New Convex Relaxations for MRF Inference with Unknown Graphs

Zhenhua Wang [†*], Tong Liu [†], Qinfeng Shi [‡], M. Pawan Kumar [♯], Jianhua Zhang [†]
[†] Zhejiang University of Technology, [‡] The University of Adelaide, [♯] University of Oxford
{zhhwang, zjh}@zjut.edu.cn, javen.shi@adelaide.edu.au, pawan@robots.ox.ac.uk

Abstract

Treating graph structures of Markov random fields as unknown and estimating them jointly with labels have been shown to be useful for modeling human activity recognition and other related tasks. We propose two novel relaxations for solving this problem. The first is a linear programming (LP) relaxation, which is provably tighter than the existing LP relaxation. The second is a non-convex quadratic programming (QP) relaxation, which admits an efficient concave-convex procedure (CCCP). The CCCP algorithm is initialized by solving a convex QP relaxation of the problem, which is obtained by modifying the diagonal of the matrix that specifies the non-convex QP relaxation. We show that our convex QP relaxation is optimal in the sense that it minimizes the ℓ_1 norm of the diagonal modification vector. While the convex QP relaxation is not as tight as the existing and the new LP relaxations, when used in conjunction with the CCCP algorithm for the non-convex QP relaxation, it provides accurate solutions. We demonstrate the efficacy of our new relaxations for both synthetic data and human activity recognition.

1. Introduction

Traditional methods for maximum *a posteriori* (MAP) inference, or energy minimization, in Markov random fields (MRFs) assume a known graph structure of the MRF [20, 11, 26]. However, the assumption is weak due to two reasons. First, it restricts the use of MRFs to problems where all the instances are *homogeneous*, that is, when the underlying graphs for all problem instances are the same. Even in the case of a time series problem, where the number of nodes of a MRF varies across instances, this assumption forces us to retain the same basic structure across two consecutive time frames (for example, see Fig. 6.1 of [10]). Second, due to the lack of information of the relationship among variables, domain knowledge or human heuristics are typically utilized to construct graphs such as trees, grids and fully-connected nets [5, 12, 3, 26, 25], which probably are not the most desirable structure. In order to overcome the deficiency of MRFs with known graph structures, recent research has started to focus on treating the graph structure as unknown and estimating proper structures jointly with labels [15, 24, 16]. Such an approach lends itself naturally to various real-world problems where the graphs are *heterogeneous*. For example, when modeling human actions in TV episodes, where each node represents an actor, the graph structure changes dynamically as different actors enter and exit the scene.

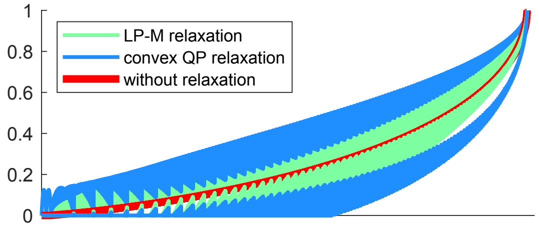


Figure 1: An illustration of the proposed new convex relaxations using a toy example. Both upper and lower bounds for these relaxations are shown. The LP-M relaxation (green) is provably tighter than the convex QP relaxation (blue), which in turn delivers significantly better solutions to the original problem (see Section 5).

The problem of simultaneous estimating labels and graphs was first introduced by Lan *et al.* [15], who suggested an alternating search strategy to obtain an approximate solution. Specifically, their method alternates between finding the best labels for a fixed graph, and finding the best graph for the fixed labels. While such an approach is computationally efficient, it is prone to bad local minima solutions. Wang *et al.* [24] cast this problem as an integer quadratic program. By dropping the integral constraints, they obtain a bilinear program, which they further relaxed to a linear program (LP). The LP relaxation of [24], which we refer to as LP-W, can either be solved over the entire feasible region, or it can be used in conjunction with a branch-and-bound strategy [2]. Each subproblem of the branch-and-bound approach requires us to solve the LP-W relaxation over a subset of the feasible region, which makes it computationally infeasible for even a smaller number of nodes. Typically, the branch-and-bound approach is stopped early, which results in sub-optimal solutions.

In order to alleviate the deficiencies of the previous methods, we propose two new convex relaxations for the problem of simultaneously estimating the MRF labels and graph structures. The first relaxation is a new LP relaxation, which replaces the upper and lower bounds of the variables in LP-W with linear marginalization constraints. We provide a sound theoretical justification of our LP, denoted by LP-M, by showing that it is provably tighter than LP-W. The second relaxation is a non-convex QP relaxation, which admits an efficient concave-convex procedure (CCCP) though it is looser than LP-M (see Figure 1 for an illustration). In order to initialize the CCCP, we propose a convexification of the non-convex QP. Similar to the convex QP of Ravikumar and Lafferty [19] for fixed graph structures, our convex QP relaxation modifies the diagonal of the matrix that is used to specify the objective of the QP. However, unlike [19], our convex QP can be shown to be optimal, that is, it minimizes the ℓ_1 norm of the diagonal modification vector. Using synthetic and real human activity recognition data, we empirically demonstrate that our relaxations provide salient improvements over existing approaches.

2. Preliminaries

Notation. We closely follow the notation of [24]. Specifically, the MRF is represented by a graph $G = (V, E)$, where the node set $V = \{1, 2, \cdots, n\}$ is known but the edge set E, which represents the graph structure, is unknown. Each node i is associated with a random variable, which needs to be assigned a label $y_i \in \mathcal{Y}$. For example, the nodes can represent the actors of a TV show, and the labels can represent their actions. The labeling of all the nodes is denoted by $Y = (y_1, \cdots, y_n)$. The node potential for assigning a node i to the label y_i is denoted by $\theta_i(y_i)$. Similarly, the pairwise (edge) potential for assigning two neighboring nodes i and j to the labels y_i and y_j respectively is denoted by $\theta_{ij}(y_i, y_j)$. Here, two nodes are said to be neighboring if they are connected by an edge in the set E.

