SDTs: Sparse Dynamic Trees

Nicholas J. Adams Christopher K. I. Williams
Institute for Adaptive and Neural Computation
Division of Informatics, University of Edinburgh
5 Forrest Hill, Edinburgh, EH1 2QL, UK
http://www.anc.ed.ac.uk/
nicka@dai.ed.ac.uk ckiw@dai.ed.ac.uk

Abstract

We introduce a class of image models which we call *sparse* dynamic trees or SDTs. These extend our previous work on Dynamic Trees Williams and Adams (1999) by introducing a top-down generative prior for the tree structures, which leads to a sparse use of nodes in the network (*cf* recent work on sparse coding). We present results showing the properties of these networks in action on 1-d patterns.

1 Introduction

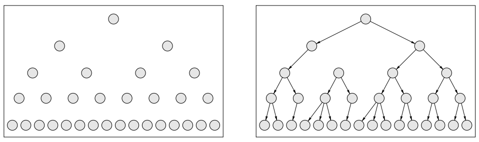
In this paper we introduce a class of image models which we call *sparse* dynamic trees or SDTs. In previous work Williams and Adams (1999) we described Dynamic Tree (DT) networks, which specify a prior over a large number of trees, each one of which is a tree-structured belief net (TSBN). Our intuition as to why DTs may be useful image models is based on the idea that most pixels in an image are derived from a single object. We think of an object as being described by a root of a tree, with the scale of the object being determined by the level in the tree at which the root occurs.

In DTs we considered a layered network structure, which was tapered in the sense that there were fewer nodes in higher levels of the tree (as shown in Figure 1). We found that DTs experienced some problems with respect to invariance of the network to translations of an input pattern. In SDTs we have reformulated the prior to allow equal numbers of nodes at each level, but we retain a prior over trees by using a top-down Markovian generative mechanism. Many nodes are now inactive, hence the name *Sparse* Dynamic Trees. This relates to recent work on sparse coding, see, e.g. Olshausen and Field (1996). Our new prior allows us to retain advantages of DTs such as the hierarchical structure of the model, while reducing the problems. One use for dynamic tree models is as prior models over labellings for image segmentation problems.

Section 2 of the paper reviews DTs and gives the theory of SDTs. The experimental results are described in section 3.

2 Theory

There are two key components to a dynamic tree network. Firstly its architecture is specified by means of a prior, and then the nodes and conditional probability tables (CPTs) between their links are defined. The former is the more interesting and less well-explored issue which we focus on here. We first of all describe a model in which all units are active (the *full-time-node-employment* prior), and then move on to discuss the main focus of this paper, the *part-time-node-employment* prior which gives rise to Sparse Dynamic Trees (SDTs).



(a) (b)

Figure 1: (a) "Naked" nodes, (b) a sample from the *full-time-node-employment* prior.

2.1 Dynamic Trees: the *full-time-node-employment* prior

We start with an arrangement of nodes organised as a number of layers, such as in Figure 1(a). To construct the tree structure we allow any child in a particular layer to connect to a parent in the layer above, but also introduce an extra null parent node in each layer such that if a child chooses this it becomes the root of a new tree. This gives rise to a forest of one or more trees. Figure 1(b) shows such a structure generated by this method.

Formally we define $\boldsymbol{z}_i$ as the indicator vector showing the parent to which node i belongs, and specify the tree structure as a matrix Z whose columns are the individual $\boldsymbol{z}_i$ vectors. The prior is thus given by $P(Z)$ and can be specified in a number of ways.

In our earlier work Williams and Adams (1999) we chose the tapered node arrangement of the balanced tree-structured belief network and set $P(Z) = \prod_i P(\boldsymbol{z}_i)$. This choice allowed direct comparison with the balanced tree-structured belief network, and is motivated by the balanced tree structure being seen as a good initial connection strategy for the nodes. We set the $P(\boldsymbol{z}_i)$'s to favour a node's "natural" parent - the parent it would have in a balanced tree structure - with connections to other parents being permitted at a cost when this better explains the evidence. Each node in the parent layer is thus assigned an "affinity" for each child node, with the "natural" parent having the highest affinity. Let a_k denote the affinity for a node k in the parent layer then we can choose $P(\boldsymbol{z}_i = \boldsymbol{e}_k) = \exp^{\beta a_k} / \sum_{j \in Pa_i} \exp^{\beta a_j}$, where β is some positive constant and $\boldsymbol{e}_k$ is the unit vector with a 1 in position k. Note that the "null" parent is considered in the same way as any other node and is included in the summation above. This is the *full-time-node-employment* prior and was so named as all the nodes participate in the creation of the tree structure to some degree.

