Fast Tree-based Learning and Inference in Markov Random Fields and Applications

Tran The Truyen [†], Dinh Q. Phung [†], Hung H. Bui [‡], Svetha Venkatesh [†]
[†]Department of Computing, Curtin University of Technology
GPO Box U 1987, Perth, WA, Australia
{trantt2,phungquo,svetha}@cs.curtin.edu.au
[‡]Artificial Intelligence Center, SRI International
333 Ravenswood Ave, Menlo Park, CA 94025, USA
bui@ai.sri.com

Abstract

Inference and learning for general structure MRFs are usually intractable problems in computer vision. In this paper, we exploit a set of tree-based methods to efficiently address this problem and evaluate these methods against some current state-of-the-art approaches in three problems: scene segmentation, stereo matching and image denoising. Our method takes advantage of the tractability of tree-structures embedded in MRFs to derive a tractable lower bound of the true likelihood, propose the use of tree-based pseudo-likelihood (PL) for parameter estimation, and the use of tree-based ICM (T-ICM) for MAP assignment. Unlike loopy belief propagation, our method is guaranteed to converge and it does so with limited memory required to store the messages. Further, unlike Graph-Cuts, our T-ICM can be applied with arbitrary cost functions such as those estimated during learning.

1. Introduction

Markov random fields (MRFs) [1] are a natural representation for vision labeling problems. Given one or more images, MRFs offer a principled way to specify prior knowledge, to learn parameters and to infer the optimal labeling, usually in the form of *maximum a posteriori* (MAP). For many applications, estimating parameters of MRFs from labeled data is desirable for many reasons. First, labeled data is often easier to acquire than expert knowledge, especially when specifying many parameters for complex problems (such as weighting many dozens of filter responses). Second, if experts are available, little can be quantified about how well a manually specified MRF matches the problem. Unfortunately, both MRFs estimation and MAP assignment are generally intractable for vision problems. As a result, many approximate methods have been proposed.

In parameter estimation, two strategies are often used: one is to use some tractable criteria as alternatives for the intractable maximum likelihood; the other is to approximately compute the true likelihood and related quantities. Pseudo-likelihood [2] (PL) falls into the first group. This is a very efficient local method which may be suboptimal with limited data. A representative method for the second group is to use Pearl's loopy sum-product algorithm [13] and more recent variants [16, 17] as approximate inference engines. The sum-product based methods, which can potentially perform better than the PL also have important drawbacks. First, they are not guaranteed to converge, and in practice, if they do converge to some approximate solution, the convergence can be slow due to the iterative nature of message passing. Second, the sum-product methods consume a fair amount of memory to store messages passing through every edge of the graph.

Likewise, MAP assignment is an intractable discrete optimisation problem. Efficient approximations to date include the iterated conditional mode (ICM) [3], Pearl's loopy max-product algorithm [13] and variants [18], and the more recent Graph-Cuts [4]. The ICM is a fast but weak local search method that may be trapped in poor local optima. The max-product, on the other hand, can often find good labeling that is close to optimal. However, as with message passing algorithms on general graphs, the max-product is not guaranteed to converge, especially in MRFs with strong interaction between sites, and it requires significant memory to store all messages for large models. Graph-Cuts have been shown to be very successful on certain classes of vision problems [4, 15]. They are, nevertheless, designed with specific cost functions in mind (i.e. *metric* and *semi-metric*), and therefore inapplicable for generic cost functions such as those resulting from learning.

These shortcomings in learning and inference motivate us to seek for alternatives that are both efficient in speed and memory, especially with some guarantee in the convergence while maintaining performance comparable with the state-of-the-art. Exploiting the fact that Markovian trees are efficient, we propose the use of *superimposed trees* and *conditional trees*, which are inherently embedded in the original MRFs. Superimposed trees are those which when put on top of each other cover the whole network. Conditional trees are those, given the values of the neighbouring variables, behave like a tree.

Using the superimposed trees, we formulate a tractable lower bound for the likelihood and propose its use for learning. Another alternative based on conditional trees is the tree-based pseudo-likelihood (T-PL) to be used as an extension of the PL. The tree-based likelihood lower bound and the T-PL can be computed efficiently using just one pass over the training data whilst consuming very limited memory. When appropriately exploited, those trees will help to decompose a complex problem so that each tree can well fit certain aspects of the data.

For MAP assignment, we use the conditional trees to extend the ICM into a stronger local method called tree-based ICM (T-ICM). Then using the iterated local search (ILS) framework [12], we design global jumps by exploiting the structure of the MRF to escape from the local optima. Like the Graph-Cuts as a local search, the T-ICM is guaranteed to converged to local minima, which are generally stronger than those found by the ICM. But unlike the Graph-Cuts, the T-ICM can be applied to any generic cost functions, especially those estimated during learning.