Problem Formulation. We would like to obtain the labeling Y and the edge set E such that it minimizes the sum of the node and pairwise potentials, that is,

$$\begin{aligned} \min_{Y,E} \quad & \textstyle\sum_{i \in V} \theta_i(y_i) + \sum_{(i,j) \in E} \theta_{ij}(y_i, y_j), \\ \text{s.t.} \quad & \text{degree}(i) \leq h, \forall i \in V. \end{aligned} \tag{1}$$

Here $\text{degree}(i)$ denotes the number of edges in E that are incident on i. By constraining the degree of each node, we ensure that we obtain a simple graph structure. Note that, unlike the MAP inference problem with a fixed graph structure, the above problem is not invariant to reparameterizations of the energy function. In other words, the above problem needs to be specified using potential functions that are scaled in such a manner as to provide accurate solutions for the corresponding application. Fortunately, Wang *et al.* [24] showed that such potentials can be learned from a training dataset using a latent support vector machine formulation [6, 27]. We refer the reader to [24] for details.

In this paper, we are concerned with the optimization of problem (1) for a given set of potential functions. To this end, we note that problem (1) can be specified as an integer quadratic program, using the three types of binary variables: (i) $\mu_i(y_i) \in \{0, 1\}$ indicates whether i is assigned the label y_i; (ii) $\mu_{ij}(y_i, y_j) \in \{0, 1\}, i < j$, indicates whether the variables i and j are assigned the labels y_i and y_j; and (iii) $z_{ij} \in \{0, 1\}$ indicates whether the edge (i, j) is present in the edge set E. Using the above variables, the problem of simultaneously estimating MRF labels and graph structure can be formulated as follows:

$$\begin{aligned} \min_{\boldsymbol{\mu}, \mathbf{z}} \quad & \sum_{i \in V} \sum_{y_i} \mu_i(y_i) \theta_i(y_i) + \\ & \sum_{i,j \in V, i<j} \sum_{y_i, y_j} \mu_{ij}(y_i, y_j) z_{ij} \theta_{ij}(y_i, y_j), \\ \text{s.t.} \quad & \mu_i(y_i) \in \{0,1\}, \mu_{ij}(y_i, y_j) \in \{0,1\}, z_{ij} \in \{0,1\}, \\ & \textstyle\sum_{y_i} \mu_{ij}(y_i, y_j) = \mu_j(y_j), \sum_{y_j} \mu_{ij}(y_i, y_j) = \mu_i(y_i), \\ & \textstyle\sum_{j>i} z_{ij} + \sum_{j<i} z_{ji} \leq h, \\ & \textstyle\sum_{y_i} \mu_i(y_i) = 1, \forall i, j \in V, i < j, y_i, y_j. \end{aligned} \tag{2}$$

Decoding. Decoding is to extract the estimated edge set E^* and labelling Y^* from μ and z solutions. To obtain Y^*, one typically let $Y^* = (y_1^*, y_2^*, \ldots, y_n^*)$, where $y_i^* = \text{argmin}_{y_i \in \mathcal{Y}} \mu_i(y_i)$. To obtain E^*, one can start with $E = \emptyset$. Then $\forall i, j \in V, i < j$, if $z_{ij} \geq 0.5$, $E^* = E^* \cup \{(i, j)\}$.

Existing LP Relaxation. Wang *et al.* [24] proposed a linear programming (LP) relaxation for the above integer quadratic program, which we denote by LP-W. The LP-W relaxation replaces the quadratic term in the objective function, namely $\mu_{ij}(y_i, y_j) z_{ij}$, by a new optimization variable $\lambda_{ij}(y_i, y_j)$. It introduces an upper and lower bound for $\lambda_{ij}(y_i, y_j)$, which are linear in $\mu_{ij}(y_i, y_j)$ and z_{ij}. By dropping the integrality constraints, the resulting convex program can be solved in a time that is polynomial in the size of the problem. Formally, the LP-W relaxation is as:

$$\begin{aligned} \min \quad & \sum_{i \in V} \sum_{y_i} \mu_i(y_i) \theta_i(y_i) + \\ & \sum_{i,j \in V, i<j} \sum_{y_i, y_j} \theta_{ij}(y_i, y_j) \lambda_{ij}(y_i, y_j), \\ \text{s.t.} \quad & \lambda_{ij}(y_i, y_j) \geq \max\{0, z_{ij} + \mu_{ij}(y_i, y_j) - 1\}, \\ & \lambda_{ij}(y_i, y_j) \leq \min\{z_{ij}, \mu_{ij}(y_i, y_j)\}, \\ & \forall i, j \in V, i < j, y_i, y_j, \ \mu, z \in \mathcal{O}\,. \end{aligned} \tag{3}$$

where $\mathcal{O}$ denotes a space which is defined as

$$\mathcal{O} = \left\{ \mu, z \;\middle|\; \begin{array}{l} \mu_{ij}(y_i, y_j), z_{ij} \in [0,1], \forall i < j, y_i, y_j, \\ \sum_{y_i} \mu_i(y_i) = 1, \forall i \in V, y_i, \\ \sum_{y_i} \mu_{ij}(y_i, y_j) = \mu_j(y_j), \forall i < j, y_j, \\ \sum_{y_j} \mu_{ij}(y_i, y_j) = \mu_i(y_i), \forall i < j, y_i, \\ \sum_{j \in V, j > i} z_{ij} + \sum_{j \in V, j < i} z_{ji} \leq h, \forall i \in V. \end{array} \right\}$$

3. LP Relaxation with Marginalization

In this section, we describe our new LP relaxation, which we denote by LP-M. Similar to LP-W, the LP-M also replaces the quadratic term $\mu_{ij}(y_i, y_j)z_{ij}$ in the objective function of problem (2) by the optimization variable $\lambda_{ij}(y_i, y_j)$. However, in contrast to LP-W, which explicitly specifies lower and upper bounds of $\lambda_{ij}(y_i, y_j)$ as linear functions of $\mu_{ij}(y_i, y_j)$ and z_{ij}, LP-M introduces a linear marginalization constraint. Specifically, since $\sum_{y_i, y_j} \mu_{ij}(y_i, y_j) = 1$ it follows that

$$\sum\nolimits_{y_i, y_j} \lambda_{ij}(y_i, y_j) = z_{ij}, \forall i < j. \tag{4}$$

In other words, by marginalizing $\lambda_{ij}(y_i, y_j)$ over all values of y_i and y_j, we recover the indicator variable for whether $(i, j) \in E$. While the above marginalization constraint specifies the relationship between $\lambda_{ij}(y_i, y_j)$ and z_{ij}, it does not depend on $\mu_{ij}(y_i, y_j)$. To address this problem, we exploit the fact that $z_{ij} <= 1$, which implies that

$$\lambda_{ij}(y_i, y_j) \leq \mu_{ij}(y_i, y_j), \forall i < j, y_i, y_j. \tag{5}$$

