Tracking Objects with Column Generation

Shaofei Wang
sfwang0928@gmail.com

Steffen Wolf
Heidelberg University, Germany
steffen.wolf@iwr.uni-heidelberg.de

Charless C. Fowlkes
University of California, Irvine
fowlkes@ics.uci.edu

Julian Yarkony
Experian Data Lab, San Diego, CA
julian.e.yarkony@gmail.com

Abstract

We formulate multi-target tracking in video as a maximum weight set packing problem where tracks correspond to sets. We then attack it with column and row generation where pricing is done efficiently using dynamic programming.

1 Introduction

Multi-target tracking in video is often formulated from the perspective of grouping disjoint sets of candidate detections into "tracks" whose underlying trajectories can be estimated using traditional single-target tracking methods such as Kalman filtering. There is a well developed literature on methods for exploring this combinatorial space of possible data associations in order to find collections of low-cost, disjoint tracks.

Our method is most closely related to the Lagrangian relaxation method of [2]. In [2] a large number of short sequences of detections (subtracks) are generated, each of which is associated with a cost. The set of subtracks form the basis from which tracks are constructed. The corresponding objective is attacked via subgradient optimization. In contrast we attack the same problem with column/row generation providing faster inference. We also tighten the bounds by optimizing over additional triplet constraints, which is inspired by [5].

2 Constraint Relaxation for Multi-target Tracking

We now consider our approach. Given a set of candidate detections $\mathcal{D}$, each with a specified space-time location, our goal is to identify a collection of tracks that describe the trajectories of objects through a scene and the subset of detections associated with each such track.

We denote the set of all possible tracks by $\mathcal{P}$ and use X to denote the detection-track incidence matrix $X \in \{0,1\}^{|\mathcal{D}|\times|\mathcal{P}|}$ where $X_{dp} = 1$ if and only if track p visits detection d. A solution to the multi-target tracking problem is denoted by the indicator vector $\boldsymbol{\gamma} \in \{0,1\}^{|\mathcal{P}|}$ where $\boldsymbol{\gamma}_p = 1$ indicates that track p is included in the solution and $\boldsymbol{\gamma}_p = 0$ otherwise. A collection of tracks specified by $\boldsymbol{\gamma}$ is a valid solution if and only if each detection is associated with at most one active track. Using $\Theta \in \mathbb{R}^{|\mathcal{P}|}$ to denote the costs associated with tracks where Θ_p describes the cost of track p, we express our tracking problem as an integer linear program:

$$\min_{\boldsymbol{\gamma}\in\bar{\boldsymbol{\Gamma}}} \Theta^t\boldsymbol{\gamma} \qquad \text{with} \qquad \bar{\boldsymbol{\Gamma}} = \{\boldsymbol{\gamma} \in \{0,1\}^{|\mathcal{P}|} : \; X\boldsymbol{\gamma} \leq 1\} \tag{1}$$

We note that this is equivalent to finding a maximum-weight set packing which is NP-hard [3].

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

2.1 Decomposing Track Scores over Subtracks

We consider a general scoring function corresponding to a model in which a track is defined by an ordered sequence of subtracks whose scores in turn depend on detections across several frames. Let $\mathcal{S}$ denote a set of subtracks, each of which contains K detections where K is a user defined modeling parameter that trades off inference complexity and modeling power. For a given subtrack $s \in \mathcal{S}$, let s_k indicate the k'th detection in the sequence $s = \{s_1, \ldots, s_K\}$ ordered by time from earliest to latest. We describe the mapping of subtracks to tracks using $T \in \{0,1\}^{|\mathcal{S}| \times |\mathcal{P}|}$ where $T_{sp} = 1$ indicates that track p contains subtrack s as a subsequence.

We decompose track costs Θ in terms of the subtrack costs $\theta \in \mathbb{R}^{|\mathcal{S}|}$ where each subtrack s is associated with cost θ_s and use θ_0 to denote a constant cost associated with instancing a track. We define the cost of a track p denoted Θ_p as $\Theta_p = \theta_0 + \sum_{s \in \mathcal{S}} T_{sp} \theta_s$.

