Diffusion Methods for Classification with Pairwise Relationships

Pedro F. Felzenszwalb*
Brown University
Providence, RI, USA
pff@brown.edu

Benar F. Svaiter†
IMPA
Rio de Janeiro, RJ, Brazil
benar@impa.br

May 22, 2015

Abstract

We define two algorithms for propagating information in classification problems with pairwise relationships. The algorithms involve contraction maps and are related to non-linear diffusion and random walks on graphs. The approach is also related to message passing and mean field methods. The algorithms we describe are guaranteed to converge on graphs with arbitrary topology. Moreover they always converge to a unique fixed point, independent of initialization. We prove that the fixed points of the algorithms under consideration define lower-bounds on the energy function and the max-marginals of a Markov random field. Our theoretical results also illustrate a relationship between message passing algorithms and value iteration for an infinite horizon Markov decision process. We illustrate the practical feasibility of our algorithms with preliminary experiments in image restoration and stereo depth estimation.

1 Introduction

In many classification problems there are relationships among a set of items to be classified. For example, in image reconstruction problems adjacent pixels are likely to belong to the same object or image segment. This leads to relationships between the labels of different pixels in an image. Energy minimization methods based on Markov random fields (MRF) address these problems in a common framework [3, 18, 14]. Within this framework we introduce two new algorithms for classification with pairwise information. Our algorithms are based on contraction maps and are related to non-linear diffusion and random walks on graphs.

The setting under consideration is as follows. Let $G = (V, E)$ be an undirected simple graph and L be a set of labels. A labeling of V is a function $x : V \to L$ assigning a label from L to each vertex in V. Local information is modeled by a cost $g_i(a)$ for assigning label a to vertex i. Information

*Partially supported by NSF under grant 1161282 and the Brown-IMPA collaboration program.
†Partially supported by CNPq grants 474996/2013-1, 302962/2011-5 and FAPERJ grant E-26/201.584/2014.

on label compatibility for neighboring vertices is modeled by a cost $h_{ij}(a,b)$ for assigning label a to vertex i and label b to vertex j. The total cost for a labeling x is defined by an energy function,

$$F(x) = \sum_{i \in V} g_i(x_i) + \sum_{\{i,j\} \in E} h_{ij}(x_i, x_j). \tag{1}$$

In the context of MRFs the energy function defines a Gibbs distribution on random variables X associated with the vertices V,

$$p(X = x) = \frac{1}{Z} \exp(-F(x)). \tag{2}$$

Minimizing the energy $F(x)$ corresponds to maximizing $p(X = x)$. This approach has been applied to a variety of problems in image processing and computer vision [10]. A classical example involves restoring corrupted images [11, 4]. In image restoration there is a grid of pixels and the problem is to estimate an intensity value for each pixel. To restore an image I one looks for an image J that is similar to I and is smooth almost everywhere. Similarity between I and J is defined by local costs at each pixel. The smoothness constraint is defined by pairwise costs between neighboring pixels in J.

1.1 Basic Definitions and Overview of Results

Let $G = (V, E)$ be an undirected, simple, connected graph, with more than one vertex. For simplicity let $V = \{1, \ldots, n\}$. Let $N(i)$ and $\mathrm{d}(i)$ denote respectively the set of neighbors and the degree of vertex i,

$$N(i) = \{j \in V \mid \{i,j\} \in E\}, \quad \mathrm{d}(i) = |N(i)|.$$

Let L be a set of labels. For each vertex $i \in V$ we have a non-negative cost for assigning label a to vertex i, denoted by $g_i(a)$. These costs capture local information about the label for each vertex. For each edge $\{i,j\} \in E$ we have a non-negative cost for assigning label a to vertex i and label b to vertex j, denoted by $h_{ij}(a,b)$ or $h_{ji}(b,a)$. These costs capture relationships between labels of neighboring vertices.

$g_i : L \to [0, \infty)$ for $i \in V$;

$h_{ij}, h_{ji} : L^2 \to [0, \infty)$ for $\{i,j\} \in E$ with $h_{ij}(a,b) = h_{ji}(b,a)$

Let $x \in L^V$ denote a labeling of V with labels from L. A cost for x that takes into account both local information at each vertex and the pairwise relationships can be defined by an energy function $F : L^V \to \mathbb{R}$,

$$F(x) = \sum_{i \in V} g_i(x_i) + \sum_{\{i,j\} \in E} h_{ij}(x_i, x_j). \tag{3}$$

This leads to a natural optimization problem where we look for a labeling x with minimum energy.

Throughout the paper we assume L is finite. The optimization problem defined by F is NP-hard even when $|L| = 2$ as it can be used to solve the independent set problem on G. It can also be used to solve coloring with k colors when $|L| = k$. The optimization problem can be solved in polynomial time using dynamic programming when G is a tree [1]. More generally dynamic programming leads to polynomial optimization algorithms when the graph G is chordal and has bounded tree-width. For general graphs G, when the pairwise functions define a metric there are polynomial time approximation algorithms [13]. In some important cases the optimization problem can be solved using graph cuts and maximum flow algorithms [12, 7, 6, 15].

Max-product belief propagation [18, 14] is a local message passing algorithm that is equivalent to dynamic programming when G is a tree. Both dynamic programming and belief propagation aggregate local costs by sequential propagation of information along the edges in E.

