Fast Memory-Efficient Generalized Belief Propagation[⋆]

M. Pawan Kumar and P.H.S. Torr

Department of Computing, Oxford Brookes University,
Oxford, UK, OX33 1HX
{pkmudigonda, philiptorr}@brookes.ac.uk
http://cms.brookes.ac.uk/computervision

Abstract. Generalized Belief Propagation (GBP) has proven to be a promising technique for performing inference on Markov random fields (MRFs). However, its heavy computational cost and large memory requirements have restricted its application to problems with small state spaces. We present methods for reducing both run time and storage needed by GBP for a large class of pairwise potentials of the MRF. Further, we show how the problem of subgraph matching can be formulated using this class of MRFs and thus, solved efficiently using our approach. Our results significantly outperform the state-of-the-art method. We also obtain excellent results for the related problem of matching pictorial structures for object recognition.

1 Introduction

Many tasks in Computer Vision, such as segmentation and object recognition, can be given a probabilistic formulation using Markov random fields (MRF). A popular method for performing inference on MRFs is Belief Propagation (BP) [1]. It is well known that on tree-structured MRFs, BP can be used to efficiently perform exact inference. For a general MRF, Yedidia *et al.* [2] proved that BP converges to stationary points of Bethe approximation of the free energy. They also proposed the Generalized Belief Propagation (GBP) algorithm which converges to stationary points of (the more accurate) Kikuchi approximation. Despite outperforming BP in terms of convergence and accuracy, there are few uses of GBP reported in the literature as it is computationally feasible only when the number of labels of the MRF is small.

Recent work has focused on tackling the problem of computational cost of message passing methods such as BP and GBP. Felzenszwalb and Huttenlocher [3, 4] put forward a method for speeding up message passing algorithms such as Viterbi, Forward-Backward and BP for a large class of pairwise potentials (e.g. Potts and linear model), when the labels are regularly discretized points in the parameter space. In our previous work [5], we extended these results to general

[⋆] This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

A. Leonardis, H. Bischof, and A. Pinz (Eds.): ECCV 2006, Part IV, LNCS 3954, pp. 451–463, 2006.

MRFs. However, little consideration thus far has been given to speeding up GBP, with the exception of Shental *et al.* [6] who describe an efficient GBP algorithm but only for the special case where the pairwise potentials form an Ising model.

The problem of reducing the large memory requirements has also met with little success. Felzenszwalb and Huttenlocher [4] observe that, when performing BP on a bipartite graph, the messages going to only a subset of sites are changed at each iteration. This allows them to reduce the memory requirements for grid graphs by half. Vogiatzis *et al.* [7] suggest a coarse-to-fine strategy for BP by grouping together similar labels. However this is restricted to labels lying on a grid. More importantly, it substantially changes the problem such that the messages and beliefs computed at any stage are not necessarily equivalent to those corresponding to the original MRF.

In this paper, show how to reduce the computational cost and memory requirements of GBP for a large class of pairwise potentials which we call the *robust truncated model*. This model divides all pairs of labels for the neighbouring sites into compatible and incompatible pairs and truncates the pairwise potentials of the incompatible pairs to a constant (see section 2 for details). Many vision applications such as object recognition [5], stereo and optical flow [4] use special cases of this model. Typically, the number of compatible labels n_C for a given label is much less than the total number of labels n_L, i.e. $n_C \ll n_L$.

We exploit the fact that, since the pairwise potentials of incompatible pairs of labels are constant, it results in many redundant computations in GBP which can be avoided. Let n_R be the number of regions formed by clustering the sites of the MRF and n_M be the size of the largest region. The main contributions of the paper are the following:

- We reduce the time complexity of GBP to $O(n_R n_M n_L^{n_M-1} n_C)$, (i.e. by a factor of n_L/n_C). Since $n_C \ll n_L$ for MRFs used in vision, this makes GBP computationally feasible (section 3).
- We observe that the approach described in [4] to reduce the memory requirements of BP by half for bipartite graphs can be extended to GBP (section 4).
- We show how the memory requirements of GBP can be reduced drastically (by a factor $(n_L/n_C)^{n_M-1}$) for a special case of the robust truncated model which can be used in various vision applications. Again, since $n_C \ll n_L$, GBP becomes memory efficient and thus, practically useful (section 4).
- We formulate the problem of subgraph matching using the special case of the robust truncated model and solve it accurately using the efficient GBP algorithm. Our results significantly outperform the state-of-the art methods (section 5).
- We obtain excellent results for the related problem of matching pictorial structures [8] for object recognition by using the efficient GBP algorithm (section 5).

It should be noted that our methods are applicable to other related message passing algorithms such as Viterbi, Forward-Backward, BP and tree-reweighted message passing [9]. For completeness, we first briefly describe the BP and GBP algorithms in the next section.