We evaluate our tree-based learning and inference methods against Pearl's message passing methods and Graph-Cuts when appropriate on three benchmark problems: scene segmentation, stereo correspondence and image denoising. We empirically show that the tree-based likelihood lower bound and the tree-based pseudo-likelihood when coupled with tree-based ICM estimates attains good performance, whilst requiring much less training time than the sum-product based learning when coupled with max-product MAP estimation. We show in both Ising simulation and image restoration that the T-ICM can be competitive with the max-product algorithm. In the stereo matching, the T-ICM, when initialised from scanline optimisation yields fast processing with a small constant overhead.

To summarise, our main contributions are the proposal and evaluation of fast and lightweight tree-based learning and inference methods in MRFs. Our choice of trees on $W \times H$ images requires only $\mathcal{O}(2D)$ memory where $D = \max\{W, H\}$ and two passes over all sites in the MRFs per iteration. These are much more economical than $\mathcal{O}(4WH)$ memory and many passes needed by belief-propagation methods.

This paper is organised as follows. The next section describes the common setting for modeling and inference in MRFs. Sections 3 and 4 present the tree superimposition and conditional tree frameworks, respectively. Experimental results are reported in Section 5 followed by conclusions and future work.

2. Preliminaries

A MRF specifies a random field $x = \{x_i\}$ over a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the collection of sites $\{i\}$, $\mathcal{E}$ is the collection of edges $\{(i,j)\}$ between sites, and $x_i \in L$ represents labels at site i. Assuming pairwise interaction between connected sites, we want to infer the conditional probability [11] of the labeling x given a set of images y as

$$\begin{aligned} P(x|y) &= \frac{1}{Z(y)}\Phi(x,y) \\ &= \frac{1}{Z(y)}\prod_{i\in\mathcal{V}}\phi(x_i,y)\prod_{(i,j)\in\mathcal{E}}\psi(x_i,x_j) \end{aligned} \quad (1)$$

where $\phi(x_i, y)$ and $\psi(x_i, x_j)$ are often referred as data potential and interaction potential respectively, $Z(y)$ is the normalisation constant.

Our inference on trees is based on Pearl's message passing schemes. Here we present the generic procedures for general graphs, while in trees, we use more specialised versions that require exactly two passes through nodes. In the sum-product scheme, messages are recursively updated as

$$\mu_{j\to i}(x_i|y) \propto \sum_{x_j}\prod_{k\in N(j),k\neq i}\mu_{k\to j}(x_j|y)\phi(x_j,y)\psi(x_i,x_j)$$

where $\mu_{j\to i}(x_i|y)$ denotes the message from node j to i evaluated at x_i, and $N(j)$ is the set of neighbours of node j. Finally, beliefs are computed by

$$b(x_i|y) \propto \prod_{j\in N(i)}\mu_{j\to i}(x_i|y)\phi(x_i,y) \quad (2)$$

where $b(x_i|y)$ are approximation of the true marginals $P(x_i|y)$. The max-product is similar, where the message updates are given as

$$\mu_{j\to i}(x_i|y) \propto \max_{x_j}\prod_{k\in N(j),k\neq i}\mu_{k\to j}(x_j|y)\phi(x_j,y)\psi(x_i,x_j) \quad (3)$$

and the maximal beliefs are computed using the same equation as in (2). The message passing schemes require $\mathcal{O}(2|\mathcal{E}||L|)$ memory to store all the messages, where $|\mathcal{E}|$ is number of edges in the graph $\mathcal{G}$ and $|L|$ is the size of the label set L. The memory will be very demanding for large images (such as those with $H = 1000$ and $W = 1000$, $|L| = 256$; and $|\mathcal{E}| \approx 2HW$).

The MAP estimation in MRF (i.e. $x^{map} = \arg\max_x P(x|y)$) is often recast as energy minimisation, where the energy is defined as

$$\begin{aligned} E(x,y) &= -\log \Phi(x,y) \\ &= \sum_{i\in\mathcal{V}} -\log\phi(x_i,y) + \sum_{(i,j)\in\mathcal{E}} -\log\psi(x_i,x_j) \end{aligned}$$

To approximately minimise this energy (or cost), early methods such as ICM seek fast local optima, where a site with the conditional distribution $P(x_i|N(i))$ is optimised at a time in a step-wise manner. Our T-ICM extends the notion of site in the ICM to a tree, thus arriving a stronger local method. The max-product that works directly on energy is often called 'min-sum', which is essentially the $-\log$ of (3). A more recent max-product variant, appeared in [18], maximises the lower bound of the energy rather than directly minimising the energy.

3. Superimposed Trees

3.1. Graph as a superimposition of trees

This subsection presents an emerging practice to consider the graph as a superimposition of (spanning) trees [16, 18, 17]. Although this is not new idea, we are not aware of previous work with the same insights as those discussed here.