Substituting the upper and lower bounds of $\lambda_{ij}(y_i, y_j)$ in the LP-W relaxation with the above two linear constraints, the LP-M relaxation can be specified as follows:

$$\begin{aligned} \min \quad & \sum_{i \in V} \sum_{y_i} \mu_i(y_i)\theta_i(y_i) + \\ & \sum_{i,j \in V, i<j} \sum_{y_i, y_j} \theta_{ij}(y_i, y_j)\lambda_{ij}(y_i, y_j), \\ \text{s.t.} \quad & \sum_{y_i, y_j} \lambda_{ij}(y_i, y_j) = z_{ij}, \forall i < j, \\ & \lambda_{ij}(y_i, y_j) \leq \mu_{ij}(y_i, y_j), \forall i < j, y_i, y_j, \\ & \lambda_{ij}(y_i, y_j) \in [0,1], \forall i, j, i < j, y_i, y_j, \\ & \mu, z \in \mathcal{O}\,. \end{aligned} \tag{6}$$

LP-M is an intuitive extension of the standard LP relaxation for MRF inference with known graph structure [7]. However, its theoretical property (tightness over LP-W) is an interesting contribution of the paper.

3.1. Comparing the LP-W and LP-M Relaxations

Problem Size. We begin by comparing the two LP relaxations in terms of the problem size. We denote the number of nodes by n, and the cardinality of the label set $\mathcal{Y}$ by c. Note that both LP-W and LP-M contain the same number of optimization variables since both the relaxations substitute the quadratic terms $\mu_{ij}(y_i, y_j)z_{ij}$ by the variables $\lambda_{ij}(y_i, y_j)$. In terms of the number of constraints, the LP-M relaxation is smaller than the LP-W relaxation. This is due to the fact that the LP-W relaxation introduces two constraints to specify the lower bound of $\lambda_{ij}(z_i, z_j)$ (specifically, $\lambda_{ij}(y_i, y_j) \geq \max\{0, z_{ij} + \mu_{ij}(y_i, y_j) - 1\}$) and two constraints to specify the upper bound of $\lambda_{ij}(z_i, z_j)$ (specifically, $\lambda_{ij}(y_i, y_j) \leq \min\{z_{ij}, \mu_{ij}(y_i, y_j)\}$). In contrast, the LP-M relaxation introduces one constraint for the lower bound of $\lambda_{ij}(y_i, y_j)$ (specifically, $\lambda_{ij}(y_i, y_j) \geq 0$), one constraint for the upper bound of $\lambda_{ij}(y_i, y_j)$ (specifically, $\lambda_{ij}(y_i, y_j) \leq \mu_{ij}(y_i, y_j)$), and one marginalization constraint for the set of variables $\{\lambda_{ij}(y_i, y_j), y_i, y_j \in \mathcal{Y}\}$. Table 1 lists the exact sizes of different relaxations.

Tightness. We now compare LP-W and LP-M in terms of their tightness. Note that a relaxation A of a problem is said to be tighter than the relaxation B of the same problem if and only if the feasible region of A is a subset of the feasible region of B. The following proposition establishes the relationship between LP-W and LP-M.

Proposition 1 *The LP-M relaxation (6) is tighter than the LP-W relaxation (3).*

The proof of the above proposition is provided in the supplementary material. As will be seen in section 5, the theoretical advantage of LP-M over LP-W also translates to better performance in practice both in terms of the energy as well as the accuracy of human action recognition.

4. Quadratic Programming Relaxation

The relaxation in the previous section replaces the quadratic objective function of the original problem (2) by a linear objective function, and linear constraints on the new variables $\lambda_{ij}(y_i, y_j)$. While the resulting LP-M relaxation is tighter than LP-W, it still increases the feasible region of problem (2) substantially. In fact, one may argue that any convex relaxation of problem (1) will not be tight in general since the feasible region of problem (1) is highly non-convex. Inspired by this observation, we propose a non-convex QP relaxation of problem (2), which is obtained by replacing the integral constraints over the optimization variables by linear constraints that allow the variables to take (possibly fractional) values between 0 and 1. While the non-convex QP relaxation cannot be solved optimally in polynomial time, we show that its local minimum or saddle point solution can be obtained efficiently. Specifically, we show that the objective function of the non-convex QP relaxation can be viewed as a difference of two convex quadratic functions, which allows us to use the concave-convex procedure (CCCP). In order to initialize the CCCP algorithm, we

	Number of variables	Number of constraints
LP-W [24]	$\frac{n(n-1)(2c^2+1)}{2} + nc$	$(cn^2+n) + (n^2-n)(3c^2+1)$
LP-M	$\frac{n(n-1)(2c^2+1)}{2} + nc$	$(cn^2+n) + (n^2-n)(3c^2+1) - \frac{n(n-1)(c^2-1)}{2}$
convex QP	$\frac{n(n-1)(c^2+1)}{2} + nc + \underbrace{n(c^2-c+1)}_{\text{removable}}$	$(cn^2+n) + (n^2-n)(3c^2+1) - 2n(n-1)c^2$

Table 1: Scales of three different relaxations. Here n and c denote the number of nodes and the cardinality of label set respectively.

propose an optimal convexification of the non-convex QP. While the convex QP relaxation is not as tight as the LP-W and LP-M relaxations, when used in conjunction with the CCCP algorithm, it provides accurate solutions.

We begin our description with the non-convex QP relaxation. In subsection 4.2, we present its convexification, which modifies the diagonal of the matrix that specifies the objective function of the non-convex QP. Unlike the convex QP relaxation for fixed graphs [19], we show that our relaxation is optimal in the sense that it minimizes the ℓ_1 norm of the diagonal modification vector. In subsection 4.3, we provide a comparison of the convex QP, LP-W and LP-M in terms of the problem size and the tightness of the relaxation. Finally, in subsection 4.4, we show how the non-convex QP can be optimized efficiently using a CCCP algorithm to obtain an accurate approximate solution to problem (2).

