Bilinear Programming for Human Activity Recognition with Unknown MRF Graphs

Zhenhua Wang, Qinfeng Shi, Chunhua Shen and Anton van den Hengel
School of Computer Science, The University of Adelaide, Australia
{zhenhua.wang01,javen.shi,chunhua.shen,anton.vandenhengel}@adelaide.edu.au

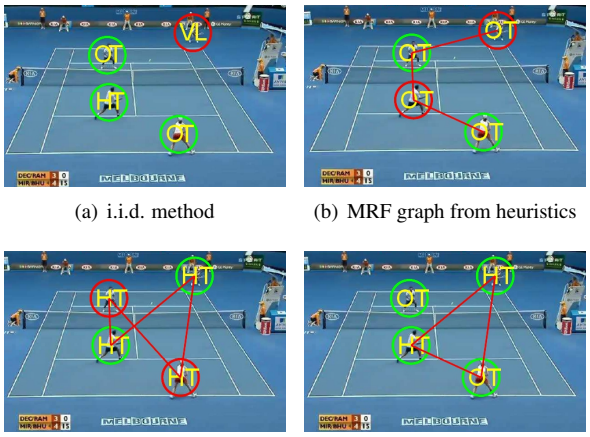
Abstract

Markov Random Fields (MRFs) have been successfully applied to human activity modelling, largely due to their ability to model complex dependencies and deal with local uncertainty. However, the underlying graph structure is often manually specified, or automatically constructed by heuristics. We show, instead, that learning an MRF graph and performing MAP inference can be achieved simultaneously by solving a bilinear program. Equipped with the bilinear program based MAP inference for an unknown graph, we show how to estimate parameters efficiently and effectively with a latent structural SVM. We apply our techniques to predict sport moves (such as serve*,* volley *in tennis) and human activity in TV episodes (such as* kiss*,* hug *and* Hi-Five*). Experimental results show the proposed method outperforms the state-of-the-art.*

1. Introduction

Human activity recognition (HAR) is an important part of many applications such as video surveillance, key event detection, patient monitoring systems, transportation control, and scene understanding. HAR involves several subfields in computer vision and pattern recognition, including feature extraction and representation, human body detection and tracking, and has attracted much research attention in recent years. A suite of methods have been proposed for this task and promising recognition rates have been achieved. For example, in [11], more than 91% of actions can be correctly classified for the KTH dataset [15]. A detailed review of different methods for HAR can be found in [1]. In this work, we focus on recognizing individual activities in videos which contain multiple persons who may interact with each other.

If one assumes that the activity of each person is an independent and identically distributed (i.i.d.) random variable from an unknown but fixed underlying distribution, one can perform activity recognition by training a multiclass classifer, or a group of binary classifiers, based on descriptors such as HoG, HoF [6] or STIP [10] extracted from body centred areas, as in [15], [12], [11]. We label these methods as i.i.d. as they classify the action of each person separately. These methods are thus very efficient, but they may not be effective as the i.i.d. assumption is often not true in practice. For example in a badminton game, a player smashing suggests that his opponent is more likely to perform bending and lobbing instead of serving.



(a) i.i.d. method (b) MRF graph from heuristics

(c) MRF, Lan's method (d) MRF, our method

Figure 1. Activity recognition within a tennis match: (a) an i.i.d. method using multiclass SVM; (b) an MRF with a graph built using heuristics; (c) an MRF using Lan's method[9]; (d) an MRF with a graph learnt by our bilinear method. All algorithms select from four moves: *normal hit* (HT), *serve* (SV), *volley* (VL) and *others* (OT). Green nodes denote successful labellings, and red nodes failure.

Unlike traditional i.i.d methods, Markov Random Fields (MRFs) [8] are able to model complex dependencies and deal with local uncertainty in a consistent and principled manner. A possible MRF is shown in Figure 1 (d), where nodes represent the activity random variables, and the edges reflect the dependencies. The reason that MRFs can achieve a superior result to i.i.d. methods is because MRFs obtain the joint optimum over all variables whereas the i.i.d.

1063-6919/13 $26.00 © 2013 IEEE
DOI 10.1109/CVPR.2013.221

methods only seek optima for each variable separately. In [5], MRFs are used to model activities like walking, queueing, crossing *etc*. The MRFs' graphs are built simply based on heuristics such as two person's relative position. These heuristics may not be always reliable. For example, persons waving to each other can be far away from each other. One may deploy other heuristics, but they typically require specific domain knowledge, and apply only to a narrow range of activities in a particular environment. Instead of using heuristics, Lan *et al*. try to learn graphs based on the potential functions of the MRFs [9]. They seek the graph and the activity labels that give the highest overall potential function value (*i.e*. the smallest energy). The joint optimisation is very different from a typical inference problem where only labels are to be predicted. Thus they solve the joint optimisation problem approximately using a coordinate ascent method by holding the graph fixed and updating the labels, then holding labels fixed and updating the graph. Though predicting each person's activities is only a by-product — predicting group activities is their task — their idea is generally applicable to HAR. A key problem with this method is that coordinate ascent only finds local optima. Since the graphs of MRFs encode the dependencies, constructing reliable graphs is crucial. A MRF with an incorrect graph (such as Figure 1 (b) (c)) may produce inferior results to an i.i.d. method (in Figure 1 (a)), whereas a MRF with a correct graph (in Figure 1 (d)) produces a superior result, at least for this example.

