HOP-MAP: Efficient Message Passing with High Order Potentials

Daniel Tarlow
Dept. of Computer Science
University of Toronto

Inmar E. Givoni
Probabilistic & Statistical
Inference Group
University of Toronto

Richard S. Zemel
Dept. of Computer Science
University of Toronto

Abstract

There is a growing interest in building probabilistic models with high order potentials (HOPs), or interactions, among discrete variables. Message passing inference in such models generally takes time exponential in the size of the interaction, but in some cases maximum a posteriori (MAP) inference can be carried out efficiently. We build upon such results, introducing two new classes, including composite HOPs that allow us to flexibly combine tractable HOPs using simple logical switching rules. We present efficient message update algorithms for the new HOPs, and we improve upon the efficiency of message updates for a general class of existing HOPs. Importantly, we present both new and existing HOPs in a common representation; performing inference with any combination of these HOPs requires no change of representations or new derivations.

1 INTRODUCTION

Probabilistic graphical models are powerful tools due to their representational power, and also due to general purpose algorithms that can be applied to any (low order) graphical model. For a broad range of problems, we can formulate a model in terms of graph structures and standard potentials. Then, without further derivation, we can automatically perform inference in the model. In particular, when the aim is to find a most likely configuration of variables (MAP), a range of efficient message passing algorithms can be applied (Wainwright et al., 2005; Werner, 2008; Globerson & Jaakkola, 2008).

When potentials begin to range over a large subset of variables, however, these methods quickly break down: for the general problem, message updates from a high order clique take time exponential in the size of the clique. One approach in such cases is to transform the problem into a pairwise problem by adding auxiliary variables. In the worst case, this will increase the problem size exponentially. Rother et al. (2009) give transformations that are practical for the special case of *sparse* potentials. Alternatively, special-purpose computations can sometimes be performed directly on the original high order—typically factor graph—representation (Givoni & Frey, 2009), which is the approach we take in this paper. This strategy applies beyond sparse potentials and is applicable for a wide range of special potential structures. Unfortunately, for a given potential it is typically not immediately clear whether it has tractable structure. Even if it does, message updates are case-specific and must be specially derived.

Our goal is to be able to use a broad range of high order potentials (HOPs) generically within MAP message passing algorithms as easily as we use low order tabular potentials. We see three issues that must be addressed:

1. Message updates need to be computed efficiently even when factors contain very large scopes.

2. It should be easy to recognize when problems contain tractable high order structure.

3. HOP constructions should be flexible and reusable, not requiring new problem-specific derivations and implementations on each use.

In Section 3, we describe the classes of atomic, building block HOPs that we consider, *cardinality-based* and *order-based*. Cardinality-based potentials have been used in several related works, and efficient message computations exist for both the general case and several restricted classes. Our first contribution, however, is showing that the efficiency can be improved even

beyond existing efficient computations (Potetz, 2007; Tarlow et al., 2008). Our algorithm computes all messages in O($N \log N$) time, a factor of $\frac{N}{\log N}$ better than existing approaches. For the novel class of order-based potentials, we present equally efficient algorithms. For the atomic potentials we consider, finding the *optimal* assignment takes O(N) or O($N \log N$) time. We show that our algorithms can compute *all* N outgoing messages in the same asymptotic time.

Next, analogous in spirit to *disciplined convex programming*, which allows many of the manipulations and transformations required to analyze and solve convex programs to be automated (Grant et al., 2006), we introduce two types of *composite* HOPs, which allow us to build more complex HOPs by applying composition rules based on maximization or simple logical switches. A complex HOP can be recognized to be tractable by decomposing it into atomic tractable units combined with allowed composition rules. Importantly, once expressed as a composition, the message updates of composite HOPs can automatically be computed from the message computations of the atomic HOPs.

Our final, more subtle contribution is the particular binary representation, message normalization scheme, and caching strategy for computing all outgoing messages from a factor at once. This "framework," which we refer to throughout, is not novel, but it is also not the standard. Each part has a purpose: the binary representation exposes more structure in potentials; the message normalization scheme yields simpler derivations; and the caching strategy leads to more efficient algorithms. It should be straightforward to apply this strategy to other structured potentials to build new atomic building block HOPs.

Section 4 shows that well-known and novel graph constructions are easily expressible using the vocabulary of potentials discussed, and that once a model is expressed in this framework, it can be used in a variety of MAP inference procedures. In Section 5 we present experimental results that illustrate the ease of constructing models of interest with this formulation.

2   REPRESENTATION

We work with a factor graph representation, which is a bipartite graph consisting of variable nodes, $\mathbf{h} = \{h_1, \ldots, h_n\}$, and factor nodes. Let $\mathcal{N}(h_j)$ be the neighbors of variable h_j in the graph. Factors, or potentials, $\theta = \{\theta_1, \ldots, \theta_n, \theta_{n+1}, \ldots, \theta_{n+k}\}$, define interactions over individual variables h_j and subsets of variables $\mathcal{C} = \{c_1, \ldots, c_k\}, c \subseteq \{h_1, \ldots, h_n\}$, which are exactly the factor's neighbors in the factor graph. With slight abuse of notation, we use θ_j to represent node potentials over single variables, and θ_c to repre-

sent HOPs over subsets. $\theta_j : h_j \in \{0, 1\} \rightarrow \mathbb{R}$ and $\theta_c : \mathbf{h}_c \in \{0, 1\}^{|c|} \rightarrow \mathbb{R}$ assign a real value to a variable or subset assignment, respectively. The potentials we present can range over any number of variables, so we use N to generically represent $|c|$.