The results in Williams and Adams (1999) showed this gave higher translation invariance over the balanced tree-structured belief network. However, it was noted that in some instances the node arrangement had too few nodes on higher levels, restricting the types of structure which could be created. An example of such a tree is shown in Figure 2(f) where the single white pixel is cut off from the rest of the white background. This problem was seen to increase in the 2d-experiments also reported in Williams and Adams (1999).

One solution to this dilemma may be to avoid modelling the background and concentrate on constructing trees only for the foreground objects. We plan to investigate this option in future work. Here we choose instead to relax the number of nodes at higher levels and allow the model to decide which nodes to use, and which nodes to leave unused and inactive. This gives rise to the *Sparse Dynamic Tree*.

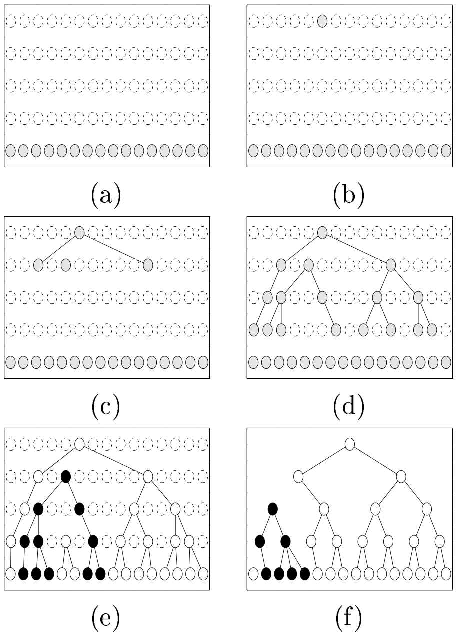


Figure 2: (a) Inactive nodes, (b) prior turns on a top level node, (c) connections are made to next level, (d) process is repeated until penultimate layer (e) bottom-up approach ensures all leaves get connected and (f) an example of too few nodes in the *full-time-node-employment* prior.

2.2 Sparse Dynamic Trees: the *part-time-node-employment* prior

Increasing the number of nodes in each layer and requiring many of them to remain inactive means that we cannot continue with the prior described above where each unit chooses its parent independently, as this would tend to lead to chains or isolated links rather than tree structures. We have solved this problem by adopting a top-down approach to tree generation, with the probability of links occurring at a given layer being conditional on the activations of nodes in the layer above; we can thus grow forests of trees from the roots to the leaves. (Montanvert *et al* (1991), and references therein describe a number of approaches to tree generation, using both top-down and bottom-up approaches.)

We start with an arrangement of nodes such as that in Figure 2(a). Initially all the nodes except those at the leaf (image) level are inactive, denoted by a broken line in the figure. A prior for activation is set for the nodes of the top layer and this is used to probabilistically activate some of the nodes and turn them on, as in Figure 2(b). Typically we wish only a single node to be activated at this level to form a master root for the image, so the prior probability needs to be set so as to encourage this.

After activating some (or none) of the top level nodes we consider the second layer. As before each child in turn probabilistically chooses a parent out of its set of potential parents, and if the parent chosen is active it connects to it and is turned on (Figure 2(c)). Choosing an inactive parent results in the node remaining inactive and disconnected. The "null" parent in each level is considered to be always active so that connecting to it results in the child becoming an active root. The affinity profile for each child at higher levels in the tree will typically be bimodal, indicating that we expect some spreading out of tree to occur, as shown in Figure 2(c) (see section 3 for details).

This procedure is repeated on successive layers down to the penultimate level (see Figure 2(d)). In this top-down approach the probability of node arrangement at level i is conditional only on the state of the nodes at level $i-1$. Denoting by Z_i the connection matrix of the ith layer and letting the leaf layer have index L, then we have the Markovian scheme $P(Z_0, Z_1, \ldots, Z_{L-1}) = P(Z_0)\prod_{i=1}^{L-1} P(Z_i|Z_{i-1})$.