2.2 LP Relaxation and Column Generation

We now attack optimization in Eq 1 using the well studied tools of LP relaxations. We use $\mathbf{\Gamma} = \{\boldsymbol{\gamma} \in [0,1]^{|\mathcal{P}|} : \; X\boldsymbol{\gamma} \leq 1\}$ to denote a convex relaxation of the constraint set $\bar{\mathbf{\Gamma}}$.

$$\min_{\boldsymbol{\gamma} \in \bar{\mathbf{\Gamma}}} \Theta^t \boldsymbol{\gamma} \quad \geq \quad \min_{\boldsymbol{\gamma} \in \mathbf{\Gamma}} \Theta^t \boldsymbol{\gamma} \tag{2}$$

This LP relaxation only contains constraints for collections of tracks that share a common detection. From the view point of maximum-weight set packing, this includes some cliques of conflicting sets but misses many others.

As a concrete example, consider four tracks $\mathcal{P} = \{p_1, p_2, p_3, p_4\}$ over three detections $\mathcal{D} = \{d_1, d_2, d_3\}$ where the first three tracks each contain two of the three detections $\{d_1, d_2\}, \{d_1, d_3\}, \{d_2, d_3\}$, and the fourth track contains all three $\{d_1, d_2, d_3\}$. Suppose the track costs are given by $\Theta_{p_1} = \Theta_{p_2} = \Theta_{p_3} = -4$ and $\Theta_{p_4} = -5$. The optimal integer solution sets $\boldsymbol{\gamma}_{p_4} = 1$, and has a cost of -5. However the optimal fractional solution sets $\boldsymbol{\gamma}_{p_1} = \boldsymbol{\gamma}_{p_2} = \boldsymbol{\gamma}_{p_3} = 0.5$; $\boldsymbol{\gamma}_{p_4} = 0$ which has cost -6. Hence the LP relaxation is loose in this case.

A tighter bound can be motivated by the following observation. For any set of three unique detections the number of tracks that pass through two or more members can be no larger than one. We now apply our tighter bound to tracking. We denote the set of groups of three unique detections (which we refer to as triplets) as $\mathcal{C}$ and index it with c. Using $C \in \{0,1\}^{|\mathcal{C}| \times |\mathcal{P}|}$ we define a tighter space for $\boldsymbol{\gamma}$ than $\mathbf{\Gamma}$.

$$\mathbf{\Gamma}^C : \{\boldsymbol{\gamma} \in \mathbb{R}^{|\mathcal{P}|} : \boldsymbol{\gamma} \geq 0, \quad X\boldsymbol{\gamma} \leq 1, \quad C\boldsymbol{\gamma} \leq 1\} \tag{3}$$
$$C_{cp} = [\sum_{d \in c} X_{dp} \geq 2] \quad \forall c \in \mathcal{C}, p \in \mathcal{P}$$

3 Optimization over $\mathbf{\Gamma}^C$

We write tracking as optimization in the primal and dual form below.

$$\min_{\boldsymbol{\gamma} \in \mathbf{\Gamma}^C} \Theta^t \boldsymbol{\gamma} = \max_{\substack{\boldsymbol{\lambda} \geq 0 \\ \boldsymbol{\lambda}^{\mathcal{C}} \geq 0 \\ \Theta + X^t \boldsymbol{\lambda} + C^t \boldsymbol{\lambda}^{\mathcal{C}} \geq 0}} -1^t \boldsymbol{\lambda} - 1^t \boldsymbol{\lambda}^{\mathcal{C}} \tag{4}$$

Given that $\mathcal{P}$ and $\mathcal{C}$ are of enormous size we use column and row generation jointly. The nascent subsets of $\mathcal{P}, \mathcal{C}$ are denoted $\hat{\mathcal{P}}, \hat{\mathcal{C}}$ respectively. We write column/row generation optimization given subroutines COLUMN$(\boldsymbol{\lambda}, \boldsymbol{\lambda}^{\mathcal{C}})$, ROW$(\boldsymbol{\gamma})$ that identify a group of violated constraints in primal and dual including the most violated in each. We write the column/row generation optimization in Alg 1.

Finding the most violated row consists of the following optimization: $\max_{c \in \mathcal{C}} \sum_{p \in \mathcal{P}} C_{cp} \boldsymbol{\gamma}_p$. We generate its rows as needed by considering only triplets over detections associated with fractional valued tracks.