A factor graph, then, defines a (log) likelihood that takes the form

$$L(\mathbf{h}) \quad = \quad \sum_{j=1}^{n} \theta_j(h_j) + \sum_{c \in \mathcal{C}} \theta_c(\mathbf{h}_c) \qquad (1)$$

The MAP inference problem—to find a setting of $\mathbf{h}$ that maximizes the likelihood $\mathbf{h}_{OPT} = \arg\max_{\mathbf{h}} L(\mathbf{h})$—is NP-hard for general loopy graphs, and we typically must resort to approximate optimization methods.

2.1   MAP MESSAGE PASSING

Max-product belief propagation (MPBP) is an iterative, local, message passing algorithm that can be used to find the MAP configuration of a probability distribution specified by a tree-structured graphical model.

When working in log space, the algorithm is known as max-sum, and the updates involve sending messages from factors to variables,

$$\tilde{m}_{\theta_c \rightarrow h_j}(h_j) = \max_{h_c \backslash \{h_j\}} \left[\theta_c(\mathbf{h}_c) + \sum_{j' \in c | j' \neq j} \tilde{m}_{h_{j'} \rightarrow \theta_c}(h_{j'}) \right],$$

and from variables to factors, $\tilde{m}_{h_j \rightarrow \theta_c}(h_j) = \sum_{c' \in \mathcal{N}(h_j) \backslash c} \tilde{m}_{\theta_{c'} \rightarrow h_j}(h_j)$. After a forward and backward pass sending messages to and from a root node, optimal assignments can be decoded from beliefs, $h_j^{OPT} = \arg\max_{h_j} b(h_j)$, where $b(h_j) = \sum_{c' \in \mathcal{N}(h_j)} \tilde{m}_{\theta_{c'} \rightarrow h_j}(h_j)$. In tree-structured graphs, beliefs defined in this way give (up to a constant) *max-marginals*: $\Phi_{j;a} = \max_{\mathbf{h}|h_j=a} L(\mathbf{h})$. In loopy graphs, beliefs produce pseudo max-marginals, which do not account for the loopy graph structure but can be decoded to give approximate solutions, which have been shown to be useful in practice.

2.2   BINARY REPRESENTATION

Note that we have restricted our attention to binary variable problems. To represent multinomial variables, we apply a simple transformation, converting variables with L states to binary variables with a 1-of-L constraint ensuring that exactly one variable in the set is on.

Since all variables are binary, we normalize messages so that the entry for $h_j = 0$ is always 0. To enforce this constraint, we subtract $\tilde{m}_{\theta_c \rightarrow h_j}(0)$ from both coordinates, giving us $\langle 0, \tilde{m}_{\theta_c \rightarrow h_j}(1) - \tilde{m}_{\theta_c \rightarrow h_j}(0) \rangle$. We can

then drop the argument for $m_{\theta_c \to h_j}(1)$ and use $m_{\theta_c \to h_j}$ to represent the scalar message difference. Similarly, we assume $\theta_j(0) = 0$ for all node potentials by setting $\theta_j = \theta_j(1) = \tilde{\theta}_j(1) - \tilde{\theta}_j(0)$.

We are then working with message differences, where a positive value indicates a variable's preference to be on, and a negative value indicates a variable's preference to be off. To recover a vector of properly scaled messages from message differences, we need the value of the optimal assignment relative to the potential of interest and the incoming messages, $M_j^{\theta_c} = \max_{\mathbf{h}_c} \left[\theta_c(\mathbf{h}_c) + \sum_{j' \in c | j' \neq j} h_j \cdot m_{h_{j'} \to \theta_c} \right]$. Define M^{θ_c} to be the maximum value when no j is left out of the sum. The correctly scaled message vector for h_j is then $\mathbf{M}_j^{\theta_c} = \left\langle M^{\theta_c} - \max(0, m_{\theta_c \to h_j}), M^{\theta_c} + \min(0, m_{\theta_c \to h_j}) \right\rangle$. Properly scaled max-marginals for the star surrounding a factor can then be seen as beliefs: $\Phi_j^{\theta_c} = \mathbf{M}_j^{\theta_c} + \langle 0, \theta_j \rangle$. We work in the scalar message difference representation, but this shows that we can convert to and from a vector representation if needed.

If we have a special purpose procedure for computing max-marginals over multinomial variables, as in Duchi et al. (2007), we can convert to a binary representation at the max-marginal level as well. Starting with max-marginals for a multinomial variable, c, which can take on one of M values, $j \in \{1, \ldots, M\}$, define binary variables $h_j = 1 \iff c = j$. A max-marginal for variable $h_j = 1$ is exactly the max-marginal for $c = j$. For $h_j = 0$, we can take the maximum max-marginal for $c = j' \mid j' \neq j$.