2 Belief Propagation and Its Generalization

This section briefly describes the standard belief propagation (BP) algorithm for performing inference on MRFs and formulates it using the *canonical* framework. This framework is then extended which results in the Generalized Belief Propagation (GBP) algorithm [2].

An MRF is defined by n_S sites along with a symmetric neighbourhood relationship $N(.)$ on them, i.e. $i \in N(j)$ if and only if $j \in N(i)$. Each site i can take a label $x_i \in \mathcal{X}_i$. We assume that the sets $\mathcal{X}_i$ are finite and discrete, i.e. $|\mathcal{X}_i| = n_L < \infty$. Associated with each configuration $\mathbf{x}$ of the MRF is its joint probability given by

$$\Pr(x_1, ..., x_{n_S}) = \frac{1}{Z} \prod_{ij} \phi_{ij}(x_i, x_j) \prod_i \phi_i(x_i). \tag{1}$$

Here, $\phi_i(x_i)$ is the unary potential of site i having label x_i, $\phi_{ij}(x_i, x_j)$ is the pairwise potential for two neighbouring sites i and j having labels x_i and x_j respectively and Z is the partition function. Note that the above equation assumes the MRF to be pairwise. However, this is not restrictive as any MRF can be converted into a pairwise MRF [2]. Performing inference on the MRF involves either determining the MAP configuration or obtaining the marginal posterior probabilities of each label. In this paper, we describe our approach in the context of max-product BP which provides the MAP configuration while noting that it is also equally applicable to sum-product BP which provides the marginal posteriors.

BP is a message passing algorithm proposed by Pearl [1]. It is an efficient approximate inference algorithm for MRFs with loops where each site i iteratively passes a message to its neighbouring site j. The message is a vector of dimension n_L whose elements are calculated as

$$m_{ij}^t(x_j) \leftarrow \alpha \max_{x_i} \phi_{ij}(x_i, x_j) \phi_i(x_i) \prod_{k \in N(i) \backslash j} m_{ki}^{t-1}(x_i), \tag{2}$$

where α is a normalization constant and $N(i) \backslash j$ is the set of all neighbouring sites of i excluding j. Note that x_j is used to index the message vector in the above equation such that $m_{ij}^t(x_j)$ corresponds to the x_j^{th} element of the vector m_{ij}^t. All messages are initialized to 1 and convergence is said to be achieved when the rate of change of all messages drops below a threshold. At convergence, the belief of a site i having a label x_i is given by

$$b_i(x_i) \leftarrow \alpha \phi_i(x_i) \prod_{j \in N(i)} m_{ji}(x_i), \tag{3}$$

and the MAP estimate is obtained by choosing the label x_i^* with the highest belief for every site i.

Yedidia *et al.* [2] proved that BP converges to the stationary points of the Bethe approximation of the free energy which clusters the sites of the MRF into *regions* of size at most 2. We denote the set of sites belonging to a region r by $\mathcal{S}(r)$. Region s is considered a sub-region of r if and only if $\mathcal{S}(s) \subset \mathcal{S}(r)$. Further, s is a *direct* sub-region of r if and only if the set $s \cup i$ is not a sub-region of r, for all regions i.

Every region r passes a message $m_{r \to s}$ to each of its direct sub-regions s. In order to compactly express what follows, we adopt the following notation. The message $m_{r \to s}(x_s)$ is simply written as $m_{r \to s}$ (i.e. the indexing is dropped). For example, $m_{ij \to j}$ stands for $m_{ij \to j}(x_j)$ in the following equations. We define $M(r)$ to be the set of messages going into a sub-region of r or going into r itself while starting outside r and its sub-regions. Let t be the set of all sites which belong to r but not to s, i.e. $t = r \backslash s$. The message update rule is given by

$$m_{r \to s} \leftarrow \alpha \max_{x_t} \phi_t(x_t) \prod_{m_{r' \to s'} \in M(r) \backslash M(s)} m_{r' \to s'}. \tag{4}$$

The potential $\phi_t(x_t)$ is defined as the product of the unary potentials of all sites in $r \backslash s$ and of all pairwise potentials between sites in r. It is easily verifiable that the above update equation is the same as equation (2). Upon convergence, the belief of r is given by

$$b_r \leftarrow \alpha \phi_r(x_r) \prod_{m_{r' \to s'} \in M(r)} m_{r' \to s'}. \tag{5}$$