Algorithm 1 Column/Row Generation

$\hat{\mathcal{P}} \leftarrow \{\}, \quad \hat{\mathcal{C}} \leftarrow \{\}$
repeat
  $\max_{\substack{\boldsymbol{\lambda}\geq 0 \\ \boldsymbol{\lambda}^{\mathcal{C}}\geq 0 \\ \Theta_{\hat{\mathcal{P}}}+X^t_{(:,\hat{\mathcal{P}})}\boldsymbol{\lambda}+C^t_{(\hat{\mathcal{C}},\hat{\mathcal{P}})}\boldsymbol{\lambda}^{\mathcal{C}}\geq 0}} -1^t\boldsymbol{\lambda} - 1^t\boldsymbol{\lambda}^{\mathcal{C}}$
  Recover $\boldsymbol{\gamma}$ from $\boldsymbol{\lambda}$ (provided by LP solver)
  $\dot{\mathcal{P}} \leftarrow \text{COLUMN}(\boldsymbol{\lambda}, \boldsymbol{\lambda}^{\mathcal{C}}) \qquad \dot{\mathcal{C}} \leftarrow \text{ROW}(\boldsymbol{\gamma})$
  $\hat{\mathcal{P}} \leftarrow [\hat{\mathcal{P}}, \dot{\mathcal{P}}] \qquad \hat{\mathcal{C}} \leftarrow [\hat{\mathcal{C}}, \dot{\mathcal{C}}]$
until $\dot{\mathcal{P}} = []$ and $\dot{\mathcal{C}} = []$

Algorithm 2 Upper Bound Rounding

while $\exists p \in \mathcal{P}$ s.t. $\boldsymbol{\gamma}_p \notin \{0, 1\}$ **do**
  $p^* \leftarrow \arg\min_{\substack{p\in\mathcal{P} \\ \boldsymbol{\gamma}_p>0}} \Theta_p\boldsymbol{\gamma}_p - \sum_{\hat{p}\in\mathcal{P}_{\perp p}} \boldsymbol{\gamma}_{\hat{p}}\Theta_{\hat{p}}$
  $\boldsymbol{\gamma}_{\hat{p}} \leftarrow 0 \quad \forall \hat{p} \in \mathcal{P}_{\perp p^*}$
  $\boldsymbol{\gamma}_{p^*} \leftarrow 1$
end while
RETURN $\boldsymbol{\gamma}$

Figure 1: (Left): Algorithm for dual-optimization of a lower bound on the optimal tracking by column generation where the notation $X_{(:,\hat{\mathcal{P}})}$ denotes selection of a subset of columns of X. (Right) We compute upper-bounds on the optimal tracking using a rounding procedure which greedily selects primal variables $\boldsymbol{\gamma}$ while removing intersecting tracks.

3.1 Computing COLUMN$(\boldsymbol{\lambda}, \boldsymbol{\lambda}^c)$ using Dynamic Programming Without Triplets

We now discuss how COLUMN$(\boldsymbol{\lambda}, \boldsymbol{\lambda}^c)$ is computed efficiently for our track cost model using dynamic programming when $\boldsymbol{\lambda}^c$ is zero valued. We later show how to use this when $\boldsymbol{\lambda}^c$ is not zero valued. We specify that a subtrack s may be preceded by a subtrack $\hat{s}$ if and only if the least recent $K-1$ detections in s correspond to the most recent $K-1$ detections in $\hat{s}$. We denote the set of valid subtracks that may precede a subtrack s as $\{\Rightarrow s\}$. We use ℓ_s to denote the cost of the cheapest track that terminates at subtrack s.

$$\ell_s \leftarrow \theta_s + \boldsymbol{\lambda}_{s_K} + \min\{\min_{\hat{s}\in\{\Rightarrow s\}} \ell_{\hat{s}}, \quad \theta_0 + \sum_{k=0}^{K-1} \boldsymbol{\lambda}_{s_k}\} \tag{5}$$

Empirically we observe large speed ups by adding the most violated track terminating at each subtrack (if one exists) which is easy to extract from the dynamic program. Thus we add many columns per solution to dynamic programming.

3.2 Generating Columns under Triplet Constraints

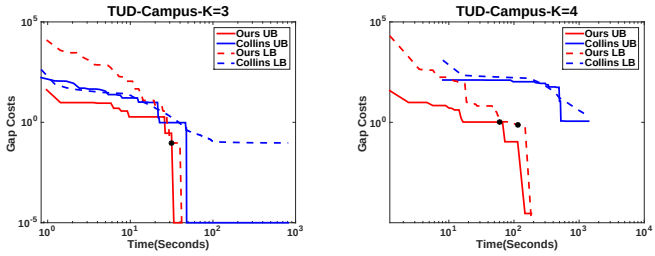
We denote the value of the slack corresponding to an arbitrary column p as $V(\Theta, \boldsymbol{\lambda}, \boldsymbol{\lambda}^{\mathcal{C}}, p)$ and the most violated as $V^*(\Theta, \boldsymbol{\lambda}, \boldsymbol{\lambda}^{\mathcal{C}})$ which we define below.

