Multi-label MRF Optimization via a Least Squares $s - t$ Cut

Ghassan Hamarneh

School of Computing Science, Simon Fraser University, Canada

Abstract. We approximate the k-label Markov random field optimization by a single binary $(s-t)$ graph cut. Each vertex in the original graph is replaced by only $ceil(log_2(k))$ new vertices and the new edge weights are obtained via a novel least squares solution approximating the original data and label interaction penalties. The $s-t$ cut produces a binary "Gray" encoding that is unambiguously decoded into any of the original k labels. We analyze the properties of the approximation and present quantitative and qualitative image segmentation results, one of the several computer vision applications of multi label-MRF optimization.

1 Introduction

Many visual computing tasks can be formulated as graph labeling problems, *e.g.* segmentation and stereo-reconstruction [1], in which one out of k labels is assigned to each graph vertex. This may be formulated as a k-way cut problem: Given graph $G(V, E)$ with $|V|$ vertices $v_j \in V$ and $|E|$ edges $e_{v_i,v_j} = e_{ij} \in E \subseteq V \times V$ with weights $w(e_{ij}) = w_{ij} > 0$, find an optimal k-cut $C^* \subset E$ with minimal cost $|C^*| = argmin_C|C|$, where $|C| = \sum_{e_{ij} \in C} w_{ij}$, such that $E \backslash C$ breaks the graph into k groups of labelled vertices. This k-cut formulation encodes the semantics of the problem at hand (e.g. segmentation) into w_{ij}. However, if the optimal label assigned to a vertex depends on the labels assigned to other vertices (e.g. to regularize the label field), setting w_{ij} $\forall i, j$ becomes less straightforward. The Markov random field (MRF) formulation captures this desired label interaction via an energy $\xi(l)$ to be minimized with respect to the vertex labels l.

$$\xi(l) = \sum_{v_i \in V} D_i(l_i) + \lambda \sum_{(v_i,v_j) \in E} V_{ij}(l_i, l_j, d_i, d_j) \quad (1)$$

where $D_i(l_i)$ penalizes labeling v_i with l_i, and V_{ij}, aka prior, penalizes assigning labels (l_i, l_j) to neighboring vertices[1]. V_{ij} may be influenced by the data value d_i at v_i (*e.g.* image intensity). λ controls the relative importance of D_i and V_{ij}.

For labeling a P-pixel image, typically a graph G is constructed with $|V| = P$. To encode $D_i(l_i)$, G may be augmented with k new terminal vertices $\{t_j\}_{j=1}^{k}$; each representing one of the k labels (Figure 2(a)) and w_{v_i,t_j} set inversely proportional to $D_i(l_j)$. When $V_{ij} = V_{ij}(d_i, d_j)$, *i.e.* independent of l_i and l_j, V_{ij} may be encoded by $w_{v_i,v_j} \propto V_{ij}\,(d_i, d_j)$. The random walker [2] globally solves a

[1] Higher order priors, e.g. 3^{rd} order $V_{ijk}(l_i, l_j, l_k)$, are also possible.

G. Bebis et al. (Eds.): ISVC 2009, Part I, LNCS 5875, pp. 1055–1066, 2009.

labeling problem of this type, *i.e.* disregarding label interaction. Solving multi-label MRF optimization for any interaction penalty remains an active research area. In [3], the globally optimal binary (k=2) labeling is found using min-cut max-flow. For $k > 2$ with convex prior, the global minimizer is attained by replacing each single k-label variable with k [4] or by using $k-1$ [5] boolean variables. However, convex priors tend to over-smooth the label field. For $k > 2$ with metric or semi-metric priors, Boykov *et al.* performed range moves using binary cuts to expand or swap labels [1]. Other range moves were proposed in [6,7]. More recent approaches to multi-label MRF optimization were proposed based on linear programming relaxation using primal-dual [8], message passing or belief propagation [9], and partial optimality [10] (see [11] for a recent survey).

In this paper, we focus on optimal encoding of the k-label MRF energy solely into the edge weights of a graph. We impose no restrictions on k, or on the order (2^{nd} or higher) or type (*e.g.* non-convex, non-metric, or spatially varying) of the label interaction penalty. The calculated edge weights are optimal in the sense that they minimize the least squares (LS) error when solving a linear system of equations capturing the original MRF penalties. Further, we transform the multi-labelling problem to a binary $s-t$ cut, in which each vertex in the original graph is replaced by the most compact boolean representation; only $ceil(log_2(k))$ vertices represent each k-label variable. In [12], a general framework for converting multi-label problems to binary ones is presented. In contrast to our work, [12] solved a system of equations *to find the boolean encoding function* (not the edge weights), they did not use LS, and their resulting binary problem can still include label interaction. We perform a single (non-iterative and initialization-independent) $s-t$ cut to obtain a "Gray" binary encoding, which is then unambiguously decoded into the k labels. Besides its optimality features, LS enables offline *pre-computation* of pseudoinverse matrices that can be re-used for different graphs.

