Energy Minimization for Linear Envelope MRFs

Pushmeet Kohli
Microsoft Research
Cambridge, UK
pkohli@microsoft.com

M. Pawan Kumar*
Computer Science Department
Stanford University, USA
pawan@cs.stanford.edu

Abstract

Markov random fields with higher order potentials have emerged as a powerful model for several problems in computer vision. In order to facilitate their use, we propose a new representation for higher order potentials as upper and lower envelopes of linear functions. Our representation concisely models several commonly used higher order potentials, thereby providing a unified framework for minimizing the corresponding Gibbs energy functions. We exploit this framework by converting lower envelope potentials to standard pairwise functions with the addition of a small number of auxiliary variables. This allows us to minimize energy functions with lower envelope potentials using conventional algorithms such as BP, TRW *and* α*-expansion. Furthermore, we show how the minimization of energy functions with upper envelope potentials leads to a difficult min-max problem. We address this difficulty by proposing a new message passing algorithm that solves a linear programming relaxation of the problem. Although this is primarily a theoretical paper, we demonstrate the efficacy of our approach on the binary (fg/bg) segmentation problem.*

1. Introduction

Markov random fields (MRF) provide a powerful framework for concisely representing the Gibbs energy of a set of random variables, which makes them immensely useful in computer vision. Specifically, they express the energy of a labeling (a particular assignment of values to the random variables) as a sum of potentials, each of which depends on a subset (more precisely, a clique) of random variables. Typically, the degree of the potential (that is, the size of the corresponding clique) is restricted to one (unary potential) or two (pairwise potential), which corresponds to a pairwise energy function. Such an energy function can be efficiently minimized using one of many accurate algorithms that have been proposed in the literature. However, despite substantial work from several communities, solutions to computer vision problems based on pairwise energy functions have met with limited success so far. This observation has led researchers to question the richness of these classical energy functions, which in turn has motivated the development of more sophisticated models. Along these lines, many have turned to the use of higher order potentials that give rise to more expressive MRFs, thereby allowing us to capture the natural image statistics.

The last few years have seen a large number of higher order potentials being proposed for different vision problems [10, 18, 23, 25, 34, 36]. Although such potentials provide us with the required modeling power, they also present a difficult energy minimization scenario (since the number of possible labelings of a potential is exponential in its degree). In order to address this difficulty, we show that (somewhat surprisingly) many higher order potentials used in computer vision have a special structure that allows us to represent them compactly. In order to fully exploit this structure, we propose a novel representation based on *upper* and *lower* envelopes of linear functions defined over the space of possible labelings for a clique of random variables.

Our representation provides a unified framework for the problem of energy minimization with higher order potentials. Specifically, as shown in previous work [26], lower envelope higher order potentials can be transformed to an equivalent sum of pairwise potentials with the addition of (typically, a small number of) auxiliary variables. Once reformulated in this manner, we can employ standard algorithms for pairwise energy functions to obtain the desired result. In contrast, converting the new upper envelope potentials to a pairwise form leads to a difficult min-max problem. To tackle this difficulty, we propose a new linear programming (LP) relaxation for minimizing energy functions with upper envelope higher order potentials. Our relaxation is a natural extension of the standard LP relaxation for pairwise energy functions [5]. Furthermore, we present an efficient algorithm for solving the relaxation.

2. Related Work

As earlier approaches in computer vision and related areas restricted themselves to pairwise MRFs, it is not surprising that most of the work on energy minimization focused on pairwise energy functions. These approaches can broadly be classified into two categories: message passing and move making. Message passing algorithms attempt

*Supported by NSF under grant IIS 0917151, MURI contract N000140710747, and the Boeing company.

to minimize approximations of the free energy associated with the MRF [8, 13, 15, 20, 29, 32, 37]. Move making approaches refer to iterative algorithms that *move* from one labeling to the other while ensuring that the energy of the labeling never increases. The move space (that is, the search space for the new labeling) is restricted to a subspace of the original search space that can be explored efficiently [4, 6, 16, 17]. Many of the above approaches (both message passing [13, 15, 32] and move making [4, 16, 17]) have been shown to be closely related to the standard LP relaxation for the pairwise energy minimization problem [5].

Although there has been work on applying message passing algorithms for minimizing certain classes of higher order energy functions [23, 30], the general problem has been relatively ignored. Traditional methods for minimizing higher order functions involve converting them to a pairwise form by addition of auxiliary variables, followed by minimization using one of the standard algorithms for pairwise functions (such as those mentioned above) [3]. However, such an approach suffers from the problem of combinatorial explosion. Specifically, a naive transformation can result in an exponential number of auxiliary variables (in the size of the corresponding clique) even for higher order potentials with special structure.

In order to avoid the undesirable scenario presented by the naive transformation, researchers have now started focusing on higher order potentials that afford efficient algorithms [10, 11, 31, 34, 36, 35]. Most of the efforts in this direction have been towards identifying useful families of higher order potentials and designing algorithms specific to them. While this approach has led to improved results, its long term impact on the field is limited by the restrictions placed on form of the potentials. To address this issue, some recent works [9, 14, 26] have attempted to characterize the higher order potentials that are amenable to optimization. Although these works have been able to exploit the *sparsity* of potentials, unlike our framework, they do not provide a convenient parameterization of tractable potentials.