In this paper, we show Maximum A Posteriori (MAP) inference for the activity labels and estimating the MRF graph structure can be carried out simultaneously as a joint optimisation and that achieving the global optimum is guaranteed. We formulate the joint optimisation problem as a bilinear program. Our bilinear formulation is inspired by recent work in Linear Program (LP) relaxation based MAP inference with known graphs in [7] and [16], and belief propagation (BP) based MAP inference with unknown graphs [9]. We show how to solve the bilinear program efficiently via the branch and bound algorithm which is guaranteed to achieve a global optimum. We then apply this novel inference technique to HAR with an unknown graph. Our experimental results on synthetic and real data, including tennis, badminton and TV episodes, show the capability of our method.

The reminder of the paper is organized as follows. We first introduce our task and the model representation in Section 2.1 followed by the MAP problem and its LP relaxation in Section 2.2. Then we give our bilinear formulation for the MAP inference with unknown graphs in Section 3.1. In Section 3.2 we show how to relax the bilinear program to an LP. In Section 3.3 we describe how to solve the bilinear program. In Section 4 we show how to train the model (*i.e*. parameter estimation) for the MRFs. Section 5 provides the experimental results followed by conclusions in Section 6.

2. Modelling HAR with unknown MRF graph

2.1. The Model

In HAR our goal is to estimate the activities of m persons $Y = (y_1, y_2, \cdots, y_m) \in \mathcal{Y}$, given an observation image $X \in \mathcal{X}$. To model the dependencies between activities, we use a MRF with the graph $G = (\mathcal{V}, \mathcal{E})$ where the vertex set $\mathcal{V} = \{1, 2, \cdots, m\}$ and the edge set $\mathcal{E}$ is yet to be determined. We cast the estimation problem as finding a discriminative function $F(X, Y, G)$ such that for an image X, we assign the activities Y and the graph G that exhibit the best score w.r.t. F,

$$(Y^*, G^*) = \underset{Y \in \mathcal{Y}, G \in \mathcal{G}}{\operatorname{argmax}} F(X, Y, G). \tag{1}$$

As in many learning methods, we consider functions linear in some feature representation Ψ,

$$F(X, Y, G; \mathbf{w}) = \mathbf{w}^\top \Psi(X, Y, G). \tag{2}$$

Here we consider the feature map $\Psi(X, Y, G)$ below,

$$\Psi(X, Y, G) = \big[\sum_{i \in \mathcal{V}} \Psi_1(X, y_i); \sum_{(i,j) \in \mathcal{E}} \Psi_2(X, y_i, y_j)\big]. \tag{3}$$

The discriminative function F can be expressed as

$$F(X, Y, G; \mathbf{w}) = \sum_{i \in \mathcal{V}} \underbrace{\mathbf{w}_1^\top \Psi_1(X, y_i)}_{-E_i(y_i)} + \sum_{(i,j) \in \mathcal{E}} \underbrace{\mathbf{w}_2^\top \Psi_2(X, y_i, y_j)}_{-E_{i,j}(y_i, y_j)}, \tag{4}$$

where $\mathbf{w} = [\mathbf{w}_1; \mathbf{w}_2]$. Now, (1) is equivalent to

$$(Y^*, G^*) = \underset{Y, G}{\operatorname{argmin}} \sum_{(i,j) \in \mathcal{E}} E_{i,j}(y_i, y_j) + \sum_{i \in \mathcal{V}} E_i(y_i), \tag{5}$$

which becomes an energy minimisation problem with unknown graph G. Here $E_{i,j}(y_i, y_j)$ is the edge energy function over edge $(i, j) \in \mathcal{E}$ and the $E_i(y_i)$ is the node energy function over vertex $i \in \mathcal{V}$.

2.2. The MAP inference and its LP Relaxation

If G is known, the MAP problem becomes

$$Y^* = \underset{Y}{\operatorname{argmin}} \sum_{(i,j) \in \mathcal{E}} E_{i,j}(y_i, y_j) + \sum_{i \in \mathcal{V}} E_i(y_i). \tag{6}$$

A typical LP relaxation [16] of the MAP problem is

$$\begin{aligned}\min_{q} & \sum_{(i,j)\in\mathcal{E}} \sum_{y_i,y_j} q_{i,j}(y_i,y_j)E_{i,j}(y_i,y_j)+ \\ & \sum_{i\in\mathcal{V}} \sum_{y_i} q_i(y_i)E_i(y_i) \qquad (7) \\ \text{s.t.} \quad & q_{i,j}(y_i,y_j)\in[0,1], \sum_{y_i,y_j} q_{i,j}(y_i,y_j)=1, \\ & \sum_{y_i} q_{i,j}(y_i,y_j)=q_j(y_j), \forall (i,j)\in\mathcal{E}, y_i, y_j. \\ & q_i(y_i)\in[0,1], \forall i\in\mathcal{V}, y_i.\end{aligned}$$

The last constraint can be removed safely since it can be derived from the other constraints. When $q_i(y_i), \forall i \in \mathcal{V}$ are integers, the solution of problem (7) is an exact solution of the problem (6).