$$V(\Theta, \boldsymbol{\lambda}, \boldsymbol{\lambda}^{\mathcal{C}}, p) = \Theta_p + \sum_{d\in\mathcal{D}} \boldsymbol{\lambda}_d X_{dp} + \sum_{c\in\hat{\mathcal{C}}} \boldsymbol{\lambda}^{\mathcal{C}}_c C_{cp} \qquad V^*(\Theta, \boldsymbol{\lambda}, \boldsymbol{\lambda}^{\mathcal{C}}) \quad = \min_{p\in\mathcal{P}} V(\Theta, \boldsymbol{\lambda}, \boldsymbol{\lambda}^{\mathcal{C}}, p) \tag{6}$$

Solving for $V^*(\Theta, \boldsymbol{\lambda}, \boldsymbol{\lambda}^{\mathcal{C}})$ can not be directly attacked using dynamic programming as in Section 3.1. However dynamic programming can be applied if we ignore the triplet term $\sum_{c\in\hat{\mathcal{C}}} \boldsymbol{\lambda}^{\mathcal{C}}_c C_{cp}$, providing a lower bound.

This invites a branch and bound (B&B) approach. The set of branches in our B&B tree is denoted $\mathcal{B}$. Each branch $b \in \mathcal{B}$ is defined by two sets $\mathcal{D}_{b+}$ and $\mathcal{D}_{b-}$. These correspond to detections that must be included on the track and those that must not be included on the track respectively. We write the set of all tracks that are consistent with a given $\mathcal{D}_{b-}$, $\mathcal{D}_{b+}$ or consistent with both $\mathcal{D}_{b-}$ and $\mathcal{D}_{b+}$ as $\mathcal{P}_{b-}, \mathcal{P}_{b+}$ and $\mathcal{P}_{b\pm}$ respectively.The initial branch b is defined by $\mathcal{D}_{b+} = \mathcal{D}_{b-} = \{\}$.

Bounding Operation: Let $V^b(\Theta, \boldsymbol{\lambda}, \boldsymbol{\lambda}^{\mathcal{C}})$ denote the value of the most violating slack over columns in $\mathcal{P}_{b\pm}$. We can compute a lower-bound for this value, denoted V^b_{lb} by independently optimizing the



(a) (b)

Figure 2: When training on a subset of motion features on MOT dataset we get instances with loose bound. For the two examples we plot the gap (absolute value of the difference) between the bounds and the final lower bound as a function of time. We indicate each time that a triplet is added with a black dot on the lower bound plot. In all examples the bound of [2] is loose and at least one triplet is needed to produce a tight bound which results in visually compelling trackings.

dynamic program and the triplet penalty.

$$V^b(\Theta,\boldsymbol{\lambda},\boldsymbol{\lambda}^{\mathcal{C}}) = \min_{p\in\mathcal{P}_{b\pm}} V(\Theta,\boldsymbol{\lambda},\boldsymbol{\lambda}^{\mathcal{C}},p) \geq \min_{p\in\mathcal{P}_{b-}} \Theta_p + \sum_{d\in\mathcal{D}} \boldsymbol{\lambda}_d X_{dp} + \min_{p\in\mathcal{P}_{b+}} \sum_{c\in\hat{\mathcal{C}}} \boldsymbol{\lambda}^{\mathcal{C}}_c C_{cp}$$
$$\geq \min_{p\in\mathcal{P}_{b-}} \Theta_p + \sum_{d\in\mathcal{D}} \boldsymbol{\lambda}_d X_{dp} + \sum_{c\in\hat{\mathcal{C}}} \boldsymbol{\lambda}^{\mathcal{C}}_c [\sum_{d\in c} [d\in\mathcal{D}_{b+}] \geq 2] = V^b_{lb}(\Theta,\boldsymbol{\lambda},\boldsymbol{\lambda}^{\mathcal{C}})$$

Observe that dynamic programming can be used to efficiently search over $\mathcal{P}_{b-}$ to minimize the first term.