3. Preliminaries

Markov Random Fields Consider a set of scene elements (such as pixels or voxels) $\mathcal{V} = \{1, 2, \cdots, n\}$ and a set of random variables $\mathbf{X} = \{X_i, i \in \mathcal{V}\}$ corresponding to them. Each random variable X_i can take a label x_i from the label set $\mathcal{L} = \{1, \cdots, h\}$. For example, in scene segmentation the labels can represent semantic classes such as building, tree or person. Clearly, in the above scenario the total number of labelings $\mathbf{x}$ (a particular assignment of labels to all the random variables) is h^n. In order to distinguish one labeling from the other quantitatively, an MRF defined over $\mathbf{X}$ specifies a Gibbs energy function of the form

$$E(\mathbf{x}; \mathbf{D}) = \sum_{c \in \mathcal{C}} \psi_c(\mathbf{x}_c), \tag{1}$$

where $\mathbf{D}$ is the observed data (such as RGB values of image pixels). The term $\psi_c(\mathbf{x}_c)$ denotes the value of the clique potential corresponding to the labeling $\mathbf{x}_c \subseteq \mathbf{x}$ for the clique c, and $\mathcal{C}$ is the set of all cliques. As noted before, the degree of the potential $\psi_c(\cdot)$ is the size of the corresponding clique c (denoted by $|c|$). For the well-studied special case of pairwise MRFs, the energy only consists of potentials of degree one or two, that is,

$$E(\mathbf{x}; \mathbf{D}) = \sum_{i \in \mathcal{V}} \psi_i(x_i) + \sum_{(i,j) \in \mathcal{E}} \psi_{ij}(x_i, x_j), \tag{2}$$

where $\mathcal{E}$ represents the set of neighboring random variables (that is, cliques of size two).

Energy Minimization Given an MRF, the problem of energy minimization consists of finding the labeling $\mathbf{x}$ that has the lowest energy. Formally, energy minimization involves solving the following problem:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} E(\mathbf{x}; \mathbf{D}). \tag{3}$$

As the Gibbs energy is related to the posterior probability of the labeling, that is,

$$\Pr(\mathbf{x}|\mathbf{D}) \propto \exp\left(-E(\mathbf{x}; \mathbf{D})\right), \tag{4}$$

energy minimization is also sometimes referred to as maximum *a posteriori* estimation. In what follows, we often consider the minimization of a single higher order potential instead of the entire energy function. However, all our results can be trivially extended to energy functions since they only involve the summation of a finite number of potentials.

4. Envelopes of Linear Functions

We will now explain our envelope representation that will be used to transform the problem of minimizing higher order potential functions to problems over pairwise functions. We define higher order potentials as lower or upper envelopes of linear functions. Each linear function is of the following form:

$$f^q(\mathbf{x}_c) = \mu^q + \sum_{i \in c} \sum_{a \in \mathcal{L}} w^q_{ia} \delta_i(a), \tag{5}$$

where the function $\delta_i(a)$ returns 1 if variable X_i takes label a (that is, $x_i = a$) and returns 0 for all other labels. The weights w^q_{ia} and the constant term μ^q are the parameters of the function $f^q(\cdot)$. Given the above definition, our higher order function representation can be written as

$$\psi_c(\mathbf{x}_c) = \otimes_{q \in \mathcal{Q}} f^q(\mathbf{x}_c), \tag{6}$$

where $\otimes \in \{\max, \min\}$. While 'min' results in a lower envelope of the linear function, 'max' results in the upper envelope (see figure 1). As will be seen later, the choice of 'min' or 'max' has a big impact on the difficulty of the corresponding energy minimization problem.

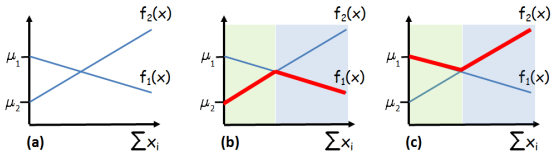

Figure 1. *Lower and upper envelopes of linear functions and their relationship to convex and concave functions. The figure shows two linear functions* $f^1 : \{0,1\}^n \rightarrow \mathbb{R}$ *and* $f^2 : \{0,1\}^n \rightarrow \mathbb{R}$. *The weights for* f^1 *are:* $\mu^1 = 0$, $w^1_{i0} = 0$ *and* $w^1_{i1} > 0$ *while those for* f^2 *are:* $\mu^2 = 0$, $w^2_{i0} = 0$ *and* $w^2_{i1} < 0$. *It can be easily seen that lower envelope results in a concave function whereas the upper envelope results in a convex function.*

Completeness of the representation The linear envelope representation is general and any higher order potential function can be written in this form. In other words, linear envelopes are *complete*. To see this, consider a potential function $\psi_c : \mathcal{L}^{|c|} \rightarrow \mathbb{R}$. This can be converted to our lower envelope representation ($\otimes = \min$) using $h^{|c|}$ linear functions as follows (where $h = |\mathcal{L}|$). We denote the q^{th} labeling (out of a possible $h^{|c|}$) of the clique as $\mathbf{l}^q = \{l^q_i | i \in c\}$. For this labeling, we define a linear function $f^q(\mathbf{x}_c)$ using

$$\mu^q = \psi_c(\mathbf{l}^q) \quad \text{and} \quad w^q_{ia} = \begin{cases} 0 & \text{if } \; l^q_i = a, \\ \infty & \text{otherwise.} \end{cases}$$

It can be easily seen that $\psi_c(\mathbf{x}_c) = \min_q f^q(\mathbf{x}_c)$.

Although completeness is a good property of the representation, it comes at a high cost of defining an exponential number of linear functions. In the next section we will show that many useful higher order potentials can be written in our envelope representation (6) using constant or polynomial (in $|c|$ and h) number of linear functions.

4.1. Lower Envelope Higher order Potentials

We now provide examples of some useful higher order potential functions that can be represented as a lower envelope of linear functions.