For $i \in V$ we define the value function $f_i : L \to \mathbb{R}$,

$$f_i(\tau) = \min_{\substack{x \in L^V \\ x_i = \tau}} F(x). \tag{4}$$

The value functions f_i are exactly what is computed by the dynamic programming and belief propagation approaches for minimizing F when G is a tree. In the context of MRFs the value functions are also known as "max-marginals". Each value function defines a cost for assigning a label to a vertex that takes into account the whole graph. If x^* minimizes $F(x)$ than x_i^* minimizes $f_i(\tau)$, and when $f_i(\tau)$ has a unique minimum we can minimize $F(x)$ by selecting

$$x_i^* = \operatorname*{argmin}_{\tau} f_i(\tau). \tag{5}$$

A local belief is a function $\varphi_i : L \to \mathbb{R}$. A field of beliefs specifies a local belief for each vertex in V, and is an element of

$$(\mathbb{R}^L)^V = \{\varphi = (\varphi_1, \ldots, \varphi_N) \mid \varphi_i : L \to \mathbb{R}\}. \tag{6}$$

We define our algorithms in terms of maps,

$$T : (\mathbb{R}^L)^V \to (\mathbb{R}^L)^V,$$
$$S : (\mathbb{R}^L)^V \to (\mathbb{R}^L)^V.$$

The maps S and T are are closely related, both are contractions, but each of them has its own unique fixed point. Each of these maps defines a fixed point iteration algorithm: for $z \in \{T, S\}$ we compute a sequence $(\varphi^0, \varphi^1, \ldots)$. The algorithm starts with an initial value φ^0 and sequentially computes $\varphi^{k+1} = z(\varphi^k)$, so that $\varphi^k = z^k(\varphi^0)$. Both $S^k(\varphi^0)$ and $T^k(\varphi^0)$ are convergent to the unique fixed points of S and T respectively. After convergence to a fixed point $\bar{\varphi}$ we select a labeling x by selecting the label minimizing the belief at each vertex,

$$x_i = \operatorname*{argmin}_{\tau} \bar{\varphi}_i(\tau). \tag{7}$$

Our algorithms depend on parameters $p \in (0,1)$, $q = 1 - p$ and weights $w_{ij} \in [0,1]$ for each $i \in V$ and $j \in N(i)$. The weights from each vertex are constrained to sum to one,

$$\sum_{j \in N(i)} w_{ij} = 1, \qquad \forall i \in V. \tag{8}$$

These weights define random walks on G where the one step transition from vertex i to vertex j has probability w_{ij}. In a uniform random walk we have $w_{ij} = 1/\mathrm{d}(i)$. Non-uniform weights can be used to capture additional information about an underlying application. For example, in the case of stereo depth estimation (Section 5.2) we have used non-uniform weights that reflect color similarity between neighboring pixels.

The maps S and T define belief updates that propagate information along the edges of G,

Definition 1.1.

$$(T\varphi)_i(\tau) = pg_i(\tau) + \sum_{j \in N(i)} \min_{u_j \in L} \frac{p}{2} h_{ij}(\tau, u_j) + q w_{ji} \varphi_j(u_j) \tag{9}$$

$$(S\varphi)_i(\tau) = pg_i(\tau) + \sum_{j \in N(i)} w_{ij} \min_{u_j \in L} p h_{ij}(\tau, u_j) + q\varphi_j(u_j) \tag{10}$$

The map defined by T corresponds to a form of non-linear diffusion of beliefs along the edges of G. The map defined by S corresponds to value iteration for a Markov decision process (MDP) [2] defined by random walks on G. Our results show that both S and T are contractions. We also show that the fixed point $\bar{\varphi}^T$ of T defines a lower bound on the energy function F, and that the fixed point $\bar{\varphi}^S$ of S defines lower bounds on the value functions f_i,

$$\sum_{i \in V} \bar{\varphi}_i^T(x_i) \leq F(x), \qquad \forall x \in L^V, \tag{11}$$

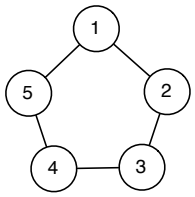
$$\bar{\varphi}_i^S(\tau) \leq f_i(\tau), \qquad \forall i \in V,\ \tau \in L. \tag{12}$$

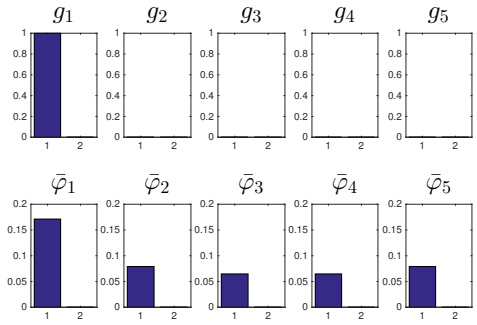
In Section 3 we study the algorithm defined by T and the relationship between $\bar{\varphi}^T$ and F. To the extent that $\sum_{i \in V} \bar{\varphi}_i^T(x_i)$ approximates $F(x)$ this justifies selecting a labeling x by minimizing $\bar{\varphi}_i^T$ at each vertex. This approach is related to a mean field approximation and variational inference with the Gibbs distribution $p(X = x)$ [18, 14].

In Section 4 we study the algorithm defined by S and the relationship between $\bar{\varphi}_i^S$ and f_i. To the extent that $\bar{\varphi}_i^S(\tau)$ approximates $f_i(\tau)$ this justifies selecting a labeling x by minimizing $\bar{\varphi}_i^S$ at each vertex. The fixed point $\bar{\varphi}^S$ defines the value of an optimal policy for an infinite horizon discounted Markov decision process. This process is defined in terms of random walks on G with transition probabilities given by the weights w_{ij}.

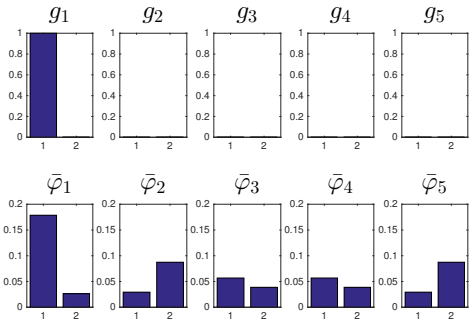
1.2 Examples

Figure 1 shows two examples of fixed points of T when the graph $G = (V, E)$ is a cycle with 5 vertices. In this case we have a binary labeling problem $L = \{1, 2\}$. The local costs are all zero





(a) Attractive relationships



(b) Repulsive relationships

Figure 1: The fixed points of T on two problems defined on the graph above. In this case $L = \{1, 2\}$. In both cases the local costs g_i are all zero except for vertex 1 who has a preference towards label 2. In **(a)** the pairwise costs encourage neighboring vertices to take the same label. In **(b)** the pairwise costs encourage neighboring vertices to take different labels.