3. Bilinear reformulation and LP relaxation

In this section we show how the MAP inference problem with an unknown graph can be formulated as a bilinear program (BLP), which can be further relaxed to an LP. We will also show how to obtain the labels and graph from the solution of the BLP.

3.1. Bilinear program reformulation

Before we solve (5), let us consider a simpler case first where the graph is unknown but the MAP solution Y^* is known. We first introduce variables $\{z_{i,j}\}_{i,j}$ indicating the edge (i,j) exists or not, for all $i,j \in \mathcal{V}$. Then the graph G can be found by seeking the set of edges that give the smallest energy as an integer program:

$$\begin{aligned}\min_{z} & \sum_{i\in\mathcal{V}} \sum_{j\in\mathcal{V}} z_{i,j}E_{i,j}(y_i^*,y_j^*) \qquad (8) \\ \text{s.t.} & \sum_{i\in\mathcal{V}} z_{i,j}\le d, z_{i,j}=z_{j,i}, z_{i,j}\in\{0,1\}, \forall i,j\in\mathcal{V}, y_i^*, y_j^*.\end{aligned}$$

Here we set $E_{i,j}(y_i,y_j)=+\infty$ when $i=j$ since (i,i) is not an edge. $E_i(y_i)$ can be ignored, since it is independent of the choice of edges. $d \in \mathbb{N}$ is a preset number that enforces the maximum degree of a vertex, allowing one to enforce a sparse structure. However dropping the degree constraint does not change the nature of the problem. This integer program can be relaxed to a linear program by changing the domain of $z_{i,j}$ from $\{0,1\}$ to $[0,1]$.

From (7) and (8), we can see that (5) can be relaxed to the problem below,

$$\begin{aligned}\min f(q,z) = & \sum_{i,j\in\mathcal{V}} \sum_{y_i,y_j} q_{i,j}(y_i,y_j)E_{i,j}(y_i,y_j)z_{i,j} \\ & + \sum_{i\in\mathcal{V}} \sum_{y_i} q_i(y_i)E_i(y_i). \qquad (9a) \\ \text{s.t.} \quad & q_{i,j}(y_i,y_j)\in[0,1], \sum_{y_i,y_j} q_{i,j}(y_i,y_j)=1, \qquad (9b) \\ & \sum_{y_i} q_{i,j}(y_i,y_j)=q_j(y_j), z_{i,j}=z_{j,i}, z_{i,j}\in[0,1], \\ & \sum_{i\in\mathcal{V}} z_{i,j}\le d, \forall i,j\in\mathcal{V}, y_i, y_j.\end{aligned}$$

This problem is a BLP with disjoint constraints *i.e.* constraints on z do not involve q, and vice versa.

Solving the BLP in (9) returns $\{q_i(y_i), \forall i \in \mathcal{V}, y_i\}$, $\{q_{i,j}(y_i,y_j), \forall i,j \in \mathcal{V}, y_i, y_j\}$ and $\{z_{i,j}, \forall i,j \in \mathcal{V}\}$. We now show how to obtain the graph G^* and the MAP Y^* (*i.e.* best activities) from BLP outcomes.

Obtaining the graph We start with $\mathcal{E}^*=\emptyset$. $\forall i,j \in \mathcal{V}, i\neq j$, if $z_{i,j}\ge 0.5$, $\mathcal{E}^*=\mathcal{E}^*\cup\{(i,j)\}$. Thus we have the estimated graph $G^*=(\mathcal{V},\mathcal{E}^*)$.

Obtaining the MAP Assume $y_i \in \{1,2,\cdots,K\}$, then $\forall i \in \mathcal{V}$, $y_i^*=\operatorname{argmax}_{k=1}^{K} q_i(k)$. We have the estimated label $Y^*=(y_1^*,y_2^*,\cdots,y_m^*)$.

3.2. LP relaxation

BLP in (9) is non-convex. Here we show how to relax it to an LP, which can be efficiently solved. By introducing $u_{i,j}(y_i,y_j)$ for each (i,j,y_i,y_j), (9) is equivalent to

$$\min_{q,z,u} \sum_{i,j\in\mathcal{V}} \sum_{y_i,y_j} u_{i,j}(y_i,y_j) + \sum_{i\in\mathcal{V}} \sum_{y_i} q_k(y_i)E_i(y_i) \quad (10a)$$

$$\text{s.t.} \begin{cases} q_{i,j}(y_i,y_j)\in[0,1] & \forall i,j\in\mathcal{V}, y_i, y_j, \\ z_{i,j}\in[0,1] & \forall i,j\in\mathcal{V}, \\ \sum_{y_i,y_j} q_{i,j}(y_i,y_j)=1 & \forall i,j\in\mathcal{V}, \\ \sum_{y_i} q_{i,j}(y_i,y_j)=q_j(y_j) & \forall i,j\in\mathcal{V}, y_j, \\ z_{i,j}=z_{j,i} & \forall i,j\in\mathcal{V}, \\ \sum_{j\in\mathcal{V}} z_{i,j}\le d & \forall i\in\mathcal{V}, \\ u_{i,j}(y_i,y_j)\ge & \\ q_{i,j}(y_i,y_j)E_{i,j}(y_i,y_j)z_{i,j} & \forall i,j\in\mathcal{V}, y_i, y_j. \end{cases} \quad (10b)$$