Region Consistency A common method to solve various image labeling problems like object segmentation, stereo and single view reconstruction is to formulate them using image segments (so called superpixels [24]) obtained from unsupervised segmentation algorithms. Researchers working with these methods have made the observation that all pixels constituting the segments often have the same label, that is they might belong to the same object or might have the same depth. This observation has motivated the proposal of higher order potentials that encourage label consistency in sets of pixels. One such potential is the recently proposed P^n Potts model [10], that is,

$$\psi_c(\mathbf{x}_c) = \begin{cases} \gamma_a & \text{if} \quad x_i = a, \;\; \forall i \in c, \\ \gamma_{\max} & \text{otherwise,} \end{cases} \tag{7}$$

where $\gamma_{\max} \geq \gamma_a$, for all $a \in \mathcal{L}$. In other words, the above potential assigns a constant penalty $\gamma_{\max}$ to all solutions

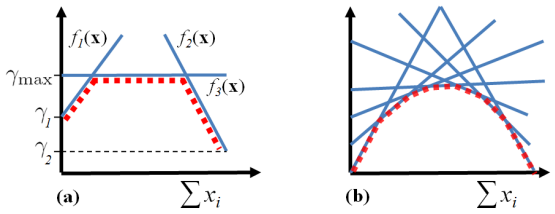

Figure 2. *(a) Robust* P^n *model for binary variables. The linear functions* f_1 *and* f_2 *represents the penalty for variables not taking the labels* 0 *and* 1 *respectively. The function* f_3 *represents the robust truncation factor. (b) The general concave form of the robust* P^n *model defined using a larger number of linear functions.*

that do not assign the same label to all pixels in a segment c. This was generalized by Kohli et al. [11] to obtain the Robust P^n model defined as

$$\psi_c(\mathbf{x}_c) = \min\{\min_{a \in \mathcal{L}}((|c| - n_a(\mathbf{x}_c))\alpha_a + \gamma_a), \gamma_{\max}\}, \tag{8}$$

where $|c|$ is the number of variables in clique c, $n_a(\mathbf{x}_c)$ denotes the number of variables in clique c that take the label a in labeling $\mathbf{x}_c$, and $\alpha_a, \gamma_a, \gamma_{\max}$ are potential function parameters that satisfy the constraints

$$\alpha_a = \frac{\gamma_{\max} - \gamma_a}{M} \text{ and } \gamma_a \leq \gamma_{\max}, \forall a \in \mathcal{L}. \tag{9}$$

The term M is called the truncation parameter of the potential and satisfies the constraint $2M < |c|$. They also showed that multiple instances of this potential can be combined to obtain any concave function over the number of segment pixels that do not take the dominant label in a labeling. This potential takes the form

$$\psi_c(\mathbf{x}_c) = \min\{\min_{a \in \mathcal{L}} \mathcal{F}_c((|c| - n_a(\mathbf{x}_c))), \gamma_{\max}\}, \tag{10}$$

where $\mathcal{F}_c$ is a non-decreasing concave function. Recall that a function $f(x)$ is concave if for any two points (u, v) and λ where $0 \leq \lambda \leq 1$: $\lambda f(u) + (1-\lambda) f(v) \leq f(\lambda u + (1-\lambda)v)$.

Lower Envelope Potential Representation It can be easily seen that the Robust P^n model (8) can be written as a lower envelope potential using $h + 1$ linear functions. The functions $f^q, q \in \mathcal{Q} = \{1, 2, \ldots, h+1\}$ are defined using

$$\mu^q = \begin{cases} \gamma_a & \text{if} \quad q = a \in \mathcal{L}, \\ \gamma_{\max} & \text{otherwise,} \end{cases}$$

$$w^q_{ia} = \begin{cases} 0 & \text{if} \quad q = h+1 \text{ or } a = q \in \mathcal{L}, \\ \alpha_a & \text{otherwise.} \end{cases}$$

The above formulation is illustrated in figure 2(a) for the case of binary variables. The representation for the more general concave form (10) is illustrated in figure 2(b).

Minimizing Lower Envelope Potentials The problem of minimizing any lower envelope higher order potential function can be transformed to the minimization of a pairwise

function with the addition of an auxiliary variable z that takes values from the index set $\mathcal{Q}$ [26]. The resulting problem can be written as

$$\min_{\mathbf{x}_c} \psi_c(\mathbf{x}_c) = \min_{\mathbf{x}_c, q} \left(\phi_z(q) + \sum_{i \in \mathcal{V}} \phi_{zi}(q, x_i) \right), \quad (11)$$

$$\text{where} \quad \phi_z(q) = \mu^q, \quad \text{and} \quad \phi_{zi}(q, x_i) = w^q_{ix_i}. \quad (12)$$

This pairwise function can be minimized using standard message passing algorithms such as TRW and BP. In fact for certain classes of lower envelope functions, the resulting pairwise function is submodular (including the Robust P^n model) and can be minimized in polynomial time using st-mincut/maxflow based algorithms. We refer the reader to [3, 10, 11] for examples.

4.2. Upper Envelope Higher order Potentials

As will be seen shortly, unlike the lower envelope case, upper envelope potentials result in a difficult energy minimization problem. Nevertheless, as the examples below illustrate, they are extremely useful for computer vision.

Size Prior In many problems we might have prior knowledge about how many scene elements (pixels or voxels) should be assigned a particular label. This knowledge can be incorporated by using a higher order potential which assigns a cost to labelings that deviate from the required distribution (counts) of labels. One example could be a potential that assigns a cost proportional to the magnitude of the deviation from the required label counts. Formally, the potential can be defined as

$$\psi(\mathbf{x}) = \sum_{a \in \mathcal{L}} \left(\tau_a | \sum_{i \in \mathcal{V}} \delta_i(a) - n_a | \right), \quad (13)$$

where n_a indicates the desired number of variables that should be assigned the label a, while the 'indicator' function $\delta_i(a)$ is zero for all values of x_i except when $x_i = a$. Examples of problems where such a potential would be useful include object reconstruction and segmentation, where we can use the potential to incorporate our prior belief that the object we are trying to reconstruct or segment is of a particular size. In other words, we can define a potential of the following form:

$$\psi(\mathbf{x}) = | \sum_{i \in \mathcal{V}} x_i - \mathcal{S} |, \quad (14)$$

where S is the desired number of voxels to be included in the reconstruction, or pixels to be included in the segmentation. A more general *size* potential would be $\psi(\mathbf{x}) = \mathcal{F}(| \sum_{i \in \mathcal{V}} x_i - \mathcal{S}|)$ where $\mathcal{F} : \mathbb{R} \rightarrow \mathbb{R}$ is general function. Similar label counting potentials have been proposed for a number of image labeling problems in [31, 35].

