THE UNIVERSITY of EDINBURGH

Edinburgh Research Explorer

Comparing Mean Field and Exact EM in Tree Structured Belief Networks

Citation for published version:
Adams, NJ, Williams, CKI & Storkey, AJ 2001, Comparing Mean Field and Exact EM in Tree Structured Belief Networks. in *In Fourth International ICSC Symposium on Soft Computing and Intelligent Systems for Industry. ICSC-NAISO Adademic.* Thoemmes Press.

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
In Fourth International ICSC Symposium on Soft Computing and Intelligent Systems for Industry. ICSC-NAISO Adademic

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 12. Sep. 2026

Comparing Mean Field and Exact EM in Tree Structured Belief Networks

Nicholas J Adams
Institute for Adaptive and
Neural Computation
University of Edinburgh
5 Forrest Hill, Edinburgh UK
nicka@dai.ed.ac.uk

Christopher K I Williams
Institute for Adaptive and
Neural Computation
University of Edinburgh
5 Forrest Hill, Edinburgh UK
c.k.i.williams@ed.ac.uk

Amos J Storkey
Institute for Adaptive and Neural Computation
University of Edinburgh
5 Forrest Hill, Edinburgh UK
a.storkey@ed.ac.uk

Abstract

We make a thorough comparison between a variationally-based learning approach and exact EM using tractable fixed architecture tree-structured belief networks, and so gain valuable insights into learning with mean field methods. We then introduce disconnections into the model showing how they can be folded into a single structure by viewing them as degeneracies in the conditional probability tables, and investigate learning with them. The results suggest that mean field performs sufficiently well to be useful in learning in more complex models where standard approaches are intractable.

1. Introduction

Dynamic Trees (DTs) are a generalisation of the fixed architecture balanced tree structured belief network (TSBN). The TSBN has has been shown [3, 4] to be attractive for image segmentation because it is hierarchically structured, so providing a multi-scale interpretation of the image. Unlike other standard techniques – such as Markov Random Fields – TSBNs also have efficient inference algorithms [6].

The fixed architecture of balanced TSBNs makes their segmentation prone to *blockiness*. DTs provide a prior over tree structures. Thus the posterior over tree structures will favour those trees compatible with the structure of a given image; as seen in [8] this reduces the blockiness problem. Nodes can also 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 techniques are no longer tractable. Simulated annealing was found [8] 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 [2] mean field was seen to find comparable MAP solutions to annealing in order of 100 times faster, and is also far more informative than annealing as it attempts to model the full posterior distribution.

To apply the DT model to useful real world problems such as the segmentation of images, learning of the model parameters from data is necessary. The speed and good performance of mean field methods for inference in the Dynamic Tree makes it an attractive candidate to base the learning algorithm upon. In this paper a mean field EM algorithm for learning in the Dynamic Tree is given. It is applied to the fixed architecture TSBN network – a special case of the DT – and contrasted with exact EM which

is tractable in these networks. This provides some interesting insights into the capabilities of the technique and is an important step towards learning in the full DT, currently work in progress.

Section 2 describes exact and mean field EM learning for the conditional probability tables (CPTs). A novel way of viewing disconnections in the DT is then introduced in Section 3 in which they are folded into a single tree structure, and the results of experiments comparing exact and variational learning in these models is given in Section 4.

2. Learning in Dynamic Trees

2.1 An EM update for learning the CPTs

The Dynamic tree model is made up of two components. A prior $P(\boldsymbol{Z}|\boldsymbol{\phi})$ defines a probability distribution over the tree structure $\boldsymbol{Z}$ 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 Z is created, with each element z_{ij} a boolean variable denoting the connectivity of node i to parent j, or disconnected.

