# Exploiting Transitivity for Learning Person Re-identification Models on a Budget

Sourya Roy, Sujoy Paul, Neal E. Young and Amit K Roy-Chowdhury
University of California, Riverside CA 92521
{sroy@ece,supaul@ece,neal.young@,amitrc@ece}.ucr.edu

## Abstract

*Minimization of labeling effort for person re-identification in camera networks is an important problem as most of the existing popular methods are supervised and they require large amount of manual annotations, acquiring which is a tedious job. In this work, we focus on this labeling effort minimization problem and approach it as a subset selection task where the objective is to select an optimal subset of image-pairs for labeling without compromising performance. Towards this goal, our proposed scheme first represents any camera network (with $k$ number of cameras) as an edge weighted complete $k$-partite graph where each vertex denotes a person and similarity scores between persons are used as edge-weights. Then in the second stage, our algorithm selects an optimal subset of pairs by solving a triangle free subgraph maximization problem on the $k$-partite graph. This sub-graph weight maximization problem is NP-hard (at least for $k \geq 4$) which means for large datasets the optimization problem becomes intractable. In order to make our framework scalable, we propose two polynomial time approximately-optimal algorithms. The first algorithm is a $1/2$-approximation algorithm which runs in linear time in the number of edges. The second algorithm is a greedy algorithm with sub-quadratic (in number of edges) time-complexity. Experiments on three state-of-the-art datasets depict that the proposed approach requires on an average only 8-15% manually labeled pairs in order to achieve the performance when all the pairs are manually annotated.*

## 1. Introduction

Person re-identification is a challenging task in computer vision which aims to recognize the same person across different cameras. In recent times, person re-id has attracted a significant amount of research interest because of its various security and surveillance related applications [8, 23]. The basic problem definition of person re-id is as follows: given a person's image from one camera (denoted as 'probe') we have to find a matching person (if it exists) in a set of images from another camera.

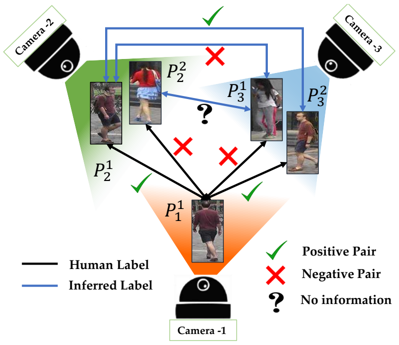


Figure 1: This figure illustrates the motivation of our approach. Here, we have a camera network with three cameras. $P_k^i$ represent the '$i$'-th person in the '$k$'-th camera. Now suppose, we ask the human to label the pairs $P_1^1 - P_2^1$ and $P_1^1 - P_3^2$ by asking a yes/no question. As both of them are positive matches, after we know the labels of these two pairs using transitivity property we can correctly infer the label of $P_2^1 - P_3^2$. Similarly, if we know labels of $P_1^1 - P_2^1$ and $P_1^1 - P_3^1$ we can precisely infer that $P_2^1 - P_3^1$ is a negative match. However, knowing the labels of pairs $P_1^1 - P_2^2$ and $P_1^1 - P_3^1$ does not give us any information about the pair $P_2^2 - P_3^1$.

Person re-id methods can be broadly categorized into three classes: supervised [11, 12, 13, 28, 33], semi-supervised [32,34] and unsupervised methods [2,4,10,15,35]. Among these approaches, supervised distance metric learning based methods are specifically popular because of their robustness towards large color variations and fast training speed. However, like other supervised methods, metric learning algorithms have their own burden of human labeling effort especially for large camera networks [27]. The total number of training pairs assumed to be available by these algorithms increases tremendously with network size and number of persons in each camera. Manual labeling of such huge number of pairs is a tedious and expensive process. So naturally a question arises: given a camera network, *can we come up with a strategy of choosing a minimal subset of image pairs for labeling without compromising on recognition performance?* This is a problem of considerable significance in the context of person re-id in multi-camera networks, especially in larger ones. However, the problem has received little attention in the literature thus far.

Transitive relations among person identities across multiple cameras and their logical consequences are strongly informative properties. These properties have been explored previously for globally consistent person re-id in several existing works [3,5,14]. Though it may not be apparent at first, we can also exploit these transitive relations to reduce manual pairwise annotation effort. To illustrate the idea, let us consider few plausible scenarios as shown in Figure 1.