The linear envelope construction for the size prior (14) is similar to the one given for the region consistency potential. The difference is that now will be using maximization (upper envelope) of linear functions rather than the minimization (lower envelope). In this case, we need only two linear functions $f_q, q \in \mathcal{Q} = \{0, 1\}$ defined with parameters $\mu_0 = \mathcal{S} - n$, $\mu_1 = -\mathcal{S}$ and $w^q_{ia} = 1$ if $q = a$, and 0 otherwise (see figure 3(b)). We note here that a recent work [34] has proposed a specialized algorithm for handling size priors. However, our framework offers a more general class of amenable potentials.

Soft Not-null Set Constraints In many problems, we need to enforce the constraint that at least one out of an arbitrary set of scene elements takes a particular label. One example that belongs to this class is the silhouette consistency constraint that has proven useful for multi-view reconstruction [12, 28]. It uses the fact that the reconstructed shape when re-projected on the view of the camera must coincide with the respective silhouettes. For every ray that intersects the silhouette, the constraint makes sure that at least one voxel on that ray is included in the reconstruction. In more detail, this 'OR' [1] constraint is defined as $\sum_{i \in c} \delta_i(`fg') \geq 1$ where the c denotes the set of indices of all voxels on the ray. See figure 3.

Several works have shown that incorporation of this constraint prevents the need for a 'ballooning' force, and leads to significant improvements in the results [12, 28]. That said, there is a problem with the above constraint. The silhouettes used as input for multi-view reconstruction are often incorrect and using such a hard constraint may lead to wrong results. To handle noisy silhouettes, we can incorporate a soft constraint by adding the following higher order penalty term to the objective function:

$$\psi_c(\mathbf{x}_c) = \begin{cases} \gamma & \text{if} \quad \sum_{i \in c} x_i = 0, \\ 0 & \text{otherwise,} \end{cases} \quad (15)$$

where $\gamma \geq 0$. The penalty parameter γ can be used to encode how confident we are about a particular ray passing through the object, and thus enables the use of soft silhouettes. It is easy to show that the above constraint is non-submodular, and thus minimizing the resulting energy minimization problem is hard.

The soft silhouette penalty function (15) can be represented as a upper envelope function using only 2 functions. These are defined as $f_q, q \in \mathcal{Q} = \{0, 1\}$ with parameters $\mu_0 = 0$, $\mu_1 = \gamma$, and

$$w^q_{ia} = \begin{cases} -\gamma & \text{if} \quad q = a = 1, \\ 0 & \text{otherwise.} \end{cases} \quad (16)$$

[1] Representing the label 'fg' by binary value 1 and the label 'bg' by binary value 0, the constraint can be seen as enforcing that application of the OR operator on the binary variables returns 1.

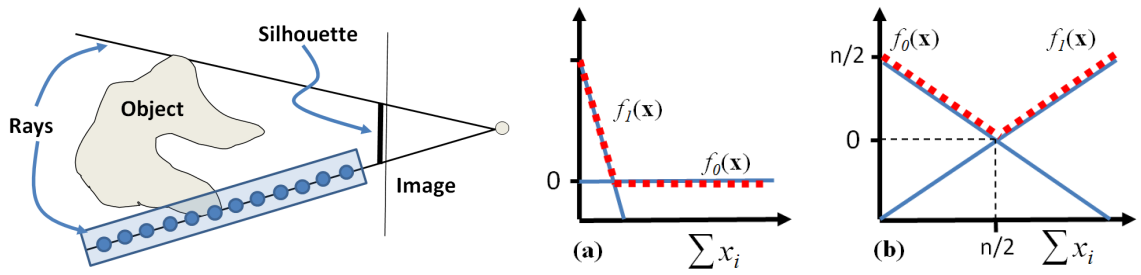

Figure 3. **(LEFT)** *Illustration of the ray constraint. At least one voxel on the ray should labelled as object.* **(RIGHT)** *(a) Graph for upper envelope representation of the ray penalty. Note that the undesirable assignment of labeling all variables to* 0 *is penalized heavily by the function* f_1*. All other assignments are favored equally using the function* f_0*. (b) Graph for upper envelope representation of size prior with* $\mathcal{S} = n/2$*. Note that the lowest energy corresponds to* $\sum_i x_i = n/2$ *(that is, assigning exactly half the random variables to label* 1*). Graphs (a) and (b) show the upper envelope definition for the ray penalty, and size prior for size* $\mathcal{S} = n/2$ *respectively.*

This representation is illustrated in figure 3(a).

Apart from object reconstruction, not-null type constraints have also appeared in recent work on imposing topological constraints such as connectivity in the object segmentation problem [19, 21].

5. Minimizing Upper Envelope Functions

The problem of minimizing an upper envelope function can be easily transformed to a min-max problem involving a pairwise function. This requires the addition of an auxiliary variable z that takes values from the index set $\mathcal{Q}$. The resulting problem can be written as

$$\min_{\mathbf{x}_c} \psi_c(\mathbf{x}_c) = \min_{\mathbf{x}_c} \max_{q} \left(\phi_z(q) + \sum_{i \in \mathcal{V}} \phi_{zi}(q, x_i) \right), \quad (17)$$

$$\text{where} \quad \phi_z(q) = \mu^q, \quad \text{and} \quad \phi_{zi}(q, x_i) = w^q_{ix_i}. \quad (18)$$

Although the above reformulation provides us with a pairwise function, the min-max form of the problem implies that we cannot use standard algorithms for solving it. To address this issue, we extend the linear programming (LP) relaxation for pairwise energy functions so that it handles upper envelope potentials.

