Dynamic Trees: Learning to Model Outdoor Scenes

Nicholas J. Adams Christopher K. I. Williams

Institute for Adaptive and Neural Computation
University of Edinburgh, 5 Forrest Hill, Edinburgh EH1 2QL, UK
http://www.anc.ed.ac.uk/
nicka@dai.ed.ac.uk *c.k.i.williams@ed.ac.uk*

Abstract. This paper considers the dynamic tree (DT) model, first introduced in [1]. A dynamic tree specifies a prior over structures of trees, each of which is a forest of one or more tree-structured belief networks (TSBN). In the literature standard tree-structured belief network models have been found to produce "blocky" segmentations when naturally occurring boundaries within an image did not coincide with those of the subtrees in the fixed structure of the network. Dynamic trees have a flexible architecture which allows the structure to vary to create configurations where the subtree and image boundaries align, and experimentation with the model has shown significant improvements.
Here we derive an EM-style update based upon mean field inference for learning the parameters of the dynamic tree model and apply it to a database of images of outdoor scenes where all of its parameters are learned. DTs are seen to offer significant improvement in performance over the fixed-architecture TSBN and in a coding comparison the DT achieves 0.294 bits per pixel (bpp) compression compared to 0.378 bpp for lossless JPEG on images of 7 colours.

1 Introduction

Dynamic Trees (DTs) are a generalisation of the fixed-architecture tree structured belief networks (TSBNs). Belief networks can be used in image analysis by grouping its nodes into visible units $\boldsymbol{X}_v$, to which the data is applied, and having hidden units $\boldsymbol{X}_h$, connected by some suitable configuration, to conduct a consistent inference. TSBNs have been shown in [2, 3] to be attractive for image segmentation because they are hierarchically structured, so provide a multi-scale interpretation of the image. Willsky and coworkers have also developed TSBNs with real-valued Gaussian variables for a variety of tasks including texture discrimination [4]. Unlike other standard techniques – such as Markov Random Fields – TSBNs also have efficient inference algorithms [5].

The fixed architecture of balanced[1] TSBNs makes their segmentation prone to *blockiness*. It can be argued that the problem with TSBNs is not the tree-structure *per se*, but the fact that it is fixed in advance. This suggests that the

[1] that is having a fully regular structure, such as a quad-tree.

model should provide a distribution over tree structures, reminiscent of parse trees obtained with a context free grammar (CFG).(See, e.g. [6,7] for work on CFG models in image analysis.)

DTs set a prior $P(\boldsymbol{Z})$ over tree structures $\boldsymbol{Z}$ which allows each node to choose its parent so as to find one which gives a higher posterior probability $P(\boldsymbol{Z}, \boldsymbol{X}_h | \boldsymbol{X}_v)$ for a given image $\boldsymbol{X}_v$. This was shown in [1] to overcome the deficiency of TSBNs. Nodes are also allowed to choose to disconnect and form their own tree, thus allowing *objects* to be represented. However, the number of possible tree structures in the DT grows exponentially with tree depth and exact inference techniques are no longer tractable. Simulated annealing was found in [1] to provide an effective (though slow) way of searching for maximum a posteriori (MAP) configurations.

The alternative to sampling is variational methods, of which mean field is the simplest, using a factorised distribution. In [8] mean field was seen to find comparable MAP solutions to annealing in order of 100 times faster, and while also being far more informative than annealing in that it attempts to model the full distribution.

The ultimate test of any model's capabilities is how it performs on real world data, and here we derive a learning algorithm for the dynamic tree and apply it to a database of outdoor scenes. Section 3 describes the exact and mean field EM learning algorithms for the DT parameters, and experiments comparing it to a fixed architecture model showing it offers a significant improvement are given in Section 4. Firstly we shall consider related work.

2 Related Work

There is of course a vast literature on the subject of image analysis. Here we concentrate particularly on probabilistic formulations of the image analysis problem. A fuller discussion of these issues can be found in Chapter 2 of [9].

Within the probabilistic modelling framework, the most popular models are MRF and TSBN models. In the statistical image modelling community these two types of model are known as non-causal and causal MRF models respectively. They are undirected and directed graphical models [10]. Early work on probabilistic image modelling focussed on non-causal MRFs, see e.g. [11,12]. These models typically have a "flat", non-hierarchical structure. They define a stationary process (thereby overcoming the problems of blockiness), but in general the inference problem in a MRF is NP-hard.

The alternative causal MRF formulation uses a directed graph, and the most commonly used form of these models is a TSBN, as described in Section 1. In contrast to flat MRFs, TSBNs provide a hierarchical multiscale model for an image and have efficient linear-time inference algorithms. However, as noted above, the fixed tree structure gives rise to a non-stationary image model and leads to problems of blockiness.

We also note that wavelet models are examples of hierarchical multiscale models for images. For example Crouse *et al* [13] have used a multiscale TSBN to model wavelet coefficients, and DeBonet and Viola [14] have used an interesting tree-structured network for image synthesis using non-Gaussian densities.