3 TRACTABLE HOPS

We now turn our attention to classes of HOP where max-marginals can be computed efficiently. The message computations require careful caching and bookkeeping; however, the efficient implementations as well as detailed derivations of the updates will be made publicly available[1].

3.1 CARDINALITY-BASED POTENTIALS

A broad class of existing tractable HOPs is where function values are based on the number of variables in the subset that are on:

$$\theta_{card}(h_1, \ldots, h_N) = f(\sum_j h_j)$$

Gupta et al. (2007) shows that the optimal single assignment can be computed for a *star-structured* model (i.e, containing a single HOP and arbitrary unary

_____
[1]http://www.cs.toronto.edu/~dtarlow/hops/

potentials) over cardinality in $O(N \log N)$ time by a simple sorting then greedy procedure. We could compute max-marginals for the star graph naively in $O(N^2 \log N)$ time by running this procedure iteratively fixing at each iteration one value of one variable.

If we use MPBP, Potetz (2007) shows that a single message can be approximately computed in $O(N)$ time, so approximate max-marginals could be computed in $O(N^2)$ time. If we compute all outgoing messages at the same time, Tarlow et al. (2008) shows that all N outgoing messages can be computed exactly in $O(N^2)$ time.

Here, we give a procedure for exactly computing all outgoing messages in $O(N \log N)$ – the same time it would take to compute the optimal assignment using the Gupta et al. (2007) procedure. We note that by using a dynamic heap and reusing previous sorts, the complexity may in reality be closer to $O(N)$ total ($O(1)$ amortized per message). Also note that this is in comparison to the $O(N \cdot 2^{N-1})$ time that it would take to compute these messages if the potential was represented in standard tabular form.

First, as usual, sort all incoming messages in descending order, yielding $m_{h_{j_b^*} \to \theta}$, where j_b^* is the index of the incoming message with b^{th} largest value (and conversely define $r(b) = j$ as the reverse index). Next, for $b \in \{0, \ldots, N\}$, in a linear pass, compute the cumulative sums $c_{-1}(b) = \sum_{b'=1}^{b} \left[m_{h_{j_{b'}^*} \to \theta} + f(b' - 1) \right]$; $c_0(b) = \sum_{b'=0}^{b} \left[m_{h_{j_{b'}^*} \to \theta} + f(b') \right]$; and $c_1(b) = \sum_{b'=0}^{b-1} \left[m_{h_{j_{b'}^*} \to \theta} + f(b' + 1) \right]$ In another linear pass, compute the cumulative *maxes* of the cumulative sums from the left, right, or both: $s_1^L(b) = \max_{b' \in \{0, \ldots, b\}} c_1(b')$; $s_{-1}^R(b) = \max_{b' \in \{b, \ldots, N\}} c_{-1}(b')$; $s_0^L(b) = \max_{b' \in \{0, \ldots, b\}} c_0(b')$; and $s_0^R(b) = \max_{b' \in \{b, \ldots, N\}} c_0(b')$.

The outgoing messages are then $m_{\theta \to h_j} = \tilde{m}_{\theta \to h_j}(1) - \tilde{m}_{\theta \to h_j}(0)$ where

$$\tilde{m}_{\theta \to h_j}(0) = \max \left(s_0^L(r(j) - 1), s_{-1}^L(r(j) + 1) - m_{h_j \to \theta} \right)$$

$$\tilde{m}_{\theta \to h_j}(1) = \max \left(s_1^R(r(j) - 1), s_0^R(r(j) + 1) - m_{h_j \to \theta} \right)$$

Intuitively, this is simply a dynamic programming procedure for computing maximizations over cumulative sums, which can be done in a few linear passes. Messages are then array lookups, so the complexity of computing all N outgoing messages is dominated by the initial sort operation, which will be $O(N \log N)$ in the worst case.

3.1.1 Special Cardinality Potentials

A simple special case of cardinality potentials is the potential that takes on value 0 if a set of variables are all on (i.e., $\sum_j h_j = N$) and a value of $-\alpha$ otherwise. When $\alpha > 0$, this encourages sets of variables to match specific patterns, so it has been referred to as a *pattern-based potential* (Komodakis & Paragios, 2009). By breaking the computation into separate cases for $\sum_j h_j = N$ and $\sum_j h_j \neq N$, it is straightforward to show that the messages take the form:

$$m_{\theta \to h_j} = \max\left(\alpha + \sum_{j':j' \neq j} \min\left(0, m_{h_{j'} \to \theta_{on}}\right),\ 0\right)$$

Due to this more restricted structure, all outgoing messages from a pattern-based potential can be computed in $O(N)$ (O(1) amortized) time by computing the full sum, $\sum_{j'} \min\left(0,\ m_{h_{j'} \to \theta_{on}}\right)$ then subtracting $\min\left(0,\ m_{h_j \to \theta_{on}}\right)$ for each individual message.