The standard BP algorithm can be considered a special case of Generalized Belief Propagation (GBP). GBP converges to the stationary points of the Kikuchi approximation (which is more accurate than Bethe approximation) by allowing for regions of size more than 2. Fig. 1 shows an example of this for an MRF with 4 sites which results in 10 regions. It also shows the corresponding messages along with their directions. We define $M(r,s)$ to be the set of all messages starting from a sub-region of r and going to s or its sub-region. Then the GBP update equation is given by

$$m_{r \to s} \leftarrow \alpha \max_{x_t} \frac{\phi_t(x_t) \prod_{m_{r' \to s'} \in M(r) \backslash M(s)} m_{r' \to s'}}{\prod_{m_{r'' \to s''} \in M(r,s)} m_{r'' \to s''}}, \tag{6}$$

where $t = r \backslash s$. Note that, like BP, the message $m_{r \to s}$ in GBP is also indexed by x_s. For example, $m_{ijk \to ij}$ stands for $m_{ijk \to ij}(x_i, x_j)$ and thus, can be interpreted as an $n_L \times n_L$ matrix. Table 1 lists all the sets $M(r)$ and $M(r,s)$ for the example MRF in Fig. 1. Using equation (6), the messages $m_{ij \to i}$ and $m_{ijk \to ij}$ are given by

$$m_{ij \to i} \leftarrow \alpha \max_{x_j} \phi_j(x_j) \phi_{ij}(x_i, x_j) m_{ijk \to ij} m_{jk \to j}, \tag{7}$$

$$m_{ijk \to ij} \leftarrow \alpha \max_{x_t} \frac{\phi_k(x_k) \prod_{p,q \in \{i,j,k\}} \phi_{pq}(x_p, x_q) m_{jkl \to jk} m_{kl \to k}}{m_{ik \to i} m_{jk \to j}}. \tag{8}$$

Robust Truncated Model. In this paper, we consider the case where the pairwise potentials $\phi_{ij}(x_i, x_j)$ form a robust truncated model such that

$$\begin{aligned} \phi_{ij}(x_i, x_j) &= f_{ij}(x_i, x_j), \quad \text{if } \; x_i \in \mathcal{C}_i(x_j), \\ &= \tau_{ij}, \quad \text{otherwise}, \end{aligned} \tag{9}$$

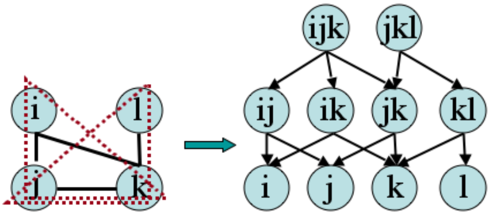


Fig. 1. *Left.* An example MRF with four sites. The solid lines show the interactions between the sites and describe a neighbourhood relationship on them. The dotted lines show the clustering of the sites into regions of size 3. *Right.* The sites are grouped to form ten regions for canonical GBP. The resulting messages and their directions are also shown using the arrows. For example, the top left arrow shows the message $m_{ijk \to ij}$ and the bottom left arrow shows the message $m_{ij \to i}$.

Table 1. Messages belonging to the sets $M(r)$ and $M(r, s)$ for each region r and its direct sub-region s shown in Fig. 1

r	$M(r)$
$\{ijk\}$	$m_{jkl \to jk}$, $m_{kl \to k}$
$\{jkl\}$	$m_{ijk \to jk}$, $m_{ij \to j}$, $m_{ik \to k}$
$\{ij\}$	$m_{ijk \to ij}$, $m_{ik \to i}$, $m_{jk \to j}$
$\{ik\}$	$m_{ijk \to ik}$, $m_{ij \to i}$, $m_{jk \to k}$, $m_{kl \to k}$
$\{jk\}$	$m_{ijk \to jk}$, $m_{jkl \to jk}$, $m_{ij \to j}$, $m_{ik \to k}$, $m_{kl \to k}$

r	$M(r)$
$\{kl\}$	$m_{jkl \to kl}$, $m_{ik \to k}$, $m_{jk \to k}$
$\{i\}$	$m_{ij \to i}$, $m_{ik \to i}$
$\{j\}$	$m_{ij \to j}$, $m_{jk \to j}$
$\{k\}$	$m_{ik \to k}$, $m_{jk \to k}$, $m_{kl \to k}$
$\{l\}$	$m_{kl \to l}$

r	s	$M(r, s)$
$\{ijk\}$	$\{ij\}$	$m_{ik \to i}$, $m_{jk \to j}$
$\{ijk\}$	$\{ik\}$	$m_{ij \to i}$, $m_{jk \to k}$
$\{ijk\}$	$\{jk\}$	$m_{ij \to j}$, $m_{ik \to k}$
$\{jkl\}$	$\{jk\}$	$m_{kl \to k}$
$\{jkl\}$	$\{kl\}$	$m_{jk \to k}$

where $\mathcal{C}_i(x_j)$ defines the subset of labels of i which are 'compatible' with x_j. In other words, the cost for an incompatible pair of labels is truncated to τ_{ij}. Included in this class are the commonly used Potts model i.e. $f_{ij}(x_i, x_j) = d_{ij}$, $\forall x_i \in \mathcal{C}_i(x_j)$, the truncated linear model i.e. $f_{ij}(x_i, x_j) = \exp(-|x_i - x_j|)$ and the truncated quadratic model i.e. $f_{ij}(x_i, x_j) = \exp(-(x_i - x_j)^2)$. In most problems, $f_{ij}(x_i, x_j) > \tau_{ij}$ and the number of labels n_C in $\mathcal{C}_i(x_j)$ are much smaller than n_L, i.e. $n_C \ll n_L$. Such MRFs have been successfully used for applications such as object recognition [5] and stereo [4]. Next, we describe our fast GBP algorithm.

