Image Processing*, **3**:pp.162–177, 1994.
3. M. R. Luettgen and A. S. Willsky. "Likelihood calculation for a class of multiscale stochastic models, with application to texture discrimination". *IEEE Transactions on Image Processing*, **4**(2):pp.194–207, 1995.
4. C. K. I. Williams and X. Feng. "Combining neural networks and belief networks for image segmentation". In *Proc. 1998 IEEE Signal Processing Society Workshop on Neural Networks for Signal Processing. Cambridge, UK*, 1998.
5. C. K. I. Williams and X. Feng. "Tree-structured belief networks as models of images", 1998. Submitted to the 13th Conference of Neural Information Processing Systems (NIPS98).
6. J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, California, 1988.
7. A. Krogh. "Hidden markov models for labeled sequences". In *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, pages pp.140–144. IEEE Computer Society Press, 1994.
8. L. R. Bahl, P. F. Brown, P. V. de Souza, and R. L. Mercer. "Maximum mutual information estimation of hidden markov model parameters for speech recognition". In *Proceedings of IEEE International Conference of Acoustic Speech Signal Processing*, pages pp.49–52, 1986.
9. P. F. Brown. *PhD Thesis: The Acoustic-Modeling Problem in Automatic Speech Recognition.* IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598, 1987.
10. P. Smyth. "Hidden markov models for fault detection in dynamic systems". *Pattern Recognition*, **27**(1):pp.149–164, 1994.
11. N. Morgan and H. A. Bourlard. "Neural networks for statistical recognition of continuous speech". In *Proceedings of the IEEE*, volume **83**(5), pages pp.742–770, 1995.
12. S. L. Lauritzen. *Graphical Models.* Oxford University Press, 1996.
13. P. S. Gopalakrishnan, D. Kanevsky, A. Nadas, and D. Nahamoo. "An inequality for rational functions with applications to some statistical estimation problems". *IEEE Transactions on Information Theory*, **37**(1):pp.107–113, 1991.
14. S. K. Riis and A. Krogh. "Hidden neural networks: a framework for HMM/NN hybrids". In *Proceedings ICASSP-97, April 21-24, Munich, Germany*, 1997.
15. M. F. Møller. "A scaled conjugate gradient algorithm for fast supervised learning". *Neural Networks*, **6**:pp.525–533, 1993.
16. C. M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, Oxford, 1995.
17. Yuichi Ohta. *Knowledge-based Interpretation of Outdoor Natural Colour Scenes.* Pitman Publishing Limited, London, 1985.
18. J. S. Weszka, C. R. Dyer, and A. Rosenfeld. "A comparative study of texture measures for terrain classification". *IEEE Transactions on Systems, Man, and Cybernetics*, **SMC-6**(4):pp.269–285, 1976.
19. R. M. Welch, M. S. Navar, and S. K. Sengupta. "The effect of spatial resolution upon texture-based cloud field classifications". *Journal of Geophysical Research*, **94**:pp.14767–14781, 1989.