4.1. Non-Convex QP Relaxation

The following notation will be helpful in describing the non-convex QP relaxation. The vector $\boldsymbol{\theta}_{ij} = [\theta_{ij}(y_i, y_j)]_{y_i, y_j \in \mathcal{Y}}$ is formed by enumerating the pairwise potentials between y_i and y_j over all possible labellings in turn. Here, $\theta_{i,i}(a,b) = 0\ \forall a, b \in \mathcal{Y}, i \in V$. The set of all pairwise potentials are defined using the matrix $\hat{\Theta}$:

$$\hat{\Theta} = \begin{bmatrix} 0 & \frac{1}{2}\Theta \\ \frac{1}{2}\Theta^\top & 0 \end{bmatrix}, \Theta = \begin{bmatrix} \boldsymbol{\theta}_{11} & 0 & \dots & 0 & 0 \\ 0 & \boldsymbol{\theta}_{12} & & 0 & 0 \\ & 0 & & \vdots & \vdots \\ & \vdots & & & \\ & & & \boldsymbol{\theta}_{n-1n} & 0 \\ & & & 0 & \boldsymbol{\theta}_{nn} \end{bmatrix}. \tag{7}$$

A less compact, but more detailed exposition of Θ can be found in the supplementary material. In order to specify the node potentials in the QP relaxation, we define a vector $\boldsymbol{\vartheta} = [\hat{\vartheta}_{ij}(y_i, y_j)]_{i \le j, y_i, y_j \in \mathcal{Y}}$ where $\vartheta_{ij}(y_i, y_j) = \theta_i(y_i)\ \forall i = j\ \&\ y_i = y_j$, and $\vartheta_{ij}(y_i, y_j) = 0$ otherwise. Furthermore, we define $\hat{\boldsymbol{\theta}} = [\boldsymbol{\vartheta}, \mathbf{0}]$, where $\mathbf{0}$ is a zero vector has $n(n+1)/2$ dimensions. The vector $\hat{\boldsymbol{\theta}}$ and the matrix $\hat{\Theta}$ represent the node and the pairwise potentials of the given MRF, that is, the input of the problem. The output of the problem, that is, the optimization variables, are denoted by the vectors $\mathbf{z} = [z_{ij}]_{i \le j}$ and $\boldsymbol{\mu} = [\mu_{ij}(y_i, y_j)]_{i \le j, y_i, y_j \in \mathcal{Y}}$, where $\mu_{ii}(y_i, y_i) = \mu_i(y_i)$. The set of all the optimization variables is denoted by $\chi = [\boldsymbol{\mu}, \mathbf{z}]$.

Using the above notation, the energy corresponding to the variables $\mathbf{z}$ and $\boldsymbol{\mu}$ can be concisely written as

$$\sum_{i \in V} \sum_{y_i} \mu_i(y_i) \theta_i(y_i) + \sum_{i<j} \sum_{y_i, y_j} \mu_{ij}(y_i, y_j) \theta_{ij}(y_i, y_j) z_{ij}$$
$$= \chi^\top \hat{\boldsymbol{\theta}} + \chi^\top \hat{\Theta} \chi. \tag{8}$$

The non-convex QP relaxation of problem (2) can therefore be specified in terms of the optimization variables χ as

$$\min_{\chi}\ \chi^\top \hat{\boldsymbol{\theta}} + \chi^\top \hat{\Theta} \chi,\ \text{ s.t. } \chi \in \mathcal{O}\,. \tag{9}$$

Note that, since the QP is obtained by relaxing the domain of z, μ from $\{0, 1\}$ to $[0, 1]$ only (without changing the objective function of the original problem (2), it is tighter than both LP-W and LP-M. The main disadvantage of problem (9) is that it is non-convex in general. However, as will be seen shortly, its local maximum or saddle point solution can be obtained efficiently using the CCCP algorithm. Before describing the CCCP algorithm in detail, it would be helpful to discuss the convexification of problem (9), which is used to initialize the CCCP algorithm.

Note we are not the first to formulate MRF inference as QP problems, see [19, 9]. However, we do provide the first QP formulation for MRF inference with unknown graphs.

4.2. Convex Approximation

We now present a convex approximation of problem (9), which is inspired by the convex relaxation of MAP inference for fixed graphs [19]. The following notation will be useful in describing the convex approximation. The number of rows (and columns) of the square matrix $\hat{\Theta}$ are denoted by N. The matrix $\mathbf{diag}(\mathbf{d})$ is a diagonal matrix, whose diagonal elements form the vector $\mathbf{d} \in \mathbb{R}^N$. The i-th element of the optimization vector χ is denoted by x_i.

For any vector $\mathbf{d} \in \mathbb{R}^N$, we can rewrite the objective of problem (9) as

$$\chi^\top \hat{\boldsymbol{\theta}} + \chi^\top \hat{\Theta} \chi =$$
$$\chi^\top (\hat{\boldsymbol{\theta}} - \mathbf{d}) + \chi^\top (\hat{\Theta} + \mathbf{diag}(\mathbf{d})) \chi + g(\mathbf{d}, \chi), \tag{10}$$

$$g(\mathbf{d}, \chi) = \chi^\top \mathbf{d} - \chi^\top \mathbf{diag}(\mathbf{d}) \chi = \sum_{i=1}^{N} d_i (x_i - x_i^2). \tag{11}$$

Clearly for $x_i \in \{0,1\}$, $g(\mathbf{d},\chi) = 0$. For $x_i \in (0,1)$, if $d_i > 0$, $g(\mathbf{d},\chi) > 0$. When $\hat{\Theta} + \mathbf{diag}(\mathbf{d}) \succeq 0$, $\chi^\top(\hat{\boldsymbol{\theta}} - \mathbf{d}) + \chi^\top(\hat{\Theta} + \mathbf{diag}(\mathbf{d}))\chi$ is a convex approximation of $\chi^\top\hat{\boldsymbol{\theta}} + \chi^\top\hat{\Theta}\chi$, with approximation gap $|g(\mathbf{d},\chi)|$. For a fixed χ, minimizing the approximation gap leads to

$$\min_{\mathbf{d}} |g(\mathbf{d},\chi)|, \ \text{ s.t. } \hat{\Theta} + \mathbf{diag}(\mathbf{d}) \succeq 0.$$

Since we do not know χ, we seek a vector $\mathbf{d}$ works well for all χ, *i.e.* minimizing $\mathbb{E}[|g(\mathbf{d},\chi)|]$. Assuming uniform prior of χ gives rise to,

$$\min_{\mathbf{d}} \|\, \mathbf{d} \,\|_1, \ \text{ s.t. } \hat{\Theta} + \mathbf{diag}(\mathbf{d}) \succeq 0, \qquad (12)$$

which means we seek a diagonal modification vector $\mathbf{d}$ such that the ℓ_1 norm of the vector is minimum.