3 Fast Generalized Belief Propagation

We now present a method for making GBP computationally efficient for MRFs whose pairwise potentials form a robust truncated model. This is a more general case than the Ising model addressed in [6]. Note that the choice of regions for GBP is not of concern in this paper since our method is independent of it. However, for clarity, we will only describe the method for MRFs that form complete graphs and where regions are formed by clustering all possible combinations of three sites. The extension to any other MRF is trivial. In this case, there are two types of messages: (i) $m_{ij \to j}$ is the message that region $\{i, j\}$ passes to site j and,

(ii) $m_{ijk \to jk}$ is the message that region $\{i, j, k\}$ passes to region $\{j, k\}$. Using equation (6), these messages are given by

$$m_{ij \to j} \leftarrow \alpha \max_{x_i} \phi_i(x_i)\phi_{ij}(x_i, x_j) \prod_{n \in \mathbf{S} \backslash \{i,j\}} m_{ni \to i} m_{nij \to ij}, \tag{10}$$

and

$$m_{ijk \to jk} \leftarrow \alpha \max_{x_i} \frac{\phi_i(x_i) \prod_{p,q \in \mathbf{R}} \phi_{pq}(x_p, x_q) \prod_{n \in \mathbf{S} \backslash \mathbf{R}} m_{ni \to i} m_{nij \to ij} m_{nik \to ik}}{m_{ij \to j} m_{ik \to k}}, \tag{11}$$

where $\mathbf{R} = \{i, j, k\}$ and $\mathbf{S}$ is the set of all n_S sites of the MRF. Obviously, most of the computational cost is contributed by messages $m_{ijk \to jk}$ which can be reduced significantly by making use of the special form of the pairwise potentials.

Since for any pair of sites i and j, most of the pairwise potential $\phi_{ij}(x_i, x_j)$ are constant (i.e. τ_{ij}), considerable speed-up is achieved by pre-computing all the terms which are common in the message update equations (10) and (11). In order to compute the messages $m_{ij \to j}$, we define

$$r_i(x_j) = \alpha \max_{x_i} \phi_i(x_i) \prod_{n \in \mathbf{S} \backslash \{i,j\}} m_{ni \to i} m_{nij \to ij}, \tag{12}$$

and

$$r'_i(x_j) = \alpha \max_{x_i \in \mathcal{C}_i(x_j)} \phi_i(x_i) f_{ij}(x_i, x_j) \prod_{n \in \mathbf{S} \backslash \{i,j\}} m_{ni \to i} m_{nij \to ij}. \tag{13}$$

The message $m_{ij \to j}$ is given by $\max\{r'_i(x_j), \tau_{ij} r_i(x_j)\}$. Note that no speed-up is obtained for the messages $m_{ij \to j}$ except in the special case of $\tau_{ij} = 0$ when each message can be computed as $\max\{r'_i(x_j)\}$ (i.e. independent of $r_i(x_j)$) in $O(n_C)$ time, where n_C is the number of labels in $\mathcal{C}_i(x_j)$. However, as noted above, the real concern is to reduce the complexity of the messages $m_{ijk \to jk}$.

We define

$$q_{ik}(x_j) = \sqrt{\alpha} \max_{x_i \notin \mathcal{C}_i(x_j)} \frac{\sqrt{\phi_i(x_i)} \prod_{n \in \mathbf{S} \backslash \{i,j,k\}} m_{nij \to ij} \sqrt{m_{ni \to i}}}{m_{ij \to j}}, \tag{14}$$

and

$$q'_i(x_j, x_k) = \alpha \max_{x_i \in \mathcal{C}_i(x_j, x_k)} \frac{\phi_i(x_i) \prod_{\{p,q\} \in \mathbf{R}} \phi_{pq} \prod_{n \in \mathbf{S} \backslash \mathbf{R}} m_{nij \to ij} m_{nik \to ik} m_{ni \to i}}{m_{ij \to j} m_{ik \to k}}, \tag{15}$$

where $\mathbf{R} = \{i, j, k\}$ and $\mathcal{C}_i(x_j, x_k) = \mathcal{C}_i(x_j) \cup \mathcal{C}_i(x_k)$. The time complexities of calculating $q_{ik}(x_j)$ and $q'_i(x_j, x_k)$ for a particular x_j and x_k are $O(n_L)$ and $O(n_C)$ respectively. Once these terms have been computed, the message $m_{ijk \to jk}$ can be obtained in $O(1)$ time as $\max\{q'_i(x_j, x_k), \phi_{jk} \tau_{ij} q_{ik}(x_j) \tau_{ik} q_{ij}(x_k)\}$.