The top-down method presents a problem at the leaf level of the network as it does not guarantee that all the leaves will be active. Clearly we cannot allow inactive leaves as they correspond to pixels in the image, so we modify the top-down strategy with a bottom-up step. To do this we simply allow each leaf node to choose its parent in the penultimate layer independently, and if any parents were not previously activated, they now become new roots. The fifth node from the left on the penultimate level of the tree in Figure 2(e) is an example of such a new root. Using this combined top-down and bottom-up scheme we can specify fully a prior over trees.

We have termed this the *part-time-node-employment* prior as the nodes can decide whether or not they wish to be employed in the tree structure or remain redundant and inactive.

When calculating $P(Z)$, we need to take into account the fact that if a node in layers 2 to $L-1$ is inactive, it could have become so by connecting to *any* inactive parent in the layer above. Hence the probability of a node becoming inactive is computed by summing the probabilities of connecting to all of the off parents.

Given that we have specified a prior over network architectures and generated a "forest" structure Z we must now translate this into a TSBN. The nodes in a TSBN take on discrete values and can be considered as C-class multinomial random variables. On each of the links there is a conditional probability table (CPT) M_k, which defines the probability of changing state during a transition between nodes, and for the root node there is a prior π. DTs and SDTs use this same methodology, however we need to associate a prior π_k with each unit as all have the potential to become a root. Currently these parameters are defined on a level by level basis. Given a particular Z matrix the probability of a particular instantiation of all of the random variables is simply the product of the probabilities of all of the trees, where the appropriate root probabilities and CPTs are picked up from the π_ls and M_ls. Figure 2(e) shows a sample generated from our tree structure.

2.3 Inference in SDTs

Inference in dynamic tree models is concerned with obtaining the posterior marginal $P(Z|X_v)$ of the network with structure Z, where X_v denotes the visible units (the image instantiated on the lowest level units). We need not consider the states of the hidden units X_h in the full posterior as we can obtain $P(X_h|X_v, Z)$ from standard techniques for TSBNs.

Our interest is in finding the maximum a posteriori (MAP) state from the posterior $P(Z|X_v) \propto P(Z)P(X_v|Z)$ and we use simulated annealing to explore this distribution. To do this we need to define a mechanism for proposing changes to the network structure which the annealer can decide to reject or accept in the normal way. As the connections of the network are conditioned on the states of the nodes in the adjacent layer and not independent as in the case of the *full-time-node-employment* prior, we cannot simply alter one link without consideration of the neighbouring units. Nodes can also be active and inactive and we need allow dormant nodes to be activated as well as redundant ones to be put to sleep. We use the move set summarized below to explore Z space:

Randomly choose a node $i \in$ {all nodes}
If Node is currently *on*
    If Node is **not** *a leaf*
        De-activate Node i;
        Break link with parent;
        Probabilistically choose new parent
            for all daughters connected to i;
    Else
        Choose another parent for the leaf;

Else
    Activate Node i;
    Probabilistically choose a parent for i;
    Ask all active daughters if they prefer
        connecting to i. Move link if *yes*;

The move set as described above is clearly only one of a number of possible strategies, and an interesting alternative includes moving or combining nodes within a layer while preserving their links (as far as possible). Consideration of the move strategy is an important issue which directly influences the coverage of the search space made during annealing and thus the quality of solutions found, and is an area of ongoing research.

An alternative to sampling from the posterior $P(Z, X_h|X_v)$ is to use approximate inference. One possibility is to use a mean-field-type approximation to the posterior of the form $Q_Z(Z)Q_h(X_h)$ (Zoubin Ghahramani, personal communication, 1998).

2.4 Comparing SDTs to other image models

The origins of our work on SDTs can be traced to two main sources. The first of these concerns the use of hierarchical tree-structured belief networks (TSBNs) as models of images, notably in the work of Bouman and Shapiro (1994) and Willsky and coworkers, see e.g. Luettgen and Willsky (1995). However, these models have a rigid architecture (usually of quad-tree type) which is unresponsive to image content. In Williams and Adams (1999) we relaxed this by creating dynamic trees (DTs) as described in section 2.1.

The notion of *sparseness* in SDTs connects to the second source for this work, sparse coding. There has been considerable interest in the idea of sparse coding, e.g. Olshausen and Field (1996). In this work an image is generated as a linear combination of basis functions $\boldsymbol{I} = \sum_i c_i \boldsymbol{\phi}_i$, where the c_i's are the coefficients on the basis functions $\{\boldsymbol{\phi}_i\}$. There is a prior on the coefficients which encourages relatively few of them to be on, but this prior is independent for each coefficient. Our work extends this by using a layered prior which means that correlations are induced in units in a layer from patterns of activation in the layer above.