Proposition 2 *The solution of problem (12) is*

$$\mathbf{d}^* = [d_k^*]_{k=1}^N, \textit{ where } d_k^* = \sum\nolimits_{j=1}^N |\hat{\Theta}_{k,j}|. \qquad (13)$$

The proof of the above proposition is provided in the supplementary material. We would like to point out that this result does not carry over to general QP problems with arbitrary quadratic matrices including the QP relaxations for MAP inference with known graph structures as in [14, 19]. However, for our problem, the above proposition provides a strong theoretical justification for approximation the non-convex QP (9) as follows:

$$\min_{\chi} \ \chi^\top \mathbf{q} + \chi^\top \mathbf{Q}\,\chi \ \text{ s.t. } \chi \in \mathcal{O}, \qquad (14)$$

where $\mathbf{Q} = \hat{\Theta} + \mathbf{diag}(\mathbf{d}^*)$, $\mathbf{q} = \hat{\boldsymbol{\theta}} - \mathbf{d}^*$.

4.3. Comparing the QP and LP Relaxations

Problem Size. We begin by comparing the relaxations in terms of the problem size. Note that, unlike the two LP relaxations, the convex QP relaxation does not introduce any additional variables $\lambda_{ij}(y_i, y_j)$. Furthermore, the variables $\mu_{ij}(y_i, y_j)\ \forall i = j, y_i \neq y_j$ and $z_{ii}\ \forall i \in V$ do not play a role in the objective function of the convex QP, and can therefore be removed. This implies that the convex QP relaxation contains significantly fewer variables than the two LP relaxations. It also contains significantly fewer constraints, namely those specified by the set $\mathcal{O}$, which is a subset of the constraints used in the two LP relaxations (see Table 1 for the exact numbers of problem sizes).

Tightness. The following proposition establishes the relationship between the convex QP relaxation and the two LP relaxations in terms of tightness.

Proposition 3 *The LP-W and LP-M relaxations are tighter than the convex QP relaxation (14).*

The proof of the above proposition is provided in the supplementary material. As will be seen in section 5, the theoretical advantage of the LP-M and LP-W relaxations over the convex QP relaxation translates to better performance in practice. However, the convex QP provides a natural way to initialize the approximate algorithm for the non-convex QP relaxation, which is described in the Section 4.4.

4.4. CCCP Algorithm for Non-Convex QP

The non-convex QP relaxation (9) can be formulated as a difference-of-convex program as follows:

$$\operatorname{argmin}_{\chi} \ F(\chi) \ \text{ s.t. } \chi \in \mathcal{O}, \qquad (15)$$

$$F(\chi) = \underbrace{\chi^\top\hat{\boldsymbol{\theta}} + \chi^\top\,\mathbf{Q}\,\chi}_{F_{vex}(\chi)} \ \underbrace{-\chi^\top\,\mathbf{diag}(\mathbf{d}^*)\chi}_{F_{cave}(\chi)}. \qquad (16)$$

It is easy to see that F_{vex}, F_{cave} are convex and concave functions of χ as $\mathbf{Q}$ and $-\,\mathbf{diag}(\mathbf{d}^*)$ are positive semidefinite and negative semidefinite respectively. The above observation allows us to obtain a local minimum or saddle point solution of problem (15) using the CCCP algorithm [28]. Starting with an initial solution $\chi^{(0)}$, the CCCP algorithm iteratively decreases the objective value of problem (15) by finding a solution $\chi^{(t+1)}$ using the current solution $\chi^{(t)}$ such that it satisfies the following condition:

$$\nabla F_{vex}(\chi^{(t+1)}) = -\nabla F_{cave}(\chi^{(t)}). \qquad (17)$$

The solution $\chi^{(t+1)}$ satisfying the above condition can be found by minimizing $F_{vex}(\chi) + \chi^\top\nabla F_{cave}(\chi^{(t)})$, that is,

$$\begin{aligned} &\min_{\chi} \ \chi^\top(\hat{\boldsymbol{\theta}} - 2\,\mathbf{diag}(\mathbf{d}^*)\chi^{(t)}) + \chi^\top\,\mathbf{Q}\,\chi \\ &\text{s.t. } \chi \in \mathcal{O}\,. \end{aligned} \qquad (18)$$

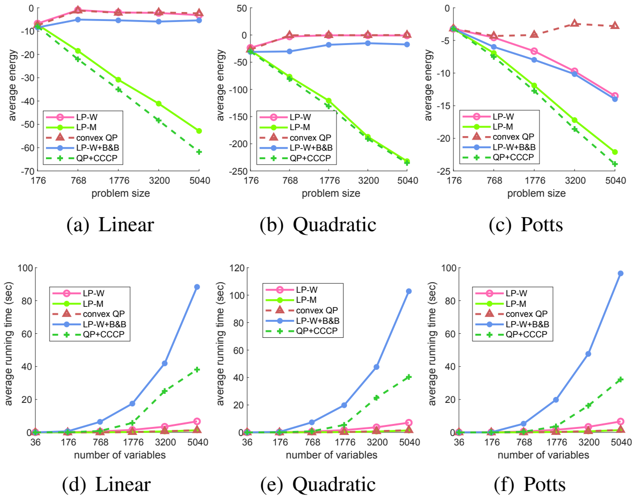
Note (18) is a convex QP which can be solved by any off-the-shelf QP solvers such as Mosek [1].

The CCCP algorithm is guaranteed to converge for any feasible initialization. We refer the reader to Theorem 10 in [22] for a proof. In our experiments, we use the solution of the convex QP (14) to initialize the CCCP algorithm. Alternatively, one can initialize by random feasible points or solutions of the LP relaxations. However, no matter what initialization is used, we still need to solve a convex QP (18) iteratively, which is based on our QP formulation (9).

QP+CCCP Algorithm. We name the above CCCP-style update procedure the QP+CCCP algorithm and its pseudocode is provided in the supplementary material.