Dynamic tree models are as examples of *dynamic* image analysis architectures, in contrast to *fixed* or static architectures. The distinction between the two is that dynamic models don't merely instantiate hidden variables in a fixed structure conditionally on the supplied data, but they seek also to find relationships between substructures of these variables and are able to modify these relationships on the fly.

von der Malsburg [15, 16] has discussed the Dynamic Link Architecture, whereby the network architecture changes dynamically in response to the input. This parallels the inference process in DT architectures, where posterior tree structures are effectively selected in accordance with how well they fit the image structure. We also note that Montanvert et al [17] have discussed irregular tree-structured networks, where the tree-structure is image dependent. Also, Geman and Geman [11] introduced line processes as an extension of the basic MRF approach. These line processes (which also form an MRF) occupy the boundaries between pixels. The line processes are dynamically combined as edge elements to describe the boundaries between regions in the image.

As well as having a dynamic architecture over a fixed number of units it can also be desirable to allow the number of units to vary. Hinton *et al* (1998) [18] adopt such a strategy with their "hierarchical community of experts", whereby the participation or non-participation of a unit in the model is determined by a gating variable. This model has a general directed acyclic graph (DAG) construction. In later work, Hinton *et al* [19] developed the "credibility network" architecture, where a single parent constraint (similar to the construction of *dynamic trees*) is used. The work on credibility networks is quite closely related to ours, although they do not use hidden state variables $\boldsymbol{X}_h$ and instead focus more on the presence or absence of nodes.

Although not an image model, the work of Geiger and Heckerman (1996) [20] on multinets is also relevant. As with the dynamic tree model Geiger and Heckerman construct a number of belief networks conditional on a structure variable $\boldsymbol{Z}$. In that work multinets were used as a way of speeding up inference, as conditional on $\boldsymbol{Z}$ each network will typically be much simpler than the equivalent network obtained by integrating out $\boldsymbol{Z}$.

3 Learning in Dynamic Trees

The dynamic tree can be considered as an ordered set U of nodes $i = 1, 2, \ldots, U$, each of which taking on one of C possible states, $1, 2, \ldots, c$. If $\boldsymbol{Z} = \{z_{ij}\}$ is used to denote the set of possible directed tree structures over these nodes, where z_{ij} is an indicator variable, then $z_{ij} = 1$ indicates that the node i is connected to parent j. By ordering the nodes such that upper level nodes have lower indices than those in the layer(s) below then it means that $z_{ij} \equiv 0$ for $j \geq i$. Finally defining $\boldsymbol{X} = \{x_i^k\}$ to be the set of all of the states of the nodes, then analogously with the z indicator variables, $x_i^k = 1$ if node i is in state k, and is zero otherwise.

There are two components to the dynamic tree model, a prior over tree structures $P(\boldsymbol{Z})$, and the likelihood $P(\boldsymbol{X}|\boldsymbol{Z})$ of being in a particular state conditioned on the structure $\boldsymbol{Z}$. $\boldsymbol{\phi}$ parameters are used to construct the prior and conditional probability tables (CPTs) $\boldsymbol{\theta}$ define the state transition probabilities across connected links acting as a conditional prior over node states. Thus the parameters of the dynamic tree fall into two distinct categories (prior $\boldsymbol{\phi}$, and CPTs, $\boldsymbol{\theta}$) each of which require their own learning algorithm.

In this Section we first define (following [1]) the structure of the dynamic tree model in Section 3.1. We then consider learning in the dynamic tree, deriving firstly an exact Expectation-Maximisation (EM) update for the CPTs (Section 3.2) which we will later use to compare with the mean field approximation. Section 3.3 then derives a mean field based EM learning algorithm for both the CPTs and the prior parameters of the dynamic tree. Finally in Section 3.4 we discuss approaches to handle missing data which is frequently an issue dealing with real-world applications.

3.1 The Dynamic Tree model

The dynamic tree model is made up of two components. A prior $P(\boldsymbol{Z}|\boldsymbol{\phi})$ defines a probability distribution over tree structures $\boldsymbol{Z}$ and is conditional on a set of parameters $\boldsymbol{\phi}$, which are used in its construction and to be learned during training. The nodes of the network are arranged hierarchically on layers with the same number on each layer as in a balanced TSBN of the same complexity. Numbering these nodes $1 \ldots n$ from top level root to the final leaf node an $n \times (n+1)$ connectivity matrix $\boldsymbol{Z}$ is created, with each element z_{ij} a boolean variable denoting the connectivity of node i to parent j, or disconnected.

Using these definitions we denote the prior as

$$P(\boldsymbol{Z}|\boldsymbol{\phi}) = \prod_{ij} \pi_{ij}^{z_{ij}}, \tag{1}$$

where π_{ij} is the probability of a given link occurring ($P(z_{ij}=1)$) and the indicator variable z_{ij} picks out the connected links under the current configuration.