except that vertex 1 has a preference for label 2. This is encoded by a cost for label 1,

$$g_1(1) = 1, \tag{13}$$

$$g_1(2) = 0, \tag{14}$$

$$g_i(a) = 0, \qquad \forall i \neq 1,\ a \in L. \tag{15}$$

In example (a) we have pairwise costs that encourage equal labels for neighboring vertices,

$$h_{ij}(a, b) = \begin{cases} 0 & a = b \\ 1 & a \neq b, \end{cases} \tag{16}$$

In example (b) we have pairwise costs that encourage different labels for neighboring vertices,

$$h_{ij}(a, b) = \begin{cases} 1 & a = b \\ 0 & a \neq b, \end{cases} \tag{17}$$

Figure 1 shows a graphical representation of the local costs for each vertex, and the value of $\bar{\varphi}$, the fixed point of T for each example. In (a) local selection of x_i minimizing $\bar{\varphi}_i$ leads to

$x = (2, 2, 2, 2, 2)$. In (b) local selection of x_i minimizing $\bar{\varphi}_i$ leads to $x = (2, 1, 2, 2, 1)$. In both examples the resulting labeling x is the global minimum of $F(x)$. For these examples we used $p = 0.1$ and $w_{ij} = 1/\mathrm{d}(i)$.

Of course in general local minimization of $\bar{\varphi}$ does not lead to a labeling minimizing $F(x)$ and it would be interesting to characterize when this happens.

1.3 Relationship to Message Passing and Mean-Field

Our algorithms are similar to message passing methods, in particular to min-sum (or equivalently max-product) belief propagation [18, 14]. When the graph G is a tree, Min-sum belief propagation converges and solves the optimization problem defined by F. Unfortunately the algorithm is not guaranteed to converge and it can have multiple fixed points for general graphs. Some form of dampening can help the algorithm converge in practice, and there are provably convergent versions of Belief propagation [17]. However, in the min-sum case even those methods do not have a unique fixed point, and the results depend on initialization. Our algorithms provide a simple alternative to min-sum belief propagation that is guaranteed to converge to a unique fixed point, regardless of initialization. Our algorithms are also guaranteed to converge "quickly".

The mean-field algorithm [18, 14] is an iterative method for approximating the Gibbs distribution $p(x)$ by a factored distribution $q(x)$,

$$q(x) = \prod_{i \in V} q_i(x_i). \tag{18}$$

The mean-field approach involves minimization of the KL divergence between p and q using fixed point iterations that repeatedly update the factors q_i defining q. A drawback of the approach is that the fixed point is not unique and the method is sensitive to initialization.

The algorithm defined by T is related to the mean-field method in the sense that the fixed points of T appear to approximate $F(x)$ by a function $H(x)$ that is a sum of local terms,

$$H(x) = \sum_{i \in V} \bar{\varphi}_i(x_i). \tag{19}$$

We do not know, however, if there is a measure under which the resulting $H(x)$ is an optimal approximation to $F(x)$ within the class of functions defined by a sum of local terms.

2 Preliminaries

Our algorithms are efficient in the following sense. Let $m = |E|$ and $k = |L|$. Each iteration in the fixed point algorithm involves evaluating T or S. This can be done in $O(mk^2)$ by "brute-force" evaluation of the expressions in Definition 1.1. In many applications, including in image restoration and stereo matching, the pairwise cost h_{ij} has special structure that allows for faster

computation using the techniques in [9]. This leads to an $O(mk)$ algorithm for each iteration of the fixed point method. Additionally, these algorithms are easily parallelizable. The fixed point iteration algorithms defined by T and S converge quickly because the maps are contractions in $(\mathbb{R}^L)^V$.

Let $z : \mathbb{R}^K \to \mathbb{R}^K$ and $\|x\|$ be a norm in $\mathbb{R}^K$. For $\gamma \in (0, 1)$, z is a γ-contraction if

$$\|z(x) - z(y)\| \leq \gamma \|x - y\|. \tag{20}$$

When z is a contraction it has a unique fixed point $\bar{x}$. It also follows directly from the contraction property that fixed point iteration $x_k = z(x_{k-1})$ converges to $\bar{x}$ quickly,

$$\|x_k - \bar{x}\| \leq \gamma^k \|x_0 - \bar{x}\|. \tag{21}$$

The weights w_{ij} in the definition of T and S define a random process that generates random walks on G. We have a Markov chain with state space V. Starting from a vertex S_0 we generate an infinite sequence of random vertices $(S_0, S_1, \ldots)$ with transition probabilities

$$p(S_{t+1} = j | S_t = i) = w_{ij}. \tag{22}$$

A natural choice for the weights is $w_{ij} = 1/\mathrm{d}(i)$, corresponding to moving from i to j with uniform probability over $N(i)$. This choice leads to uniform random walks on G [16].

We consider in $(\mathbb{R}^L)^V$ the partial order

$$\varphi \leq \psi \iff \varphi_i(\tau) \leq \psi_i(\tau) \ \ \forall i \in V, \ \forall \tau \in L. \tag{23}$$

It follows trivially from the definitions of T and S that both maps preserve order in $(\mathbb{R}^L)^V$, that is

$$\varphi \leq \psi \Rightarrow T\varphi \leq T\psi, \ S\varphi \leq S\psi. \tag{24}$$

We claim that for any $\alpha \in \mathbb{R}^V$,

$$\sum_{i \in V} \sum_{j \in N(i)} w_{ji} \alpha_j = \sum_{j \in V} \alpha_j. \tag{25}$$

This follows from re-ordering the double summation and the constraints that the weights out of each vertex sum to one,

$$\sum_{i \in V} \sum_{j \in N(i)} w_{ji} \alpha_j = \sum_{j \in V} \sum_{i \in N(j)} w_{ji} \alpha_j = \sum_{j \in V} \alpha_j$$

We note that the algorithms defined by T and S are related in the following sense. For a regular graph with degree d, if we let $w_{ij} = 1/d$ the maps T and S are equivalent up to rescaling if the costs in T and S are rescaled appropriately.