5. Experiments

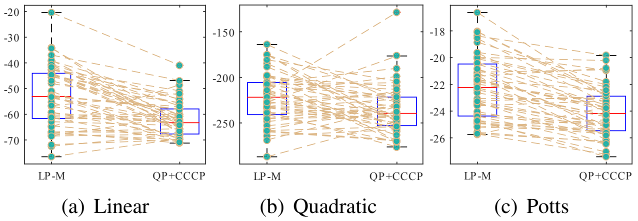
In this section we first evaluate different inference algorithms on synthetic data. In each case we report both the



(a) Linear (b) Quadratic (c) Potts

(d) Linear (e) Quadratic (f) Potts

Figure 2: Comparisons of energy (the objective in (1), top row) an run time (bottom) using three types of synthetic data (Linear, Potts, Quadratic). Our LP-M relaxation and QP+CCCP algorithm perform increasingly better than other methods as the problem size goes larger. For time-consuming, LP-M is the fastest and QP is marginally worse than LP-M, see the text for explanation.



(a) Linear (b) Quadratic (c) Potts

Figure 3: Paired-sample t-test on synthetic data. Estimated energies for each paired samples are connected by dashed lines. Here p equals 0.98×10^{-7}, 0.96×10^{-2}, 0.4×10^{-9} respectively for linear, quadratic and Potts results. Since $p < 0.05$ for all tests the null hypothesis at the 5% significance level is rejected.

final value of the potential function (equation (1)) and the running time. We then show how to apply the inference technique to the human activity recognition task. Additional results are provided in the supplementary material.

To solve the inference problem (2), we use: 1) LP-W, solving the problem (3) proposed in [24]; 2) LP-W+B&B–the branch and bound algorithm proposed in [24], where the bounds are computed by solving LP-W problems; 3) LP-M; 4) QP–the convex QP; 5) QP+CCCP. The number of iterations for LP-W+B&B and QP+CCCP is 50. One may argue that the original optimization (2) can be exactly solved using off-the-shelf integer quadratic programming (IQP) solvers such as CPLEX MIQP. However, even for optimizations with 6 nodes and 5 classes, based on our tests IQP takes around 2 minutes compared to 0.1 seconds for QP+CCCP, while the obtained objectives of them are quite close, which makes the use of IQP solvers undesirable.

Synthetic Data. We generate synthetic data using a method similar to that used in [19]. The node potential $\theta_i(y_i) \sim \mathcal{U}(-1, 1)$, while the edge potentials are created as $\theta_{ij}(y_i, y_j) = s \times \text{dis}(y_i, y_j)$. Here the couple strength $s \sim \mathcal{U}(-\eta, \eta)$, and $\text{dis}(y_i, y_j)$ is one of three types of distance functions including linear: $\text{dis}(y_i, y_j) = |y_i - y_j|$, quadratic: $\text{dis}(y_i, y_j) = (y_i - y_j)^2$, Potts: $\text{dis}(y_i, y_j) = \mathbb{1}(y_i = y_j)$. Here $\eta = 1$, the graph degree parameter $h = 2$. More results using different η and h are provided in the supplementary material.

Two comparisons are made here. First, we compare the estimated potential, *i.e.* the value of (1). We report results for problems with a range of different sizes. For a problem with n nodes and c classes (assuming the cardinalities of the spaces of all random variables are the same), the size is calculated by $\frac{n(n-1)}{2}(1 + c^2) + nc$. Here $n = \{4, \cdots, 20\}$ and $c = 5$. For each problem size, twenty examples are generated and we report the mean potential of these examples, see Figure 2. Tighter relaxations translate to better performance: LP-M performs better than LP-W which performs slightly better than (initial convex) QP. Overall, QP+CCCP performs the best, due to its iterative approximation to the non-convex QP, which is the tightest among all relaxations here. We observe that QP+CCCP converges within 50 iterations for all data here, but LP-W+B&B does not converge within 50 iterations in most cases, which leads to inferior results. Statistically, the differences between our QP+CCCP method and LP-M are significant since the null hypothesis is usually rejected at the 0.05 level in the standard paired t-tests (ttest in Matlab), see Figure 3. In other words, our QP+CCCP method is more accurate. Second, we compare the running time of different methods on problems with a size of 5,040. For each distance function 100 examples are tested and the average running time is reported in Figure 2. Overall LP-W+B&B and QP+CCCP are much slower than other methods as expected. The fastest is LP-M. Although both QP and LPs are solved using the interior point method, the interior point method for QP is computationally more expensive than that for LP. This might be because: 1) both problems translate to solving linear systems, essentially; 2) however, the coefficient matrix (see equation (16.58) in [17]) in the system corresponding to QP is more costly to factor than that of the LPs (see (14.9) in [17]), because of the hessian matrix of QP. We refer the reader to Page 482 of [17] for details.

Predict Human Group Activities. We now consider human activity recognition on the CAD dataset [3], which is a benchmark for this task. For clarity, the term *activity* is

	Cross	Wait	Queue	Walk	Talk	Precision	Recall	F1-score	Time (second)
tree-structured	45.0	47.2	95.3	65.2	96.1	71.6	69.8	70.7	$\mathbf{1.0 \times 10^{-2}}$
Lan [15]	55.9	59.7	94.6	62.2	99.5	73.3	74.4	73.8	6.0×10^{-2}
LP-W [24]	60.7	60.4	93.6	47.3	99.5	72.6	72.3	73.6	4.1×10^{-2}
LP-W+B&B [24]	55.9	61.8	95.7	55.4	99.5	73.6	73.7	73.6	4.0×10^{-1}
QP (ours)	55.9	61.8	92.5	48.7	99.5	71.8	71.7	71.7	3.5×10^{-2}
LP-M (ours)	60.7	59.7	93.6	56.8	99.5	73.9	74.0	73.9	3.0×10^{-2}
QP+CCCP (ours)	62.1	61.1	95.7	55.4	98.9	**74.3**	**74.6**	**74.4**	2.0×10^{-1}

Table 2: Group activity recognition performance on the CAD dataset. Accuracies for different action-classes span from column two to six from left to right. Column seven to nine reports overall precision, recall and F1-score for all classes. QP+CCCP is the best except for time.