We could simply define a single π value for each connection, however such a scheme is inefficient and there is likely to be far more πs than the usually limited training data would allow for good training. So in preference we adopt the notion of *affinity* parameters and assign each parent node (nodes in the layer above) a particular *affinity* a_{ij}. By doing this we can then share *affinities* between nodes a similar distance from a particular child and also children can reuse the same parent *affinity map* translated to take into effect their different position in the layer. To allow disconnections there is a *null* parent denoted by index 0, who's *affinity* is a_{i0}. We use the same scheme for assigning *affinities* as described in [1]. The *affinities* are then realised as the π probabilities through using a softmax function – which ensures all the probabilities sum to unity – as given by the following

$$\pi_{ij} = \frac{\exp(a_{ij})}{\sum_{j'} \exp(a_{ij'})}. \tag{2}$$

A particular tree-structure $\boldsymbol{Z}$ organizes the nodes into a corresponding TSBN. Each node X_i can take on one of C states. Each layer l of the structure has associated with it a prior probability vector P_l and conditional probability table (CPT) θ_l which defines the probability of the child of X_i being in a particular state given that X_i is in a particular state.

The image vector is instantiated at the leaves of the tree, $\boldsymbol{X}_v$ and all other nodes are hidden units $\boldsymbol{X}_h$. The joint distribution for the whole tree $P(\boldsymbol{X}, \boldsymbol{Z})$, where $\boldsymbol{X} = \boldsymbol{X}_v \cup \boldsymbol{X}_h$ is conditioned on the CPTs $\boldsymbol{\theta}$.

3.2 An EM update for learning the CPTs

Using the notation defined above and a training set of $p = 1 \ldots P$ patterns, the log-likelihood of the data under the model is given by

$$\sum_p \log P(\boldsymbol{X}_v^p) = \sum_{p=1}^{P} \log \sum_{\boldsymbol{Z}^p, \boldsymbol{X}_h^p} P(\boldsymbol{X}_v^p, \boldsymbol{X}_h^p | \boldsymbol{Z}^p, \boldsymbol{\theta}) P(\boldsymbol{Z}^p | \boldsymbol{\phi}). \tag{3}$$

Note that the $\boldsymbol{Z}$s are summed over T tree configurations, and for each there will be a different $\boldsymbol{X}_h$. Notation for this is omitted for clarity.

To assign each parent-child combination its own unique CPT would lead to massive over-parameterisation for the limited training data usually available, so it was deemed sensible to share the CPTs among nodes on the same level (scale). θ_I is used to denote the shared CPT for the set of nodes $\boldsymbol{X}_I$.

Standard calculus and the use of Lagrange multipliers to ensure that the CPTs are valid probabilities, produces the following EM update for the CPT element θ_{Ij}^{kl} representing the transition probability $P(x_i^k | x_j^l)$

$$\hat{\theta}_{Ij}^{kl} = \frac{\sum_{p, \boldsymbol{Z}^p} \sum_{x_i \in \boldsymbol{X}_I} P(x_i^{k(p)}, x_j^{l(p)} | \boldsymbol{X}_v^p, \boldsymbol{Z}^p, \boldsymbol{\theta}) P(\boldsymbol{Z}^p | \boldsymbol{X}_v^p, \phi)}{\sum_{p, \boldsymbol{Z}^p} \sum_{x_i \in \boldsymbol{X}_I} \sum_{k'} P(x_i^{k'(p)}, x_j^{l(p)} | \boldsymbol{X}_v^p, \boldsymbol{Z}^p, \boldsymbol{\theta}) P(\boldsymbol{Z}^p | \boldsymbol{X}_v^p, \phi)}, \tag{4}$$

where

$$P(x_i^k, x_j^l | \boldsymbol{X}_v^p, \boldsymbol{Z}^p, \boldsymbol{\theta}) = \frac{1}{\sum_{l'} \pi(x_j^{l'}) \lambda(x_j^{l'})} \lambda(x_i^k | \boldsymbol{\theta}) \theta_{Ij}^{kl} \pi(x_j^l | \boldsymbol{\theta}) \prod_{y \in s(x_i)} \lambda_y(x_j^l | \boldsymbol{\theta}). \tag{5}$$

The λs and πs are the Pearl messages used to pass information to a node about the states of its children and parents respectively [5], and $s(x_i)$ is the set of siblings of node i. This derivation is an extension of that of [3] used for fixed architecture TSBNs, full details of which are given in [9].

3.3 Mean Field EM in Dynamic Trees

In the mean field DT [8] the true posterior distribution $P(\boldsymbol{X}_h, \boldsymbol{Z}|\boldsymbol{X}_v, \boldsymbol{\theta}, \boldsymbol{\phi})$ is approximated by a factorising distribution, $Q(\boldsymbol{X}_h, \boldsymbol{Z}|\boldsymbol{X}_v) = Q(\boldsymbol{X}_h)Q(\boldsymbol{Z})$. This can be used to find a lower bound on the log-likelihood of the data

$$\sum_p \log P(\boldsymbol{X}_v^p) \geq \sum_{p,\boldsymbol{X}_h^p,\boldsymbol{Z}^p} Q(\boldsymbol{X}_h^p, \boldsymbol{Z}^p|\boldsymbol{X}_v^p) \log \frac{P(\boldsymbol{X}_v^p, \boldsymbol{X}_h^p, \boldsymbol{Z}^p|\boldsymbol{\theta}, \boldsymbol{\phi})}{Q(\boldsymbol{X}_h^p, \boldsymbol{Z}^p|\boldsymbol{X}_v^p)} \stackrel{def}{=} \mathcal{L}(Q, \boldsymbol{\theta}, \boldsymbol{\phi}) \tag{6}$$