2 Method

2.1 Reformulating the Multi-label MRF as an $s-t$ Cut

Given a graph $G(V,E)$, the objective is to label each vertex $v_i \in V$ with a label $l_i \in \mathcal{L}_k = \{l_0, l_1, ..., l_{k-1}\}$. Rather than labeling v_i with $l_i \in \mathcal{L}_k$, we replace v_i with b vertices $(v_{ij})_{j=1}^{b}$, and binary-label them with $(l_{ij})_{j=1}^{b}$, *i.e.* $l_{ij} \in \mathcal{L}_2 = \{l_0, l_1\}$. b is chosen such that $2^b \geq k$ or $b = ceil(log_2(k))$, *i.e.* a long enough sequence of *bits* to be decoded into $l_i \in \mathcal{L}_k$[2]. To this end, we transform $G(V,E)$ into a new graph $G_2(V_2,E_2)$ with additional source s and sink t nodes, *i.e.*$|V_2| = b|V| + 2$. E_2 includes terminal links $E_2^{tlinks} = E_2^t \cup E_2^s$ where $|E_2^t| = |E_2^s| = |V_2|$; neighborhood links $E_2^{nlinks} = E_2^{ns} \cup E_2^{nf}$ where $|E_2^{nlinks}| = b^2|E|$, $|E_2^{ns}| = b|E|$, and $|E_2^{nf}| = (b^2 - b)|E|$; and intra-links E_2^{intra} where $|E_2^{intra}| = \binom{b}{2}|V|$. Figure 1 shows these different types of edges. Following an $s-t$ cut on G_2, vertices v_{ij} that remain connected to s are assigned label 0, and the remaining are connected

[2] We distinguish between the decimal (base 10) and binary (base 2) encoding of the labels using the notation $(l_i)_{10}$ and $(l_i)_2 = (l_{i1}, l_{i2}, \cdots, l_{ib})_2$, respectively.

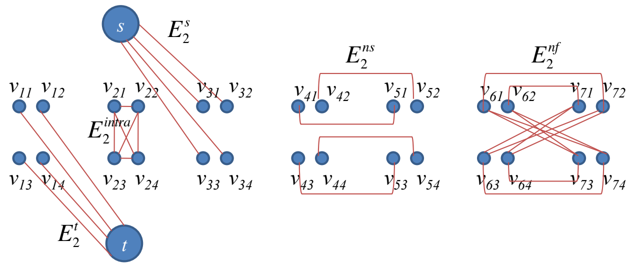

Fig. 1. Edge types in the $s - t$ graph. Shown are seven groups of vertex quadruplets, b=4, and only sample edges from $E_2^t, E_2^s, E_2^{ns}, E_2^{nf}$, and E_2^{intra}.

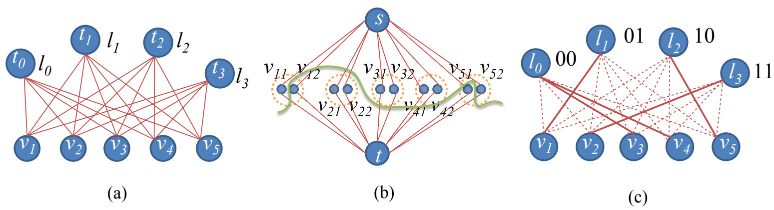

Fig. 2. Reformulating the multi-label problem as an $s - t$ cut. (a) Labeling vertices $\{v_i\}_{i=1}^5$ with labels $\{l_j\}_{j=0}^3$ (only t-links are shown). (b) New graph with 2 terminal nodes $\{s, t\}$, $b = 2$ new vertices (v_{i1} and v_{i2} inside the dashed circles) replacing each v_i in (a), and 2 terminal edges for each v_{ij}. An $s-t$ cut on (b) is depicted as the green curve. (c) Labeling v_i in (a) is based on the $s - t$ cut in (b): Pairs of (v_{i1}, v_{i2}) assigned to (s, s) are labeled with binary string 00, (s, t) with 01, (t, s) with 10, and (t, t) with 11. The binary encodings {00,01,10,11} in turn reflect the original 4 labels $\{l_j\}_{j=0}^3$.

to t and assigned label 1. The string of b binary labels $l_{ij} \in \mathcal{L}_2$ assigned to v_{ij} are then decoded back into a decimal number indicating the label $l_i \in \mathcal{L}_k$ assigned to v_i (Figure 2).

It is important to set the edge weights of E_2 in such a way that decoding the binary labels resulting from the $s - t$ cut of G_2 results in optimal (or close to optimal) labels for the original multi-label problem. To achieve this, we derive a system of linear equations capturing the relation between the original multi-label MRF penalties and the $s - t$ cut cost incurred when generating different label configurations. We then calculate the weights of E_2 as the LS error solution to these equations. The next sections expose the details.

2.2 Data Term Penalty: Severing T-Links and Intra-Links

The 1^{st} order penalty $D_i(l_i)$ in (1) is the cost of assigning l_i to v_i in G, which entails assigning a corresponding sequence of binary labels $(l_{ij})_{j=1}^b$ to $(v_{ij})_{j=1}^b$ in G_2. To assign $(l_i)_2$ to a string of b vertices, appropriate terminal links must be cut. To assign a 0 (resp. 1) label to v_{ij}, the edge connecting v_{ij} to t (resp.