Consider a particular graph representation of the image, for example, the grid where each site may correspond to a pixel. Alternatively we can view this grid as a collection of overlapping spanning trees (e.g. see Figure 1). That is, the trees share all the vertexes $\mathcal{V}$ of the graph but each of them covers only a subset $\mathcal{E}_t$ of edges. Intuitively, we consider each tree represents a particular Markov process, and the original MRF is a result of interaction between these processes. For example, Figure 1d shows the horizontal Markov chains, which can be used to specify the *left-to* and *right-to* relations between objects/labels such as *car* and *person*. Similarly, vertical Markov chains as in Figure 1e can encode the *above* and *below* relations between objects/labels such as *sky* and *water*.

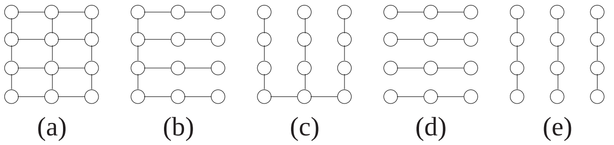

Figure 1. Superimposed trees. (a): original graph, (b,c): spanning trees, and (d,e): another decomposition.

One way to specify the interaction between overlapping trees $\{t\}$ is to define tree-based MRFs in the same way as Equation 1 and to require that the product of the tree potentials is equal to the that of the original MRF

$$P_t(x|y) = \frac{1}{Z_t(y)}\Phi_t(x,y) \qquad (4)$$

$$\Phi(x,y) = \prod_t \Phi_t(x,y) \qquad (5)$$

where $\Phi_t(x,y) = \prod_{i\in\mathcal{V}}\phi_t(x_i,y)\prod_{(i,j)\in\mathcal{E}_t}\psi_t(x_i,x_j)$. For example, using the set of two trees in Figures 1d,e (each tree has three chains) we can specify the tree potentials as

$$\phi_t(x_i,y) = \phi(x_i,y)^{\alpha_t} \text{ and } \psi_t(x_i,x_j) = \psi(x_i,x_j) \qquad (6)$$

where $\sum_t \alpha_t = 1, \alpha_t \geq 0$.

Substituting (5,4) into (1) we have the conditional Products-of-Experts [9] (PoE) form

$$P(x|y) = \frac{\prod_t P_t(x|y)}{\sum_x \prod_t P_t(x|y)} \qquad (7)$$

The PoE framework is attractive because it potentially combines different 'experts', each of which focuses on certain aspects of the problem. For example, in our image labeling case, the horizontal and vertical trees in Figure 1d,e may respectively capture the left/right and above/below relations between objects.

3.2. A lower bound on the likelihood

Observation 1 *Given the definitions (1,4) and the constraint (5), the following holds:*

$$P(x|y) \geq \prod_t P_t(x|y) \qquad (8)$$

Proof: From (7), it is clear that we need to prove

$$\sum_x \prod_t P_t(x|y) \leq 1 \qquad (9)$$

Since the LHS is in the sum-product form, we apply the generalised Hölder's inequality [7, Theorem 11]

$$\sum_x \prod_t P_t(x|y) \leq \prod_t (\sum_x P_t(x|y)^{r_t})^{1/r_t} \qquad (10)$$

where $\sum_t 1/r_t = 1$ and $r_t > 0$. By choosing $r_t = |T|$ where $|T|$ is the number of trees and by making use of the simple inequality $\sum_k z_k^{|T|} \leq (\sum_k z_k)^{|T|}$ for $z_k > 0$ and $|T|$ is an integer, we have

$$\begin{aligned} \sum_x \prod_t P_t(x|y) &\leq \prod_t (\sum_x P_t(x|y)^{|T|})^{1/|T|} \\ &\leq \prod_t (\sum_x P_t(x|y))^{|T|/|T|} = 1 \blacktriangle \end{aligned}$$

As an immediate result of the Observation 1, $\log P(x|y) \geq \sum_t \log P_t(x|y)$. Using this fact, we propose a learning method that explicitly maximises the lower log-bound instead of the original log-likelihood. This is equivalent to training component tree-based models *jointly*. By maximising the joint likelihood, the trees are specifically adapted for the specific aspects for which they are designed. The net effect is that, the learned ensemble of trees is expected to fit the data effectively.

Remark: Substituting the result of (10) into (7) would give us a tractable lower bound, which is clearly tighter than the one in Observation 1. More specifically, let $Q_t(x|y) = P_t(x|y)^{r_t} / \sum_x P_t(x|y)^{r_t}$, we obtain $P(x|y) \geq \prod_t Q_t(x|y)^{1/r_t}$. Another way to prove this bound is to make use of Wainwright's upper bound on the log-partition function $\log Z(y)$ [17] in the exponential family setting. However, it is interesting that this tighter bound did not translate into better final performance in our experiments of scene segmentation, and hence we decided to follow with the simpler solution. Our early experience also indicated that Wainwright's message passing variant [17] of the sum-product does not offer clear advantages over the sum-product when used as underlying inference in learning.

4. Conditional Trees

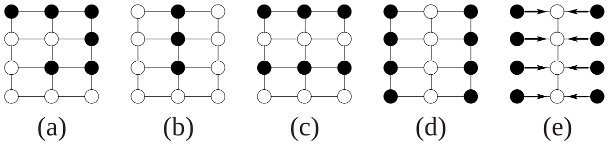

Figure 2. (a-d): conditional trees (connecting empty nodes). Filled nodes are known sites. (e): absorbing the interaction potentials into the data potential. Black nodes are fixed, white nodes are conditional on the black nodes.