3 Algorithm defined by T (Diffusion)

In this section we study the fixed point algorithm defined by T. We show that T is a contraction in $(\mathbb{R}^L)^V$ and that the fixed point of T defines a "factored" lower bound on F.

We start by showing that T is a contraction with respect to the norm on $(\mathbb{R}^L)^V$ defined by

$$\|\varphi\|_{\infty,1} = \sum_{i \in V} \|\varphi_i\|_\infty. \tag{26}$$

Lemma 3.1. *(Contraction) For any $\varphi, \psi \in (\mathbb{R}^L)^V$*

$$\|(T\varphi)_i - (T\psi)_i\|_\infty \leq q \sum_{j \in N(i)} w_{ji} \|\varphi_j - \psi_j\|_\infty \qquad \forall i \in V, \tag{27}$$

$$\|(T\varphi) - (T\psi)\|_{\infty,1} \leq q \|\varphi - \psi\|_{\infty,1}. \tag{28}$$

Proof. Take $i \in V$ and $\tau \in L$. For any $x \in L^V$

$$\begin{aligned}
(T\varphi)_i(\tau) &= p g_i(\tau) + \sum_{j \in N(i)} \min_{u_j \in L} \frac{p}{2} h_{ij}(\tau, u_j) + q w_{ji} \varphi_j(u_j) \\
&\leq p g_i(\tau) + \sum_{j \in N(i)} \frac{p}{2} h_{ij}(\tau, x_j) + q w_{ji} \varphi_j(x_j) \\
&\leq p g_i(\tau) + \sum_{j \in N(i)} \frac{p}{2} h_{ij}(\tau, x_j) + q w_{ji} (\psi_j(x_j) + |\varphi_j(x_j) - \psi_j(x_j)|) \\
&\leq p g_i(\tau) + \sum_{j \in N(i)} \frac{p}{2} h_{ij}(\tau, x_j) + q w_{ji} (\psi_j(x_j) + \|\varphi_j - \psi_j\|_\infty)
\end{aligned}$$

If we let x minimize the right hand side of the last inequality we conclude

$$(T\varphi)_i(\tau) \leq (T\psi)_i(\tau) + q \sum_{j \in N(i)} w_{ji} \|\varphi_j - \psi_j\|_\infty.$$

Since this inequality holds interchanging φ with ψ we have

$$|(T\varphi)_i(\tau) - (T\psi)_i(\tau)| \leq q \sum_{j \in N(i)} w_{ji} \|\varphi_j - \psi_j\|_\infty.$$

Taking the τ maximizing the left hand side proves (27). To prove (28), we sum the inequalities (27) for all $i \in V$ and use (25). □

The contraction property above implies the fixed point algorithm defined by T converges to a unique fixed point independent on initialization. It also implies the distance to the fixed point decreases quickly, and we can bound the distance to the fixed point using either the initial distance to the fixed point or the distance between consecutive iterates (a readily available measure).

Theorem 3.2. *The map T has a unique fixed point $\bar{\varphi}$ and for any $\varphi \in (\mathbb{R}^L)^V$ and integer $k \geq 0$,*

$$\|\bar{\varphi} - T^k\varphi\|_{\infty,1} \leq q^k \|\bar{\varphi} - \varphi\|_{\infty,1}, \tag{29}$$

$$\|\bar{\varphi} - \varphi\|_{\infty,1} \leq \frac{1}{p}\|T\varphi - \varphi\|_{\infty,1}. \tag{30}$$

Proof. Existence and uniqueness of the fixed point, as well as the first inequality follows trivially from Lemma 3.1. To prove the second inequality observe that since $T^k\varphi$ converges to $\bar{\varphi}$,

$$\|\bar{\varphi} - \varphi\|_{\infty,1} \leq \sum_{k=0}^{\infty} \|T^{k+1}\varphi - T^k\varphi\|_{\infty,1} \leq \sum_{k=0}^{\infty} q^k \|T\varphi - \varphi\|_{\infty,1}. \tag{31}$$

Now note that since $p \in (0,1)$ and $p + q = 1$,

$$\sum_{k=0}^{\infty} q^k p = 1 \implies \sum_{k=0}^{\infty} q^k = \frac{1}{p}. \tag{32}$$

□

The map T and the energy function F are related as follows.

Proposition 3.3. *For any $\varphi \in (\mathbb{R}^L)^V$ and $x \in L^V$*

$$\sum_{i \in V} (T\varphi)_i(x_i) \leq pF(x) + q \sum_{i \in V} \varphi_i(x_i). \tag{33}$$

Proof. Direct use of the definition of T yields

$$\begin{aligned}
\sum_{i \in V} (T\varphi)_i(x_i) &= \sum_{i \in V} p g_i(x_i) + \sum_{j \in N(i)} \min_{u_j \in L} \frac{p}{2} h_{ij}(x_i, u_j) + q w_{ji} \varphi_j(u_j) \\
&\leq \sum_{i \in V} p g_i(x_i) + \sum_{j \in N(i)} \frac{p}{2} h_{ij}(x_i, x_j) + q w_{ji} \varphi_j(x_j) \\
&= p \left(\sum_{i \in V} g_i(x_i) + \sum_{i \in V} \sum_{j \in N(i)} \frac{1}{2} h_{ij}(x_i, x_j) \right) + q \sum_{i \in V} \sum_{j \in N(i)} w_{ji} \varphi_j(x_j) \\
&= pF(x) + q \sum_{j \in V} \varphi_j(x_j),
\end{aligned}$$

where the last equality follows from the fact that $h_{ij}(x_i, x_j) = h_{ji}(x_j, x_i)$ and Equation (25). □

Now we show the fixed point of T defines a lower bound on F in terms of a sum of local terms.

Theorem 3.4. *Let $\bar{\varphi}$ be the fixed point of T and*

$$H(x) = \sum_{i \in V} \bar{\varphi}_i(x_i).$$

Then $0 \leq \bar{\varphi}$ and $H(x) \leq F(x)$.

Proof. The fact that $H(x) \leq F(x)$ follows directly from Proposition 3.3.