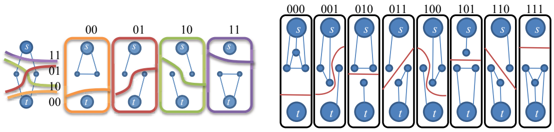

Fig. 3. The 2^b ways of cutting through $\{v_{ij}\}_{j=1}^{b}$ for $b = 2$ (left) and $b = 3$ (right) with the resulting binary codes $\{00, 01, 10, 11\}$ and $\{000, 001, \cdots, 111\}$

s) must be severed (Figure 3). Therefore, the cost of severing t-links in G_2 to assign l_i to vertex v_i in G is calculated as

$$D_i^{tlinks}(l_i) = \sum_{j=1}^{b} l_{ij} w_{v_{ij},s} + \bar{l}_{ij} w_{v_{ij},t} \tag{2}$$

where $\bar{l}_{ij}$ denotes the unary complement (NOT) of l_{ij}. The G_2 $s-t$ cut severing the t-links, as per (2), will also result in severing edges in E_2^{intra} (Figure 1). In particular, $e_{im,in} \in E_2^{intra}$ will be severed iff the $s-t$ cut leaves v_{im} connected to one terminal, say s (resp. t), while v_{in} remains connected to the other terminal t (resp. s). If this condition holds, then $w_{v_{im},v_{in}}$ will contribute to the cost. Therefore, the cost of severing intra-links in G_2 to assign l_i to vertex v_i in G is

$$D_i^{intra}(l_i) = \sum_{m=1}^{b} \sum_{n=m+1}^{b} (l_{im} \oplus l_{in})\, w_{v_{im},v_{in}} \tag{3}$$

where $\oplus$ denotes binary XOR. The total data penalty is the sum of (2) and (3),

$$D_i(l_i) = D_i^{tlinks}(l_i) + D_i^{intra}(l_i). \tag{4}$$

2.3 Prior Term Penalty: Severing N-Links

The interaction penalty $V_{ij}(l_i, l_j, d_i, d_j)$ for assigning l_i to v_i and l_j to neighboring v_j in G must equal the cost of assigning a sequence of binary labels $(l_{im})_{m=1}^{b}$ to $(v_{im})_{m=1}^{b}$ and $(l_{jn})_{n=1}^{b}$ to $(v_{in})_{n=1}^{b}$ in G_2. The cost of this cut can be calculated as (Figure 4)

$$V_{ij}(l_i, l_j, d_i, d_j) = \sum_{m=1}^{b} \sum_{n=1}^{b} (l_{im} \oplus l_{jn})\, w_{v_{im},v_{jn}}. \tag{5}$$

This effectively adds the edge weight between v_{im} and v_{jn} to the cut cost iff the cut results in one vertex of the edge connected to one terminal (s or t) while the other vertex connected to the other terminal (t or s). Note that we impose no restrictions on the left hand side of (5), *e.g.* it could reflect non-convex or non-metric priors, spatially-varying, or even higher order label interaction.

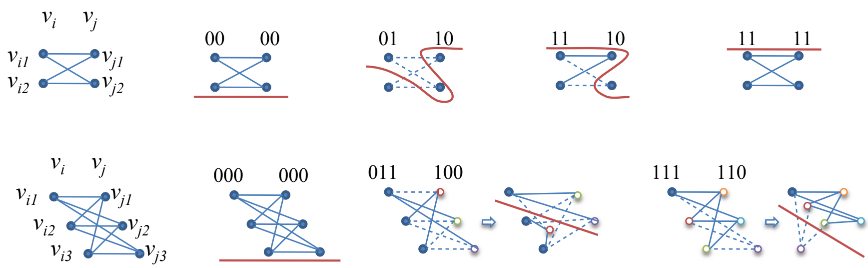

Fig. 4. Severing n-links between neighboring vertices v_i and v_j for $b = 2$ (four examples are shown in the top row) and $b = 3$ (three examples in the bottom row). The cut is depicted as a red curve. In the last two examples for $b = 3$, the colored vertices are translated while maintaining the n-links in order to clearly show that the severed n-links for each case follow (5).

2.4 Edge Weight Approximation with Least Squares

Equations (4) and (5) dictate the relationship between the penalty terms (D_i and V_{ij}) of the original multi-label problem and the severed edge weights $w_{ij,mn}$; $\forall e_{ij,mn} \in E_2$ of the $s-t$ graph G_2. What remains missing before applying the $s-t$ cut, however, is to find these edge weights.

Edge weights of t-links and intra-links. For $b = 1$ (*i.e.* binary labelling), (3) simplifies to $D_i^{intra}(l_i) = 0$ and (4) simplifies to $D_i(l_i) = l_{i1}w_{v_{i1},s} + \bar{l}_{i1}w_{v_{i1},t}$. With $l_i = l_{i1}$ for $b = 1$, substituting the two possible values for $l_i = \{l_0, l_1\}$, we obtain

$$\begin{array}{l} l_i = l_0 \Rightarrow D_i(l_0) = l_0 w_{v_{i1},s} + \bar{l}_0 w_{v_{i1},t} = 0w_{v_{i1},s} + 1w_{v_{i1},t} \\ l_i = l_1 \Rightarrow D_i(l_1) = l_1 w_{v_{i1},s} + \bar{l}_1 w_{v_{i1},t} = 1w_{v_{i1},s} + 0w_{v_{i1},t} \end{array} \tag{6}$$

which can be written in matrix form $A_1 X_1^i = B_1^i$ as $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} w_{v_{i1},s} \\ w_{v_{i1},t} \end{pmatrix} = \begin{pmatrix} D_i(l_0) \\ D_i(l_1) \end{pmatrix}$ where X_1^i is the vector of unknown edge weights connecting vertex v_{i1} to s and t, B_1^i is the data penalty for v_i, and A_1 is the matrix of coefficients. The subscript 1 in A_1, X_1^i, and B_1^i indicates that this matrix equation is for $b = 1$. Clearly, the solution is trivial and expected: $w_{v_{i1},s} = D_i(l_1)$ and $w_{v_{i1},t} = D_i(l_0)$