Each node can take on one of C states and $P(x_i^k)$ is defined as the probability that node i is in state k. 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 $\boldsymbol{\theta}$. These describe the state transition probabilities between parent and child on connected links. Using this notation and a training set of $p = 1 \ldots P$ patterns, the log likelihood of the data under the model is given by

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

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 at 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 $P(x_i^k | x_j^l)$

$$\tilde{\theta}_{Ij}^{kl} = \frac{\sum_{p, \boldsymbol{Z}^p, x_i \in \boldsymbol{X}_I} P(x_i^{k(p)}, x_j^{l(p)} | \boldsymbol{X}_v^p, \boldsymbol{Z}^p, \boldsymbol{\theta}) P(\boldsymbol{Z}^p | \boldsymbol{X}_v^p, \boldsymbol{\phi})}{\sum_{p, \boldsymbol{Z}^p, x_i \in \boldsymbol{X}_I} \sum_{k'} P(x_i^{k'(p)}, x_j^{l(p)} | \boldsymbol{X}_v^p, \boldsymbol{Z}^p, \boldsymbol{\theta}) P(\boldsymbol{Z}^p | \boldsymbol{X}_v^p, \boldsymbol{\phi})} \tag{2}$$

where

$$P(x_i^k, x_j^l | \boldsymbol{X}_v^p, \boldsymbol{Z}^p, \boldsymbol{\theta}) = \frac{1}{\sum_{l'} \pi(x_j^{l'}) \lambda(x_j^{l'})} \lambda(x_i^k | \boldsymbol{\theta}) \theta_{Ij}^{kl} \pi(x_j^l | \boldsymbol{\theta}) \prod_{y \in s(x_i)} \lambda_y(x_j^l | \boldsymbol{\theta}) \tag{3}$$

The λs and πs are the Pearl messages used to pass information to a node about the states of its children and parents respectively [6], and $s(x_i)$ is the set of siblings of node i. This derivation is an extension of that of [4] used for fixed architecture TSBNs, full details of which are given in [1].

2.2 Mean Field EM in Dynamic Trees

In the mean field Dynamic Tree [2] the true posterior distribution $P(\boldsymbol{X}_h, \boldsymbol{Z} | \boldsymbol{X}_v, \boldsymbol{\theta}, \boldsymbol{\phi})$ is approximated by a factorised 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

$$\log P(\boldsymbol{X}_v) \geq \sum_{p, \boldsymbol{X}_h^p, \boldsymbol{Z}^p} Q(\boldsymbol{X}_h^p, \boldsymbol{Z}^p | \boldsymbol{X}_v^p) \log \frac{P(\boldsymbol{X}_v^p, \boldsymbol{X}_h^p, \boldsymbol{Z}^p | \boldsymbol{\theta}, \boldsymbol{\phi})}{Q(\boldsymbol{X}_h^p, \boldsymbol{Z}^p | \boldsymbol{X}_v^p)} \tag{4}$$

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 [5]. In the E-step the bound (variational log-likelihood) 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 $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$. For DTs this optimisation gives the following update rule for the CPTs

$$\tilde{\theta}_{Ij}^{kl} = \frac{\sum_{p,x_i \in \boldsymbol{X}_I} Q(x_i^{k(p)}) Q(x_j^{l(p)}) Q(\boldsymbol{Z}^{(p)}) z_{ij}}{\sum_{k'} \sum_{p,x_i \in \boldsymbol{X}_I} Q(x_i^{k'(p)}) Q(x_j^{l(p)}) Q(\boldsymbol{Z}^{(p)}) z_{ij}} \tag{5}$$

The derivation of the update uses a similar methodology to that of exact EM (see [1]), and so it is no surprise that they are of a similar form.

3. The Disconnecting Tree

The disconnecting tree is an intermediate step between a single fixed architecture TSBN and the Dynamic Tree. In it each node is allowed the choice of connecting to a single parent (we restrict this to the *natural* parent which is the one it would have if it were part of a balanced TSBN) or being disconnected.

In the Dynamic Tree disconnections are modelled by having a *null* parent on each layer which a node can choose to connect to with a particular probability. This gives excellent interpretability as it is immediately obvious from a node's indicator vector $\boldsymbol{z}_i$ whether it has disconnected or chosen to connect to a particular parent, however for learning this presents a problem. The reason is clear, that for a disconnecting tree of n nodes with each having the choice of connecting or disconnecting, then there are 2^{n-1} distinct configurations (the top level node can only be a root), which quickly becomes intractable to enumerate.