- In camera pair 1-2 and 1-3, if we know from human labeling person that pairs $P_1^1 - P_2^1$ and $P_1^1 - P_3^2$ are positive matches, then from transitivity we can directly infer that $P_2^1$ and $P_3^2$ also have same identity.
- Similarly, if we have labels of $P_1^1 - P_2^1$ (+ve) and $P_1^1 - P_3^1$ (-ve), we can infer that $P_2^1 - P_3^1$ is negative.
- However, given that we already know labels of $P_1^1 - P_2^2$ (-ve) and $P_1^1 - P_3^1$ (-ve), we still cannot conclude anything about pair $P_2^2 - P_3^1$.

So, from the examples above we can make a simple observation, i.e., if we don't ask human for the label of the third pair/s in the first two cases described above, required labeling effort will be considerably reduced. However, this seemingly simple strategy implicitly makes an invalid assumption that we already have access to the pair-labels from human. Also, note that, if we arbitrarily choose subsets of pairs for labeling there is no guarantee that we will be able to take advantage of pairwise-relations as we will end up frequently in situations like the third scenario (occurrence probability of this scenario is significantly higher than the other two). So, in order to actually reduce annotation effort using this transitivity based approach, we have to choose image pairs in a judicious manner.

Towards this objective, in this work, we first formulate this pair subset selection as a combinatorial optimization problem on edge-weighted $k$-partite graph. This combinatorial optimization can be represented as a binary integer program which we can solve exactly for smaller datasets using standard techniques such as branch and cut [21], cutting plane algorithms [16], etc. However, as it is an NP-hard optimization problem, solving it with exact algorithms takes exponential order time and for larger datasets it becomes intractable. So, in order to scale up the proposed methodology for large camera networks, we propose two polynomial-time sub-optimal algorithms for our optimization problem. The first proposed algorithm is a pure greedy algorithm and second one is a $1/2$-approximation algorithm.

### 1.1. Main contributions

1) We propose a pairwise subset selection framework to minimize human labeling effort for person re-id in camera networks. Our method does not require us to make any assumption about the topology of the camera network or the learning algorithm. Thus, even though in this specific work we present our framework in conjunction with KISSME [11], our algorithm can be used with any supervised algorithm.
2) To cope with the 'NP' hardness of our formulated optimization problem, we propose two polynomial time algorithms for solving it sub-optimally for large networks.
3) To demonstrate the efficacy of the proposed method, we conduct extensive experiments on three benchmark multi-camera person re-id datasets. The results show that our algorithm can significantly reduce annotation effort without adversely affecting recognition performance.

## 2. Related works

**Metric Learning in Person Re-id.** Metric learning based methods focus on learning a discriminative projection which will helps to cluster similar and dissimilar pairs separately. In person re-identification literature numerous metric learning based methods have been proposed. KISSME [11] is one such popular metric learning method which uses log-likelihood ratio test to construct a Mahalanobis type distance metric. XQDA [13] uses quadratic discriminant analysis to derive the metric. LMNN [33] learns the distance metric via penalizing closeness between dissimilar samples. Various other metric learning methods are proposed in [20, 26],etc. A comprehensive survey on this topic can be found in [37].

**Scalable Person Re-id.** Authors in [1] propose a scalable re-id framework using manifold smoothing. Active learning is introduced for incremental updates in [31]. Another scalable re-id framework which incorporates human machine interaction is proposed in [32]. In [6], an entropy based selection approach is proposed for reducing manual annotation. In [17],the authors uses a dominant clustering based approach for probe relevant set selection and utilizes it for pair selection in a dynamic setting.

**Transitivity in Re-id.** Transitivity is utilized in [5] for increasing performance by re-organizing the predicted assignment matrix. The method proposed in [14] also uses similar ideas to structure a deep learning based framework.

**Budget Constrained Learning.** The problem of video analysis under budget constraints have been studied by few researchers [24, 25, 30] in the recent past, however none look into the problem of re-identification under budget constraints. Activity detection under a computational budget is considered in [29].

## 3. Proposed Method

### 3.1. $k$-Paritite Graph Based Representation

In our work, we represent any camera network as a edge weighted complete $k$-partite graph $G_k = (V, E)$ [see Figure 2]. This section describes in detail how this partite graph is constructed from a camera network consisting of $k$ cameras and total $n$ persons across all cameras.
**Vertex:** Each vertex in $G_k$ denotes a person in the camera network. To be precise, vertex $v_{k'}^i$ represents the $i$-th person from $k'$-th camera. From now on, throughout rest of the work we will use the terms 'person' and 'vertex' interchangeably.