For $b = 2$, we address multi-label problems of $k = \{3, 4\}$, or $2^{b-1} = 2 < k \leq 2^b = 4$ labels. Substituting the $2^b = 4$ possible label values, ((0,0),(0,1),(1,0), and (1,1)), of $(l_i)_2 = (l_{i1}, l_{i2})$ in (4) we obtain

$$\begin{array}{l} (0,0) \Rightarrow D_i(l_0) = 0w_{v_{i1},s} + 1w_{v_{i1},t} + 0w_{v_{i2},s} + 1w_{v_{i2},t} + 0w_{v_{i1},v_{i2}} \\ (0,1) \Rightarrow D_i(l_1) = 0w_{v_{i1},s} + 1w_{v_{i1},t} + 1w_{v_{i2},s} + 0w_{v_{i2},t} + 1w_{v_{i1},v_{i2}} \\ (1,0) \Rightarrow D_i(l_2) = 1w_{v_{i1},s} + 0w_{v_{i1},t} + 0w_{v_{i2},s} + 1w_{v_{i2},t} + 1w_{v_{i1},v_{i2}} \\ (1,1) \Rightarrow D_i(l_3) = 1w_{v_{i1},s} + 0w_{v_{i1},t} + 1w_{v_{i2},s} + 0w_{v_{i2},t} + 0w_{v_{i1},v_{i2}} \end{array} \tag{7}$$

which can be written in matrix form $A_2 X_2^i = B_2^i$ as

$$\begin{pmatrix} 0\,1\,0\,1\,0 \\ 0\,1\,1\,0\,1 \\ 1\,0\,0\,1\,1 \\ 1\,0\,1\,0\,0 \end{pmatrix} \begin{pmatrix} w_{v_{i1},s} \\ w_{v_{i1},t} \\ w_{v_{i2},s} \\ w_{v_{i2},t} \\ w_{v_{i1},v_{i2}} \end{pmatrix} = \begin{pmatrix} D_i(l_0) \\ D_i(l_1) \\ D_i(l_2) \\ D_i(l_3) \end{pmatrix}. \tag{8}$$

In general, for any b, we have

$$A_b X_b^i = B_b^i \tag{9}$$

where X_b^i is a column vector of length $2b+\binom{b}{2}$ (superscript ^{t} denotes transpose),

$$\begin{aligned} X_b^i = (&w_{v_{i1},s}, w_{v_{i1},t}, w_{v_{i2},s}, w_{v_{i2},t}, \cdots, w_{v_{ib},s}, w_{v_{ib},t}, \\ &w_{v_{i1},v_{i2}}, w_{v_{i1},v_{i3}}, \cdots, w_{v_{i1},v_{ib}}, w_{v_{i2},v_{i3}}, \cdots, w_{v_{i2},v_{ib}}, \cdots, w_{v_{i,b-1},v_{ib}})^t \end{aligned} \tag{10}$$

A_b is a $2^b \times (2b + \binom{b}{2})$ matrix whose jth row $A_b(j,:)$ is given by

$$\begin{aligned} A_b(dec(l_{i1}l_{i2}\cdots l_{ib}),:) = (&l_{i1}, \bar{l}_{i1}, l_{i2}, \bar{l}_{i2}, \cdots, l_{ib}, \bar{l}_{ib}, \\ &l_{i1} \oplus l_{i2}, l_{i1} \oplus l_{i3}, \cdots, l_{i1} \oplus l_{ib}, l_{i2} \oplus l_{i3}, l_{i2} \oplus l_{ib}, \cdots, l_{i,b-1} \oplus l_{ib}) \end{aligned} \tag{11}$$

where $dec(.)$ is the decimal equivalent of its binary argument. B_b^i is a 2^b-long column vector given by $B_b^i = (D_i(l_0), D_i(l_1), D_i(l_2), \cdots, D_i(l_{2^b-1}))^t$.

We now solve the linear system of equations in (9) to find the optimal, in a LS sense, t-links and intra-links edge weights $\hat{X}_b^i$ related to every vertex v_i using

$$\hat{X}_b^i = A_b^+ B_b^i \tag{12}$$

where A^+ is the (Moore-Penrose) pseudoinverse of A calculated using singular value decomposition (SVD)[13].