However, the above problem is still non-convex due to the bilinear term $q_{i,j}(y_i,y_j)z_{i,j}$. The bilinear term can be further substituted, however. Following the relaxation techniques in [13, 4], we relax (10) to the following LP,

$$\min_{q,z,u,\gamma} \sum_{i,j\in\mathcal{V}} \sum_{y_i,y_j} u_{i,j}(y_i,y_j) + \sum_{k\in\mathcal{V}} \sum_{y_k} q_k(y_k)E_k(y_k), \quad (11a)$$

$$\text{s.t.} \begin{cases} q_{i,j}(y_i, y_j) \in [0,1] & \forall i,j \in \mathcal{V}, y_i, y_j, \\ z_{i,j} \in [0,1] & \forall i,j \in \mathcal{V}, \\ \sum_{y_i,y_j} q_{i,j}(y_i,y_j) = 1 & \forall i,j \in \mathcal{V}, \\ \sum_{y_i} q_{i,j}(y_i,y_j) = q_j(y_j) & \forall i,j \in \mathcal{V}, y_j, \\ z_{i,j} = z_{j,i} & \forall i,j \in \mathcal{V}, \\ \sum_{j \in \mathcal{V}} z_{i,j} \le d & \forall i \in \mathcal{V}, \\ u_{i,j}(y_i,y_j) \ge & \\ E_{i,j}(y_i,y_j)\gamma_{i,j}(y_i,y_j) & \forall i,j \in \mathcal{V}, y_i, y_j, \\ \gamma^l \le \gamma_{i,j}(y_i,y_j) \le \gamma^u & \forall i,j \in \mathcal{V}, y_i, y_j, \end{cases} \tag{11b}$$

where γ^l is

$$\max\{q^l_{i,j}(y_i,y_j)z_{i,j} + z^l_{i,j}q_{i,j}(y_i,y_j) - q^l_{i,j}(y_i,y_j)z^l_{i,j}, \\ q^u_{i,j}(y_i,y_j)z_{i,j} + z^u_{i,j}q_{i,j}(y_i,y_j) - q^u_{i,j}(y_i,y_j)z^u_{i,j}\},$$

and γ^u is

$$\min\{q^u_{i,j}(y_i,y_j)z_{i,j} + z^l_{i,j}q_{i,j}(y_i,y_j) - q^u_{i,j}(y_i,y_j)z^l_{i,j}, \\ q^l_{i,j}(y_i,y_j)z_{i,j} + z^u_{i,j}q_{i,j}(y_i,y_j) - q^l_{i,j}(y_i,y_j)z^u_{i,j}\}.$$

The LP relaxation (11) provides an efficient way of computing lower bounds for the BLP (9). In order to solve (9), we resort to branch and bound [3] detailed in the next section.

3.3. Branch and bound solution

Branch and bound [3] is an iterative approach for finding global ϵ-close solutions to non-convex problems. Consider minimising a function $f : \mathbb{R}^n \to \mathbb{R}$, over an n-dimensional rectangle $\mathcal{Q}_{\text{init}}$. Any $\mathcal{Q} \subseteq \mathcal{Q}_{\text{init}}$ can be expressed as $\prod_{i=1}^n [l_i, u_i]$ where l_i and u_i are the smallest and largest input values at the i-th dimension. Here we define the length of $\mathcal{Q}$ as $\mathcal{L}(\mathcal{Q}) = \max_{i=1}^n (u_i - l_i)$. The branch and bound method requires two functions Φ_{lb} and Φ_{ub} over any $\mathcal{Q} \subseteq \mathcal{Q}_{\text{init}}$, such that

$$\Phi_{\text{lb}}(\mathcal{Q}) \le \min_{v \in \mathcal{Q}} f(v) \le \Phi_{\text{ub}}(\mathcal{Q}), \tag{12a}$$

$$\forall \epsilon > 0, \exists \delta > 0, \mathcal{L}(\mathcal{Q}) < \delta \Rightarrow \Phi_{\text{ub}}(\mathcal{Q}) - \Phi_{\text{lb}}(\mathcal{Q}) < \epsilon. \tag{12b}$$

Here we consider $f(q,z)$ in (9) and assume $\mathcal{Q}_{\text{init}} = [0,1]^n$, thus $v = (q,z) \in \mathcal{Q}_{\text{init}}$.

Branch strategy Since there are fewer $z_{i,j}$ variables than $q_{i,j}(y_i,y_j)$ variables, we always split $\mathcal{Q}$ along the $z_{i,j}$ variables as suggested in [4].

Bound strategy For any $\mathcal{Q} \subseteq \mathcal{Q}_{\text{init}}$, we let $\Phi_{\text{lb}}(\mathcal{Q})$ be the solution of (11) when restricting $(q,z) \in \mathcal{Q}$. Denoting its solution (q^*, z^*), we let $\Phi_{\text{ub}}(\mathcal{Q}) = f(q^*, z^*)$ in (9). Clearly $\min_{q,z \in \mathcal{Q}} f(q,z) \le f(q^*,z^*)$ since $(q^*,z^*) \in \mathcal{Q}$. Since (11) is a relaxation of (9), $\Phi_{\text{lb}}(\mathcal{Q}) \le \min_{q,z \in \mathcal{Q}} f(q,z)$. Thus condition (12a) is satisfied. By the argument of Lemma 1 in [4], Condition (12b) is also satisfied. Hence the convergence holds.