Another type of tree can be obtained by fixing some site variables to particular labels (Figure 2a-d). Let $x = (x_t, x_{-t})$ where $x_t \in \mathcal{V}_t \in \mathcal{V}$ is tree-structured and not known, and $x_{-t} = \{x_j | j \in \mathcal{V}, j \notin \mathcal{V}_t\}$ is known. We can rewrite the MRF distribution as

$$P(x|y) = P(x_t|x_{-t}, y)P(x_{-t}|y) \qquad (11)$$

A conditional tree x_t is different from the superimposed trees in that it interacts with the neighbouring sites of the tree t, and x_t is always a subset of x. Due to Markov property, $P(x_t|x_{-t}, y) = P(x_t|N(t), y)$ where $N(t)$ denotes the set of neighbour sites of the tree t. Given this fact and fixed neighbour labels, the neighbourhood interacting potentials can be absorbed into the conditional tree as follows (Figure 2e)

$$\phi_t(x_i, y) = \phi(x_i, y) \prod_{j|j \notin \mathcal{V}_t, (i,j) \in \mathcal{E}} \psi(x_i, x_j) \qquad (12)$$

Inference can then be carried out using these modified potentials as in normal trees.

4.1. Tree-based pseudo-likelihood

Given the efficiency of conditional trees, it is quite straightforward to introduce the concept of tree-based pseudo-likelihood (T-PL) $\prod_t P(x_t|x_{-t}, y)$ as an alternative criterion to maximum likelihood. The choice of trees $\{t\}$ is problem dependent. Intuitively, we can think of x_t as a mega-site but we treat it just like an ordinary site. Since at the moment, we do not have any theoretical guarantee on asymptotic consistency of the T-PL, we rely on domain knowledge and careful regularisation practice to control overestimating interaction potentials.

4.2. Conditional trees for strong local search

We want to find the MAP assignment in a step-wise manner, that is to optimise one conditional tree (with conditional distribution $P(x_t|x_{-t}, y)$) at a time. For a fixed configuration of x_{-t}, we have

$$\min_x E(x, y) \leq \min_{x_t} E(x_t, x_{-t}, y) \qquad (13)$$

Observation 2 *Let* $x_t^* = \arg\min_{x_t} E(x_t, x_{-t}, y)$, *if* $x_{-t} \in x^{map}$ *then* $x_t^* \in x^{map}$.

Proof: Assume that $x_t^* \notin x^{map}$, so there must exist $x_t' \in x^{map}$ that $x_t^* \neq x_t'$ and $E(x_t', x_{-t}, y) > E(x_t^*, x_{-t}, y)$, or equivalently $E(x^{map}, y) > E(x_t^*, x_{-t}, y)$, which contradicts with (13).

Locally minimising the energy over x_t is straightforward with the application of the max-product algorithm. This suggests an iterative minimisation method that repeatedly chooses one tree x_t to minimise the energy at a time. Once the local optimal configuration x_t^* is found, we proceed to a new tree, which is selected in some sensible manner. The energy E is guaranteed to decrease until reaching a local minimum.

This method includes the ICM as a special case when the tree is reduced to a single site, so we call it the tree-based ICM (T-ICM) algorithm. Although the T-ICM only finds local minima of the energy, we can expect the quality to be better than the original ICM because each tree covers many sites. For example, as shown in Figures 2a,b, a tree in the grid can cover as many as half of all the sites, which is very significant compared to only one site used by the ICM. The number of configurations of the tree t is $|L|^{|\mathcal{V}_t|}$, whilst it is $|L|$ for the ordinary ICM. Although it is a fairly simple extension, we are not aware of its previous use in MRFs.

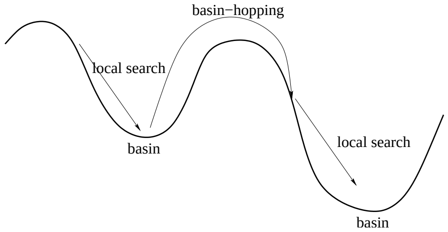

Figure 3. Iterated Local Search (Basin-hopping).

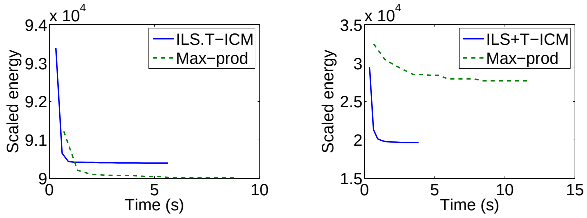

Figure 4. Performance of ILS.T-ICM and Max-product algorithm in minimising Ising energy with $\lambda = 0.5$ (left) and $\lambda = 1.0$ (right).