Edge weights of n-links. For $b = 1$ (*i.e.* binary labelling), (5) simplifies to $(l_i \oplus l_j)w_{ij} = V_{ij}(l_i, l_j, d_i, d_j)$, where $w_{v_{i1},v_{j1}}$ has been replaced by $w_{i,j}$ and l_{i1} and l_{j1} have been replaced by l_i and l_j, since they are equivalent for $b = 1$. If $V_{ij}(l_i, l_j, d_i, d_j) = V_{ij}(d_i, d_j)$, *i.e.* label-independent, we can simply ignore the outcome of $l_i \oplus l_j$ by setting it to a constant. Then, the solution is trivial and as expected (Section 1): $w_{i,j} \propto V_{ij}(d_i, d_j)$. However, in the general case when V_{ij} depends on the labels l_i and l_j of the neighboring vertices v_i and v_j, a single edge weight is insufficient to capture such elaborate label interactions, intuitively, because $w_{i,j}$ needs to take on a different value for every pair of labels. To address this problem, we substitute in (5) each of the $2^b 2^b = 2^{2b} = 2^2 = 4$ possible combinations of pairs of labels $(l_i, l_j) \in \{l_0, l_1\} \times \{l_0, l_1\} = \{0,1\} \times \{0,1\}$, and obtain the following system of linear equations:

$$\begin{aligned} (l_0, l_0) = (0,0) &\Rightarrow V_{ij}(l_0, l_0, d_i, d_j) = (0 \oplus 0)\, w_{i,j} = 0 \\ (l_0, l_1) = (0,1) &\Rightarrow V_{ij}(l_0, l_1, d_i, d_j) = (0 \oplus 1)\, w_{i,j} = w_{i,j} \\ (l_1, l_0) = (1,0) &\Rightarrow V_{ij}(l_1, l_0, d_i, d_j) = (1 \oplus 0)\, w_{i,j} = w_{i,j} \\ (l_1, l_1) = (1,1) &\Rightarrow V_{ij}(l_1, l_1, d_i, d_j) = (1 \oplus 1)\, w_{i,j} = 0 \end{aligned} \tag{13}$$

which is written in matrix form $S_1 Y_1^{ij} = T_1^{ij}$ as

$$\begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} (w_{i,j}) = \begin{pmatrix} V_{ij}(l_0, l_0, d_i, d_j) \\ V_{ij}(l_0, l_1, d_i, d_j) \\ V_{ij}(l_1, l_0, d_i, d_j) \\ V_{ij}(l_1, l_1, d_i, d_j) \end{pmatrix} \tag{14}$$

where Y_1^{ij} is the unknown n-link weight $w_{i,j}$ connecting v_i to neighboring v_j. The 1^{st} and 4^{th} equations capture the condition that in order to guarantee the same label for neighboring vertices then the edge weight connecting them should be infinite $(0/V_{ij})$ and, hence, never severed. Solving for w_{ij} using pseudoinverse gives $w_{ij} = S_1^+ T_1^i = \frac{1}{2}(V_{ij}(l_0, l_1, d_i, d_j) + V_{ij}(l_1, l_0, d_i, d_j))$ since $S_1^+ = (0, 0.5, 0.5, 0)$, *i.e.* w_{ij} is equal to the average between the interaction penalties of the two cases when the labels are different.

For $b = 2$, (5) simplifies to

$$\begin{aligned} V_{ij}(l_i, l_j, d_i, d_j) = (l_{i1} \oplus l_{j1})\, w_{v_{i1},v_{j1}} + (l_{i1} \oplus l_{j2})\, w_{v_{i1},v_{j2}} + \\ (l_{i2} \oplus l_{j1})\, w_{v_{i2},v_{j1}} + (l_{i2} \oplus l_{j2})\, w_{v_{i2},v_{j2}} \end{aligned} \tag{15}$$

We can now substitute all possible $2^b 2^b = 2^{2b} = 16$ combinations of the pairs of interacting labels $(l_i, l_j) \in \{l_0, l_1, l_2, l_3\} \times \{l_0, l_1, l_2, l_3\}$, or equivalently, $((l_i)_2, (l_j)_2) \in \{00, 01, 10, 11\} \times \{00, 01, 10, 11\}$. Here are a few examples,

$$\begin{aligned} (l_0, l_0) = (00, 00) \Rightarrow V_{ij}(l_0, l_0, d_i, d_j) = 0 w_{v_{i1},v_{j1}} + 0 w_{v_{i1},v_{j2}} + 0 w_{v_{i2},v_{j1}} + 0 w_{v_{i2},v_{j2}} \\ (l_1, l_2) = (01, 10) \Rightarrow V_{ij}(l_1, l_2, d_i, d_j) = 1 w_{v_{i1},v_{j1}} + 0 w_{v_{i1},v_{j2}} + 0 w_{v_{i2},v_{j1}} + 1 w_{v_{i2},v_{j2}} \\ (l_3, l_3) = (11, 11) \Rightarrow V_{ij}(l_3, l_3, d_i, d_j) = 0 w_{v_{i1},v_{j1}} + 0 w_{v_{i1},v_{j2}} + 0 w_{v_{i2},v_{j1}} + 0 w_{v_{i2},v_{j2}} \end{aligned} \tag{16}$$

Writing all the 16 equations, we obtain the system of linear equations in matrix format as $S_2 Y_2^{ij} = T_2^{ij}$, where $Y_2^{ij} = (w_{v_{i1},v_{j1}}, w_{v_{i1},v_{j2}}, w_{v_{i2},v_{j1}}, w_{v_{i2},v_{j2}})^t$ is the 4×1 vector of unknown n-link edge weights, T_2^{ij} is a 16×1 vector whose entries are the different possible interaction penalties $((V_{ij}(l_i, l_j, d_i, d_j))_{i=0}^3)_{j=0}^3$, and S_2 is a 16×4 matrix with 0 or 1 entires resulting from $\oplus$.