To prove $0 \leq \bar{\varphi}$ consider the sequence $(0, T0, T^2 0, \ldots)$. The non-negativity of g_i and h_{ij} implies $0 \leq T0$. By the monotonicity of T it follows by induction that $T^k 0 \leq T^{k+1} 0$ for all $k \geq 0$. Since the sequence is pointwise non-decreasing and converges to $\bar{\varphi}$ we have $0 \leq \bar{\varphi}$. □

3.1 Linear Programming Formulation

Let a be a vector of coefficients in $(\mathbb{R}^L)^V$ and consider the linear programming problem in $(\mathbb{R}^L)^V$,

$$\max_{\varphi} a^T \varphi$$
$$\varphi_i(u_i) \leq p g_i(u_i) + \sum_{j \in N(i)} \frac{p}{2} h_{ij}(u_i, u_j) + q w_{ji} \varphi_j(u_j) \qquad \forall i \in V, \forall u \in L^V.$$

Note that the constraints in the LP are equivalent to $\varphi \leq T\varphi$. Now we show the fixed point of T corresponds to an optimal solution for the LP.

Theorem 3.5. *If a is a non-negative vector of coefficients the fixed point of T is an optimal solution for the LP. If a is a positive vector the fixed point of T is the unique optimal solution for the LP.*

Proof. Let $\bar{\varphi}$ be the fixed point of T. First note that $\bar{\varphi}$ is a feasible solution since $\bar{\varphi} \leq T\bar{\varphi}$.

Let $\varphi \in (\mathbb{R}^L)^V$ be any feasible solution for the LP. The linear constraints are equivalent to $\varphi \leq T\varphi$. Since T is monotone it follows by induction that $T^k \varphi \leq T^{k+1} \varphi$ for all $k \geq 0$. Since the sequence $(\varphi, T\varphi, T^2\varphi, \ldots)$ converges to $\bar{\varphi}$ and it is pointwise non-decreasing we conclude $\varphi \leq \bar{\varphi}$.

If a is non-negative we have $a^T \varphi \leq a^T \bar{\varphi}$ and therefore $\bar{\varphi}$ must be an optimal solution for the LP. If a is positive and $\varphi \neq \bar{\varphi}$ we have $a^T \varphi < a^T \bar{\varphi}$. This proves the fixed point is the unique optimal solution for the LP. □

4 Algorithm defined by S (Optimal Control)

In this section we study the algorithm defined by S. We start by showing that S corresponds to value iteration for an infinite horizon discounted Markov decision process (MDP) [2].

An infinite horizon discounted MDP is defined by a tuple (Q, A, c, t, γ) where Q is a set of states, A is a set of actions and γ is a discount factor in $\mathbb{R}$. The cost function $c : Q \times A \to \mathbb{R}$ specifies a cost $c(s, a)$ for taking action a on state s. The transition probabilities $t : Q \times A \times Q \to \mathbb{R}$ specify the probability $t(s, a, s')$ of moving to state s' if we take action a in state s.

Let o be an infinite sequence of state and action pairs, $o = ((s_1, a_1), (s_2, a_2), \ldots) \in (Q \times A)^\infty$. The (discounted) cost of o is

$$c(o) = \sum_{k=0}^{\infty} \gamma^k c(s_k, a_k). \tag{34}$$

A policy for the MDP is defined by a map $\pi : Q \to A$, specifying an action to be taken at each state. The value of a state s under the policy π is the expected cost of an infinite sequence of state and action pairs generated using π starting at s,

$$v_\pi(s) = E[c(o)|\pi, s_1 = s]. \tag{35}$$

An optimal policy π^* minimizes $v_\pi(s)$ for every starting state. Value iteration computes v_{π^*} as the fixed point of $\mathcal{L} : \mathbb{R}^Q \to \mathbb{R}^Q$,

$$(\mathcal{L}v)(s) = \min_{a \in A} c(s, a) + \gamma \sum_{s' \in Q} t(s, a, s')v(s'). \tag{36}$$

The map $\mathcal{L}$ is known to be a γ-contraction with respect to the $\|\cdot\|_\infty$ norm.

Now we show that S is equivalent to value iteration for an MDP defined by random walks on G. Intuitively we have states defined by a vertex $i \in V$ and a label $a \in L$. An action involves selecting a different label for each possible next vertex, and the next vertex is selected according to the random walk defined by the weights w_{ij}.

Lemma 4.1. *Define an MDP (Q, A, c, t, γ) as follows. The states are pairs of vertices and labels $Q = V \times L$. The actions specify a label for every possible next vertex $A = V^L$. The discount factor is $\gamma = q$. The transition probabilities and cost function are defined by*

$$t((i, \tau), u, (j, \tau')) = \begin{cases} w_{ij} & j \in N(i),\ \tau' = u_j \\ 0 & \text{otherwise} \end{cases} \tag{37}$$

$$c((i, \tau), u) = pg_i(\tau) + \sum_{j \in N(i)} pw_{ij}h_{ij}(\tau, u_j) \tag{38}$$

The map S is equivalent to value iteration for this MDP. That is, if $\varphi_i(\tau) = v((i, \tau))$ then

$$(S\varphi)_i(\tau) = (\mathcal{L}v)((i, \tau)).$$

Proof. The result follows directly from the definition of the MDP, $\mathcal{L}$ and S.

$$(\mathcal{L}v)((i, \tau)) = \min_{u \in V^L} c((i, \tau), u) + \gamma \sum_{(j,\tau') \in Q} t((i, \tau), u, (j, \tau'))v(j, \tau') \tag{39}$$

$$= \min_{u \in V^L} pg_i(\tau) + \sum_{j \in N(i)} pw_{ij}h_{ij}(\tau, u_j) + q \sum_{j \in N(i)} w_{ij}v(j, u_j) \tag{40}$$

$$= pg_i(\tau) + \sum_{j \in N(i)} w_{ij} \min_{u_j \in L} ph_{ij}(\tau, u_j) + qv(j, u_j) \tag{41}$$

$$= (S\varphi)_i(\tau) \tag{42}$$

□

The relationship to value iteration shows S is a contraction and we have the following results regarding fixed point iterations with S.