4.3. From local to global search

Although the T-ICM is an improvement over the ICM, some drawbacks still remain: (i) it is sensitive to initialisation, and (ii) it can get stuck in poor local minima. An effective strategy, commonly known in stochastic optimisation as iterated local search (ILS) (or basin hoping) [12], advocates *jumps* to escape from such minima. The jump step-size has to be large enough to successfully escape from the *basin* that traps the local search (see Figure 3). At the same time, the jump should not be too large so that search history can be exploited. After a jump, the local search is invoked, followed by an acceptance decision to accept or reject the jump. Convergence guaranteed acceptance criteria such as those used in simulated annealing can be used, but it is likely to be slow. A common practice is to accept the jump if a better local minimum is found.

In this study, we design a simple jump by randomly flipping labels of a small subset of sites (says, about 10-20%). Differing from most ILS practice, our T-ICM is a strong local search, and thus reduces the slow random-walk behaviour significantly. Although the ILS framework never guarantees to find global minima within limited time, we can expect it to find good local minima with the T-ICM as local search. Let us call this hybrid algorithm ILS.T-ICM.

Simulation: We simulate a 500×500 grid Ising model where $\{x_i\} = \pm 1$, and $\phi(x_i) = \exp(\theta_i x_i)$; $\psi(x_i, x_j) = \exp(\theta_{ij} x_i x_j)$. The parameters θ are set as follows

$$\theta_i \sim \mathcal{U}(-1,1) \text{ and } \theta_{ij} \sim \lambda \times \mathcal{U}(-1,1)$$

where $\mathcal{U}(-1,1)$ denotes the uniform distribution in the range $(-1,1)$, and $\lambda > 0$ specifies the interaction strength. The choice of trees for the T-ICM is discussed in Section 4.4. The result of minimising energy is shown in Figure 4. When the interaction is weak (e.g. $\lambda = 0.5$), the max-product performs well, but when the interaction is strong (e.g. $\lambda = 1.0$), the ILS coupled with T-ICM has a clear advantage.

4.4. Choice of trees and complexity

We employ the commonly used 4-neighbour grid MRF to model images. Although there is a wide range of choices for both superimposed and conditional trees, we limit them to simple horizontal rows and vertical columns for simplicity and efficiency.

Tree-based inference takes $\mathcal{O}(2|\mathcal{E}_t|)$ time to pass up and down messages, and requires $\mathcal{O}(2|\mathcal{E}_t|)$ memory to hold messages. So the time to incorporate all the trees to cover the whole image is $\mathcal{O}(2|\mathcal{E}|) = \mathcal{O}(4HW)$, where H, W are image dimensions. Thus, the time complexity per iteration of tree-based inference is about the same as that of Pearl's belief propagation. However, the memory in our case is still $\mathcal{O}(2|\mathcal{E}_t|) = \mathcal{O}(2D)$ where $D = \max\{H, W\}$, which is much smaller than the memory required by Pearl's belief propagation ($\mathcal{O}(4HW)$). In addition, inference in trees takes exactly 2 passes, while the number of iterations of belief propagation for the whole image, if the method does converge, is unknown and parameter dependent.

There are two superimposed trees (Figure 1d,e), where the horizontal tree has disconnected rows, and the vertical has disconnected columns. Strictly speaking, there are edges connecting parallel rows and columns to form graphically correct trees, potentials of these edges are just 1 and can be ignored in actual computation. To specify the remaining tree potentials, we make use of those in (6), where $\alpha_{t_1} = \alpha_{t_2} = 0.5$. Optimising α_t may give a better bound, but it is likely to hurt efficiency.

The conditional trees are just either rows or columns (Figure 2c,d). In particular, for the row pass in the T-ICM, we fix the labels of all rows except one, then update the labels for that row using the T-ICM. The process is repeated until convergence.

5. Experiments

5.1. Outdoor scene segmentation

In the first experiment we evaluate our learning and MAP estimation methods on the outdoor scene modeling problem. We use the Sowerby dataset[1] [6, 8], which has 104 labeled images of size 96×64. We choose 60 images for learning and 44 for testing. There are seven label classes to recognise: *sky, vegetation, road marking, road surface,*

[1] www.cs.toronto.edu/∼hexm/data/sowerby_mod.mat

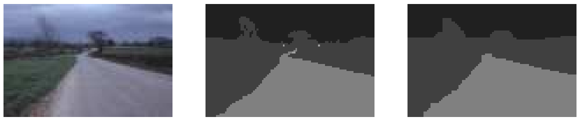

Figure 5. An image (left) in the Sowerby dataset, scene groundtruth (middle) and scene segmentation of likelihood lower-bound + T-ICM (right).

building, *street objects*, and *cars*. Methods that only exploit local image features may miss the intrinsic relations between class regions. For example, the sky is usually ‘above’ the rest of image regions, and road marking must be ‘within’ the road.