Another special case of cardinality potentials is the potential that constrains exactly b variables in a set to be on. Givoni and Frey (2009) gives updates to compute all messages from a single potential in $O(N)$ time, showing that additional structure allows us to compute messages more efficiently than the general case.

3.2 ORDER-BASED POTENTIALS

We now introduce examples from a new class of tractable HOP. These potentials depend on the ordering of variables within a factor, and they arise when trying to represent spatial, temporal, or other ordering considerations.

3.2.1 Convex Set Potentials

A convex set in one dimension is a contiguous subset. We define a *convex set* potential, θ_{cvx}, to be a potential that requires the variables with value 1 in an ordered set to form a convex set. Before deriving message updates, we must develop a notion of maximum weight contiguous sequences and an algorithm for efficiently finding them.

Suppose we have a vector of real-valued weights $w = \langle w_1, \ldots, w_N \rangle$. We define the maximum weight contiguous sequence problem as $\max_h w \cdot h$ such that $h_j \in \mathbf{h}$ with value 1 form a convex set. Define $MS(\text{range, constraint})$ as the value of the maximum weight contiguous subsequence in the given range obeying the constraint (or unconstrained if constraint is left out). For notational simplicity, if $MS(\text{range}, h_j = 1) < 0$, let the value be 0 instead. The scalar message $m_{\theta_{cvx} \to h_j}$ is

$$MS(h_{1:j-1}, h_{j-1} = 1) + MS(h_{j+1:N}, h_{j+1} = 1)$$
$$- \max\left(MS(h_{1:j-1}), MS(h_{j+1:N})\right)$$

We can compute the constrained and unconstrained maximum weight contiguous subsequences we need for all outgoing messages in linear time using dynamic programming. Given this, computing a message requires only four array lookups, so the total time to compute messages is $O(N)$ (O(1) amortized).

3.2.2 Before-After Potentials

We define a *before-after* potential over two ordered subsets of variables that encourages one to come before the other:

$$\theta_{\to}(\mathbf{h}_x, \mathbf{h}_y) = \begin{cases} -\infty & \text{if } (\sum_{i \in x} h_i) \neq 1 \vee (\sum_{j \in y} h_j) \neq 1 \\ 0 & \text{if } i > j | i \in x, j \in y, h_i = h_j = 1 \\ -\alpha & \text{otherwise} \end{cases}$$

The key to computing messages for this factor is to condition on the hard constraint being satisfied then into cases $i > j$ and $i < j$. We compute maximums independently, then we take the maximum over cases where α is subtracted from the case $i < j$.

As with the cardinality potentials, we take cumulative maxes over the incoming x messages and incoming y messages from both the left and the right, respectively: $s_x^L(k), s_x^R(k), s_y^L(k), s_y^R(k)$ for $k \in \{1, \ldots, N\}$. Ignoring edge cases and only dealing with messages to $h_i | i \in x$ for simplicity, the basic form of the messages are $\tilde{m}_{\theta_{\to} \to h_i}(1) = \max(s_y^L(i-1) - \alpha, s_y^R(i+1))$ and $\tilde{m}_{\theta_{\to} \to h_i}(0) = M^{\theta_{\to}}$ for all cases except to h_{i^*}, where $M^{\theta_{\to}} = \max\left(\max_k \left[s_x^L(k) + s_y^R(k+1)\right], \max_k \left[s_y^L(k) + s_x^R(k+1) - \alpha\right]\right)$ for $k \in \{1, \ldots, N-1\}$ and i^* is the choice of i used in the optimal assignment of i and j. For messages to h_{i^*}, we need the maximum setting of i and j where $i \neq i^*$. This can easily be computed in another linear pass. As usual, $m_{\theta_{\to} \to h_i} = \tilde{m}_{\theta_{\to} \to h_i}(1) - \tilde{m}_{\theta_{\to} \to h_i}(0)$. Updates to $h_j \in y$ are almost identical but reversed.

This computation leverages the same caching ideas as the general cardinality computations, which should be broadly applicable across other classes of HOP as well. Again, after the proper linear processing, the messages we need to send are just array lookups, so the total complexity is $O(N)$ to compute all messages, or O(1) per message amortized.

3.3 COMPOSITION OF POTENTIALS

In many cases, we may wish to switch between HOPs, based on some simple logical rules. Typically, this would require deriving new message updates, even for small changes. To address this problem, we present the class of composite potentials., which can be viewed as an extension of context-specific independencies, where rules specify tractable HOPs instead of constant values (Boutilier et al., 1996).

Suppose we have two disjoint subsets of $\mathbf{h}_c$, $\mathbf{h}_s \cup \mathbf{h}_p = \mathbf{h}_c$ and $\mathbf{h}_s \cap \mathbf{h}_p = \emptyset$. Define a composite potential

$$\theta_\phi(\mathbf{h}_c) = \theta_{g(\mathbf{h}_s)}(\mathbf{h}_p)$$

where $g : \{0,1\}^{|\mathbf{h}_s|} \to \{0,\ldots,K-1\}$ assigns each setting of $\mathbf{h}_s$ to one of K partitions, and there is a different active potential $\theta_{g(\mathbf{h}_s)}$ depending on the partition.