5.1. LP Relaxation for Min-Max Labeling

We now consider the energy minimization problem in its entirety, that is the minimization of the function that consists of unary, pairwise and upper envelope potentials. Using the discussion at the beginning of the section, this energy function can be written as

$$\begin{aligned} E(\mathbf{x}; \mathbf{D}) \quad &= \quad \sum_{i \in \mathcal{V}} \psi_i(x_i) + \sum_{(i,j) \in \mathcal{E}} \psi_{ij}(x_i, x_j) \quad (19) \\ &+ \quad \sum_{z \in \mathcal{Z}} \max_{q \in \mathcal{Q}} \left(\phi_z(q) + \sum_{i,(z,i) \in \mathcal{C}} \phi_{zi}(q, x_i) \right). \end{aligned}$$

Here, $\mathcal{Z}$ refers to the set of all auxiliary variables (which is equal to the number of upper envelope potentials in the energy function). With a slight abuse of notation, we denote the set of neighboring auxiliary and random variables as $\mathcal{C}$ since it defines the cliques.

In order to formulate the above problem as an integer program, we define binary variables $y_i(a)$ that indicate whether variable X_i takes a label a, that is,

$$y_i(a) = \begin{cases} 1 & \text{if} \quad x_i = a, \\ 0 & \text{otherwise.} \end{cases} \quad (20)$$

Similarly, we define variables $y_{ij}(a, b) = y_i(a) y_j(b)$ for all $(a, b) \in \mathcal{E}$. In addition, we also specify a slack variable t_z to represent the clique potential corresponding to the auxiliary variable z. Using these variables, the integer program is specified as

$$\begin{aligned} \arg\min_{\mathbf{y}, t_z} \quad & \sum_{i \in \mathcal{V}, a \in \mathcal{L}} \psi_i(a) y_i(a) + \quad (21) \\ & \sum_{(i,j) \in \mathcal{E}, a, b \in \mathcal{L}} \psi_{ij}(a, b) y_{ij}(a, b) + \sum_{z \in \mathcal{Z}} t_z \\ \text{s.t.} \quad & t_z \geq \phi_z(q) + \sum_{i,(z,i) \in \mathcal{C}} t_{zi}(q), \\ & t_{zi}(q) \geq \phi_{zi}(q, a) y_i(a), \\ & \sum_a y_i(a) = 1, \\ & \sum_b y_{ij}(a, b) = y_i(a), \\ & y_i(a) \in \{0, 1\}, y_{ij}(a, b) \in \{0, 1\}. \end{aligned}$$

The first two constraints ensure that $t_z = \max_q(\phi_z(q) + \sum_i \phi_{zi}(q, x_i))$. The next two constraints imply that each random variable takes a unique label and that $y_{ij}(a, b) = y_i(a) y_j(b)$ respectively. The final set of constraints enforces the variables $\mathbf{y}$ to be binary. These binary constraints make the above integer program NP-hard to optimize. However, we can obtain an approximate solution to this problem by relaxing $\mathbf{y}$ to take (possibly fractional) values between 0 and 1. The resulting LP relaxation is similar to the standard relaxation for pairwise MRFs with the addition of slack variables t_z.

Although the above LP relaxation can be solved in polynomial time using standard interior point algorithms this is not a practical solution for the problem due to large time and memory requirements. In this paper, taking inspiration from previous works on energy minimization of pairwise MRFs (that is, where the energy function contains no auxiliary variables), we develop an efficient iterative strategy for solving the Lagrangian dual (hereby referred to as simply the dual) of the above LP. We begin by providing the exact formulation of the dual.

Dual The dual of the LP relaxation (21) is specified as

$$\max_{\boldsymbol{\theta},\lambda} \quad \left(\sum_{i \in \mathcal{V}} \min_{a} \theta_i(a) + \sum_{(i,j) \in \mathcal{E}} \min_{a,b} \theta_{ij}(a,b) + \sum_{z \in \mathcal{Z}} \sum_{q} \lambda_z(q) \phi_z(q) \right) \quad \text{s.t.} \;\; \boldsymbol{\theta} \equiv \boldsymbol{\theta}^{\lambda}. \tag{22}$$

Here $(\equiv)$ denotes reparameterization and the parameter $\boldsymbol{\theta}^{\lambda}$ is given by

$$\begin{aligned} &\theta_i^{\lambda}(a) = \psi_i(a) + \sum_{z,(z,i) \in \mathcal{C}} \sum_{q} \lambda_{zi}(q,a) \phi_{zi}(q,a), \\ &\forall i \in \mathcal{V}, a \in \mathcal{L}, \\ &\theta_{ij}^{\lambda}(a,b) = \psi_{ij}(a,b), \;\; \forall (i,j) \in \mathcal{E}, a, b \in \mathcal{L}, \end{aligned} \tag{23}$$

where λ satisfies the following constraints:

$$\begin{aligned} &\sum_{q} \lambda_z(q) = 1, \;\; \forall v_z \in \mathcal{Z}, \\ &\sum_{a} \lambda_{zi}(q,a) = \lambda_z(q), \;\; \forall (z,i) \in \mathcal{C}, q \in \mathcal{Q}, \\ &\lambda_{zi}(q,a) \geq 0, \lambda_z(q) \geq 0, \;\; \forall (z,i) \in \mathcal{C}, q \in \mathcal{Q}, a \in \mathcal{L}. \end{aligned} \tag{24}$$

Next, we design an efficient diffusion algorithm for optimizing the dual (22).