There are many other related papers. For example de Bonet and Viola (1998) use a tree-structured model for wavelet coefficients, but they do not have an overcomplete basis and do not consider sparsity. Sallans *et al.* (1998) consider dynamic networks in which the structure can switch, but allow general DAGs rather than just trees; we believe that it is useful to find out what can be achieved using TSBNs. We also note the similarity between posterior tree configurations and parses generated by a context-free grammar[1].

Markov Random Field (MRF) models are also popular image models; however, one of their main limitations is that inference in a MRF is NP-hard. Also, they lack an hierarchical structure. On the other hand,

[1] CFGs have a $O(n^3)$ algorithm to determine the MAP parse; however, this algorithm depends crucially on the one-dimensional ordering of the inputs. We believe that the possibility of crossed links in the (S)DT architecture means that this kind of algorithm is not applicable to the DT case. Also, dynamic tree models can be applied to 2-d images, where the $O(n^3)$ algorithm is not applicable.

stationarity of the process they define can be easily ensured, which is not the case for fixed-structure TSBNs.

3 Experiments

The experiments have been designed to compare the performance of the *sparse* dynamic trees with that of the *full-time-employment* prior of the DT, and the balanced TSBN.

We consider a stimulus of 4 and 5 pixel bars in a 1d image of 16 pixels as used in Williams and Adams (1999). The corresponding balanced TSBN for such an image is thus a 5-layer binary tree and we choose the node organisation of the *part-time-employment* prior such that the TSBN fits into this arrangement. This still gives a wide scope and we select the arrangement of Figure 2(a) with a uniform number of 15 nodes in each layer above the leaves so that the number of nodes does not limit the type of structures we can create.

Each node in the tree is a binary variable taking on values of either white or black. We set the M_l's to be equal for all layers l, with values of 0.99 on the diagonal and 0.01 off diagonal, and the π_l's $= (0.75, 0.25)$ on all layers. In the π_l vectors the first element is the prior of the node being white so we are favouring white roots by a factor of 3 to 1. This is determined by the statistics of the images under consideration, which in this instance have approximately three times as many white pixels as black. The

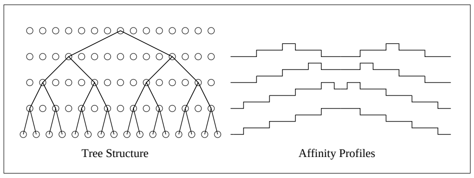


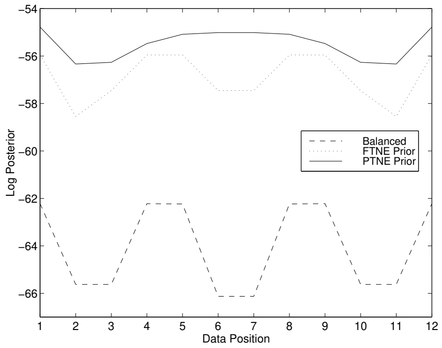
Figure 3: Affinity profile to encourage formation of tree structures.

affinities a_k and βs are relevant to all the layers except the very top as any active node at this level only has the option of connecting to a null parent. For the sparse node arrangement we no longer have the notion of a "natural" parent as used by the *full-time-node-employment* prior, so the affinity profiles must be designed to encourage formation of tree structures. To get coherent trees such as in Figure 3, nodes at higher levels must connect to parents further away than those in the lower layers, so each layer is treated independently. We adopt the notation of affinity profiles and replace the individual affinities a_k that a node has for a particular parent with templates encompassing all nodes in the parent layer. These templates are set so that a node favours parents at or near the desired distance indicated by the peaks in the profiles of Figure 3. We use a reference affinity of 1 for the most favoured node(s), decreasing by 1 for pairs of nodes further from these points, eg. the 4th level affinity profile is of the form $(\ldots -1\ -1\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ -1\ -1 \ldots)$.