If we condition on $g(\mathbf{h}_s) = k$, the maximizations we need decouple into $\max_{\mathbf{h}_s | g(\mathbf{h}_s)=k} \left[\sum_{i'} m_{h_{i'} \to \theta_\phi}(h_{i'}) \right] + \max_{\mathbf{h}_p \backslash h_j} \left[\theta_k(\mathbf{h}_p) + \sum_{j' \neq j} m_{h_{j'} \to \theta_\phi}(h_{j'}) \right]$. In most practical cases, $|\mathbf{h}_s|$ is small (and as it grows, the potential becomes an intractable non-structured HOP), so we do the first maximizations by enumerating all possible values of $\mathbf{h}_s$, yielding $M^{\theta_s}(k)$—the maximum of the first term conditioned on $g(\mathbf{h}_s) = k$.

We use efficient message computations to compute properly scaled max-marginals for the subset of variables in $\mathbf{h}_p$ relative to θ_k—this is just computing max-marginals for the star graph around θ_k, ignoring $\mathbf{h}_s$. We know how to do this efficiently because θ_k is assumed to be a tractable HOP. We define these values to be $M_{j;h_j}^{\theta_k}(k)$ for potential k and h_j fixed to take on value of 0 or 1. Finally, compute

$$\tilde{m}_{\theta_\phi \to h_j}(0) = \max_k \left[M^{\theta_s}(k) + M_{j;0}^{\theta_k}(k) \right]$$

$$\tilde{m}_{\theta_\phi \to h_j}(1) = \max_k \left[M^{\theta_s}(k) + M_{j;1}^{\theta_k}(k) \right]$$

by evaluating all combinations of $k \in \{0,\ldots,K-1\}$ and $h_j \in \{0,1\}$, which requires computing all outgoing messages from K tractable HOPs, yielding an amortized message cost of $O(2^{|\mathbf{h}_s|}+K)$ or $O(2^{|\mathbf{h}_s|}+K\log N)$ depending on the type of HOP. Note that the optimal value of k may be different for each maximization. The messages to variables in $\mathbf{h}_s$ are similar, but we omit them due to space.

A related composite factor is one where there is no preference for $\mathbf{h}_s$ to take any particular value (i.e., for all $h \in \mathbf{h}_s$, h is only in the scope of one composite factor). In this case, the HOPs are switched between based only on which is most likely, and since messages to $\mathbf{h}_s$ will only be useful to infer which potential was chosen, we can choose not to compute them if we are only interested in the assignment of the $\mathbf{h}_p$ variables. This is a *max-composition* potential, and the truncated linear deviation pattern potentials of Rother et al. (2009) can be seen as a special case of this composition.

The binary formulation of affinity propagation (Givoni & Frey, 2009), for example, can be thought of as using a composite potential for columns, where $\mathbf{h}_s = \{h_{jj}\}$, $\mathbf{h}_p = \{h_{j-j}\}$, $g(0) = 0$, $g(1) = 1$, θ_0 is an all-off potential, and $\theta_1 = 0$.

4 USING HOPS

Graph Constructions

There are many existing and novel models that can be constructed by combining HOPs in different ways. Here we give a few examples of the range of models that can be constructed using the HOP vocabulary.

Pattern potentials are becoming popular in several computer vision applications. b-of-N potentials can be used: in general b-matching problems; to represent the set cover problem in conjunction with a composite potential that enforces an all-on or all-off constraint; and as hard degree constraints on nodes in a structure learning framework.

Cardinality potentials are useful in many cases beyond b-of-N potentials or pattern potentials, though. Hard or soft parity potentials can be expressed by setting $f(k) = 0$ when k is even, and $f(k) = -\alpha$ when k is odd. They can further be used to express nonparametric priors over cluster sizes in an exemplar-based clustering setting; they can encourage a specific percentage of pixels to be on in an image segmentation setting; and they can be used to represent a learned distribution of sizes for empirical image segmentation priors.

Before-after potentials are useful for representing user preferences of document i over document j in a ranking setting; temporal ordering information in a time-warped signal matching setting; soft word ordering preferences in a language parsing setting; or the soft temporal ordering of subactivities in an activity recognition setting.

Convex set potentials are useful for representing constraints that parts be convex in an image segmentation setting; words to be contiguous in a language parsing setting; or objects appear in a contiguous region of time in a recognition-based tracking setting.

Importantly, because the underlying mechanism is message passing on standard factor graphs, any combination of these potentials or composed variations can readily be combined and seamlessly integrated with standard local potentials.

Outer Loop Algorithms

Our contribution in this work lies in the inner loop of inference—in the subroutines used to calculate messages in the message passing framework.