The computational complexity of the overall algorithm is $O(n_S^3 n_L^2 n_C)$ (i.e. the number of messages). This is significantly better than the $O(n_S^3 n_L^3)$ time taken by ordinary GBP when n_L is very large. Again, for the special case of $\tau_{ij} = 0$, the messages can be computed even more efficiently as $q'_i(x_j, x_k)$, without computing the terms $q_{ik}(x_j)$. Note that the terms $q_{ik}(x_j)$ and $q_{ij}(x_k)$ would be computed using the same label $x_i \notin \mathcal{C}_i(x_j, x_k)$ in equation (14) as the pairwise potentials $\phi_{ij}(x_i, x_j)$ and $\phi_{ik}(x_i, x_k)$ are constant for all such x_i (proof in appendix). Thus, the messages computed would be exactly equal to the messages in equation (11).

In general, this approach reduces the time complexity of GBP from $O(n_R n_M n_L^{n_M})$ to $O(n_R n_M n_L^{n_M-1} n_C)$, where n_R is the number of regions and n_M is size of the largest region. For example, in the case of BP over a complete graph, the only messages are of the form $m_{ij \to j}$ which can be computed efficiently using the above method in $O(n_S^2 n_L n_C)$ time instead of $O(n_S^2 n_L^2)$ time required by ordinary BP. Note that this is the same factor of speed-up obtained by the method described in [3] which cannot be extended to the GBP algorithm. Algorithm 1 shows the main steps involved in reducing the computational cost of GBP. Next, we describe our memory-efficient GBP algorithm.

Algorithm 1. Fast Generalized Belief Propagation

```
1. Using equations (12) and (13), calculate $r_i(x_j)$ and $r'_i(x_j)$, $\forall$ sites $i$, $j$ and labels $x_j$.
2. Compute $m_{ij \to j} \leftarrow \max\{r'_i(x_j), \tau_{ij} r_i(x_j)\}$.
3. Using equations (14) and (15), calculate $q_{ik}(x_j)$ and $q'_i(x_j, x_k)$, $\forall$ $i$, $j$, $k$, $x_j$ and $x_k$.
4. Compute $m_{ijk \to jk} \leftarrow \max\{q'_i(x_j, x_k), \phi_{jk} \tau_{ij} q_{ik}(x_j) \tau_{ik} q_{ij}(x_k)\}$.
5. Obtain the beliefs using equation (5).
```

4 Memory-Efficient Generalized Belief Propagation

We now present two approaches to reduce the memory requirements of GBP. The first approach extends the method of Felzenszwalb and Huttenlocher [4] for reducing the memory requirements of BP by half on bipartite graphs. The basic idea is that for a bipartite graph with the set of regions $A \cup B$, the message that a region in A passes to its sub-regions depends only on the messages coming from the regions in B and vice versa. In other words, if we know the messages coming from B, we can compute the messages within A. This suggests the strategy of alternating between computing messages for regions in A and B, thereby reducing the memory requirements by half.

We now describe the second approach which requires that $\tau_{ij} = 0$ for all pairs of site i and j. It is not surprising that further constraints need to be imposed on the robust truncated model. As mentioned above, the problem of reducing memory requirements has proven to be more difficult than that of reducing the time complexity and has met with limited success so far. However, we will demonstrate in section 5 that this restricted robust truncated model is still useful in a wide variety of vision applications.

The basic idea is to reduce the state space of the original MRF by dividing it into smaller MRFs whose labels are a subset of the labels of the original MRF.

However, these subsets are chosen such that the messages and beliefs computed on them are equivalent to those that would be obtained for the original problem. Specifically, we observe that when $\tau_{ij} = 0$, the messages $m_{ij \to j}$ and $m_{ijk \to jk}$ can be calculated using only $r'_i(x_j)$ and $q'_i(x_j, x_k)$ for all iterations of the GBP algorithm. Since $r'_i(x_j)$ and $q'_i(x_j, x_k)$ (and therefore the messages and the beliefs) are computed using $\mathcal{C}_i(x_j)$ and $\mathcal{C}_i(x_k)$, it would be sufficient to only include these in the smaller MRFs. Thus, each of smaller MRFs contains a subset of labels such that if x_j is included in an MRF, then $\mathcal{C}_i(x_j)$ is also included in that MRF, for all sites i. These MRFs can then be solved one at a time using Algorithm 1 thereby greatly reducing the memory requirements since $n_C \ll n_L$. Moreover, this approach does not increase the computational cost of the fast GBP algorithm described in the previous section. Algorithm 2 illustrates the main steps of memory-efficient GBP.