4. Training

We now present a maximum margin training method for predicting structured output variables, such as human activity labels. Given ℓ image-activities pairs, $\{(X^i, Y^i)\}_{i=1}^\ell$ (note that the graphs are not known), we estimate $\mathbf{w}$ via Latent Structural Support Vector Machine (LSSVM) [19],

$$\min_{\mathbf{w}} \frac{1}{2} \| \mathbf{w} \|^2 + C \sum_{i=1}^{\ell} \Big[\max_{Y,G'} \big[\mathbf{w}^\top \Psi(X^i, Y, G') + \\ \Delta(Y^i, Y) \big] - \max_{G} \mathbf{w}^\top \Psi(X^i, Y^i, G) \Big]_+, \tag{13}$$

where $[a]_+ = \max\{0, a\}$. Here C is the trade-off between the regulariser and the risk, and Δ is the label cost that penalises incorrect activity labels. Here we use a well known hamming distance for the label cost, that is

$$\Delta(Y^i, Y) = \frac{1}{m} \sum_{j=1}^{m} \delta(y^i_j \neq y_j), \tag{14}$$

where the indicator $\delta(\cdot) = 1$ if the statement is true, 0 otherwise. We follow [19] in solving (13) via Convex-Concave Procedure (CCCP) [20] due to non-convexity of (13).

CCCP requires solving the following two problems:

$$\max_{G} \mathbf{w}^\top \Psi(X^i, Y^i, G). \tag{15}$$

$$\max_{Y,G} \mathbf{w}^\top \Psi(X^k, Y, G) + \Delta(Y^k, Y). \tag{16}$$

Here (15) is efficiently solved via (8). Clearly (16) is equivalent to

$$\min_{Y,G} \sum_{(i,j) \in \mathcal{E}} E_{i,j}(y_i, y_j) + \sum_{i \in \mathcal{V}} E'_i(y_i), \tag{17}$$

where $E'_i(y_i) = E_i(y_i) - \delta(y^k_i \neq y_i)$. Note that (17) has the same form as (5), and thus can also be formulated as a bilinear program which in turn can be solved by the Branch and Bound method.

Lan's method The approach closest to ours is that proposed by Lan *et al*. in [9] using LSSVM via CCCP to recognise human activities. They also are required to solve (15) and (16). They use an LP similar to (8) to solve (15), and use a coordinate ascent style algorithm to approximately

solve (16). The coordinate ascent style algorithm iterates the following two steps: (1) holding G^* fixed and solving,

$$Y^* = \underset{Y}{\operatorname{argmax}}\, \mathbf{w}^\top \Psi(X^k, Y, G) + \Delta(Y^k, Y) \qquad (18)$$

via belief propagation; (2) holding Y^* fixed and solving

$$G^* = \underset{G}{\operatorname{argmax}}\, \mathbf{w}^\top \Psi(X^i, Y^*, G) \qquad (19)$$

via an LP (8). It is known that coordinate ascent style algorithms are prone to converging to local optima. In the experiment section we provide an empirical comparison between Lan's method and ours.

5. Experiments

We apply our method to a synthetic dataset and three real datasets. The real datasets include two sports competition datasets (tennis and badminton), and a TV episode dataset.

5.1. Synthetic data

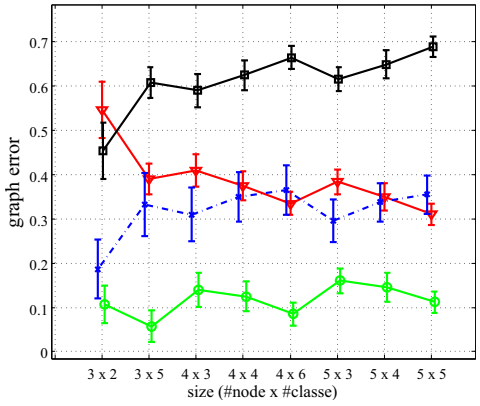
To quantitatively evaluate the performance of different methods for MAP inference and graph estimation, we randomly generate node (unary) and edge (binary) energies for different scales of problems. Specifically, 8 groups of energies are generated and each group corresponds to a fixed number of nodes and a fixed number of activities. For each group we randomly generate energies 50 times. The ground truth graphs and activities labels are obtained by exhaustive search for (5). For the true graph $G = (\mathcal{V}, \mathcal{E})$, predicted graph $G' = (\mathcal{V}, \mathcal{E}')$ with m nodes, true labels Y and predicted label Y', we define two errors below,

$$e_g(G, G') = \frac{1}{m(m-1)} \sum_{i,j \in \mathcal{V}} \mid z_{i,j} - z'_{i,j} \mid, \qquad (20)$$

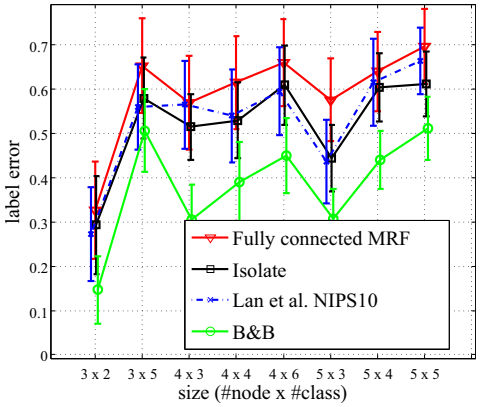
$$e_l(Y, Y') = \frac{1}{m} \sum_{i=1}^{m} \delta(y_i \neq y'_i), \qquad (21)$$

for which we call them the graph error and the label error respectively.