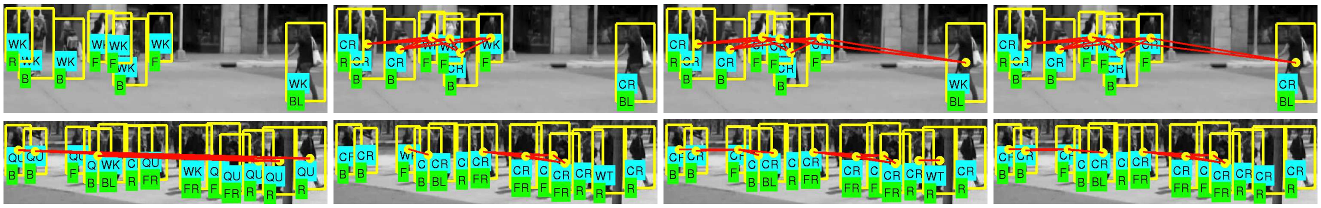


Figure 4: Visualisation of recognition results by LP-W+B&B, Lan, LP-M and QP+CCCP (from left to right), using two examples (each row corresponds to one example) in CAD. The predicted action and pose labels are shown in cyan and green boxes. The red edges represent the learned graph structures within the action layer. For action names, *CR, WK, QU, WT* indicate *cross, walk, queue* and *wait*. For poses, *B, L, R, F, BL, BR, FR, FL* denote *back, left, right, front, back-left, back-right, front-right* and *front-left* respectively. Note our approaches (two rightmost columns) can predict meaningful long-range connections between targets, which helps to predict consistent action labels for different people within the same group.

used to describe the behavior of a group of people, while the term *action* refers to the behavior of an individual. CAD contains 44 videos and 5 action classes: *cross*, *wait*, *queue*, *walk* and *talk*. In each image most people perform the same action. Like [15], the activity label for an image is defined as the dominant action performed by these people. Our aim is to assign each testing image an activity label. The problem is modeled by MRFs in a manner similar to that used in [15] [1]. Specifically, the MRF has two layers. The first layer is the activity layer that contains one node representing the activity variable. The second layer, the action layer, contains a number of nodes representing the action variables corresponding to different people. During both training and prediction the dependency among action variables, thus the graph structure in the action layer will be estimated together with the activity and action variables. The interaction between both layers is fixed by connecting each action node to the activity node. The potentials used here are similar to those used in [15] with differences including: 1) rather than predicting human body poses jointly with actions, we use fixed human body poses during training and testing, which reduces the computational burden significantly. To estimate poses, we train a multi-class SVM classifier based on HoG features [4] extracted from human body areas; 2) to obtain the image features we extract HoG features from the whole image, as compared to taking an average of HoG features extracted from all human body areas as in [15]. Details of the potential function can be found in [15].

The inference problem here is estimating the best activity, action labels and graph structure in the action layer. Since there is only one activity variable, we exhaustively search all possible activities. For each fixed activity, the problem reduces to finding the best actions and graph structure in the action layer, which is solved via different methods. For training, we use the method employed in [15].

For comparison we also consider MRFs with known graphs. In particular, we have used minimum spanning trees that maintain the smallest Euclidean distance over all body detections. In such cases the inference problems can be exactly solved via belief propagation, and the model parameters are learned by using structured SVM [23]. Results are reported in Table 2. Clearly the methods which estimate the graph structure outperform those using fixed tree-structured graphs significantly. Among all methods that estimate the graphs, QP+CCCP performs best in terms of precision, recall and F1-score. LP-M performs second best in terms of precision and F1-score, and performs best in terms of speed. We visualize some recognition results in Figure 4 using LP-W+B&B, Lan, LP-M and QP+CCCP, from which one can observe our new relaxations yield competitive results in terms of action classification, consequently better activity recognition results are achieved.

[1] Feature extraction and Lan method are implemented by ourselves.

	No-Int	Handshake	Highfive	Hug	Kiss	Precision	Recall	F1-score	Time (second)
tree-structured	20.2	51.1	61.3	58.2	46.4	48.4	47.4	47.9	$\mathbf{1.0 \times 10^{-3}}$
two stream Net [21]	54.7	38.4	35.4	54.8	58.6	49.3	48.4	48.8	–
Lan [15]	11.3	52.1	58.4	55.0	67.2	49.5	48.8	49.1	5.0×10^{-3}
LP-W [24]	59.0	49.0	49.2	65.2	71.5	59.5	58.8	59.1	5.0×10^{-3}
LP-W+B&B [24]	49.1	56.3	63.0	70.0	69.2	60.7	**61.5**	61.1	5.6×10^{-2}
QP (ours)	65.3	40.8	54.8	62.0	66.8	60.8	58.0	59.4	4.1×10^{-3}
LP-M (ours)	57.5	50.7	48.7	66.8	74.7	60.6	59.7	60.1	3.2×10^{-3}
QP+CCCP (ours)	61.4	60.8	45.2	69.0	71.0	**62.9**	**61.5**	**62.2**	1.5×10^{-2}

Table 3: Person-wise instantaneous interaction recognition results (%) on TVHI dataset. Here "No-Int" means no-interaction. Our QP+CCCP performs best in terms of overall precision, recall and F1-score.

	No-Int	BX	HS	HF	HG	KK	BD	PT	PS	Precision	Recall	F1-score	Time (second)
ResNet [8]	95.7	5.8	53.6	15.5	66.9	76.4	59.6	27.5	36.5	60.3	48.6	53.8	–
LP-W+B&B [24]	95.5	5.6	53.6	15.4	69.0	75.3	62.0	28.7	36.5	60.4	49.1	54.2	3.0×10^{-1}
LP-M (ours)	95.6	5.8	53.2	15.2	69.8	77.0	62.2	28.5	36.5	**60.5**	49.3	54.3	$\mathbf{6.9 \times 10^{-3}}$
QP+CCCP (ours)	95.7	6.0	53.6	15.5	69.9	76.7	62.4	28.5	36.7	**60.5**	**49.4**	**54.4**	3.0×10^{-2}

Table 4: Person-wise instantaneous interaction recognition results (%) on BIT dataset. Here "No-Int, BX, HS, HF, HG, KK, BD, PT, PS" means no-interaction, box, handshake, highfive, hug, kick, bend, pat, push respectively.

Estimated Energy. We compute the mean of the estimated energies using the entire CAD-testing set. For fair comparison we use the same potential functions (θ) across different inference algorithms. The results for Lan, LP-M and QP+CCCP, are $-9.62, -10.13, -10.28$ (lower is better) respectively, which indicates that the proposed solvers (both LP-M and QP+CCCP) are more accurate.