For the purposes of comparing different learning and inference algorithms, we use a simple flat conditional MRF model (also known as Conditional Random Field (CRF) [11]) with 4-neighbour grid structure, where each site corresponds to a pixel, although sophisticated hierarchies [8, 10] may be used to capture multiscale structures.

The data potentials in Equation 1 incorporate image features including colour descriptions (the R, G, B colour components, $R - B, 2G - R - B$ [6], saturation, $\max\{R, G, B\} - \min\{R, G, B\}$), gradient information (first-order, second-order at two scales in 0,45,90,135 degrees), and contextual information (difference of mean intensities of 3×3 regions near the site in 0,45,90,135 degrees). We also count the number of edge pixels falling in 15×15 regions centred at the site for each colour, and take the difference between R and B components, and between $2G$ and $R + B$ components. These image statistics are normalised to have zero mean and unit standard deviation over the training data.

For the interaction potentials, we use simple indicator features between pair of labels, but distinguish between horizontal and vertical label pairs. The feature weights $\mathbf{w}$ are parameters to be estimated during learning. The result is the Gibbs distribution with potentials given as

$$\begin{aligned}\phi(x_i, y) &= \exp(\mathbf{w}_\phi \mathbf{f}_\phi(x_i, y)) \\ \psi(x_i, x_j) &= \exp(\mathbf{w}_\psi \mathbf{f}_\psi(x_i, x_j))\end{aligned}$$

where $\mathbf{w}$ is the parameter vector and $\mathbf{f}$ is the feature vector.

Readers may want to consult ([11, 8, 10]) for details of estimating parameters in CRFs. Here we employ the stochastic gradient method to optimise the likelihood, its lower bound and pseudo-likelihood. This simple method appears to be both fast and effective, especially for the sum-product because optimisation methods relying on exact gradient (e.g. conjugate gradient) may be corrupted and stopped prematurely. We empirically found that 5 passes over training data are enough to achieve good performance for all learning algorithms.

For pseudo-likelihood learning, regularisation on interaction potentials is critical to avoid overestimating. This is to encounter the assumption commonly made by the pseudo-likelihood framework that the neighbour configuration of a given tree x_t is (nearly) optimal. This may be true at training time, but not at testing time when all configurations are only suboptimal. We use a Gaussian prior with diagonal covariance matrix $\mathbf{w} \sim \mathcal{N}(0, \sigma)$, and setting $\sigma_\psi = 3.0$ yields good performance, where σ_ψ corresponds to the interaction features. For other features and learning algorithms, no regularisation is needed.

The performance in pixel-wise classification rate of different training algorithms (in rows) coupled with MAP estimation methods (in columns) are presented in Table 1. The sum-product has a damping factor of 0.3 and is stopped after 30 iterations unless the messages have reached the convergence rate of 10^{-4}. Stronger convergence criteria may help to stabilize the gradient of the log-likelihood but can be too time consuming. All the MAP estimation methods are run for 100 iterations unless the energy has converged at the rate of 10^{-5}. The second column in Table 1 shows the training time on an ordinary Intel 2.6GHz PC.

As shown in Table 1, the sum-product is the slowest due to its iterative nature and lack of convergence guarantee. It is interesting to see that the sum-product offers no advantage over tree-based training methods. However, exact inference on trees is much faster and more predictable. Further, the tree-based methods for training and MAP estimation consistently outperform their site-based counterparts.

The learning method using likelihood lower bound coupled with the MAP estimation method T-ICM achieves 90.0% pixel-wise accuracy, which is comparable with the previous results reported in [8] (89.5%), and in [6] (90.7%). The difference is that techniques used in these papers are specifically tailored for scene segmentation problem while we just use the flat CRF with the focus on efficiency. In particular, [8] designs a sophisticated multiscale CRF with hidden variables, and [6] proposes a hybrid method between a neural network and a hierarchical Bayesian network together with feature selection and careful pixel sampling to handle class imbalance in the training data.

	Time(m)	ICM	T-ICM	MaxProd
PL	1.0	87.2	87.4	87.5
T-PL	1.6	87.4	88.0	88.1
SumProd	10.0	87.5	89.3	89.2
LL-LB	1.5	88.5	90.0	89.6

Table 1. Pixel-wise classification rate (%) of different training and MAP estimation methods on the Sowerby dataset. PL = Pseudo-likelihood, T-PL = Tree-based PL, LL-LB = Likelihood lower bound, SumProd = Sum-product, MaxProd = Max-product.

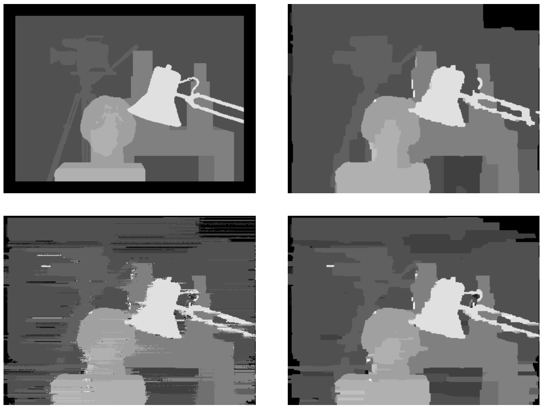

Figure 6. Stereo results: (top left) the groundtruth of the Tsukuba dataset, (top-right): α-expansion graph cut, (bottom-left): scanline and (bottom-right): scanline + one step T-ICM.