Here we compare four methods: our bilinear method (BLP), Lan's method in [9] (Lan's), the isolated graph (0 edge connections) with node status decided according to node potentials (we call it the Isolate method), belief propagation on fully connected graphs (BP+Full). We report the average errors (for 50 runs) for the 12 groups of synthetic data in Fig. 2. As expected, our bilinear method (the green curves) outperforms all other methods on both graph prediction and activity prediction. The superior results of our method are due to the use of global optimisation techniques, whereas Lan's method achieves only local optima. The errors for both Isolate and BP+Full are high, because their graphs are not learnt. This shows the importance of learning



(a) Graph errors



(b) Label errors

Figure 2. A comparison of the graph error (a) and the label error (b) by different methods. The average graph and label errors are plotted with error bars indicating the standard errors. Note the two subfigures share the same legend in (b). The black, red, blue and green curves correspond to errors by using the Isolate method, the BP+Full method, the Lan's method, and our BLP method respectively.

the graph structure. An interesting observation is that the label prediction error of the Isolate method is much lower than that of BP+Full, though their performance on graph estimation is reversed. This may be caused by (loopy) belief propagation on graphs with many loops — fully connected graphs are used in BP+Full.

5.2. Predict moves in sports

The task is to find sporting activity labels for m persons $Y = (y_1, y_2, \cdots, y_m) \in \mathcal{Y}$, given an observation image $X \in \mathcal{X}$. The bounding box for each athlete is already given.

5.2.1 Datasets

Two datasets are used here. The first one is the UIUC badminton match dataset [17]. We only use the annotated

men's singles video with 3,072 frames. This dataset includes four moves: *smash* (SM), *backhand* (BH), *forehand* (FH) and *others* (OT). The second dataset is created by us. This dataset is a mixed-doubles tennis competition video of 2961 frames. We have annotated this video frame by frame. The annotation includes the position and the size of each body-centred bounding box, the pose and the action of each player. This dataset includes four moves: *normal hit* (HT), *serve* (SV), *volley* (VL) and *others* (OT). Thus the moves for the i-th person can be expressed as a discrete variable $y_i \in \{1, 2, 3, 4\}$.

5.2.2 Features

We use the LSSVM (Section 4) to train a discriminative model to predict moves. The feature Ψ in (3) consists of two local features Ψ_1, Ψ_2 defined below,

$$\Psi_1(X, y_i) = \mathbf{s}_i \otimes \mathbf{e}_1(y_i), \tag{22}$$

$$\Psi_2(X, y_i, y_j) = \mathbf{t}_i \otimes \mathbf{t}_j \otimes \mathbf{e}_1(y_i) \otimes \mathbf{e}_1(y_j) \otimes \mathbf{e}_2(r_{i,j}) \tag{23}$$

- Move Local Appearance: Similar to [9], $\mathbf{s}_i = (s_{i,1}, s_{i,2}, s_{i,3}, s_{i,4}) \in \mathbb{R}^4$, where $s_{i,j}, (j = 1, 2, 3, 4)$ is a move confidence score of assigning the i-th person the j-th move. The the score is the discriminative function value of a SVM classifier trained on the local image descriptor similar to that in [17]. The only difference is that we use histogram the dense gradients, rather than the silhouette of the human body area, since estimating silhouette is tricky when both camera and people are moving. All settings for the feature extraction process are as suggested in the literature. We pick this descriptor for two reasons. First, the descriptor is extracted from a stack of 15 consecutive frames, which accounts for the temporal action context. Second the length of descriptor vector is well controlled via the PCA projection.

- Pose: $\mathbf{t}_i$ is a vector of body pose confidence scores based on a SVM classifier trained on the same local descriptor used for $\mathbf{s}$. Here five discrete body poses are considered: *profile-left*, *profile-right*, *frontal-left*, *frontal-right* and *backwards*, thus $\mathbf{t}_i \in \mathbb{R}^5$.

- Relative position: Here $r_{i,j} \in \{1, 2, \cdots, 6\}$ is the relative position of person i and person j. There are six possible relative positions including *overlap*, *near-left*, *near-right*, *adjacent-left*, *adjacent-right* and *far*. The relative position of two persons is determined according to their 2D Euclidean distance as in [14].

- Tensor product: Here $\mathbf{e}_1(y_i)$ is a 4-dimensional vector with 1 in the y_i-th dimension, and 0 elsewhere. Likewise, $\mathbf{e}_2(r_{i,j})$ is a 6-dimensional vector with 1 in the $r_{i,j}$-th dimension, and 0 elsewhere. $\otimes$ denotes the Kronecker tensor.

Intuitively, Ψ_1 reflects the confidence of assigning one person to different moves, Ψ_2 captures the co-occurrence of the related persons' body poses, relative 2D position and their moves. Based on this joint feature representation, our model predict the moves via (5).