There is an alternative way of handling disconnections which arises by viewing the prior state vector as a degenerate CPT made up of identical row vectors each a copy of the prior. By making the assumption that any degeneracy in a CPT is a contribution to the prior then it is possible to fold the prior into the CPT, and after training exactly recover the prior, disconnection probability and CPT from the learned CPT. Defining P_d as the prior disconnection probability for a node, $\boldsymbol{\pi}$ as the prior (a row vector), and θ the CPT, then we can fold in disconnections using

$$\theta_{eff} = (1 - P_d)\theta + P_d \mathbf{1}\boldsymbol{\pi} \tag{6}$$

to get an effective CPT, θ_{eff}. To recover the probabilities after learning use

$$\hat{P}_d = \sum_k min_l(\theta_{eff}^{kl}) \tag{7}$$

$$\hat{\pi}_k = min_l(\theta_{eff}^{kl}) / \hat{P}_d \tag{8}$$

$$\hat{\theta} = (\theta_{eff} - \hat{P}_d \mathbf{1}\hat{\boldsymbol{\pi}}) / (1 - \hat{P}_d) \tag{9}$$

where $\mathbf{1}$ is a vector of 1s, and θ_{eff}^{kl} the effective CPT whose rows index the child state k, and columns the parent state l. Equation (7) extracts the degenerate probability component for each child state and attributes it to the disconnection probability. Equation (8) finds the normalised ratio of these which is the prior vector, and in (9) these are weighted by $\hat{P}_d$ and subtracted from the effective CPT. By folding in disconnections in this way the disconnecting tree can be represented in a single structure and learning then is identical to the fixed architecture TSBN.

4. Experiments

We now wish compare mean field EM learning against exact EM. We first apply the algorithms to a fixed architecture TSBN before assessing the effect of introducing disconnections has on learning. The theory described in the previous section is for the most general case, the Dynamic tree, but is readily simplified to each of the above scenarios. For the fixed architecture tree the $\boldsymbol{Z}$s are superfluous as there is only a single structure. In the disconnecting tree probabilities of disconnection P_{d_i}, are assigned to each node i and z_i is a binary indicator variable over the two connectivity states, taking on a 1 for connection to the parent and zero for disconnection.

In Section 4.1 we evaluate the performance of mean field EM learning against exact EM on a fixed architecture balanced TSBN, before seeing how it learns in the disconnecting tree in Section 4.2.

4.1 Learning in a fixed architecture tree

For the experiments a generative model was used to provide the training data. It was constructed to be a 4 level binary tree having binary node states. The CPTs were set to be 0.9 on diagonal and 0.1 off diagonal, with uniform priors for roots. With this architecture we have 1-d images of 8 pixels, giving a maximum of 256 possible image vectors counting reflections as different. The training set comprised of all 256 of these images vectors, each weighted by the

probability of seeing it under the generative model. Thus we effectively use an infinite training set. This was done so as to find the best possible performance that each model can achieve, and so set an upper bound on what can be realistically achieved in practice.