which can be shown to be tightest when the KL-divergence between the approximating distribution and the true posterior is minimised, and suggests an iterative EM-style algorithm for variational methods [21]. In the E-step the variational log-likelihood $\mathcal{L}(Q, \boldsymbol{\theta}, \boldsymbol{\phi})$ is maximised wrt Q holding $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ fixed (by minimising the KL-divergence). Then in the M-step Q is fixed and the bound is maximised wrt to the model parameters. The CPTs $\boldsymbol{\theta}$ and the affinities $\boldsymbol{\phi}$ constitute two very distinct parameter types and as such require different treatment.

Learning the CPTs: The derivation uses a similar methodology to that of exact EM (see [9]). Performing this optimisation on the DT thus gives rise to the following update rule for the CPTs

$$\hat{\theta}_{Ij}^{kl} = \frac{\sum_{p,\boldsymbol{Z}^{(p)}} \sum_{X_i \in \boldsymbol{X}_I} Q(X_i^{k(p)}) Q(X_j^{l(p)}) Q(\boldsymbol{Z}^{(p)}) z_{ij}^{(p)}}{\sum_{k'} \sum_{p,\boldsymbol{Z}^{(p)}} \sum_{X_i \in \boldsymbol{X}_I} Q(X_i^{k'(p)}) Q(X_j^{l(p)}) Q(\boldsymbol{Z}^{(p)}) z_{ij}^{(p)}}. \tag{7}$$

Note the similarity of this update to Equation 4, except that the exact quantity $P(x_i^{k(p)}, x_j^{l(p)}|\boldsymbol{X}_v^p, \boldsymbol{Z}^p, \boldsymbol{\theta}) P(\boldsymbol{Z}^p|\boldsymbol{X}_v^p, \boldsymbol{\phi})$ has been replaced by its mean-field equivalent.

Learning the Affinities: The affinities set a prior over tree structures. As for the CPTs affinities also can be shared between nodes to reduce parameterisation. We define $\boldsymbol{\phi}$ to denote such sets of shared affinities, and a_{ij} is the individual *affinity* a node i has for connecting to parent j. The raw affinity values are translated into a probability distribution using Equation (2).

To obtain an update we maximise the bound on the log-likelihood (Equation (6)) with respect to the affinities. Differentiating Equation (6) with respect to a shared affinity ϕ produces

$$\frac{\partial \mathcal{L}(Q, \boldsymbol{\theta}, \boldsymbol{\phi})}{\partial \phi} = \sum_p \sum_{i,j:a_{ij} \in \phi} [\mu_{ij} - \pi_{ij}], \tag{8}$$

where $\mu_{ij} = \langle z_{ij} \rangle_{Q(\boldsymbol{Z})}$. Thus $\mathcal{L}$ can be optimised wrt $\boldsymbol{\phi}$ using standard gradient optimisation techniques such a conjugate gradients. (See [9] for the complete derivation.)

The Complete Learning Algorithm therefore consists of fixing the model parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, and running mean field until the Qs reach convergence. Then we fix the Qs and calculate the update for the CPT using Equation (7). A gradient based optimiser can then be used on Equation (8) to traverse one or more steps along the affinity gradient. The $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ parameters are then updated, and the process repeated.

3.4 Handling Missing Data

A reality of dealing with real world is that frequently we are given examples which are incomplete, broaching the issue of how do we deal with missing data? If we simply set the missing pixel to a random value we may well end up learning noise, and even a uniform instantiation with all states equi-probable could still be saying something which is untrue.

Assuming that the data is missing at random (see [22]) the correct solution is to marginalize out the uninstantiated variables. For exact inference using Pearl message passing in belief networks this is achieved automatically by the algorithm when the λ value of the uninstantiated node is set to $(1, 1, \ldots, 1)$. With the mean field approximation a little care is required as it is unclear as to whether an uninstantiated leaf node may exert any unintended influence on the resultant equilibrium distribution. The solution we have used is to temporarily modify the connection probability table so that missing data leaf nodes have a probability of disconnection of 1.0. They then do not contribute anything towards the mean field equilibrium distribution. The implications for the learning rule update equations is that such nodes should be ignored for the given example.

4 Experiments

We now consider a real-world application of the dynamic tree. In Section 4.1 we start by introducing the dataset and discussing some of its features, before using it in Section 4.2 where we learn the dynamic tree model parameters using the EM learning algorithms described earlier.

4.1 The Image Data

We use a set of colour images of out-door scenes from the Sowerby Image database[2] for our experiments. These feature both rural and urban examples and contain many of the typical objects you would expect to find – such as the roads, cars and buildings of urban environments to the country lanes and fields of the rural. The original scenes were photographed using small-grain 35mm transparency film under carefully controlled conditions at a number of locations in

[2] This database can be made available to other researchers. Please contact Dr Andy Wright or Dr Gareth Rees, Advanced Information Processing Department, Advanced Technology Centre - Sowerby, BAE SYSTEMS Ltd, PO Box 5 Filton, Bristol, BS34 7QW, UK, email:`gareth-s.rees@baesystems.com` for details.