Theorem 4.2. *The map S has a unique fixed point $\bar{\varphi}$ and for any $\varphi \in (\mathbb{R}^L)^V$ and integer $k \geq 0$,*

$$\|\bar{\varphi} - S^k\varphi\|_\infty \leq q^k \|\bar{\varphi} - \varphi\|_\infty, \tag{43}$$

$$\|\bar{\varphi} - \varphi\|_\infty \leq \frac{1}{p}\|S\varphi - \varphi\|_\infty. \tag{44}$$

Proof. The first inequality follows directly from Lemma 4.1 and the fact that $\mathcal{L}$ is a γ-contraction with $\gamma = q$. The proof of the second inequality is similar to the proof of the analogous result for the map T in Theorem 3.2. □

4.1 Random Walks

The formalism of MDPs is quite general, and encompasses the fixed point algorithm defined by S. In this section we further analyze this fixed point algorithm and provide an interpretation using one-dimensional problems defined by random walks on G.

The weights w_{ij} define a random process that generates infinite walks on G. Starting from some vertex in V we repeatedly move to a neighboring vertex, and the probability of moving from $i \in V$ to $j \in N(i)$ in one step is given by w_{ij}.

An infinite walk $\omega = (\omega_1, \omega_2, \ldots) \in V^\infty$ can be used to define an energy on an infinite sequence of labels $z = (z_1, z_2, \ldots) \in L^\infty$,

$$F_\omega(z) = \sum_{t=0}^{\infty} pq^t g_{\omega_t}(z_t) + pq^t h_{\omega_t\omega_{t+1}}(z_t, z_{t+1}). \tag{45}$$

The energy $F_\omega(z)$ can be seen as the energy of a pairwise classification problem on a graph $G' = (V', E')$ that is an infinite path,

$$V' = \{1, 2, \ldots\}, \tag{46}$$

$$E' = \{\{1, 2\}, \{2, 3\}, \ldots\}. \tag{47}$$

The graph G' can be interpreted as a one-dimensional "unwrapping" of G along the walk ω. This unwrapping defines a map from vertices in the path G' to vertices in G.

Consider a policy $\pi : V \times L \times V \to L$ that specifies z_{k+1} in terms of ω_k, z_k and ω_{k+1},

$$z_{k+1} = \pi(\omega_k, z_k, \omega_{k+1}). \tag{48}$$

Now consider the expected value of $F_\omega(z)$ when ω is a random walk starting at $i \in V$ and z is a sequence of labels defined by the policy π starting with $z_1 = \tau$,

$$v_\pi(i, \tau) = E[F_\omega(z) | \omega_1 = i, z_1 = \tau, z_{k+1} = \pi(\omega_k, z_k, \omega_{k+1})]. \tag{49}$$

There is an optimal policy π^* that minimizes $v_\pi(i, \tau)$ for every $i \in V$ and $\tau \in L$. Let $\bar{\varphi}$ be the fixed point of S. Then $\bar{\varphi}_i(\tau) = v_{\pi^*}(i, \tau)$. This follows directly from the connection between S and the MDP described in the last section.

4.2 Bounding the Value Functions of F

Now we show that the fixed point of S defines lower bounds on the value functions f_i defined in (4). We start by showing that f_i can be lower bounded by f_j for $j \in N(i)$.

Proposition 4.3. *Let $i \in V$ and $j \in N(i)$.*

$$f_i(u_i) \geq pg_i(u_i) + \min_{u_j} ph_{ij}(u_i, u_j) + qf_j(u_j), \tag{50}$$

$$f_i(u_i) \geq pg_i(u_i) + \sum_{j \in N(i)} w_{ij} \min_{u_j} ph_{ij}(u_i, u_j) + qf_j(u_j). \tag{51}$$

Proof. The second inequality follows from the first one by taking a convex combination over $j \in N(i)$. To prove the first inequality note that,

$$f_i(u_i) = \min_{\substack{x \in L^V \\ x_i = u_i}} F(x) \tag{52}$$

$$= \min_{u_j \in L} \min_{\substack{x \in L^V \\ x_i = u_i, x_j = u_j}} F(x) \tag{53}$$

$$= \min_{u_j \in L} \min_{\substack{x \in L^V \\ x_i = u_i, x_j = u_j}} pF(x) + qF(x) \tag{54}$$

$$\geq pg_i(u_i) + \min_{u_j \in L} ph_{ij}(u_i, u_j) + \min_{\substack{x \in L^V \\ x_i = u_i, x_j = u_j}} qF(x) \tag{55}$$

$$\geq pg_i(x_i) + \min_{u_j \in L} ph_{ij}(u_i, u_j) + \min_{\substack{x \in L^V \\ x_j = u_j}} qF(x) \tag{56}$$

$$= pg_i(x_i) + \min_{u_j \in L} ph_{ij}(u_i, u_j) + qf_j(u_j). \tag{57}$$

The first inequality above follows from $F(x) \geq g_i(x_i) + h_{ij}(x_i, x_j)$ since all the terms in $F(x)$ are non-negative. The second inequality follows from the fact that we are minimizing $F(x)$ over x with fewer restrictions. □

The map S and the value functions are related as follows.

Proposition 4.4. *Let $f = (f_1, \ldots, f_N) \in (\mathbb{R}^L)^V$ be the field of beliefs defined by the value functions.*

$$Sf \leq f. \tag{58}$$

Proof. The result follows directly from Proposition 4.3. □

Now we show that the fixed point of S defines lower bounds on the value functions.

Theorem 4.5. *Let $\bar{\varphi}$ be the fixed point of S. Then*

$$0 \leq \bar{\varphi}_i(\tau) \leq f_i(\tau).$$

Proof. Since the costs g_i and h_{ij} are non-negative we have $0 \leq S0$. Using the monotonicity of S we can conclude $0 \leq \bar{\varphi}$.