Branch Operation: We now consider the branch operation. We describe an upper bound on $V^b(\Theta,\boldsymbol{\lambda},\boldsymbol{\lambda}^{\mathcal{C}})$ as $V^b_{ub}(\Theta,\boldsymbol{\lambda},\boldsymbol{\lambda}^{\mathcal{C}})$. This is constructed by adding in the active $\boldsymbol{\lambda}^{\mathcal{C}}$ terms ignored when constructing $V^b_{lb}(\Theta,\boldsymbol{\lambda},\boldsymbol{\lambda}^{\mathcal{C}})$. Let $p_b = \arg\min_{p\in\mathcal{P}_{b-}} \Theta_p + \sum_{d\in\mathcal{D}} \boldsymbol{\lambda}_d X_{dp}$.

$$V^b_{ub}(\Theta,\boldsymbol{\lambda},\boldsymbol{\lambda}^{\mathcal{C}}) = V^b_{lb}(\Theta,\boldsymbol{\lambda},\boldsymbol{\lambda}^{\mathcal{C}}) + \sum_{c\in\hat{\mathcal{C}}} \boldsymbol{\lambda}^{\mathcal{C}}_c C_{cp_b} [\sum_{d\in c} [d\in\mathcal{D}_{b+}] < 2] \quad (7)$$

$$= \Theta_{p_b} + \sum_{d\in\mathcal{D}} \boldsymbol{\lambda}_d X_{dp_b} + \sum_{c\in\hat{\mathcal{C}}} \boldsymbol{\lambda}^{\mathcal{C}}_c [\sum_{d\in c} [d\in\mathcal{D}_{b+}] \geq 2] + \sum_{c\in\hat{\mathcal{C}}} \boldsymbol{\lambda}^{\mathcal{C}}_c C_{cp_b} [\sum_{d\in c} [d\in\mathcal{D}_{b+}] < 2]$$
$$\geq V(\Theta,\boldsymbol{\lambda},\boldsymbol{\lambda}^{\mathcal{C}},p_b) \geq V^b(\Theta,\boldsymbol{\lambda},\boldsymbol{\lambda}^{\mathcal{C}}) \quad (8)$$

Now consider the largest triplet constraint term $\boldsymbol{\lambda}^{\mathcal{C}}_c$ that is included in $V^b_{ub}(\Theta,\boldsymbol{\lambda},\boldsymbol{\lambda}^{\mathcal{C}},p_b)$ but not $V^b_{lb}(\Theta,\boldsymbol{\lambda},\boldsymbol{\lambda}^{\mathcal{C}})$. $c^* \leftarrow \arg\max_{c\in\hat{\mathcal{C}}} \boldsymbol{\lambda}^{\mathcal{C}}_c C_{cp_b} [\sum_{d\in c} [d\in\mathcal{D}_{b+}] < 2]$

We create eight child branches of b where there is one for each of the eight different ways of splitting the detections in the triplet corresponding to c^* between the include $(+)$ and exclude $(-)$ sets.

We compute upper bounds using a fast principled method that avoids resolving the LP. We round fractional $\boldsymbol{\gamma}$ via a greedy iterative approach described in Alg 2. We rely on the technique of [6] (supplement Section 4) to produce the following lower bound: $\min_{\boldsymbol{\gamma}\in\bar{\boldsymbol{\Gamma}}} \Theta^t\boldsymbol{\gamma} \geq -1^t\boldsymbol{\lambda} + \sum_{d\in\mathcal{D}} \min\{0, \min_{\substack{s\in\mathcal{S}\\ s_K=d}} \ell_s\}$.

4 Experiments

We use a part of MOT 2015 training set [4] to train and evaluate real-world tracking models. For $K = 2$ we observe 48.5% Multiple Object Tracking Accuracy [1], 11 identity switches and 9 track fragments or for short hand (48.5,11,9). However when setting $K = 3, 4$ the performance is (49,10,7), and (49.9,9,7) which constitutes noticeable improvements over all three metrics. In Fig 2 we compared the timing/cost performance of our algorithm with the baseline algorithm of [2].

References

[1] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: The clear mot metrics. *Journal on Image Video Processing*, 2008.

[2] A. Butt and R. Collins. Multi-target tracking by lagrangian relaxation to min-cost network flow. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1846–1853, June 2013.

[3] R. M. Karp. Reducibility among combinatorial problems. The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972.

[4] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler. MOTChallenge 2015: Towards a benchmark for multi-target tracking. *arXiv:1504.01942 [cs]*, Apr. 2015. arXiv: 1504.01942.

[5] D. Sontag, T. Meltzer, A. Globerson, T. Jaakkola, and Y. Weiss. Tightening lp relaxations for map using message passing. In *Proceedings of the Twenty-Fourth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-08)*, pages 503–510, july 2008.

[6] J. Yarkony and C. Fowlkes. Planar ultrametrics for image segmentation. In *Neural Information Processing Systems*, 2015.