Predict Instantaneous Human Interactions. Here the task is to predict an interaction label for each person in each frame, which is much more challenging than video-wise action recognition predicts each video an interaction category. We use two datasets. The first is television human interaction (TVHI) dataset introduced in [18]. Each video in TVHI contains at least two persons who either are interacting (*handshake, highfive, hug or kiss*) with each other or simply have no interactions. The second is BIT dataset [13] which consists of nine classes of human interactions, *i.e.* box, handshake, highfive, hug, kick, bend, pat, push and others (*i.e.* interactions beyond the first eight classes). Each class contains 50 videos with cluttered backgrounds. The train-test splits for both dataset follow the suggestions of their authors. For TVHI, since MRFs with unknown graphs have been used to model the human interactions [24], for fair comparison we use the same MRF representation and experimental settings as [24], while the estimation of labels and graph structures is solved using the methods proposed in this work. For BIT, we use the same MRF model as TVHI, but extract features using ResNet [8]. Here we set $h = 1$. Recognition results are presented in Table 3 and Table 4, for TVHI and BIT respectively. Clearly the proposed QP+CCCP performs best. Overall LP-M is only second to QP+CCCP, and is occasionally worse than LP+B&B (see Table 3). However, it is the best in terms of running time among all methods learning MRF-structures. In addition, we experiment MRF with known fully connected graphs (with inference solved via tree-reweighted message passing [11]) on TVHI. The resulting precision, recall and F1-score are 56.5, 54.8 and 55.6 respectively, which are much worse then our best results. Hence it is beneficial to infer MRF graphs for human interaction recognition compared against using fixed known graphs.

6. Conclusion

We presented two relaxations to the problem of jointly estimating the labels and the structure of Markov random field, a new LP relaxation, and a non-convex QP relaxation both tighter than the existing relaxation. The non-convex QP can be efficiently solved by using CCCP by solving a number of convex QP problems. We show that our convex QP is optimal in some sense. Experimental results on both synthetic data and human activity recognition tasks demonstrate that our QP in conjunction with CCCP performs best in terms of accuracy and objective value. The proposed new LP relaxation performs second best in terms of accuracy and objective value, and best in terms of running time.

Acknowledgement. We thank Anton van den Hengel and Chunhua Shen for their suggestions on the exposition of this paper. This work is partially supported by National Natural Science Foundation of China (61802348, 61876167 and U1509207), National Key R&D Program of China (2018YFB1305200), and ARC discovery grant (DP160100703).

References

[1] Erling Andersen and Knud Andersen. Mosek (version 8). Academic version available at www.mosek.com, 2019.

[2] Manmohan Chandraker and David Kriegman. Globally optimal bilinear programming for computer vision applications. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[3] Wongun Choi, Khuram Shahid, and Silvio Savarese. What are they doing?: Collective activity classification using spatio-temporal relationship among people. In *International Conference on Computer Vision Visual Surveillance Workshops*, 2009.

[4] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[5] James Diebel and Sebastian Thrun. An application of markov random fields to range sensing. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2005.

[6] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[7] Amir Globerson and Tommi Jaakkola. Fixing max-product: Convergent message passing algorithms for MAP LP-relaxations. *Conference on Neural Information Processing Systems (NeurIPS)*, 2007.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[9] Jeremy Jancsary, Sebastian Nowozin, and Carsten Rother. Learning convex QP relaxations for structured prediction. In *International Conference on Machine Learning (ICML)*, 2013.

[10] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[11] Vladimir Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1568–1583, 2006.

[12] Vladlen Koltun. Efficient inference in fully connected CRFs with gaussian edge potentials. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2011.

[13] Yu Kong, Yunde Jia, and Yun Fu. Learning human interaction by interactive phrases. In *European Conference on Computer Vision (ECCV)*, 2012.

[14] M. Pawan Kumar, Vladimir Kolmogorov, and Philip H. S. Torr. An analysis of convex relaxations for map estimation of discrete MRFs. *Journal of Machine Learning Research*, 10:71–106, 2009.

[15] Tian Lan, Yang Wang, and Grey Mori. Beyond actions: Discriminative models for contextual group activities. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2010.

[16] Evgeny Levinkov, Jonas Uhrig, Siyu Tang, Mohamed Omran, Eldar Insafutdinov, Alexander Kirillov, Carsten Rother, Thomas Brox, Bernt Schiele, and Bjoern Andres. Joint graph decomposition and node labeling by local search. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[17] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization (Second Edition)*. Springer Press, 2006.

[18] Alonso Patron-Perez, Marcin Marszalek, Andrew Zisserman, and Ian Reid. High Five: Recognising human interactions in TV shows. In *British Machine Vision Conference (BMVC)*, 2010.

[19] Pradeep Ravikumar and John Lafferty. Quadratic programming relaxations for metric labeling and markov random field map estimation. In *International Conference on Machine Learning (ICML)*, 2006.

[20] Ishant Shanu, Chetan Arora, and S. N. Maheshwari. Inference in higher order MRF-MAP problems with small and large cliques. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[21] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2014.

[22] Bharath K. Sriperumbudur and Gert R. G. Lanckriet. On the convergence of the concave-convex procedure. In *Conference on Neural Information Processing Systems (NeurIPS)*, volume 9, pages 1759–1767, 2009.

[23] Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6(2):1453–1484, 2006.

[24] Zhenhua Wang, Qinfeng Shi, Chunhua Shen, and Anton van den Hengel. Bilinear programming for human activity recognition with unknown mrf graphs. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[25] Rui Yao, Guosheng Lin, Qinfeng Shi, and Damith C. Ranasinghe. Efficient dense labelling of human activity sequences from wearables using fully convolutional networks. *Pattern Recognition*, 78:252–266, 2018.

[26] Rui Yao, Shixiong Xia, Zhen Zhang, and Yanning Zhang. Real-time correlation filter tracking by efficient dense belief propagation with structure preserving. *IEEE Transactions on Multimedia*, 19(4):772–784, 2017.

[27] Chun-Nam John Yu and Thorsten Joachims. Learning structural SVMs with latent variables. In *International Conference on Machine Learning (ICML)*, 2009.

[28] A. L. Yuille and Anand Rangarajan. The concave-convex procedure (CCCP). In *Conference on Neural Information Processing Systems (NeurIPS)*, 2002.