Since $Sf \leq f$ and S is monotone, $S^k f \leq f$ for all k. To end the proof, take the limit $k \to \infty$ at the left hand-side of this inequality. □

5 Numerical Experiments

In this section we demonstrate the practical feasibility of our algorithms with preliminary experiments in the problems of image restoration and depth estimation.

5.1 Image Restoration

In many applications we expect idealized images to be piecewise smooth, changing slowly almost everywhere, but with potentially large intensity discontinuities across object boundaries. Let I be a grayscale image that has been corrupted by noise. To restore the image I we can look for an image J that is similar to I and changes slowly almost everywhere. In this case the graph $G = (V, E)$ is a grid. The vertices V corresponds to the pixels in I and J. The edges E connect neighboring pixels. The set of labels L are possible intensity, or brightness, values for pixels in J. For our numerical experiments we use $L = \{0, \ldots, 255\}$ corresponding to the possible values in an 8-bit image.

In image restoration with a piecewise smooth model such as the weak string and membrane in [5] the local costs $g_i(a)$ penalize differences between I and J while the pairwise costs $h_{ij}(a, b)$ penalize intensity differences for neighboring pixels in J.

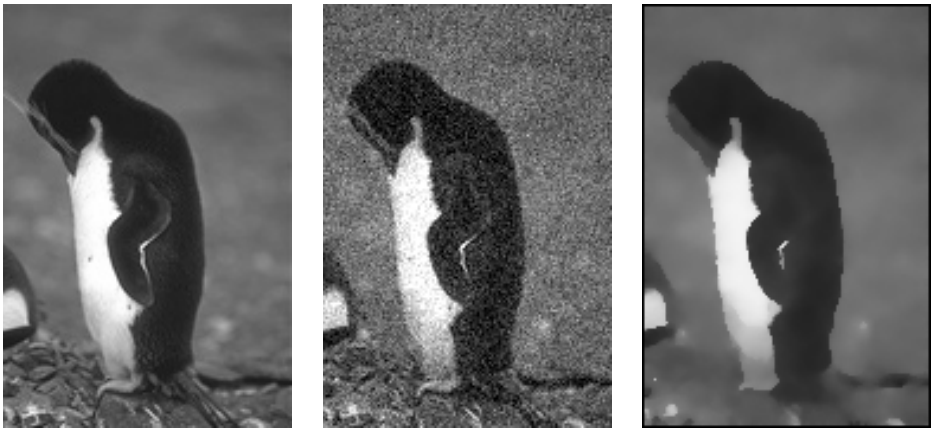
Let $I(i)$ be the intensity of pixel i in image I. The model we used in our image restoration experiments is defined by,

$$g_i(a) = (I(i) - a)^2; \tag{59}$$

$$h_{ij}(a, b) = \begin{cases} 0 & a = b, \\ \alpha & |a - b| = 1, \\ \beta & |a - b| > 1. \end{cases} \tag{60}$$

The discontinuity cost $h_{ij}(a, b)$ depends on two parameters $\alpha, \beta \in \mathbb{R}$ with $\alpha \leq \beta$. These discontinuity costs encourage two neighboring pixels in J to have the same value (cost 0), or differ by one (cost α), but also allows for arbitrarily large discontinuities (cost β).

Figure 2 shows an example result of the algorithm defined by T for image restoration. In this example the noisy image was obtained from a clean image by adding independent Gaussian noisy to each pixel, with standard deviation $\sigma = 20$. We used $\alpha = 500$ and $\beta = 10,000$ to define the discontinuity costs above. The weights w_{ij} were defined to be uniform, $w_{ij} = 1/\mathrm{d}(i)$ and $p = 0.001$. The fixed-point algorithm was run for 100 iterations which took 3 seconds on a 1.6Ghz Intel Core i5 laptop computer. The example illustrates the algorithm is able to recover a clean image that



Original Image Noisy Image Restoration Result

Figure 2: Image restoration using the fixed point algorithm defined by T. The algorithm was run for 100 iterations.

is smooth almost everywhere while at the same time recovering the sharp discontinuities that are present at the boundaries of objects.

The results in Figure 2 are similar to results obtained using max-product belief propagation [8] and other standard algorithms.

5.2 Stereo Depth Estimation

In stereo matching we have two images I_l and I_r taken at the same time from different viewpoints. Most pixels in one image have a corresponding pixel in the other image, corresponding to the projection of the same three-dimensional point. The difference in the coordinates of corresponding pixels is called the disparity. We assume the images are rectified such that a pixel (x, y) in I_l matches a pixel $(x - d, y)$ in I_r with $d \geq 0$. For rectified images the distance of a three-dimensional point to the image plane is inversely proportional to the disparity.

In practice we consider the problem of labeling every pixel in I_l with an integer disparity in $L = \{0, \ldots, D\}$. In this case a labeling x is a disparity map for I_l. The local costs $g_i(a)$ encourage pixels in I_l to be matched to pixels of similar color in I_r. The pairwise costs $h_{ij}(a, b)$ encourage piecewise smooth disparity maps.

The model we used in our stereo experiments is defined by,

$$g_i(a) = \min(\gamma, ||I_l(i) - I_r(i - (a, 0))||_1); \tag{61}$$

$$h_{ij}(a, b) = \begin{cases} 0 & a = b, \\ \alpha & |a - b| = 1, \\ \beta & |a - b| > 1. \end{cases} \tag{62}$$

Here $I_l(i)$ is the value of pixel i in I_l while $I_r(i-(a,0))$ is the value of the corresponding pixel in I_r assuming a disparity a for i. The ℓ_1 norm $||I_l(i) - I_r(i-(a,0))||_1$ defines a distance between RGB values (matching pixels should have similar color). The color distance is truncated by γ to allow for some large color differences which occur due to specular reflections and occlusions. The discontinuity costs are the same as the ones used for image restoration. These costs encourage the disparity values of neighboring pixels to be similar, but also allows for large discontinuities which occur at object boundaries.