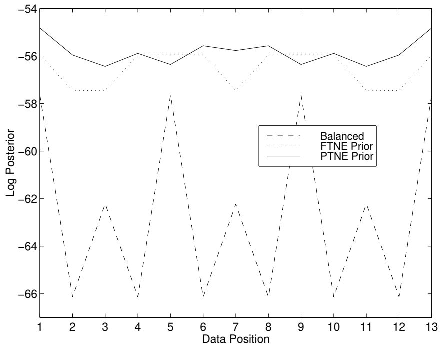
Within this framework we similarly set the affinities for the "null" parents for each of the lower layers. We choose, $a_{null} = 0.0$ for all levels, and set the β's to 1.5. The prior for activating the nodes on the top level was set to 1/7.

We illustrate the effects of translation by applying the black bar stimuli at successive positions in the image and find the unnormalised log posterior $\log P(Z|X_v) \propto \log P(Z) + \log P(X_v|Z)$ for the particular Z configurations. Our interest is in the maximum a posteriori (MAP) Z for each image and we use simulated annealing combined with our move set to search for these. We also calculate the posterior of the balanced TSBN architecture and compare it with the MAP configurations in Figures 4(a) and (b). The affinity templates and other model parameters have been set such that the highest prior probability $P(Z)$ will be that of the balanced TSBN arrangement. In the figures the x-axis denotes the position of the left hand end of the bar, and the y-axis shows the log posterior probability.

We see from these figures that plot of the posterior of the *part-time-node-employment* prior is almost flat indicating very little translation variation compared to the heavily peaked and troughed posterior of the balanced TSBN. Notice also from the figures that the *part-time-node-employment* prior achieves a flatter posterior than the *full-time-node-employment* prior. This is explained by extra nodes being available, enabling similar structures to be formed at each of the different bar positions, as opposed to the shortage of nodes at some po-



(a) 5 black nodes



(b) 4 black nodes

Figure 4: Plots of the unnormalised log posterior vs position of the input pattern for (a) the 5-black-nodes pattern and (b) 4-black-nodes pattern.

sitions encountered in the posterior of the *full-time-node-employment* prior.

Some exploration of the parameter space has yielded even flatter posterior plots for the *sparse* dynamic tree than the ones shown in Figure 4, however setting the parameters is not trivial. It should be possible to learn the model parameters using an approach such as EM in much the same way that the CPTs and priors are learned for fixed architecture networks, and this is an area of ongoing research.

4 Discussion

Above we have demonstrated that SDT models have greater translation invariance than the balanced TSBN model, and that simulated annealing is successful at finding trees that have high posterior probability. We are keen to extend the work to 2-d, and there are several other directions to explore including the learning of parameters in the networks (e.g. using EM), and the introduction of additional information at the nodes; for example one might use real-valued variables in addition to the multinomial variables considered above. These additional variables might be used to encode information such as that concerning the instantiation parameters of objects.

Acknowledgements

We thank Zoubin Ghahramani, Geoff Hinton and Peter Dayan for helpful conversations and Lewis Griffin for drawing our attention to the work of Montanvert *et al.* NJA is supported by an EPSRC research studentship, and the work of CW is partially supported by EPSRC grant GR/L03088, *Combining Spatially Distributed Predictions From Neural Networks.*

References

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

de Bonet, J. S. and P. A. Viola (1998). A Non-Parametric Multi-Scale Statistical Model for Natural Images. In M. I. Jordan, M. J. Kearns, and S. A. Solla (Eds.), *Advances in Neural Information Processing Systems 10*. Cambridge, MA: MIT Press.

Luettgen, M. R. and A. S. Willsky (1995). Likelihood Calculation for a Class of Multiscale Stocahstic Models, with Application to Texture Discrimination. *IEEE Trans. Image Processing* **4(2)**, 194–207.

Montanvert, A., P. Meer, and A. Rosenfeld (1991). Hierarchical Image Analysis Using Irregular Tesselations. *IEEE Trans. Pattern Analysis and Machine Intelligence* **13(4)**, 307–316.

Olshausen, B. A. and D. J. Field (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* **381**, 607–609.

Sallans, B., G. E. Hinton, and Z. Ghahramani (1998). A Hierarchical Community of Experts. In C. M. Bishop (Ed.), *Neural Networks and Machine Learning*. Springer.

Williams, C. K. I. and N. J. Adams (1999). DTs: Dynamic Trees. In M. J. Kearns, S. A. Solla, and D. A. Cohn (Eds.), *Advances in Neural Information Processing Systems 11*. MIT Press.