5.2. Stereo correspondence

The problem in stereo correspondence is to estimate the depth of the field given two or more 2-D images of the same scene taken from different cameras. It is often translated into estimating *disparity* between images. For simplicity, we only mention the two image problem. In the MRF-based stereo framework, a configuration of x realises the *disparity map*, which in this case is represented by a 4-neighbour grid network. The disparity set (or the label set) is often pre-specified. For example, in the two standard datasets[2] we use in this experiment, the Tsukuba has 16 labels, and the Venus has 20.

The data potential at each pixel location usually measures the similarity of pixel intensity between the left/right images, and the interaction potential ensures the smoothness of the disparity map. For comparison purposes, we use the simple truncated linear Potts cost model, although much more involved cost models and pre/post-processing should be used in real deployment. Let $y = (I^l, I^r)$ where I^l and I^r are intensities of the left and right images respectively, and $i = (i_X, i_Y)$ where i_X and i_Y are horizontal and vertical coordinates of the pixel i. The potentials are defined as

$$\begin{aligned}\phi(x_i, y) &= \exp(-\min(\Delta I(i, x_i), \tau)) \\ \psi(x_i, x_j) &= \exp(-\lambda \times \delta[x_i \neq x_j])\end{aligned}$$

where $\Delta I(i, x_i) = |I^l(i_X, i_Y) - I^r(i_X - x_i, i_Y)|$ and $\delta[.]$ is indicator function. The constant $\tau > 0$ is to control the effect of noise, and $\lambda > 0$ specifies the smoothness of the disparity map. For comparison, we choose $\tau = 15, \lambda = 10$ although ideally these parameters should be automatically estimated from data. Our implementation employs the technique described in [5] for fast message-passing and is based on the software framework of [15][3].

The are wide range of techniques available for stereo estimation [14]. Winning methods in term of accuracy are currently based on variants of either the max-product or Graph-Cuts. However, fast methods like scanline optimisation are still widely used for real-time implementation. The scanline is equivalent to taking independent 1-D rows of the MRF and running the forward-backward passes (Figure 1d). The result, however, has the inherent horizontal 'streaking' effect since no 2-D constraints are ensured (Figure 6, bottom-left).

Here we show that it is indeed very useful to combine the scanline method and our T-ICM algorithm. For this problem, the scanline provides a good initialisation for the T-ICM and the job of the T-ICM is to refine the solution of the scanline. More importantly, the T-ICM largely removes the 'streaking' effect introduced by the scanline (Figure 6, bottom-right). The energy, runtime, and error rate of algorithms are reported in Table 2. The errors are for all pixels, measured by an evaluation tool described in [14]. The scanline potentials are from (6) where there is an extra parameter $\alpha_t \in [0, 1]$ to help adapting to the 2-D cost model. Table 2 shows the effect of changing from $\alpha_t = 1.0$ to $\alpha_t = 0.4$ in term of reducing 2-D energy. As we can see, the T-ICM initialised from the scanline ($\alpha_t = 0.4$) is fast and effective in minimising energy as well as in increasing the performance of the scanline. Typically one step of T-ICM is about 3 times slower than the scanline. This is due to one scan of rows with the additional one scan of columns in the T-ICM, coupled with extra overhead needed for absorbing neighbour interaction potentials as in (12). The α-expansion Graph-Cuts, also shown in extensive experiments previously gives the very smooth result. This is not surprising because the algorithm is specifically designed for smooth labeling, especially with Potts model [4]. Interestingly, for the Tsukuba dataset, the Scanline+T-ICM achieves slightly lower error than the Graph-Cuts although the Graph-Cuts minimises the energy better.

5.3. Image denoising

The 122×179 noisy gray Penguin image[4] is shown in Figure 7. The labels of the MRF correspond to 256 intensity levels. Similar to the stereo correspondence problem, we use a simple truncated Potts model for the energy as follows

$$\begin{aligned}\phi(x_i, y) &= \exp(-\min(|x_i - y_i|, \tau)) \\ \psi(x_i, x_j) &= \exp(-\lambda \times \delta[x_i \neq x_j])\end{aligned}$$

where $\tau = 100$ and $\lambda = 25$. In addition, fast message passing for Potts models from [5] are used. For this particular

[2] Available at: http://vision.middlebury.edu/stereo/

[3] The C++ code is available at http://vision.middlebury.edu/MRF/

[4] Available at: http://vision.middlebury.edu/MRF/

Method	Data	Energy	Time(s)	Error
SL(α=1.0)	Tsukuba	629	0.2	11.9
SL(α=0.4)	Tsukuba	503	0.2	8.0
SL+1T-ICM	Tsukuba	340	0.8	4.9
GC	Tsukuba	317	15.0	5.4
SL(α=1.0)	Venus	1037	0.4	12.8
SL(α=0.4)	Venus	837	0.4	7.3
SL+1T-ICM	Venus	512	1.4	4.2
GC	Venus	489	33.6	2.5

Table 2. Energy($\times 10^3$) and error(%) vs runtime(s) of stereo algorithms evaluated on a Intel 2.6GHz machine. SL=Scanline, 1T-ICM=one step T-ICM, GC=Graph-Cuts.