In general, for any b, we obtain the following linear system of equations

$$S_b Y_b^{ij} = T_b^{ij} \tag{17}$$

where Y_b^{ij} is the $b^2 \times 1$ vector of unknown n-link edge weights, S_b is $2^{2b} \times b^2$ matrix of 0s and 1s, and T_b^{ij} is a $2^{2b} \times 1$ vector of interaction penalties.

We now solve the linear system of equations in (17) to find the optimal, in a LS sense, n-links edge weights $\hat{Y}_b^{ij}$ related to a pair of vertices v_i and v_j using

$$\hat{Y}_b^{ij} = S_b^+ T_b^{ij}. \tag{18}$$

Solving (18) for every pair of neighboring vertices v_i and v_j, we obtain the weights of all edges in E_2^{inter}, and solving (12) for every vertex v_i, we obtain the weights of all edges in $E_2^{tlinks} \cup E_2^{intra}$, *i.e.* $w_{ij,mn}, \forall e_{ij,mn} \in E_2$ are now known. It is important to note that some of these resulting edge weights may turn out negative. In order to guarantee positive weights and hence guarantee a globally optimal cut of G_2 in polynomial time, we simply add the same constant to all the edge weights in G_2 to translate all the values to become larger than zero. We now calculate the minimal $s-t$ cut of G_2 to obtain the binary labeling of every vertex in $V_2 = \{\{v_{ij}\}_{i=1}^{|V|}\}_{j=1}^b$. Finally, every sequence of b binary labels $(v_{ij})_{j=1}^b$ is decoded to a decimal label $l_i \in \mathcal{L}_k = \{l_0, l_1, ..., l_{k-1}\}, \forall v_i \in V$, *i.e.* the solution to the original multi-label MRF problem.

2.5 Gray Encoding of Extra Labels

When $k < 2^b$, an $s - t$ cut may generate *extra labels* (the nth label l_{n-1} is extra iff $k < n \leq 2^b$), which must be replaced or merged with a *non-extra* label (the mth label l_{m-1} is non-extra iff $2^{b-1} < m \leq k$). To replace l_n with l_m, we replace $D_i(l_n)$ with $D_i(l_m)$ in (4), $V_{ij}(l_n, l_j, d_i, d_j)$ with $V_{ij}(l_m, l_j, d_i, d_j)$, and $V_{ij}(l_i, l_n, d_i, d_j)$ with $V_{ij}(l_i, l_m, d_i, d_j)$ in (5). Rather than merging arbitrary labels, we adopt a Gray encoding scheme that minimizes the Hamming distance between the binary codes of merged labels. We first note that the most significant bit of any extra label l_n will always be 1. Then, l_n is merged with the non-extra label whose binary code is identical to l_n *except* for having 0 as its most significant bit: *i.e.* $(l_n)_2 = (1, l_2, \cdots, l_b)_2$ is merged with $(l_m)_2 = (0, l_2, \cdots, l_b)_2$.

3 Results

3.1 LS Error and Rank Deficiency Analysis

LS approximation error is a well studied topic (e.g. [13]). Table 1 summarizes the main properties (number of equations, unknowns, and rank) when solving for the weights of t-links and intra-links (A_b in (9)) and for inter-links (S_b in (17)) for increasing bits b. We note that, not surprisingly, the only full-rank case is A_1 (*i.e.* binary segmentation). A_b is underdetermined for $b = 2, 3$ and overdetermined for $b \geq 4$. All cases of S_b are rank deficient and overdetermined.

In Figure 5, we present empirical results of LS error $e_b = |B_b^i - \hat{B}_b^i|/|B_b^i| = |(I - A_b A_b^+) B_b^i|/|B_b^i|$ (when solving for t-links and intra-links) and $e_t = |T_b^{ij} - \hat{T}_b^{ij}|/|T_b^{ij}| = |(I - S_b S_b^+) T_b^{ij}|/|T_b^{ij}|$ (for n-links), for increasing number of labels k, where I is the identity matrix and $|.|$ is the l^2-norm. The plots are the result of a Monte Carlo simulation of 500 random realizations of B_b^i and T_b^{ij} for every k. Note how e_b starts at exactly zero for binary segmentation ($b = 1$), as expected. As k increases, the average of e_b increases with an (empirical) upper bound of 0.5, while its variance decreases. e_t is non-zero even for $b = 1$ (Section 2.4) and converges to 0.5 with increasing k.

Table 1. Properties of the systems of linear equations (9) and (17). For increasing b, the number of equations e, number of unknowns u, and ranks r of A_b and S_b are shown. For A_b, u_0 is when intra-links are not used ($E_2^{intra} = \emptyset$) and, for S_b, u_0 is when only sparse n-links are used ($E_2^{nf} = \emptyset$).

b	A_b in (9)					S_b in (17)				
bits	$e = 2^b$	$u = 2b + \binom{b}{2}$	r	$u_0 = 2b$	r_0	$e = 2^{2b}$	$u = b^2$	r	$u_0 = b$	r_0
1	2	2	2	2	2	4	1	1	1	1
2	4	5	4	4	3	16	4	4	2	2
3	8	9	7	6	4	64	9	9	3	3
4	16	14	11	8	5	256	16	16	4	4
5	32	20	16	10	6	1024	25	25	5	5
6	64	27	22	12	7	4096	36	36	6	6
7	128	35	29	14	8	16384	49	49	7	7
8	256	44	37	16	9	65536	64	64	8	8