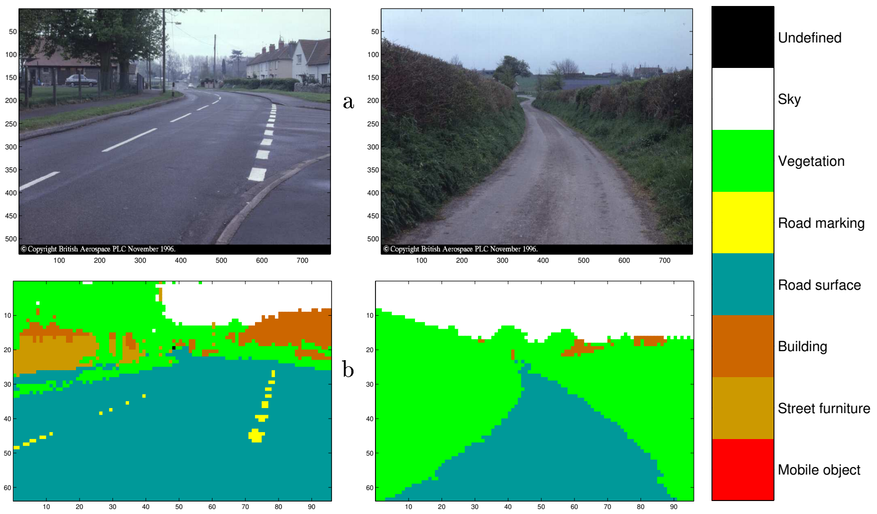


Fig. 1. (a) An urban and a rural scene from the Sowerby database and (b) their corresponding labelled images. To the right is a key defining the labels used.

and around Bristol, UK. The analogue transparencies were then digitised using a calibrated scanner to produce high quality 24-bit colour representations.

In addition to the raw images the database also contains corresponding labelled examples created by over-segmenting the images and then hand labelling each region produced. This gave rise to 92 labels, organised hierarchically. A fixed TSBN model ws applied to this database in [23], in which the 92 class labels were merged down to 7 super classes. Since the fixed TSBN can be viewed a special case of the dynamic tree where the architecture is not allowed to change this provides an excellent model for comparison and consequently we chose to adopt the same class labels.

The labels distinguish all of the key regions of interest in the image representing, "sky", "vegetation", "road markings", "road surface", "building", "street furniture" and "mobile object." Such is the nature of gathering real data that circumstances inevitably arise where there is missing data, so a further dummy "unlabelled" class is added to accommodate this. Unlike the others the unlabelled class is not learned since it is an artifact of the labelling strategy adopted and is dealt with as described in Section 3.4. Figure 1(a) shows one urban and one rural scene from the database.

The dynamic tree as formulated operates with discrete classes hence it is necessary to use the labelled images instead of the real valued data. However there are a wealth of established techniques which can be used to produce a mapping from real valued pixels to a discrete number of classes. For example in [3] a multi-layer perceptron (MLP) is trained to do this. The reverse mapping from labels to real valued pixels is equally straightforward, eg. using Gaussian mixture models. The MLP and Gaussian mixture model methods are compared in [23].

4.2 Experimental Procedure

The Sowerby database contains 104 images which were randomly divided in [3] into a training set of 61 images and the rest allocated as a test set. For comparative purposes we use the same training set to learn the dynamic tree model parameters.

The full size images comprise of 768×512 pixels, which Feng *et al* [23] reduce by sub-dividing them into regions and adopting a majority voting strategy to chose the winning class label. In cases where there was a tie a label was chosen probabilistically from the competing classes based on the prior probabilities of the given labels being seen in the images. The class label prior probabilities for the training and test sets are given in Table 1.

Table 1. Image pixel class priors for the training and test sets.

Class	**P(Label)**	
	Training Set	Test Set
Unlabelled	0.0036	0.0418
Atmospheric phenomena	0.1443	0.1140
Vegetation	0.3703	0.3899
Road surface markings	0.0012	0.0008
Road surface	0.4210	0.3804
Building	0.0473	0.0569
Street furniture	0.0056	0.0112
Mobile object	0.0067	0.0050

We adopt this procedure and down-sample to an image size of 96×64 pixels. An urban and rural example at this resolution are shown in Figure 1(b) and it can be clearly seen that most of the detail is still present.

For the experiments we use a 7-level model. The node arrangement is quadtree for all bar the second level where there is 6 instead of 4 nodes. The latter is to accommodate the $3:2$ aspect ratio of the training images and produces the desired image size of 96×64 pixels.

Setting the initial model parameters is an open question and probably a paper in itself. This issue has already been explored by [24, 23] and in keeping with the philosophy that a child node would favour being in the same state as its parent we make the CPTs strongly diagonal with probability 0.9 and the rest of the probability mass is shared equally among the 6 other states. The prior for root nodes being in a particular state was set to be uniform. All models were initialised with the above CPTs and state prior.