We emphasize that several "outer loop" inference routines can take advantage of these message-computation subroutines. Most obviously, MPBP defined on a cluster graph with single variable separator sets can use the presented message computations
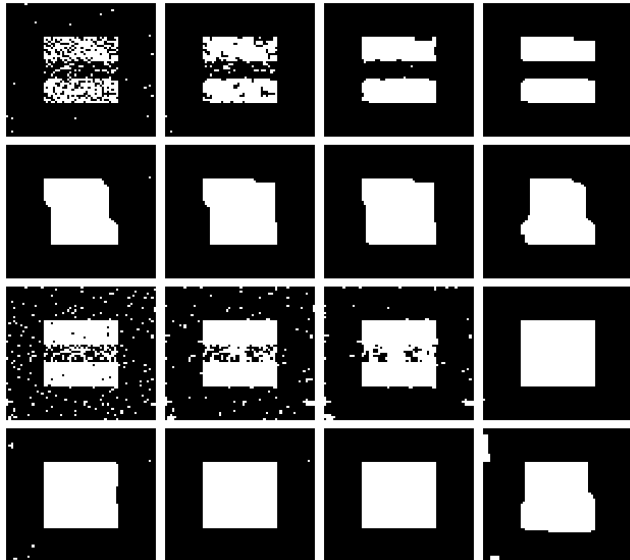
Figure 1: Image segmentation results for a 64 x 64 square using asynchronous max-product belief propagation and combinations of high order cardinality and convexity potentials. Rows from top to bottom: only unary and pairwise; unary, pairwise, and convexity; unary, pairwise, and cardinality; unary, pairwise, convexity, and cardinality. Columns from left to right: pairwise strength 0, .05, .1, .5.

without modification, though schedules that compute all outgoing messages from a high order factor at once will be most efficient. Globerson and Jaakkola (2008) give a generalized edge-based variant of max-product linear programming (GEMPLP) that can use standard HOP computations but requires that (easily calculated) backwards messages be added to the output. Komodakis and Paragios (2009) mention a high order variant of tree-reweighted max-product (TRW) (Wainwright et al., 2005) that can use these computations to directly compute the needed star graph max-marginals.

5 EXPERIMENTS

We have explored HOP constructions for several problems in combinatorial optimization, ranking, and vision. However, we leave a thorough exploration of these applications and comparison of outer-loop MAP algorithms to future work. Here, we focus on an image segmentation and a ranking example, demonstrating that high order models can be constructed easily and used within a variety of outer loop algorithms.

Image Segmentation

We generated 64 x 64 pixel synthetic data for a foreground-background image segmentation task that simulates a fairly strong foreground response in a

square region of the image, then a semi-occluded region (horizontal bar) where there is little distinction from the background. We added uniform pairwise potentials to the 4-neighborhood of each pixel, and we experimented with different combinations of HOPs: no extra potentials; convex set potentials along horizontal, vertical, $45°$, and $135°$ diagonals; cardinality potentials $f(k) = -|\frac{64^2}{4} - k|$; and a combination of both cardinality and convex set potentials.

We run asynchronous belief propagation, using a typical grid schedule, where we pass messages down then up along columns, to neighboring columns, then moving across the image and repeating for the next column. We follow this with a similar scheme for rows. After one round of these messages, we iterate through all HOPs and update all messages going into the current high order factor, then all messages outgoing from the HOP. We use damping of .5, and run until messages converge.

Fig. 1 shows decoded beliefs for a variety of pairwise potential strengths and HOP combinations. Note that there is no setting of pairwise potentials that identifies the region as one square (top row). However, when we add the HOPs, it becomes easy to express priors that encourage the single square interpretation. Additionally, the algorithm is reasoning in a non-trivial way, especially in the cardinality case as pairwise potentials get stronger (third row). When pairwise potentials are weak (left columns), the algorithm chooses the pixels with highest singleton potential to turn on. As the pairwise potentials get stronger (right columns), it must balance singleton potentials with spatial contiguity.

Rank Aggregation

When ranking documents for information retrieval, it is common to use a set of document- and query-specific features to learn a score for each document-query combination. When eliciting preferences from users, however, it is difficult for them to produce a real-valued score for a query-document pair. Instead, users typically can express pairwise preferences (i.e., they prefer document i to document j for some query).

We simulate this setting by building a query-specific model where N documents can take on one of N ranks, represented as an N x N binary grid of variables h_{ij}. Each document has a score, $s_i \sim \text{Uniform}(0, 1)$, and we use unary potentials of the form $\theta_{ij}(h_{ij}) = (N-j) \cdot s_i$ to express that we prefer documents with higher scores to be ranked higher (closer to 1). We simulate user preferences by adding before-after potentials of random strength over 10% of randomly chosen pairs of rows. We also scale the overall strength of all before-after potentials by multiplying by a common factor, λ,