All datasets are randomly split into two parts: one for training and the other for testing. We compare four methods:

- MCSVM uses a multiclass SVM trained on the local image descriptor.

- SSVM uses a structural SVM [18] for training. Finding the most violated constraint requires inference which is done via BP. The graph is constructed by finding the minimum spanning tree weighted by the 2D Euclidean distance between persons.

- Lan's method described in Section 4. The degree of the vertex d is set to 1.0. This means each node can have at most two edges ($z_{i,j} \geq 0.5$ adds an edge).

- BLP (our method) described in Section 4. We use Mosek package [2] to solve LP involved in our bilinear method. We set the degree d in Eq. (11b) being 1.0.

The confusion matrices for two datasets are presented in Table 1 and 2. We can see that in tennis dataset, our BLP outperforms the other methods on all four moves. In badminton dataset, our BLP outperforms the other methods on 3 moves: FH, BH and SM. SSVM achieves the best recognition rate on OT, and our BLP achieves the second best. As expected, MCSVM performs the worst in general, since it relies on the local descriptors. An interesting observation is that Lan's method does not always outperform SSVM.

5.3. Predict activities in TV episodes

The TVHI dataset proposed in [14] is a benchmark for predicting HAR in real TV episodes. It contains 300 short videos collected from TV episodes and includes five activities: *handshake* (HS), *hug* (HG), *High-Five* (HF), *kiss* (KS) and *No-Interaction* (NO). Each of the first four activities consists of 50 video clips. *No-Interaction* contains 100 video clips. We also bisect this dataset into training and testing parts as was done in the sports competition datasets. We use the same four methods as in the sports competition datasets. The confusion matrices are presented in Table 3. Our BLP outperforms the others on all activities. Lan's method performs reasonably well on four activities: HS, HG, HF, and KS, but performs poorly on NO.

We show prediction results in Fig. 3 for four methods on different datasets. A green node means correct prediction and a red node means wrong prediction. We can

Table 1. Confusion matrices of the tennis dataset (sports match)

Alg.	MCSVM				SSVM				Lan's				BLP			
A/A	**OT**	**SV**	**HT**	**VL**	**OT**	**SV**	**HT**	**VL**	**OT**	**SV**	**HT**	**VL**	**OT**	**SV**	**HT**	**VL**
OT	0.43	0.31	0.11	0.15	0.48	0.17	0.20	0.14	0.30	0.29	0.20	0.21	**0.59**	0.08	0.22	0.11
SV	0	0.59	0.17	0.24	0.15	0.75	0	0.09	0.10	0.74	0.15	0.01	0.17	**0.76**	0.06	0
HT	0.23	0.27	0.23	0.27	0.31	0.18	0.31	0.21	0.31	0.09	0.35	0.25	0.39	0.07	**0.39**	0.15
VL	0.08	0.06	0.06	0.80	0.15	0.14	0.06	0.65	0.15	0	0	0.85	0	0	0.10	**0.90**

Table 2. Confusion matrices of the badminton dataset (sports match)

Alg.	MCSVM				SSVM				Lan's				BLP			
A/A	**OT**	**FH**	**BH**	**SM**	**OT**	**FH**	**BH**	**SM**	**OT**	**FH**	**BH**	**SM**	**OT**	**FH**	**BH**	**SM**
OT	0.38	0.15	0.12	0.34	**0.65**	0.21	0.03	0.11	0.56	0.21	0.06	0.17	0.61	0.19	0.02	0.18
FH	0.38	0.40	0	0.22	0.14	0.45	0.22	0.18	0.07	0.42	0.24	0.27	0.10	**0.52**	0.14	0.24
BH	0.14	0.23	0.44	0.19	0.32	0.24	0.39	0.05	0.01	0.16	0.65	0.18	0	0.27	**0.66**	0.07
SM	0.08	0.04	0.16	0.71	0.16	0.12	0.08	0.64	0.09	0.06	0.08	**0.78**	0.08	0.07	0.07	**0.78**

see that when the graph is learnt correctly, MRF-based approaches (see in the 2nd row, column 3, 4) outperform the i.i.d. method MCSVM (see in the 2nd row, column 1). We also see that Lan's method often learns better graphs (see the 3rd column) than the graphs built from heuristics in SSVM (see the 2nd column). Overall, our BLP learns the most accurate graph and therefore makes the most accurate activity prediction.

6. Conclusion and future work

The structure of the graph used is critical to the success of any MRF-based approach to human activity recognition, because they encapsulate the relationships between the activities of multiple participants. These graphs are often manually specified, or automatically constructed by heuristics, but both approaches have their limitations. We have thus shown that it is possible to develop a MAP inference method for unknown graphs, and reformulated the problem of finding MAP solution and the best graph jointly as a bilinear program, which is solved by branch and bound. An LP relaxation is used as a lower bound for the bilinear program. Using the bilinear program based MAP inference, we have shown that it is possible to estimate parameters efficiently and effectively with a latent structural SVM. Applications in predicting sport moves and human activities in TV episodes have shown the strength of our method.