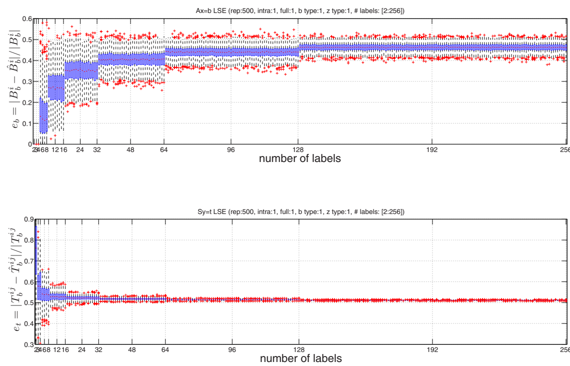

Fig. 5. LS error for increasing number of labels. (top) Error e_b in estimating the weights of $E_2^{tlinks} \cup E_2^{intra}$. (bottom) Error e_t in estimating the weights of E_2^{nlinks}. The general behavior of the error is clear from the figure. The reader may refer to the electronic copy of this paper for color and scalable graphics.

3.2 Effect of LS Error in Edge Weights on the $s-t$ Cut

The LS error in edge weights induces error in the $s-t$ cut or binary labeling, which is decoded into a suboptimal solution to the multi-label problem. To quantify the cut cost error $\triangle|C|$ and the labeling accuracy ACC due to edge weight error, we create a graph G with a proper topology (*i.e.* reflecting the 4-connectedness of 2D image pixels) and edge weights sampled from a uniform probability distribution function (PDF) with support $[0, 1]$. We then construct G_{LSE}, a noisy version of G, by adding uniformly distributed noise with support $[0, \text{noise level}]$ to the edge weights. Figure 6 shows the results of $\triangle|C| = ||C| - |C_{LSE}||/|C|$ and $ACC = (TP+TN)/|V|$, where $|C| = \sum_{e_{ij} \in C} w_{ij}$ is the cut cost of G, $|C_{LSE}|$ is the cut cost of G_{LSE}, and $TP + TN$ is the number of correctly labelled vertices (*i.e.* true positive and true negatives), and $|V|$ is the number of vertices in G. The plots are the results of a Monte Carlo simulation of 20 realizations of G and G_{LSE} each with 10,000 vertices.

3.3 Image Segmentation Results

We evaluated our method's segmentation accuracy by calculating the average (over all labels) Dice similarity coefficient $\overline{DSC}$ [14](Figure 7(left)) on synthetic (with known ground truth) images: $I(x, y) : R^2 \rightarrow [0, 1]$, containing ellipses with random major and minor axes and varying pixel intensities (Figure 7(right)). We tested increasing levels of Gaussian noise $\sim \mathcal{N}(0, \sigma \in \{0, 0.05, 0.10, \cdots, 0.40\})$, labels $k = \{2, 3, \cdots, 16\}$, and with non-convex Pott's label interaction weighted by a spatially varying Gaussian image intensity penalty [3]. We ran 10 realization for each test case. For pixel i with intensity d_i, $D_i(l_i) = (p_l(\mu) - p_l(x_i))/p_l(\mu)$, where $p_l(d) \sim \mathcal{N}(\mu_l, \sigma_l)$ is a Gaussian PDF learned from 50% of the pixels of

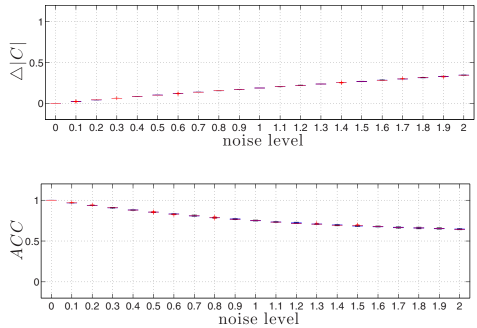

Fig. 6. Cut cost error $\triangle|C|$ and labeling accuracy ACC as we corrupt the edge weights with increasing levels of noise

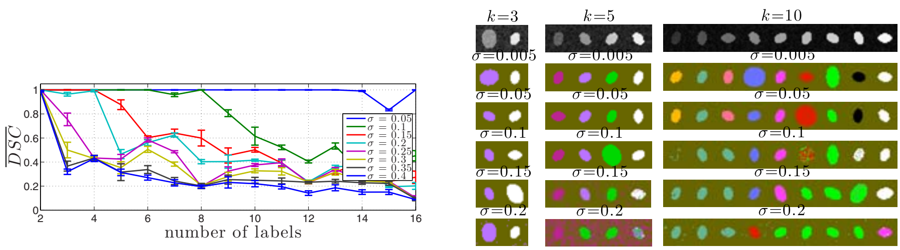

Fig. 7. Segmentation results on images of ellipses. (left) $\overline{DSC}$ between the ground truth and our method's segmentation with increasing number of labels and noise levels (different colors). (right) Sample qualitative results with k labels ($k-1$ ellipses plus background) and noise level σ. (top row) sample intensity images; (remaining rows) labeling results.