$$\lambda = 0.01 \qquad \lambda = 0.05 \qquad \lambda = 0.25$$
$$\rho = 13/18 \qquad \rho = 15/18 \qquad \rho = 18/18$$
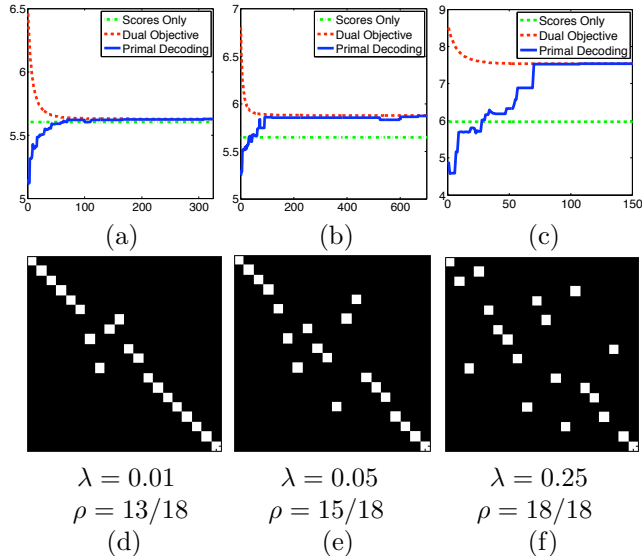$$\text{(d)} \qquad\qquad \text{(e)} \qquad\qquad \text{(f)}$$

Figure 2: **(a - c)** Primal and dual objective for ranking using high order TRW. λ gives the strength of before-after potentials representing binary user preferences of one document over another. **(d - f)** Final assignments found when λ is varied, all of which are globally optimal. Documents are rows, ranks are columns. Document-specific scores encourage document i to be ranked in position i. ρ gives the fraction of soft pairwise order constraints that are satisfied.

giving us $\theta_\rightarrow(\mathbf{h}_{i:}, \mathbf{h}_{k:}; \lambda)$. This objective incorporates both document-specific scores and elicited user preferences. Note that if all document scores are zero, this can represent the NP-hard Kemeny rank aggregation problem (Bartholdi et al., 1989).

In addition, we add the full bipartite matching potential described in Duchi et al. (2007), enforcing the constraint that each document can only take on one rank, and each rank can only be occupied by one document. We do inference with the simple high-order generalization of TRW given in Komodakis and Paragios (2009). To decode assignments, we employ a greedy strategy that is guaranteed to enforce the uniqueness constraints: find the largest belief, b_{ij}, set the corresponding $h_{ij} = 1$, then set beliefs in row i and column j to $-\infty$ and repeat until we have a full assignment. For comparison, we also compute the objective of an algorithm that ranks based solely on score. The dual objective always met the primal decoded objective, so we stopped inference when we were guaranteed to have found the global optimum. Fig. 2 shows results for several λ.

6 RELATED WORK

High Order Message Passing

In this work, we put into a single framework and ex-

panded upon several recent works. See Fig. 3 for an overview. Givoni and Frey (2009) use efficient updates for various b-of-N potentials in a MPBP setting; Huang and Jebara (2007) develop an efficient MPBP algorithm for weighted bipartite b-matching and prove that the algorithm finds the optimal assignment; Duchi et al. (2007) shows how combinatorial algorithms can efficiently compute messages in models that have a full bipartite matching or associative MRF as a subproblem; and Komodakis and Paragios (2009) shows how to compute max-marginals for pattern-based potentials. Potetz (2007) and Tarlow et al. (2008) show how to compute messages for the general cardinality case and Werner (2008) briefly sketches a simple algorithm for doing so, but none gives the more efficient algorithm we have presented here.

Other Decompositions

Dual decomposition (DD) (Bertsekas, 1999) is a framework for combining optimal subproblem solutions rather than max-marginals. Our potentials could easily be used as subproblems in this framework, but it would be simpler to directly compute subproblem assignments rather than doing so via max-marginals. DD has become popular very recently as an alternative to belief propagation, but few direct comparisons have been done on equivalent problem decompositions. Our formulation provides a means to perform a comparison of DD and message passing approaches, because it allows message passing algorithms to be defined on equivalent problem decompositions for many of the problems that DD has been applied to.

Finally, certain restricted types of HOP have also been used in a graph cuts framework (Kohli et al., 2007, 2009; Delong & Boykov, 2009). When a problem can be formulated to be submodular, these approaches will always find the global optimum. As models become more complex, however, it takes more effort to manipulate the model into a compliant form—often involving transformations, creation of auxiliary variables, and sometimes non-intuitive restrictions on the problem formulation. Our work, in contrast, applies uniformly to a broader range of problems, and it is intuitive to construct a model using the message passing formulation. However, we see discovering further connections and developing hybrid algorithms that work with ideas from both approaches as an open direction of work.