5.2. The Diffusion Algorithm

We begin by describing the standard diffusion algorithm [27, 33] for the case where the energy function has no auxiliary variables (that is, energy minimization for pairwise MRFs). We then generalize the algorithm to solve problem (22). When there are no auxiliary variables present in the energy function, the dual of the LP relaxation can be further simplified to

$$\max_{\boldsymbol{\theta}} \left(\sum_{i \in \mathcal{V}} \min_{a} \theta_i(a) + \sum_{(i,j) \in \mathcal{E}} \min_{a,b} \theta_{ij}(a,b) \right) \quad \text{s.t.} \;\; \boldsymbol{\theta} \equiv \boldsymbol{\psi}, \tag{25}$$

where $\boldsymbol{\psi}$ is the parameter of the given random field. An efficient algorithm for optimizing problem (25) is the following: (1) Choose a random variable i, (2) Run the AVERAGING procedure for i (described in table 1) using the current parameter $\boldsymbol{\theta}$, and (3) Repeat till the dual objective (25) cannot be increased for any choice of i. Note that the AVERAGING procedure is guaranteed not to decrease the value of the dual. Since the dual is bounded, it follows that the above algorithm will converge. Upon convergence, we obtain the value of the primal problem (that is, the desired labeling $\mathbf{x}$) in a similar manner to [13, 15].

- Define $\mathcal{N}(i) = \{j | (i,j) \in \mathcal{E}\}$ as the set of all neighbors of i.
- Compute $\beta_{ij}(a) = \min_b \theta_{ij}(a,b)$ for all $j \in \mathcal{N}(i)$.
- Compute $\alpha_i(a) = \frac{1}{|\mathcal{N}(i)|+1}(\theta_i(a) + \sum_{j \in \mathcal{N}(i)} \beta_{ij}(a))$.
- Update parameter to $\boldsymbol{\theta}'$ as follows:
  $\theta'_i(a) = \alpha_i(a), \quad \theta'_{ij}(a,b) = \theta_{ij}(a,b) + \alpha_i(a) - \beta_{ij}(a).$

Table 1. *The* AVERAGING *procedure.*

Handling Auxiliary Variables For auxiliary variables, we need to determine the value of λ that increases the dual. In order to formulate this problem, let us denote the current value of λ as λ^c. Since the solution is feasible, it follows that $\boldsymbol{\theta}^c \equiv \overline{\boldsymbol{\theta}}^{\lambda^c}$ where $\boldsymbol{\theta}^c$ is the current value of the parameter $\boldsymbol{\theta}$. For a given auxiliary variable z, we define

$$\phi_i(a) = \theta_i^c(a) + \sum_{q} \left(1 - \lambda_{zi}^c(q,a)\right) \phi_{zi}(q,a),$$

Note that the value of $\phi_i(a)$ depends on λ^c. However, in order to avoid cluttered notation, we do not make this dependency explicit. Now given a new λ^n, we define a new parameter $\boldsymbol{\theta}^n$ as

$$\theta_i^n(a) = \phi_i(a) - \sum_{q} \left(1 - \lambda_{zi}^n(q,a)\right) \phi_{zi}(q,a),$$

It follows that $\boldsymbol{\theta}^n \equiv \overline{\boldsymbol{\theta}}^{\lambda^n}$. Ideally, we would like to find the λ^n that solves the following problem:

$$\begin{aligned} \max_{\lambda^n} \quad & \left(\sum_{i,(z,i) \in \mathcal{C}} \min_{a} \theta_i^n(a) + \sum_{q} \lambda_z^n(q) \phi_z(q) \right) \\ \text{st.} \quad & \theta_i^n(q) = \phi_i(a) - \sum_{q} \left(1 - \lambda_{zi}^n(q,a)\right) \phi_{zi}(q,a), \\ & \sum_{q} \lambda_z^n(q) = 1, \\ & \sum_{a} \lambda_{zi}^n(q,a) = \lambda_z^n(q), \\ & \lambda_z^n(q) \geq 0, \lambda_{zi}^n(q,a) \geq 0, \end{aligned} \tag{26}$$

that is, the value of λ^n that provides the maximum increase in the dual (22). However, although solvable in polynomial time, it may still be computationally expensive to optimize the above problem for λ^n directly. Instead we use an efficient dual decomposition strategy. Each *slave* problem (corresponding to a particular variable i) for dual decomposition

is defined as

$$\begin{aligned}
\max_{\lambda^n} \quad & \min_a \theta_i^n(a) + \sum_q \left(\nu_{zi}(q) + \frac{\phi_z(q)}{|\mathcal{N}(z)|} \right) \lambda_z^n(q) \\
\text{s.t.} \quad & \theta_i^n(q) = \phi_i(a) - \sum_q \left(1 - \lambda_{zi}^n(q,a)\right) \phi_{zi}(q,a), \\
& \sum_q \lambda_z^n(q) = 1, \\
& \sum_a \lambda_{zi}^n(q,a) = \lambda_z^n(q), \\
& \lambda_z^n(q) \geq 0, \lambda_{zi}^n(q,a) \geq 0, \qquad (27)
\end{aligned}$$

where $\mathcal{N}(z) = \{i | (z,i) \in \mathcal{C}\}$, and $\nu_{zi}(q)$ are Lagrange multipliers of the dual decomposition which ensure that the value of $\lambda_z^n(q)$ is the same in all problems corresponding to variables i such that $(z,i) \in \mathcal{C}$. The Lagrange multipliers for dual decomposition satisfy the following constraint:

$$\sum_{i \in \mathcal{N}(z)} \nu_{zi}(q) = 0, \quad \forall q \in \mathcal{Q}. \qquad (28)$$