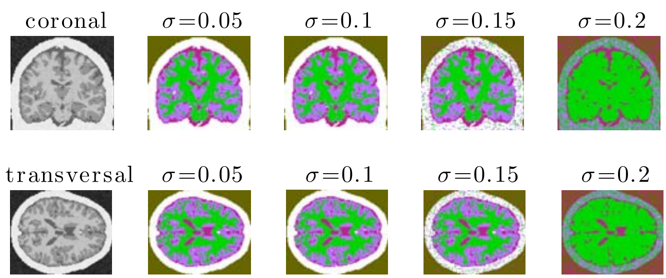

Fig. 8. Brain MRI segmentation on coronal (top) and transversal (bottom) slices for increasing noise σ

each region (or label) l of the noisy image (mimicking seeding). Note that $\overline{DSC}$ gradually decreases from unity with increasing σ or k, *e.g.* the topmost curve (blue, $\sigma = 0.05$) shows almost perfect segmentation; $\overline{DSC} = 1\ \forall k$, whereas $\overline{DSC}$ drops below 1 for $k \geq 9$ for the second-from-top curve (green, $\sigma = 0.1$), and for $k \geq 5$ for third curve (red, $\sigma = 0.15$).

We present qualitative segmentation results on synthetic data (Figure 7(right)) and on magnetic resonance brain images (Figure 8) from BrainWeb [15].

4 Conclusions

Multi-label MRF optimization with non-trivial priors is a challenging problem with several computer vision applications. In our proposed approach, rather than labeling a vertex with one of k labels, the vertex is replaced by $b = ceil(log_2(k))$ new vertices that are binary-labelled to encode the original k labels; effectively approximating the multi-label problem with a globally and non-iteratively solvable $s-t$ cut. The new $s-t$ graph is optimal in a least squares sense because its edge weights are the LS error solution of a system of linear equations capturing the original multi-label MRF energy, without any restrictions on the interaction priors. To the best of our knowledge, this is the first work to use LS to approximate the multi-label MRF with any order of label interaction solely via the edge wights of a graph (with no label interaction). Offline pre-computation of A_b^+ and S_b^+ in (12) and (18) is performed only once for each b value then re-used for different vertices and graphs. We quantitatively evaluated different properties of the proposed approximation and demonstrated its application to image segmentation (with qualitative and quantitative results on synthetic and brain images). More elaborate analysis of the algorithm (*e.g.* error bounds, value of the minimized energy, computational complexity, running times) and comparison with state-of-the-art approaches on standard benchmarks is left for future work. Further, we are exploring the use of non-negative least squares (*e.g.* Chapter 23 in [16]) to guarantee non-negative edge weights as well as quantifying the benefits of the Gray encoding.

Acknowledgements

We thank Yuri Boykov, Vladimir Kolmogorov, and Olga Veksler for the MRF optimization code and Shai Bagon for its MATLAB wrapper; Marc Koppert and Ben Smith for discussions and assistance in conducting some initial experiments; and the reviewers for their valuable feedback.

References

1. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE PAMI 23, 1222–1239 (2001)
2. Grady, L.: Random walks for image segmentation. IEEE PAMI 28, 1768–1783 (2006)
3. Boykov, Y., Funka-Lea, G.: Graph cuts and efficient N-D image segmentation. International Journal of Computer Vision 70, 109–131 (2006)
4. Ishikawa, H.: Exact optimization for markov random fields with convex priors. IEEE PAMI 25, 1333–1336 (2003)

5. Schlesinger, D., Flach, B.: Transforming an arbitrary min-sum problem into a binary one. Tech. Report TUD-FI06-01, Dresden University (2006)
6. Veksler, O.: Graph cut based optimization for MRFs with truncated convex priors. In: IEEE CVPR, pp. 1–8 (2007)
7. Lempitsky, V., Rother, C., Blake, A.: Logcut - efficient graph cut optimization for markov random fields. In: IEEE ICCV, pp. 1–8 (2007)
8. Komodakis, N., Tziritas, G., Paragios, N.: Performance vs computational efficiency for optimizing single and dynamic MRFs: Setting the state of the art with primal-dual strategies. Comput. Vis. Image Underst. 112, 14–29 (2008)
9. Kolmogorov, V.: Convergent tree-reweighted message passing for energy minimization. IEEE PAMI 28, 1568–1583 (2006)
10. Kohli, P., Shekhovtsov, A., Rother, C., Kolmogorov, V., Torr, P.: On partial optimality in multi-label MRFs. In: International Conference on Machine learning (ICML), pp. 480–487. ACM, New York (2008)
11. Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M., Rother, C.: A comparative study of energy minimization methods for markov random fields with smoothness-based priors. IEEE PAMI 30, 1068–1080 (2008)
12. Ramalingam, S., Kohli, P., Alahari, K., Torr, P.: Exact inference in multi-label CRFs with higher order cliques. In: IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008, pp. 1–8 (2008)
13. Björk, A.: Numerical Methods for Least Squares Problem. SIAM, Philadelphia (1996)
14. Dice, L.R.: Measures of the amount of ecologic association between species. Ecology 26, 279–302 (1945)
15. Cocosco, C.A., Kollokian, V., Kwan, R.K.-s., Pike, G.B., Evans, A.C.: Brainweb: Online interface to a 3D mri simulated brain database. NeuroImage 5, 425 (1997)
16. Lawson, C., Hanson, R.: Solving Least Squares Problems. Society for Industrial Mathematics (1974)