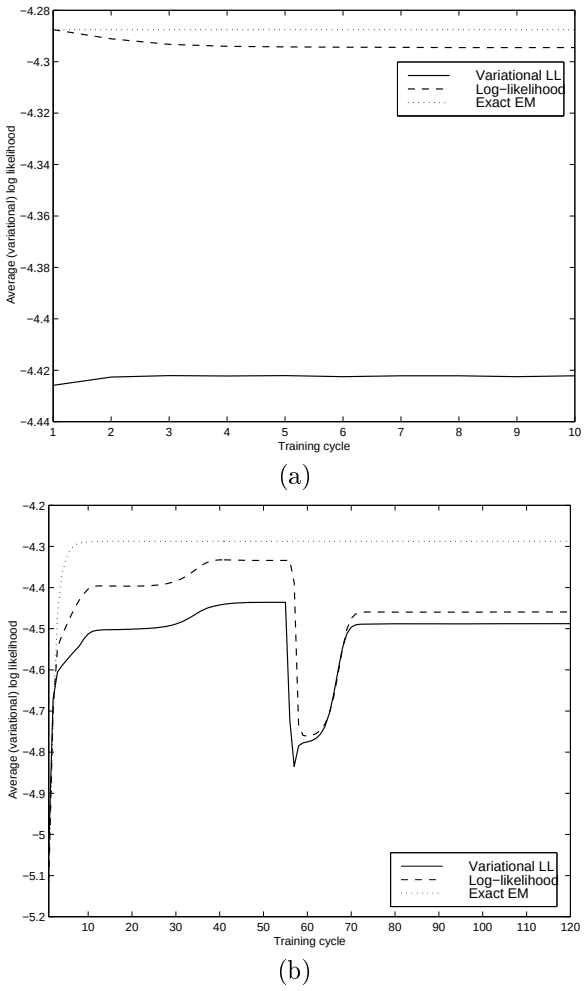

Figure 1. Comparison of mean field with exact EM learning starting at (a) the generative model, and (b) perturbed CPTs of 0.7 on the diagonal.

Experimental runs were performed with both exact EM and mean field EM for two distinct starting positions. The first was at the parameters of the generative model and designed to test the stability of the algorithms. The second was with CPTs of 0.7 on the diagonal, some way from the generative model, to assess whether the generative model parameters could be learned. Each run was performed for a maximum of 300 iterations, though in practice all had converged long before this time. The results are plotted in Figure 1 showing the learning curves prior to convergence, where the variational log-likelihood and true log-likelihood of the mean field EM approach are given by the continuous and dashed lines respectively, and the dotted line is the log-likelihood for exact EM.

From Figure 1(a) it can be seen that the exact EM run starting at the generative model does not deviate. This is to be expected as exact EM is working on the true posterior and will not be able to find a higher likelihood solution than this given that we are effectively using an infinite data set so the posterior is exact. For mean field EM the log-likelihood decreases slightly before stabilising. This would appear worrisome until it is noted that with variational methods we are only approximating the true posterior and though it was shown in [2] that mean field performed rather well, the fact that it uses a fully factorised distribution whereas the true joint distribution is definitely not fully factorisable makes it far from perfect. Thus it is not possible to find a $Q \equiv P$ and reduce the KL-divergence to zero so the M-step is then optimising with respect to a slightly different distribution to the true posterior. The VLL which it is maximising (Equation (4)) does increase as expected.

Examination of the final learned CPTs of mean field EM shows them to have remained fairly close with the worst probability only differing by 0.0334, and most being significantly less.

Starting perturbed from the generative model can be seen in Figure 1(b) to produce quite interesting results. Exact EM again performs as expected, monotonically increasing the log-likelihood until by training cycle 26 it has recovered the generative model parameters exactly and can do no better. Ordinarily we would not necessarily expect EM to do so well, except in the limit of an infinite data set as we have here. Mean field EM starts off well, but on cycle 57 we see a dramatic fall in the variational log-likelihood. For exact EM, the log-likelihood is guaranteed not to decrease on each iteration, but we have no such assurance for mean field EM. A close investigation of the CPTs around this point show that at cycle 55 the prior on the root has a probability of $P(White) = 0.46$, so favouring black states and of the 256 images the means found by mean field at the root node prefer to be white for only 81 patterns. In the next step $P(White) = 0.54$ and for 142 of the patterns the root node prefers being white, but the slight perturbation off the uniform prior $P(White) = P(Black) = 0.5$ pushes mean field

from its unstable equilibrium of not favouring any particular state and in the next cycle 255/256 patterns prefer to have a *white* root node, an example of spontaneous symmetry breaking well known in mean field.

After 120 iterations the resultant prior has a $P(White) = 0.91$, and at the lower levels the CPT entry for $P(X_i = Black|Pa_i = Black)$ is forced to 1 to try to offset this. Prior to cycle 57 the CPTs were moving towards that of the generative model. Clearly then it is possible to over-train using mean field EM even on an infinite data set, and spotting and stopping training prior to the point of spontaneous symmetry breaking could be the answer.