The values of $\nu_{zi}(q)$ are initialized to 0 (thereby satisfying the above constraints). Note that problem (27) is an instance of the well-studied *fractional packing* problem for which there exist several efficient algorithms. In this work, we solve problem (27) using the method described in [22] to obtain the values of $\lambda_z^n(q)$ and $\lambda_{zi}^n(q,a)$ for the current values of $\nu_{zi}(q)$. We note here that although in theory the method of [22] requires several iterations, in practice we found it to be very efficient. However, a speed up may be achieved by using more recently approaches for fractional packing such as [2]. The multipliers are then updated as $\nu_{zi}(q) = \nu_{zi}(q) - \eta_t \sum_a \lambda_{zi}^n(q,a)$, that is, using subgradient descent. Here η_t is the learning rate at iteration t. The multipliers then projected to satisfy constraint (28), that is, their average value of subtracted from them so that they sum to 0. The new values of $\nu_{zi}(q)$ are used to compute the new values of $\lambda_z^n(q)$ and $\lambda_{zi}^n(q,a)$. This procedure is repeated until convergence. Note that the dual decomposition strategy is guaranteed to provide the globally optimal solution under fairly mild conditions [1, 7]. Furthermore, due to the convexity of problem (26), at convergence all slave problems agree on the value of λ^n, which implies that we can trivially obtain the primal solution from the dual.

The overall diffusion algorithm for the dual (22) is as follows:

- Choose a random variable i or auxiliary variable z.
- If i, then run the AVERAGING procedure for i using the current parameter $\boldsymbol{\theta}$.
- If z, then compute λ^n and use it to define a new parameter $\boldsymbol{\theta}^n$.
- Repeat until convergence.

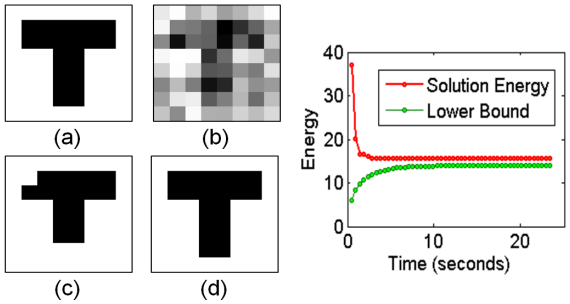

Figure 4. *Binary image segmentation with a size prior. (a) Original Image. (b) Image with random noise used as input. (c) Best result from pairwise model. (d) Result with size prior (20 pixels). The algorithm is able to achieve the exact number of foreground pixels. The graph shows how the solution energy and lower-bound change in different iterations.*

Note that both the dual decomposition strategy and the AVERAGING procedure are guaranteed to converge thereby implying that the overall algorithm will converge.

6. Experiments

We demonstrate the efficacy of our algorithm on the binary (fg/bg) segmentation problem with gray scale images. The classical Markov Random Field model for the problem is defined as

$$E(\mathbf{x}) \quad = \quad \sum_{i \in \mathcal{V}} \psi_i(x_i) + \sum_{(i,j) \in \mathcal{E}} \psi_{ij}(x_i, x_j), \qquad (29)$$

where $\psi_i(x_i) = \tau_i x_i$ and $\psi_{ij}(x_i, x_j) = \tau_{ij}(x_i \overline{x}_j + x_i \overline{x}_j)$ where $G = (\mathcal{V}, \mathcal{E})$ represents an image grid with $n = |V|$ pixels. The unary parameters are defined as $\tau_i = 125 - I_i$, where I_i is the gray scale value of pixel i. The pairwise parameters τ_{ij} encode a simple Ising model, and are set to a constant. We extend the pairwise energy function defined above by incorporating a size prior potential (14) that encourages the segmentation to contain the exact number of pixels that were present in the ground truth.

The algorithms is applied on a noisy image that is generated by adding random noise to the original image. The result for the classical and new problem formulation (with size prior) are shown in figure 4 (c) and (d) respectively. The size prior is able to ensure a segmentation with the correct number of foreground pixels (20 pixels). The accompanying graph shows how the lower bound and solution energy change with iterations of the diffusion algorithm.

Random Synthetic Problems We also tested our algorithm on randomly generated energy functions containing a size prior potential. These functions were defined on 10 node complete graphs with parameter samples $\psi_i \sim \mathcal{U}[-10, 10]$, and $\psi_{ij} \sim \mathcal{U}[0, w]$. The values of parameters of the size prior (equation (14)) are taken from $\psi_c \sim \mathcal{U}[0, v]$, while $\mathcal{S} = 5$. The $x-axes$ of the graphs shows the strength of the size prior parameter θ. For each data point in the

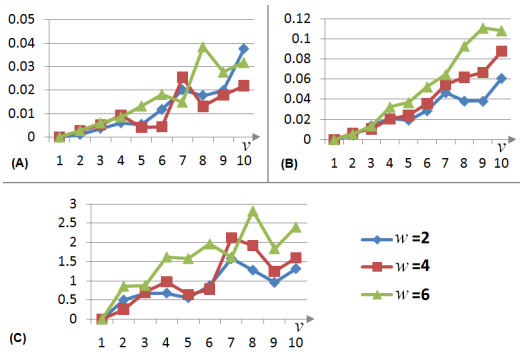

Figure 5. *Graphs showing the results of randomly generated problems. w and v denote the strength of the pairwise and size potentials respectively. (A) Error in energy $|E-E_o|/E_o$. (B) Difference between lower bound and solution energy $|E-L_b|/E_o$. (C) l_1 label error in solution $\sum_i (x_i \neq x_i^o)$.*

graphs we averaged the results over 50 trials. Let E and E_o denote the energies of our result $\mathbf{x}$ and the optimal result $\mathbf{x}^o$. The results are shown in figure 5. Graph (A) shows the error in energy $|E - E_o|/E_o$, (B) shows the difference between lower bound and solution energy as a fraction of the optimal energy $|E - L_b|/E_o$. (C) shows the l_1 label error in solution $\sum_i (x_i \neq x_i^o)$. It can be seen that the diffusion algorithm is very effective, and is able to obtain solutions close to the optimal solution of the problem.