Algorithm 2. Memory-Efficient Generalized Belief Propagation

```
1. Choose a subset of labels $x_i$ for $i$. Choose all the labels $x_j \in \mathcal{C}_j(x_i)$, $\forall$ sites $j$.
2. Solve the resultant small MRF using Algorithm 1. Note that $r_i(x_j)$ and $q_{ik}(x_j)$ need
   not be calculated.
3. Repeat step 2 with a different subset until all beliefs have been computed.
```

Note that our second approach achieves a considerable reduction in memory (of factor $(n_L/n_C)^{n_M - 1}$) by restricting the form of the robust truncated model. Further, it is applicable to any general topology of the MRF, i.e. it is not restricted to only bipartite graphs. We now demonstrate our approach for subgraph matching and object recognition.

5 Experiments

In order to demonstrate the effectiveness of our approach, we generated several complete MRFs whose pairwise potentials form a robust truncated model with $\tau_{ij} = 0$. The regions are formed by clustering all possible combinations of three sites. Fig. 2 shows the average time and memory requirements for different values of the n_C/n_L (averaged over 100 MRFs). Note that when $n_C = n_L$ our approach reduces to the standard GBP algorithm. However, when $n_C \ll n_L$, it provides a significant reduction in time and memory requirements.

We now formulate two important problems, subgraph matching and object recognition, using the special case of the robust truncated model (i.e. $\tau_{ij} = 0$). It is observed that in both cases $n_C \ll n_L$ which allows us to solve these problems accurately using our fast, memory-efficient GBP algorithm.

5.1 Subgraph Matching

We use the fast, memory-efficient GBP algorithm to solve the problem of subgraph matching. Given two graphs $\mathcal{G}_1 = \{\mathcal{V}_1, \mathcal{E}_1\}$ and $\mathcal{G}_2 = \{\mathcal{V}_2, \mathcal{E}_2\}$, subgraph

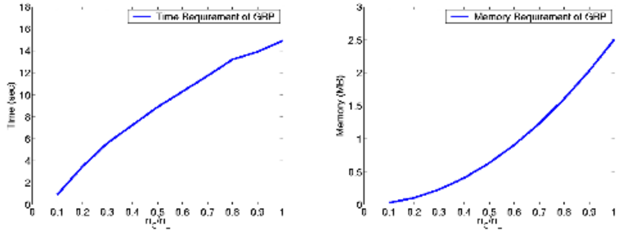


Fig. 2. Left: Average time taken by the efficient GBP algorithm for 100 random complete MRFs whose pairwise potentials satify the special case of the robust truncated model. The time complexity scales almost linearly with the factor n_C/n_L. Right: Average memory requirements which scales quadratically with n_C/n_L.

matching involves finding a mapping $f : \mathcal{V}_1 \rightarrow \mathcal{V}_2$ which minimizes the following energy function:

$$\sum_{v_i,v_j \in \mathcal{V}_1} \|l^1_{ij} - l^2_{f(i)f(j)}\|, \tag{16}$$

where l^k_{ij} is the distance between vertices i and j of the k^{th} graph. Many important computer vision problems, such as matching part-based models for object recognition can be thought of as special cases of this problem.

We define an MRF for determining the mapping $f(.)$ such that each site i represents a vertex v^1_i in $\mathcal{V}_1$. Each label x_i represents a vertex v^2_i in $\mathcal{V}_2$. For our example, we assume that all points $v^1_i \in \mathcal{V}_1$ are equally likely to map to a point in $\mathcal{V}_2$, and hence the likelihood terms $\phi_i(x_i)$ are set to 0.5 (however this is not generally the case). The sites of the MRF form a complete graph as distances between all pairs of vertices should be preserved by the mapping. We define the pairwise potentials as

$$\phi_{ij}(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} d \text{ if } |l^1_{ij} - l^2_{x_i x_j}| \leq \epsilon \\ 0 \quad\quad \text{otherwise,} \end{cases} \tag{17}$$

where ϵ is a constant which depends on the (expected) level of noise. In our experiments, we use $d = 1$. This favours the preservation of distance between corresponding pairs of vertices. Figure 3 shows an example of this formulation when $|\mathcal{V}_1| = 3$ and $|\mathcal{V}_2| = 4$.

Our problem thus reduces to obtaining the MAP estimate given the above MRF. For this purpose, we use the efficient GBP algorithm described in Algorithm 2. By restricting the region size to two, we obtain a time and memory efficient BP. Although less accurate, efficient BP is faster than efficient GBP. We compare the results with ordinary GBP and BP algorithms. For complete graphs, we found that GBP works well when the regions form a *star* pattern, i.e. the regions are of the form $\{1, i, j\}$ for all pairs $i > 1$ and $j > 1$. The common site '1' is chosen randomly. Note that this observation is consistent with that reported in [10].

We generated 1000 pairs of random graphs $\mathcal{G}_1$ and $\mathcal{G}_2$, with $|\mathcal{V}_1| = 0.25|\mathcal{V}_2|$ on an average. The number of vertices $|\mathcal{V}_2|$ were varied between 30 and 60.