The BLP formulation is not only applicable to MRFs, but also to graphical models with factor graphs and directed graphs (*i.e*. Bayesian networks). One possible future work is to seek BP style algorithms to replace LP relaxations in solving BLP. Another future direction is to consider temporal dependencies among video frames using tracking techniques and dynamic hierarchy graphical models.

Acknowledgement This work was supported by Australian Research Council grants DP1094764, DE120101161, FT120100969, and DP11010352.

References

[1] J. K. Aggarwal and M. S. Ryoo. Human activity analysis: A review. *ACM Comput. Surv.*, 43(3), 2011.

[2] E. D. Andersen, C. Roos, and T. Terlaky. On implementing a primal-dual interior-point method for conic quadratic optimization. *Mathematical Programming*, 95, 2003. `http://www.mosek.com`.

[3] S. Boyd and J. Mattingley. Branch and bound methods, 2003.

[4] M. Chandraker and D. Kriegman. Globally optimal bilinear programming for computer vision applications. In *Proc. IEEE Conf. Comp. Vis. & Pattern Recogn.*, 2008.

[5] W. Choi, K. Shahid, and S. Savarese. Learning context for collective activity recognition. In *Proc. IEEE Conf. Comp. Vis. & Pattern Recogn.*, 2011.

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. IEEE Conf. Comp. Vis. & Pattern Recogn.*, 2005.

[7] A. Globerson and T. Jaakkola. Fixing max-product: Convergent message passing algorithms for map lp-relaxations. In *Proc. Adv. Neural Info. Process. Syst.*, 2007.

[8] R. Kinderman and J. L. Snell. *Markov Random Fields and their applications*. Amer. Math. Soc., Providence, RI, 1980.

[9] T. Lan, Y. Wang, and G. Mori. Beyond actions: Discriminative models for contextual group activities. In *Proc. Adv. Neural Info. Process. Syst.*, 2010.

[10] I. Laptev. On space-time interest points. *Int. J. Comput. Vis.*, 64(2):107–123, 2005.

[11] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Proc. IEEE Conf. Comp. Vis. & Pattern Recogn.*, 2008.

[12] O. Masoud and N. Papanikolopoulos. A method for human action recognition. *Image Vision Comput.*, 21(8):729–743, 2003.

[13] G. P. McCormick. Computability of global solutions to factorable nonconvex programs. *Mathematical programming*, 10(1):147–175, 1976.

[14] A. Patron-Perez, M. Marszalek, A. Zisserman, and I. Reid. High five: Recognising human interactions in tv shows. In *Proc. British Comp. Vis. Conf.*, 2010.

[15] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In *Proc. Int. Conf. Pattern Recogn.*, 2004.

Table 3. Confusion matrices of the TVHI dataset (television episodes)

Alg.	MCSVM					SSVM					Lan's					BLP				
A/A	**NO**	**HS**	**HF**	**HG**	**KS**	**NO**	**HS**	**HF**	**HG**	**KS**	**NO**	**HS**	**HF**	**HG**	**KS**	**NO**	**HS**	**HF**	**HG**	**KS**
NO	0.37	0.07	0.21	0.11	0.24	0.20	0.40	0.27	0.06	0.06	0.11	0.36	0.19	0.20	0.13	**0.49**	0.20	0.13	0.13	0.05
HS	0.01	0.55	0.06	0.17	0.21	0.10	0.51	0.21	0.11	0.06	0.09	0.52	0.14	0.15	0.10	0.18	**0.56**	0.09	0.08	0.09
HF	0.09	0.03	0.52	0.21	0.14	0.08	0.11	0.61	0.08	0.12	0.02	0.14	0.58	0.18	0.08	0.11	0.09	**0.63**	0.07	0.10
HG	0.02	0.14	0.20	0.49	0.15	0.05	0.15	0.11	0.58	0.11	0.03	0.06	0.11	0.55	0.26	0.03	0.10	0.10	**0.70**	0.06
KS	0.07	0.11	0.09	0.05	0.67	0.02	0.26	0.14	0.12	0.46	0.01	0.07	0.15	0.11	0.67	0.06	0.08	0.04	0.13	**0.69**

Figure 3. Prediction results of different methods on tennis data (1st two rows), badminton data (the 3rd row) and TV episode data (the last 2 rows). **First Column:** the i.i.d. method using MCSVM; **Second Column:** SSVM; **Third Column**: Lan's method; **Last Column:** our BLP method. Both the nodes and edges of the MRFs are shown. For the i.i.d. method, there are no edges. The green node indicates correct predictions, and the red nodes incorrect predictions.

[16] D. Sontag, T. Meltzer, A. Globerson, Y. Weiss, and T. Jaakkola. Tightening LP relaxations for MAP using message-passing. In *Proc. Conf. Uncertainty in Artificial Intelli.*, 2008.

[17] D. Tran and A. Sorokin. Human activity recognition with metric learning. In *Proc. Eur. Conf. Comput. Vis.*, 2008.

[18] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *J. Mach. Learn. Res.*, 6(2):1453–1484, 2006.

[19] C. N. J. Yu and T. Joachims. Learning structural svms with latent variables. In *Proc. Int. Conf. Mach. Learn.*, 2009.

[20] A. Yuille, A. Rangarajan, and A. L. Yuille. The concave-convex procedure (cccp. In *Proc. Adv. Neural Info. Process. Syst.* MIT Press, 2002.