7. Conclusions

We presented a framework that compactly encodes many useful classes of higher order potentials [9, 10, 11, 14, 26]. Our representation makes these potentials amenable to efficient inference algorithms. One of the contributions of this work was to show how inference with size and not-null potential functions can be formulated as min-max problems. We proposed a message passing algorithm for solving this problem, thereby allowing us to incorporate these useful family of potentials in classical formulations. We believe that the min-max representation of these potentials would inspire more research in the development of new algorithms for solving such optimization problems in both the computer vision and machine learning communities.

References

[1] D. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

[2] D. Bienstock and G. Iyengar. Solving fractional packing in $O(1/\epsilon)$ iterations. In *STOC*, 2004.

[3] E. Boros and P. Hammer. Pseudo-boolean optimization. *Discrete Applied Mathematics*, 2002.

[4] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 2001.

[5] C. Chekuri, S. Khanna, J. Naor, and L. Zosin. A linear programming formulation and approximation algorithms for the metric labelling problem. *SIAM Journal on Discrete Mathematics*, 2005.

[6] S. Gould, F. Amat, and D. Koller. Alphabet SOUP: A framework for approximate energy minimization. In *CVPR*, 2009.

[7] M. Guignard and S. Kim. Lagrangean decomposition: A mdoel yielding stronger lagrangean bounds. *Mathematical Programming*, 1987.

[8] T. Hazan and A. Shashua. Convergent message-passing algorithms for inference over general graphs with convex free energy. In *UAI*, 2008.

[9] H. Ishikawa. Higher-order clique reduction in binary graph cut. In *CVPR*, 2009.

[10] P. Kohli, M. Kumar, and P. Torr. P^3 and beyond: Solving energies with higher order cliques. In *CVPR*, 2007.

[11] P. Kohli, L. Ladicky, and P. Torr. Robust higher order potentials for enforcing label consistency. In *CVPR*, 2008.

[12] K. Kolev and D. Cremers. Integration of multiview stereo and silhouettes via convex functionals on convex domains. In *ECCV*, 2008.

[13] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *PAMI*, 2006.

[14] N. Komodakis and N. Paragios. Beyond pairwise energies: Efficient optimization of higher-order MRFs. In *CVPR*, 2009.

[15] N. Komodakis, N. Paragios, and G. Tziritas. MRF optimization via dual decomposition: Message-passing revisited. In *ICCV*, 2007.

[16] N. Komodakis, G. Tziritas, and N. Paragios. Fast, approximately optimal solutions for single and dynamic MRFs. In *CVPR*, 2007.

[17] M. P. Kumar and D. Koller. MAP estimation of semi-metric MRFs via hierarchical graph cuts. In *UAI*, 2009.

[18] X. Lan, S. Roth, D. Huttenlocher, and M. Black. Efficient belief propagation with learned higher-order Markov random fields. In *ECCV*, 2006.

[19] V. Lempitsky, P. Kohli, C. Rother, and T. Sharp. Image segmentation with a bounding box prior. In *ICCV*, 2009.

[20] T. Meltzer, A. Globerson, and Y. Weiss. Convergent message passing algorithms - a unifying view. In *UAI*, 2009.

[21] S. Nowozin and C. Lampert. Global connectivity potentials for random field models. In *CVPR*, 2009.

[22] S. Plotkin, D. Shmoys, and E. Tardos. Fast approximation algorithms for fractional packing and covering problems. *Mathematics of Operations Research*, 1995.

[23] B. Potetz. Efficient belief propagation for vision using linear constraint nodes. In *CVPR*, 2007.

[24] X. Ren and J. Malik. Learning a classification model for segmentation. In *ICCV*, 2003.

[25] S. Roth and M. Black. Fields of experts: A framework for learning image priors. In *CVPR*, 2005.

[26] C. Rother, P. Kohli, W. Feng, and J. Jia. Minimizing sparse higher order energy functions of discrete variables. In *CVPR*, 2009.

[27] M. Schlesinger. Syntactic analysis of two-dimensional visual signals in noisy conditions. *Kibernetika*, 1976.

[28] S. Sinha and M. Pollefeys. Multi-view reconstruction using photo-consistency and exact silhouette constraints: A maximum-flow formulation. In *ICCV*, 2005.

[29] D. Sontag, T. Meltzer, A. Globerson, T. Jaakkola, and Y. Weiss. Tightening LP relaxations for MAP using message passing. In *UAI*, 2008.

[30] D. Tarlow, R. Zemel, and B. Frey. Flexible priors for exemplar-based clustering. In *UAI*, 2008.

[31] S. Vicente, V. Kolmogorov, and C. Rother. Joint optimization of segmentation and appearance models. In *ICCV*, 2009.

[32] Y. Weiss, C. Yanover, and T. Meltzer. MAP estimation, linear programming and belief propagation with convex free energies. In *UAI*, 2007.

[33] T. Werner. A linear programming approach to max-sum problem: A review. *PAMI*, 2007.

[34] T. Werner. High-arity interactions, polyhedral relaxations, and cutting plane algorithm for soft constraint optimisation (MAP-MRF). In *CVPR*, 2008.

[35] O. Woodford, C. Rother, and V. Kolmogorov. A global perspective on MAP inference for low-level vision. In *ICCV*, 2009.

[36] O. Woodford, P. Torr, I. Reid, and A. Fitzgibbon. Global stereo reconstruction under second order smoothness priors. In *CVPR*, 2008.

[37] J. Yedidia, W. Freeman, and Y. Weiss. Generalized belief propagation. In *NIPS*, 2001.