4.2 Learning in the disconnecting tree

Introducing disconnections to the fixed architecture produces a more interesting model, while folding them into the CPTs as described in Section 3 allows the 2^{n-1} configurations to be represented in a single structure. This makes it tractable to compare mean field with exact EM, as was done with the fixed architecture model.

We use the same generative model as in the previous Section with CPTs of 0.9 on the diagonal and uniform node prior. The disconnection probabilities P_d considered will be 0.1, and 0.5. Substituting these parameters into Equation (6) thus gives an effective CPT θ_{eff} for the folded-in model

$$\theta_{eff} \quad = \quad (1 - P_d) \begin{pmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{pmatrix} + P_d \begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} 0.5 & 0.5 \end{pmatrix}.$$

Equations (7)–(9) can then be used to determine the set of parameters where all CPT degeneracy is absorbed into the disconnections. This produces $\hat{\theta} = I_2$, the identity matrix, and $\hat{\boldsymbol{\pi}} = (0.5, 0.5)$. The disconnection probabilities P_d of 0.1 and 0.5 produce $\hat{P}_d$ of 0.28 and 0.6 respectively.

Runs were made for mean field keeping the prior and CPTs distinct[1] (Standard MF EM), mean field with CPTs folded in (CPT_{eff} MF EM) and exact EM with folded in CPTs (Exact EM), for the two disconnection probabilities. Their learning curves are compared in Figure 2.

Standard mean field EM performs the worst in each case. This is perhaps not surprising as by keeping the CPTs and priors distinct gives more degrees of freedom. Folding in the CPTs greatly improves the mean field performance, though exact EM is still noticeably better.

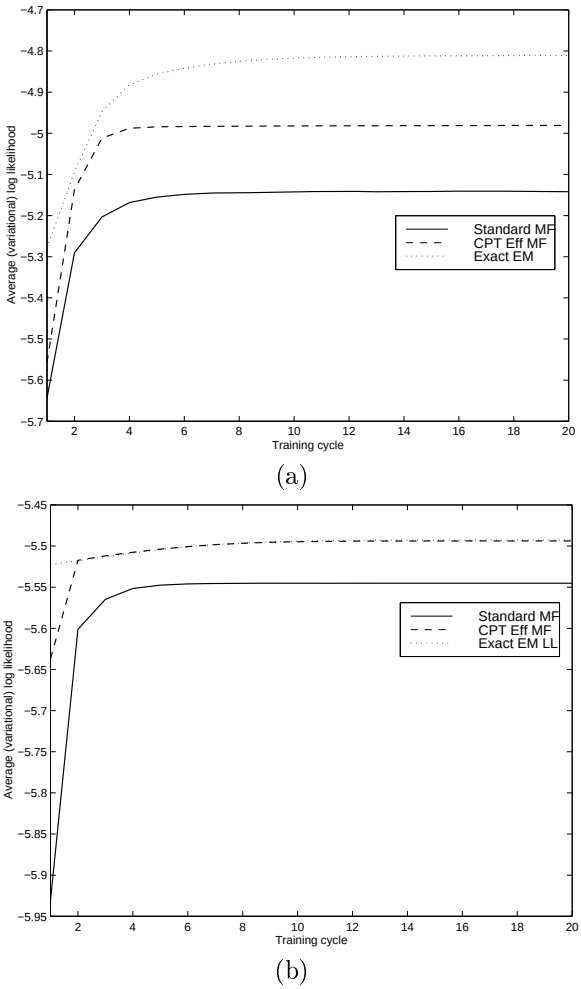

Figure 2. Learning of the CPTs from a start CPT of 0.7, for generative model disconnection probabilities $\hat{P}_d$ of (a) 0.28, and (b) 0.6.

An examination of the effective disconnection probabilities found after 40 training cycles is interesting. They were extracted from the final CPTs using Equation (7) and are given in the table for levels 2 to the leaves at level 4, for each run performed[2]. The prior and CPT associated with each was as for the generative model.