The dynamic tree model has a further set of parameters called affinities, which are used to set the prior over tree structure ([1]). They are given relative to a parameter called the *natural parent* affinity which is the parent a given node would have if it were part of a balanced tree. This is set to 0 for reference purposes. Important other connections a node may wish to make are to its nearest neighbour(s) or to disconnect and become a root (*null* node). Equation (2) is used to map the affinity values into probabilities. In [9] preliminary investigation showed a useful working range for the dynamic tree was for affinity values of 0 and -3 for these two respectively. Other connections are given a probability of zero. The interpretation for this is that nearest neighbour connections are equiprobable with the natural parent and disconnections are possible but with a far lower probability.

We make a comparison between the fixed architecture balanced tree-structured belief network of [23] and the dynamic tree. In the experiments on the DT the mean field EM learning algorithm described in Section 3.3 was used. Firstly mean field was run over all of the training examples to produce an approximation of the joint distribution for the E-step of the algorithm. This involved updating the means cyclically for a number of iterations until equilibrium was reached, then updating the structure $Q(\boldsymbol{Z})$ in single step and repeating until convergence. In practice the 5 complete iterations were sufficient but the algorithm was allowed to terminate early if the variational log-likelihood altered by less than 0.05 between cycles.

The M-step of the algorithm is a single step update for the CPTs, but for the affinities a gradient method is necessary. Conjugate gradients was used with the optimiser being allowed to take up to 3 steps. Three steps were necessary in order to take advantage of the conjugate gradients – performing only one would simply be gradient descent. After calculating new estimates for the CPTs and affinities all of the model parameters were updated and the process repeated.

A comparison was made between three types of model, the fixed quad-tree (fixed architecture), a dynamic tree where only the CPTs were learned (CPT-only DT) and the dynamic tree model where all parameters were learned (full DT). All used the mean field EM algorithm as described in Section 3.3, and additionally exact EM learning was performed on a fixed architecture quad-tree model using the exact EM learning update given in Section 3.2. (For a comparison of mean field EM against the exact method in fixed trees see [24].)

4.3 Experimental Results

The learning curves on the training set of 61 patterns are given in Figure 2, showing the variational log-likelihoods obtained by the three DT models learned using the mean field EM approach, and the log-likelihood of the fixed architecture quad-tree model learned by exact EM.

Considering firstly mean field EM learning, it can be seen from the Figure that while the fixed architecture mean field EM model shows good improvement in variational log-likelihood during training it can only go so far. Learning only the CPTs of a dynamic tree model whose affinities were set from observed optimal parameters in toy data (see [24]) appears to offer an advantage over the fixed architecture model with increased variational log-likelihood over the training set. The full DT model does better still, indicating that having variable architecture offers an advantage over the fixed architecture model.

Exact EM on the fixed architecture model can be seen to obtain a higher log-likelihood than variational log-likelihood bound given by both the fixed

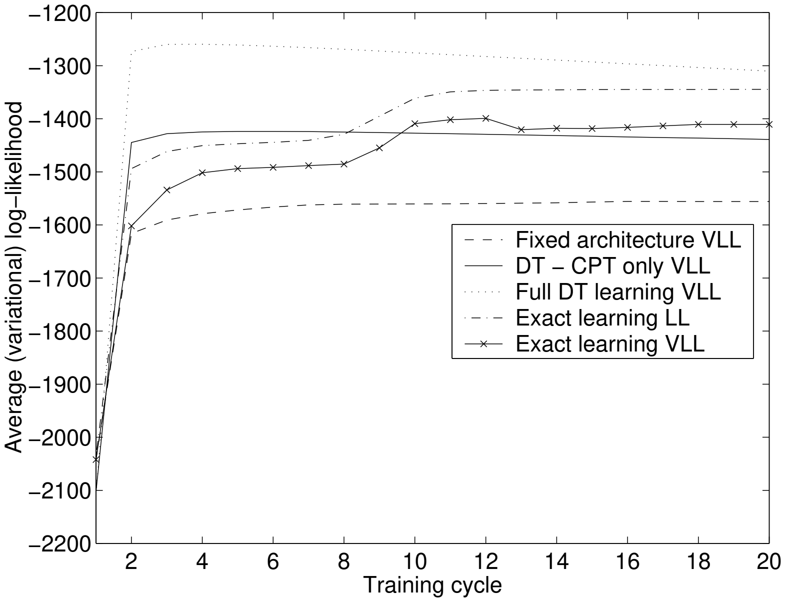


Fig. 2. Learning curves on the 61 image training set for the fixed tree, CPT only learn dynamic tree, fully learned dynamic tree models, and exact EM learning on the fixed tree.

and CPT-only mean field learning models. This is not surprising as the exact approach using the true probability distribution is able give the actual log-likelihood where-as mean field gives only a lower bound on the likelihood of the data $P(\boldsymbol{X}_v)$, and since mean field is only an approximation it may not be that tight. To confirm this on the real data the mean field variational log-likelihood was calculated during training at each iteration of the model learned under exact EM. This is shown as the crossed line in the plot of Figure 2 and we see clearly that the variational log-likelihood does indeed lie significantly below the exact log-likelihood. However, we observe that the variational log-likelihood obtained by mean field learning on the full dynamic tree beats the exact log-likelihood for EM training of the fixed architecture model, so clearly even though we can't perform exact learning on the dynamic tree model – which we would like to do – it still out-performs the fixed architecture model learned by exact approaches.