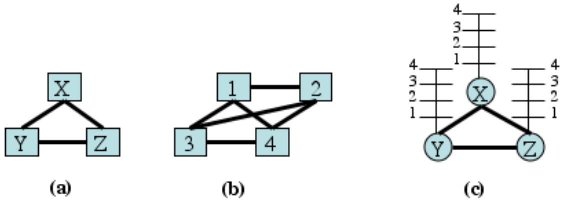


Fig. 3. Subgraph Matching. (a) Graph $\mathcal{G}_1$ with three vertices which is a rigidly transformed subgraph of graph $\mathcal{G}_2$ shown in (b). (c) The corresponding MRF formulation for subgraph matching. The MRF consists of three sites corresponding to the vertices X,Y and Z of $\mathcal{G}_1$. Each site has four possible labels corresponding to vertices 1,2,3 and 4 of $\mathcal{G}_2$. The interactions between the sites is shown using solid lines.

Table 2. Average time and space requirements of various methods for subgraph matching. Columns 4 and 5 show the requirements for smaller graphs with $|\mathcal{V}_2| = 20$.

Method	Time	Memory	Time (Small)	Memory (Small)	Accuracy (%)
BP	2 sec	4 MB	0.009 sec	0.08 MB	78.61
GBP	-	> 350 MB	6 sec	0.5 MB	95.79
Efficient BP	0.2 sec	0.4 MB	0.006 sec	0.008 MB	78.61
Efficient GBP	1.5 sec	3.5 MB	0.6 sec	0.07 MB	95.79
[11]	4.3 sec	0.1 MB	2.2 sec	0.02 MB	20.00

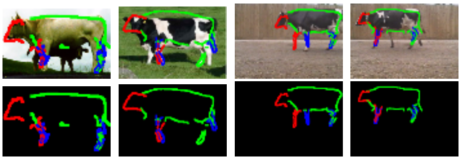
The vertices $|\mathcal{V}_1|$ were randomly selected subset of $|\mathcal{V}_2|$ with 7% noise added to them. The average number of correct matches for the vertices in $\mathcal{V}_1$ found using GBP were 95.79% (9421 out of 9835) compared to 78.61% (7732 out of 9835) found using BP. Thus, GBP provides much more accurate results than BP which should encourage its use in practice. We also significantly outperformed the state-of-the-art method by Chui and Rangarajan [11] (tested using their publically available code) on our challenging dataset. Table 2 summarizes the average time and space requirements for the various methods used. Note that due to large memory requirements of GBP, we ran another set of experiments on smaller graphs, i.e. $|\mathcal{V}_2| = 20$. The time and memory requirements for these smaller graphs are shown in the fourth and fifth column.

5.2 Object Recognition

We tested our approach for object recognition using a parts-based model called pictorial structures (PS) introduced by Fischler and Elschlager [8] and extended in [5]. PS are compositions of 2D patterns, i.e. parts, under a probablistic model for their shape, appearance and spatial layout (see [5] for details).

The connections between the parts of the PS form a complete graph. The pairwise potentials are defined as

$$\phi_{ij}(x_i, x_j) = \begin{cases} d \text{ if valid configuration,} \\ 0 \qquad \text{otherwise.} \end{cases} \tag{18}$$



Method	Time	Memory
BP	59 sec	0.7 MB
GBP	240 sec	38 MB
Efficient BP	2 sec	0.09 MB
Efficient GBP	16 sec	0.5 MB

Fig. 4. Results of obtaining the MAP estimate of the parts of cows using the fast, memory-efficient GBP. The first row shows the input images. The detected parts are shown in the second row. The table on the right shows the average time and space requirements of various methods for object recognition.

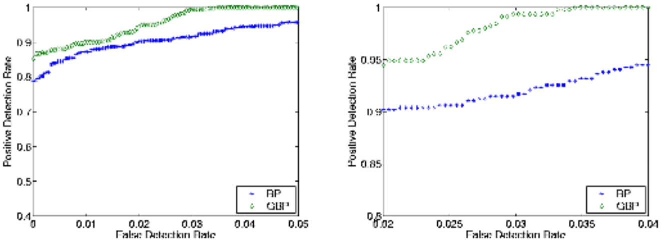


Fig. 5. Left: ROC curves for cow recognition. Right: Zoomed versions of a part of the ROC curve. Results indicate that better recognition performance is obtained using GBP compared to BP.

A configuration is valid if $x_{ij}^{min} \leq ||x_i - x_j|| \leq x_{ij}^{max}$. In all our experiments, we used $d = 1$. The parameters of the model are learnt in an unsupervised manner from videos as described in [5]. During recognition, the putative poses of the parts are found using a *tree cascade of classifiers* (see [5] for details). This allows us to efficiently prune the undesirable poses which result in a low potential $\phi_i(x_i)$. Again, for the above MRF, the regions form a *star* pattern with the *torso* part being the common site [10]. The MAP estimate of the pose for each part is obtained by performing inference using the fast, memory-efficient GBP algorithm.