7 CONCLUSION

In many low order models where we can compute global optimums, the assignments still do not match ground truth. This indicates we need richer, possibly higher order representations. A challenge in working with HOPs is that it is often difficult to recognize when

	Special Cardinality		Pattern	General Cardinality	Order	General Composition
	b of N	>/< b of N				
MPBP	[Giv] [H] [C]	[Giv]	*	[T]**	*	*
MPBP Variants	*	*	[Kom]	[W]**	*	*
DD (OPT only)	[Tor]	*	[Kom]	[Vic] [Gup]	*	*

Figure 3: Combinations of potentials and inference algorithms in the literature. OPT only means the method does not use max-marginals. * Can express using presented HOPs and to our knowledge is novel. ** We improve upon the efficiency of these calculations. [C] (Cheng et al., 2006), [G] (Givoni & Frey, 2009), [Gup] (Gupta et al., 2007), [H] (Huang & Jebara, 2007), [Kom] (Komodakis & Paragios, 2009), [T] (Tarlow et al., 2008), [Tor] (Torresani et al., 2008), [W] (Werner, 2008), [Vic] (Vicente et al., 2009).

a problem formulation will be tractable. Our vocabulary of common HOPs should make it easier to identify problem structures and formulate appropriate potentials that can be incorporated into efficient message passing MAP inference algorithms.

We emphasize that the HOPs and composition rules that we have presented are not exhaustive. There are likely to be many other classes of tractable HOP and compositions. In these cases, the strategies presented here can likely be applied to build new classes of HOPs.

A limitation of our work as presented here is that factors only communicate with the rest of the network via single variable interfaces. To get communication, for example, between planar faces in a generalized MPBP setting, and to get tighter LP relaxations, we would like our factors to communicate via separator sets of several variables. In theory, there is nothing preventing us from deriving such algorithms; this is an interesting direction of future work.

Acknowledgements

References

Bartholdi, J., Tovey, C. A., & Trick, M. A. (1989). Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare*, *6*(3).

Bertsekas, D. (1999). *Nonlinear Programming*.

Boutilier, C., Friedman, N., Goldszmidt, M., & Koller, D. (1996). Context-specific independence in bayesian networks. In *Uncertainty in artificial intelligence (uai)* (pp. 115–123).

Cheng, Y.-S., Neely, M. J., & Chugg, K. M. (2006). Iterative message passing algorithm for bipartite maximum weighted matching. In *Proc. of IEEE International Symposium on Information Theory*.

Delong, A., & Boykov, Y. (2009). Globally optimal segmentation of multi-region objects. In *International Conference on Computer Vision, (ICCV)*.

Duchi, J., Tarlow, D., Elidan, G., & Koller, D. (2007). Using combinatorial optimization within max-product belief propagation. In *Advances in Neural Information Processing Systems 19* (pp. 369–376). Cambridge, MA: MIT Press.

Givoni, I., & Frey, B. J. (2009). A binary variable model for affinity propagation. *Neural Computation*, *21*(6), 1589-600.

Globerson, A., & Jaakkola, T. (2008). Fixing max product: Convergent message passing algorithms for MAP LP-relaxations. In *Neural Information Processing Systems*.

Grant, M., Boyd, S., & Ye, Y. (2006). *Global optimization: From theory to implementation* (L. Liberti & N. Maculan, Eds.).

Gupta, R., Diwan, A., & Sarawagi, S. (2007). Efficient inference with cardinality-based clique potentials. In *International Conference on Machine Learning ICML* (Vol. 227, p. 329-336).

Huang, B., & Jebara, T. (2007). Loopy belief propagation for bipartite maximum weight b-matching. In *The Eleventh International Conference on Artificial Intelligence and Statistics*.

Kohli, P., Kumar, M. P., & Torr, P. (2007). P3 and beyond: Solving energies with higher order cliques. In *Computer Vision and Pattern Recognition (CVPR)*.

Kohli, P., Ladický, L., & Torr, P. H. (2009). Robust higher order potentials for enforcing label consistency. *Int. J. Comput. Vision*, *82*(3), 302–324.

Komodakis, N., & Paragios, N. (2009). Beyond pairwise energies: Efficient optimization for higher-order MRFs. In *Computer Vision and Pattern Recognition (CVPR)*.

Potetz, B. (2007). Efficient belief propagation for vision using linear constraint nodes. In *Cvpr 2007: Proceedings of the 2007 ieee computer society conference on computer vision and pattern recognition*. Minneapolis, MN, USA: IEEE Computer Society.

Rother, C., Kohli, P., Feng, W., & Jia, J. (2009). Minimizing sparse higher order energy functions of discrete variables. In *Computer Vision and Pattern Recognition (CVPR)*.

Tarlow, D., Zemel, R., & Frey, B. (2008). Flexible priors for exemplar-based clustering. In *Uncertainty in Artificial Intelligence*.

Torresani, L., Kolmogorov, V., & Rother, C. (2008). Feature correspondence via graph matching: Models and global optimization. In *European Conference on Computer Vision, (ECCV)*.

Vicente, S., Kolmogorov, V., & Rother, C. (2009). Joint optimization of segmentation and appearance models. In *International Conference on Computer Vision, (ICCV)*.

Wainwright, M. J., Jaakkola, T., & Willsky, A. S. (2005). MAP estimation via agreement on trees: message-passing and linear programming. *IEEE Transactions on Information Theory*, *51*(11), 3697-3717.

Werner, T. (2008). High-arity interactions, polyhedral relaxations, and cutting plane algorithm for soft constraint optimisation (MAP-MRF). In *Computer Vision and Pattern Recognition (CVPR)*.