Though apparently fairly stable, it can be seen from the plot that the average variational log-likelihood for both variants of the DT appears to decline after 3–4 training iterations. A known weakness of the mean field learning algorithm ([24]) is that it tends to settle at unstable equilibria where even a slight perturbation of the parameters can cause "spontaneous symmetry breaking." On the toy data learning of [24] it was observed that in subsequent iterations the model tries to correct this by hardening the CPTs resulting in a drop in performance, and this is probably what is happening here.

We can evaluate the quality of the learned models by calculating the variational log-likelihood on the test set of 43 images. This can then be used to obtain the coding cost in bits/pixel, where

$$\text{Coding cost} = -\frac{\log_2 P(\boldsymbol{X}_v^p)}{\#\ \text{labelled pixels in image}}. \tag{9}$$

An important point to remember is that with variational methods we are calculating a lower bound on the likelihood (an upper bound on coding cost), and the likelihood can only be at least as good or better.

Table 2. Performance on the test set of 43 images.

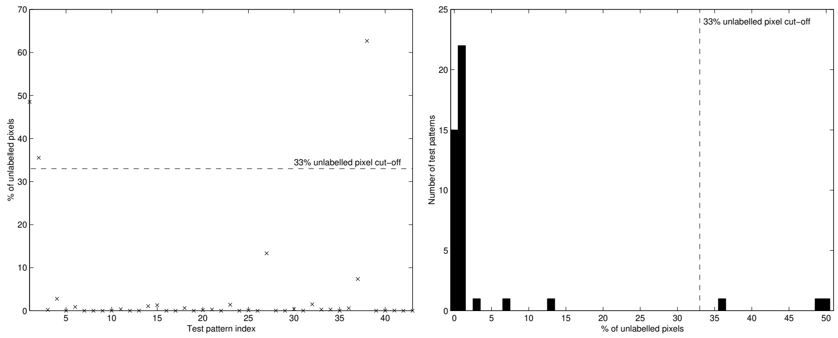
Model		**After Training Cycle**	**Bound on Coding Cost (bpp)**	
			Full	Less than 33% missing
Mean field	Fixed architecture	15	0.8588	0.3918
	DT - CPT only	2	0.4089	0.3228
	Full dynamic tree	2	0.3805	0.2942
Exact EM	Fixed architecture	10	0.3421	0.3253
JPEG-LS		–	0.3810	0.3782

Table 2 gives these coding costs for the various models and is compared with the lossless JPEG-LS codec[3] – which is available from `http://www.hpl.hp.com /loco/`. The second column of the table gives the number of training epochs used to train each of the models before applying them on the test data. In an attempt to minimise over-training this usually occurred at the point where the training error first peaked (see Figure 2). We see from the third column that on the full test set JPEG-LS outperforms all except the full DT and exact EM models.

However, an examination of the percentage of unlabelled pixels in each of the images of the test set (Figure 3) shows 3 images to have greater than 33% unlabelled pixels which is extremely unusual. Ordinarily we might expect some unlabelled pixels and so need a robustness to them, but images degraded to that extent could reasonably be rejected as bad data. Removing these 3 images gives average coding costs as listed in the final column of the table. As can be seen now even the fixed tree is comparable to JPEG-LS and the full dynamic tree offers significant improvements over them both – the DT model was found to have a higher variational log-likelihood than the fixed architecture model in 42/43 of the test images.

Exact EM learning of the fixed architecture quad-tree tree also performs well. Over the full set of test images it achieves a lower coding cost than even the full dynamic tree. It has already been noted that the mean field approximation is quite poor when there is little data so it is likely to perform badly on images with

[3] In the case of JPEG-LS the coding cost was obtained by compressing the images and measuring the size of the compressed files.



(a) (b)

Fig. 3. Percentage of unlabelled pixels in each of the 43 test set images (a) per image and, (b) as histogram.

many missing pixels, and indeed we see this to be the case. (The variational log-likelihoods of the 3 images with more than 33% missing pixels are significantly lower than those of the other 40 images.) So it is not surprising that the fixed architecture model learned by exact EM achieves a slightly lower coding cost than the dynamic tree over the full test set of 43 images where there are many unlabelled pixels – especially as it has the advantage of using the true probability distribution for the model. However, after removing the 3 images with more than 33% missing pixels the dynamic tree model achieves a lower coding cost (0.2942 bpp) than the exact EM learned quad-tree model (0.3253 bpp). On this set of 40 images, lossless JPEG achieves 0.3782 bpp.

Acknowledgements NJA is supported by an EPSRC research studentship. The work of CW is supported through EPSRC grant GR/L78161 *Probabilistic Models for Sequences*. The authors would like to thank British Aerospace for making the Sowerby Image Database available to us.