Figure 3 shows an example result of disparity estimation using the fixed point algorithm defined by S. In this example we used non-uniform weights w_{ij} to emphasize the relationships between neighboring pixels of similar color, since those pixels are most likely to belong to the same object/surface. The specific parameters we used for the results in Figure 3 were defined by,

$$w_{ij} \propto 0.01 + e^{-0.2||I_l(i)-I_l(j)||_1}, \tag{63}$$

$p = 0.00001$, $\alpha = 500$, $\beta = 1000$ and $\gamma = 20$. The fixed-point algorithm was run for 1,000 iterations which took 13 seconds on a laptop computer.

As in the case of image restoration, the results in Figure 3 are similar to results obtained using max-product belief propagation [8] and other standard algorithms.

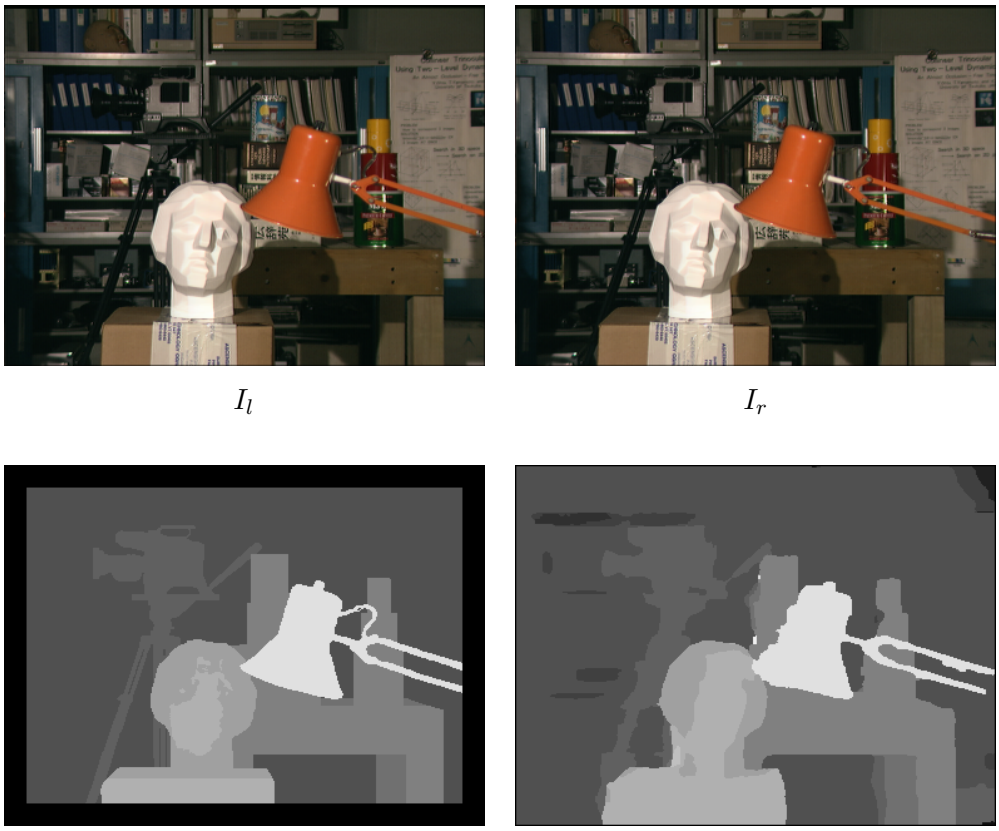
6 Conclusion and Future Work

The experimental results in the last section illustrate the practical feasibility of our algorithms. Our theoretical results prove these algorithms are guaranteed to converge to unique fixed points on graphs with arbitrary topology and with arbitrary pairwise relationships. This includes the case of repulsive interactions which often leads to convergence problems for message passing methods.

Our results can be extended to other contraction maps similar to T and S and alternative methods for computing the fixed points of these maps. Some specific directions for future work are as follows.

1. *Asynchronous updates.* It is possible to define algorithms that update the beliefs φ_i one vertex at a time in a sequential order. As long as all vertices are updated in a "sweep", the resulting algorithms converge to the same fixed point as the parallel update methods we assumed here.

2. *Non-backtracking random walks.* The algorithms defined by S and T can be understood in terms of random walks on G. It is possible to define alternative algorithms based on non-backtracking random walks. In particular, starting with the MDP in Section 4 we can increase the state-space Q to keep track of the last vertex visited in the walk and define transition probabilities that avoid the previous vertex when selecting the next one. The resulting value iteration algorithm becomes very similar to belief propagation and other message passing methods that involve messages defined on the edges of G.



I_l I_r

Ground truth Result of S

Figure 3: Stereo disparity estimation using the fixed point algorithm defined by S on the Tsukuba image pair. The algorithm was run for 1,000 iterations.

References

[1] U. Bertele and F. Brioschi. *Nonserial Dynamic Programming*. Academic Press, 1972.

[2] Dimitry P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 2005.

[3] Julian Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 192–236, 1974.

[4] Julian Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 259–302, 1986.

[5] Andrew Blake and Andrew Zisserman. *Visual reconstruction*. MIT press, 1987.

[6] Endre Boros and Peter L Hammer. Pseudo-boolean optimization. *Discrete applied mathematics*, 123(1):155–225, 2002.

[7] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[8] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient belief propagation for early vision. *International journal of computer vision*, 70(1):41–54, 2006.

[9] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Distance transforms of sampled functions. *Theory of computing*, 8(1):415–428, 2012.

[10] Pedro F. Felzenszwalb and Ramin Zabih. Dynamic programming and graph algorithms in computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4):721–740, 2011.

[11] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[12] D. M. Greig, B. T. Porteous, and A. H. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 271–279, 1989.

[13] Jon Kleinberg and Eva Tardos. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and markov random fields. *Journal of the ACM*, 49(5):616–639, 2002.

[14] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models*. MIT Press, 2009.

[15] Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, 2004.

[16] László Lovász. Random walks on graphs: A survey. *Combinatorics, Paul erdos is eighty*, 2(1):1–46, 1993.

[17] Talya Meltzer, Amir Globerson, and Yair Weiss. Convergent message passing algorithms: a unifying view. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 393–401, 2009.

[18] Martin J. Wainwright and Michael I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305, 2008.