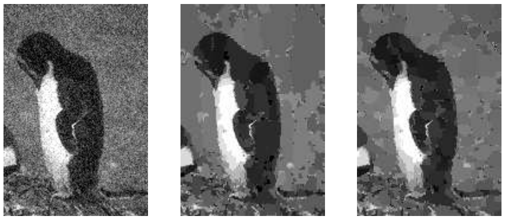

Figure 7. Penguin (left) noisy image, (middle) restored image with ILS.T-ICM, (right) restored image with Max-product.

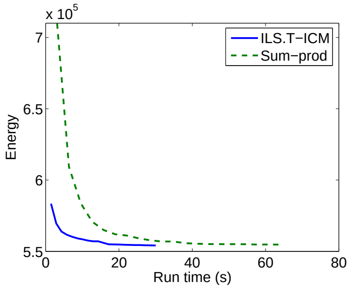

Figure 8. Performance of ILS.T-ICM and Max-product on Penguin image.

problem, the ILS.T-ICM runs faster than the max-product, yielding (slightly) lower energy (Figure 8) and (slightly) smoother restoration (Figure 7).

6. Conclusion

We have proposed a set of fast alternatives for supervised learning and inference in (conditional) Markov random fields by exploiting tree structures embedded in the network. For learning, we show that by using a tree-based pseudo-likelihood and a lower bound of the true likelihood as learning criteria, and coupling them with tree-based approximate MAP estimation, we can gain significant speed up in training without performance loss. For MAP estimation, we proposed a strong local search operator T-ICM and a global stochastic search operator in the iterated local search framework. We have shown in both Ising simulation and image restoration that the T-ICM can be competitive with the well-known max-product algorithm. In the stereo matching, the T-ICM is coupled well with the scanline optimisation yielding fast processing with a small overhead.

Future work includes theoretical consistency analysis of the proposed tree-based likelihoods and adapting the T-ICM for certain cost functions. Currently, the T-ICM is designed as a generic optimisation method, making no assumptions about the the nature of optimal solution. In contrast, label maps in vision are often smooth almost everywhere except for sharp boundaries.

References

[1] J. Besag. Spatial interaction and the statistical analysis of lattice systems (with discussions). *Journal of the Royal Statistical Society Series B*, 36:192–236, 1974. 1

[2] J. Besag. Statistical analysis of non-lattice data. *The Statistician*, 24(3):179–195, 1975. 1

[3] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society B*, 48(3):259–302, 1986. 1

[4] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 23(11):1222–1239, 2001. 1, 7

[5] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *International Journal of Computer Vision*, 70(1):41–54, Oct 2006. 7

[6] X. Feng, W. C. K. I., and S. N. Felderhof. Combining belief networks and neural networks for scene segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 24(4):467–483, 2002. 5, 6

[7] G. Hardy, J. Littlewood, and G. Pólya. *Inequalities*. Cambridge University Press, Cambridge, 2nd edition, 1952. 3

[8] X. He, R. Zemel, and M. Carreira-Perpinan. Multiscale conditional random fields for image labeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 695–702, 2004. 5, 6

[9] G. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14:1771–1800, 2002. 3

[10] S. Kumar and M. Hebert. A hierarchical field framework for unified context-based classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 1284–1291, Oct 2005. 6

[11] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine learning (ICML)*, pages 282–289, 2001. 2, 6

[12] H. R. Lourenco, O. C. Martin, and T. Stutzle. Iterated local search. *International Series in Operations Research and Management Science*, (57):321–354, 2003. 2, 5

[13] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Francisco, CA, 1988. 1

[14] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, 2002. 7

[15] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for Markov random fields. In *Proceedings of the European Conference on Computer Vision (ECCV)*, number 3952 in Lecture Notes in Computer Science, pages 16–29, 2006. 1, 7

[16] M. J. Wainwright, T. Jaakkola, and A. S. Willsky. Tree-based reparameterization framework for analysis of sum-product and related algorithms. *IEEE Transactions on Information Theory*, 45(9):1120–1146, 2003. 1, 3

[17] M. J. Wainwright, T. Jaakkola, and A. S. Willsky. A new class of upper bounds on the log partition function. *IEEE Transactions on on Information Theory*, 51:2313–2335, Jul 2005. 1, 3, 4

[18] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. MAP estimation via agreement on (hyper)trees: Message-passing and linear-programming approaches. *IEEE Transactions on Information Theory*, 51(11):3697–3717, 2005. 1, 3