References

1. Williams, C.K.I., Adams, N.J.: DTs: Dynamic Trees. In Kearns, M.J., Solla, S.A., Cohn, D.A., eds.: Advances in Neural Information Processing Systems 11. MIT Press (1999) 634–640
2. Bouman, C.A., Shapiro, M.: A Multliscale Random Field Model for Bayesian Image Segmentation. IEEE Transactions on Image Processing **3(2)** (1994) 162–177
3. Feng, X., Williams, C.K.I.: Training Bayesian Networks for Image Segmentation. In: Proceedings of SPIE. Volume 3457. (1998)
4. Luettgen, M.R., Willsky, A.S.: Likelihood Calculation for a Class of Multiscale Stochastic Models, with Application to Texture Discrimination. IEEE Transactions on Image Processing **4(2)** (1995) 194–207
5. Pearl, J.: Probabilistic Reasoning in Intelligent Systems. 2nd edn. Morgan Kaufman Publishers Inc., San Francisco, USA (1988)

6. Chou, P.A.: Recgonition of Equations Using a Two-Dimensional Stochastic Context-Free Grammar. Visual Communications and Image Processing IV **1199** (1989) 852–863
7. Geman, S., Manbeck, K.: Experiments in Syntactic Recognition. Technical Report CICS-P-411, Division of Applied Mathematics, Brown University, Providence, RI 02912 USA (1994)
8. Adams, N.J., Storkey, A.J., Ghahramani, Z., Williams, C.K.I.: MFDTs: Mean Field Dynamic Trees. In Sanfeliu, A., Villanueva, J.J., Vanrell, A., Alquézar, R., Huang, T., Serra, J., eds.: Proceedings of 15th International Conference Pattern Recognition. Volume 3, *Image speech and Signal Processing.*, IEEE Computer Society (2000) 151–154
9. Adams, N.J.: Dynamic Trees: A Hierarchical Probabilistic Approach to Image Modelling. PhD thesis, Institute for Adaptive and Neural Computation, Artificial Intelligence, Division of Informatics, University of Edinburgh, 5 Forrest Hill, Edinburgh, EH1 2QL, UK (2001) Available at: `http://www.anc.ed.ac.uk/code/adams/`.
10. Lauritzen, S.L.: Graphical Models. Oxford University Press (1996)
11. Geman, S., Geman, D.: Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. In: IEEE Transactions on Pattern Analysis and Machine Intelligence. Volume 6, no. 6. (1984) 721–741
12. Chellappa, R., Chatterie, S.: Classification of Textures using Guassian Markov Random Fields. In: IEEE Trans. Accoust., Speech and Signal Processing. Volume 33. (1985) 959–963
13. Crouse, M., Nowak, R., Baraniuk, R.: Wavelet-based statistical signal proccessing using hidden Markov models. IEEE Transactions on Signal Processing **46** (1998) 886–902
14. De Bonet, J.S., Viola, P.A.: A Non-Parametric Multi-Scale Statistical Model for Natural Images. In Jordan, M.I., Kearns, M.J., Solla, S.A., eds.: Advances in Neural Information Processing Systems 10. MIT Press, Cambridge, MA (1998)
15. von der Malsburg, C.: The correlation theory of brain function. Internal Report 81-2, Max-Planck-Institut für Biophysikalische Chemie (1981) Reprinted in *Models of Neural Networks*, eds. K. Schulten and H.-J. van Hemmen, 2nd. ed, Springer, 1994.
16. von der Malsburg, C.: Dynamic link architecture. In Arbib, M.A., ed.: Handbook of Brain Theory and Neural Networks. MIT Press (1995) 329–331
17. Montanvert, A., Meer, P., Rosenfeld, A.: Hierarchical Image Analysis Using Irregular Tessellations. IEEE Trans. Pattern Analysis and Machine Intelligence **13(4)** (1991) 307–316
18. Hinton, G.E., Sallans, B., Ghahramani, Z.: A Hierarchical Community of Experts. In Bishop, C.M., ed.: Neural Networks and Machine Learning. Springer-Verlag New York inc. (1998)
19. Hinton, G., Ghahramani, Z., Teh, Y.W.: Learning to Parse Images. In Solla, S.A., Leen, T.K., Müller, K.R., eds.: Advances in Neural Information Processing Systems 12. MIT Press (2000) 463–469
20. Geiger, D., Heckerman, D.: Knowledge Representation and Inference in Similarity Networks and Bayesian Multinets. Artificial Intelligence **82** (1996) 45–74
21. Jordan, M.I., Ghahramani, Z., Jaakkola, T.S., Saul, L.K.: An Introduction to Variational Methods For Graphical Models. In Jordan, M.I., ed.: Learning in Graphical Models. Kluwer Academic Publishers (1998) 105–161
22. Little, R.J.A., Rubin, D.B.: Statistical Analysis with Missing Data. John Wiley, New York, USA (1987)

23. Feng, X., Williams, C.K.I., Felderhof, S.N.: Combining Belief Networks and Neural Networks for Scene Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence (2001) Accepted for publication.
24. Adams, N.J., Williams, C.K.I., Storkey, A.J.: Comparing Mean Field and Exact EM in Tree Structured Belief Networks. In: Fourth International ICSC Symposium on Soft Computing and Intelligent Systems for Industry. ICSC-NAISO Adademic Press (2001)