From the table it can be seen that mean field tends to choose to make all the upper levels independent by preferring to disconnect and the structure of the

[1] Unlike the folded-in case we are required to use the $Q(Z)$ distribution in mean field. The disconnection probabilities were also fixed to avoid the over-parameterisation problem.

[2] The top level contains only the root node which is permanently disconnected.

	Target $\hat{P}_d = 0.28$	
Level	CPT_{eff} MF	Exact EM
2	1.000	0.281
3	0.991	0.280
4	0.28	0.28

	Target $\hat{P}_d = 0.6$	
Level	CPT_{eff} MF	Exact EM
2	1.0	0.797
3	1.0	0.734
4	0.6	0.590

Table 1. Comparison of learned disconnection probabilities from exact and mean field EM approaches for a start CPT of 0.7, with generative model disconnection probabilities $\hat{P}_d$ of (a) 0.28 and (b) 0.6.

CPTs learned is degenerate. In the lowest level (closest to the data) it recovers the generative model parameters exactly. This type of behaviour is not really surprising considering that mean field uses a factorised approximation. Exact EM nearly finds the generative model for low disconnection probabilities, but for high disconnections it struggles on the higher levels, getting progressively worse the further from the data the parameters are.

5. Discussion

We have seen that an EM style learning algorithm based upon mean field performs encouragingly in a comparison with exact EM in fixed architecture trees, and shows good potential for use in larger structures where exact EM becomes intractable. Using small tractable models has enabled us to make a thorough comparison between the two approaches and given valuable insights into the capabilities of mean field EM learning invaluable for future work. Spontaneous symmetry breaking was seen to be a weakness of mean field which can affect learning, but with careful monitoring of training error can be avoided.

In the disconnecting tree mean field EM finds the generative model parameters at the level nearest the data, but higher levels become degenerate. It means that mean field has collapsed down the hierarchical model into a single layer as it can still obtain good log-likelihoods for the toy dataset considered. It was noted in [2] that mean field driven by its factorised approximation has a tendency to do this, but for more complex datasets it made use of higher levels. A more structured variational approach such as in [7] could improve this, and is an active area of interest.

Armed with the insights gained by comparing a variational learning approach with an exact method, and given the promise it has shown we are currently extending the work to larger models for real images and moving on to allowing nodes to choose their own parent with probabilities that we also hope to learn by the same method. This will complete an implementation of learning in the full *Dynamic Tree* model which it is hoped will make it tractable on real world images.

Acknowledgements

NJA is supported by an EPSRC research studentship. The work of AS and CW is supported through EPSRC grant GR/L78161 *Probabilistic Models for Sequences.*

References

[1] N. J. Adams. *Dynamic Trees: A Hierarchical Probabilistic Approach to Image Modelling.* PhD thesis, Institute for Adaptive and Neural Computation, University of Edinburgh, 5 Forrest Hill, Edinburgh, UK, 2001. Forthcoming.

[2] N. J. Adams, A. J. Storkey, Z. Ghahramani, and C. K. I. Williams. MFDTs: Mean Field Dynamic Trees. In A. Sanfeliu, J. J. Villanueva, A. Vanrell, R. Alquézar, T. Huang, and J. Serra, editors, *Proceedings of 15th International Conference Pattern Recognition*, volume 3, *Image speech and Signal Processing*, pages 151–154. IEEE Computer Society, September 2000.

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

[4] X. Feng and C. K. I. Williams. Training Bayesian Networks for Image Segmentation. In *Proceedings of SPIE*, volume 3457, July 1998.

[5] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An Introduction to Variational Methods For Graphical Models. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 105–161. Kluwer Academic Publishers, 1998.

[6] J. Pearl. *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufman Publishers Inc., San Francisco, USA, 1988.

[7] A. J. Storkey. Dynamic Trees: A Structured Variational Approach Giving Efficient Propagation Rules. In *Uncertainty in Artificial Intelligence (UAI2000).* 2000.

[8] C. K. I. Williams and N. J. Adams. DTs: Dynamic Trees. In M. J. Kearns, S. A. Solla, and D. A. Cohn, editors, *Advances in Neural Information Processing Systems 11*, pages 634–640. MIT Press, 1999.