Fig. 4 shows the results of our approach on some images containing cows. The cascade efficiently obtains approximately one hundred putative poses per part in 2 minutes. The MAP estimate of each of the parts obtained using GBP is shown in the second row. The table on the right summarizes the time and space requirements of the various methods for object recognition. Fig. 5 shows the ROC curves obtained using 450 positive and 2400 negative examples. Note that, as in the case of subgraph matching, GBP performs better than BP.

6 Summary and Conclusions

We have presented methods to overcome the problems of large computational complexity and space requirements in using GBP for the important case where the pairwise potentials form a robust truncated model. Specifically,

- We reduce the time complexity of GBP to $O(n_R n_M n_L^{n_M-1} n_C)$ for the case of robust truncated models.
- We reduce the memory requirements of GBP over bipartite MRFs by half.
- We further reduce the memory requirements of GBP for a general MRF by a factor of $(n_L/n_C)^{n_M-1}$ for a special case of the robust truncated model.

Further, we have demonstrated how the important problems of subgraph matching and object recognition can be formulated using the robust truncated model and solved efficiently using our approach. Our results significantly outperform the state-of-the-art method. We plan to investigate whether some restrictions can be relaxed (e.g. $\tau_{ij} = 0$). Other applications such as segmentation and optical flow also need to be explored.

References

1. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kauffman (1998)
2. Yedidia, J., Freeman, W., Weiss, Y.: Bethe free energy, kikuchi approximations, and belief propagation algorithms. Technical Report TR2001-16, MERL (2001)
3. Felzenszwalb, P., Huttenlocher, D.: Fast algorithms for large state space HMMs with applications to web usage analysis. In: NIPS. (2003) 409–416
4. Felzenszwalb, P., Huttenlocher, D.: Efficient belief propagation for early vision. In: CVPR. (2004) I: 261–268
5. Kumar, M.P., Torr, P.H.S., Zisserman, A.: OBJ CUT. In: CVPR. (2005) I:18–25
6. Shental, N., Zomet, A., Hertz, T., Weiss, Y.: Learning and inferring image segmentation with the GBP typical cut algorithm. In: ICCV. (2003) 1243–1250
7. Vogiatzis, G., Torr, P.H.S., Seitz, S., Cipolla, R.: Reconstructing relief surfaces. In: BMVC. (2004) 117–126
8. Fischler, M., Elschlager, R.: The representation and matching of pictorial structures. TC **22** (1973) 67–92
9. Wainwright, M., Jaakkola, T., Willsky, A.: MAP estimation via agreement on (hyper)trees. Technical Report UCB/CSD-03-1226, UC Berkeley (2003)
10. Minka, T., Qi, Y.: Tree-structed approximations by expectation propagation. In: NIPS. (2003)
11. Chui, H., Rangarajan, A.: A new point matching algorithm for non-rigid registration. CVIU **89** (2003) 114–141

Appendix. The terms $q_{ik}(x_j)$ and $q_{ij}(x_k)$ described in equation (14) are obtained using the same label x_i.

Proof. The only terms which differ in $q_{ik}(x_j)$ and $q_{ij}(x_k)$ are $m_{nij\rightarrow ij}$ and $m_{nik\rightarrow ik}$ in the right-hand side of equation (14). Since all messages are initialized to 1 the proposition holds true for the first iteration. For subsequent iterations, consider the following equations:

$$m_{nij\to ij} \leftarrow \alpha \max_{x_n} \frac{\phi_n(x_n)\prod_{p,q\in\mathbf{R}_1}\phi_{pq}(x_p,x_q)\prod_{l\in\mathbf{S}\setminus\mathbf{R}_1} m_{ln\to n}m_{lni\to ni}m_{lnj\to nj}}{m_{ni\to i}m_{ni\to j}}, \tag{19}$$

$$m_{nik\to ik} \leftarrow \alpha \max_{x_n} \frac{\phi_n(x_n)\prod_{p,q\in\mathbf{R}_2}\phi_{pq}(x_p,x_q)\prod_{l\in\mathbf{S}\setminus\mathbf{R}_2} m_{ln\to n}m_{lni\to ni}m_{lnj\to nk}}{m_{ni\to i}m_{nk\to k}}, \tag{20}$$

where $\mathbf{R}_1 = \{n, i, j\}$ and $\mathbf{R}_2 = \{n, i, k\}$. The pairwise potentials $\phi_{ij}(x_i, x_j)$ and $\phi_{ik}(x_i, x_k)$ are constants for all $x_i \in \mathcal{C}_i(x_j)$ and $x_i \in \mathcal{C}_i(x_k)$ (over which the terms $q_{ik}(x_j)$ and $q_{ij}(x_k)$ are computed). The term $m_{lni\to ni}$ is common to both equations (19) and (20) and all other terms are constants for a particular pair of labels x_j and x_k. Thus, the above two messages are equivalent and it follows that $q_{ik}(x_j)$ and $q_{ij}(x_k)$ will be computed using the same label x_i.