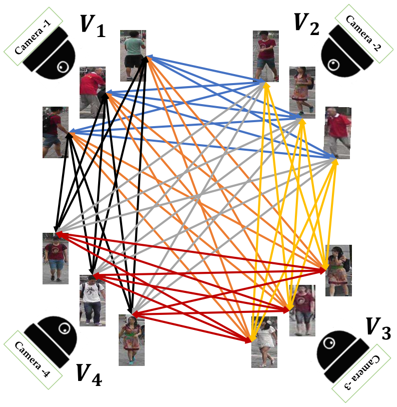


Figure 2: This figure demonstrates representation of a camera network with four cameras as a $k$-partite graph with $k$=4.

**Edge:** An edge $E^{i,j}_{k_1,k_2} = (v^i_{k_1}, v^j_{k_2})$ denotes probable correspondence between $i$-th person in camera $k_1$ and $j$-th person in camera $k_2$.

**Vertex Set Partitions:** As per our definition, the set of all the persons in a camera network forms the vertex set $V$ of $G_k$. Now in our framework, we assume the intra-camera vertices are not connected to each other, i.e., they form an independent vertex set. So, $k$ sets of vertices from each different camera form $k$ different partitions. More formally, $V = (V_1, V_2, ..., V_k)$ where $V_{k'} = \left\{v^1_{k'}, v^2_{k'}, \dots, v^{n_{k'}}_{k'}\right\}$ is the set of $n_{k'}$ persons in $k'$-th camera. So, if we have $n_1$,$n_2$,. . . ,$n_k$ persons in camera 1, camera 2, . . . , camera $k$ respectively, the cardinality of the set $V$ is

$$|V| = \sum_{i=1}^{k} |V_i| = n_1 + n_2 + ... + n_k = n \qquad (1)$$

Now, $G_k$ is a complete multipartite graph as we have probable correspondences (i.e. an weighted edge) between every pair of vertices from different partitions. So the total number of edges in the graph, can be computed as follows:

$$|E| = \sum_{\substack{\forall k_1 \in \{1,2,\dots,k\} \\ \forall k_2 \in \{1,2,\dots,k\} \\ s.t.\ k_1 < k_2}} n_{k_1} n_{k_2} \qquad (2)$$

**Edge weight:** We define our edge weight function $\mathcal{F}_w : E \to \mathbb{R}$, as follows:

$$\mathcal{F}_w\left(E^{i,j}_{k_1,k_2}\right) = \mathcal{S}\left(v^i_{k_1}, v^j_{k_2}\right) \qquad (3)$$

where $\mathcal{S}$ is a function which computes similarity or association score between two persons $v^i_{k_1}$ and $v^j_{k_2}$. It may be noted that our framework can be used with any kind of similarity measure. As we define our objective function [see Section 3.3] over non-negative edge weights, the proposed scheme will scale any negative valued similarity score into a non-negative value using the sigmoid function. In this work, we compute similarity scores between a pair of shots of two persons as follows:

$$\mathcal{S}\left(v^i_{k_1}, v^j_{k_2}\right) = \frac{1}{1 + \exp\left(\mathcal{D}(\boldsymbol{f}^i_{k_1}, \boldsymbol{f}^j_{k_2}) - \mu\right)} \qquad (4)$$

where $\boldsymbol{f}^i_{k_1}, \boldsymbol{f}^j_{k_2}$ are the feature vectors of the corresponding persons $v^{k_1}_i, v^{k_2}_j$ respectively, $\mathcal{D}$ is a distance function giving distance between two feature vectors, and $\mu$ is a threshold.

**Triangle:** Complete subgraphs (or clique) of size 3 are termed as triangle in any graph. Naturally, whenever we have three persons(vertices), $v^i_{k_1}, v^j_{k_2}, v^l_{k_3}$ from three different cameras (camera $k_1$, camera $k_2$ and camera $k_3$), they form a triangle, $T^{i,j,l}_{k_1,k_2,k_3} = \left\{v^i_{k_1}, v^j_{k_2}, v^l_{k_3}\right\}$. As we progress, we will see that triangles are the central objects around which our whole framework evolves.

### 3.2. Pair Selection as a Combinatorial Optimization

With the initial setup in place, we can now formulate the image pair selection task as an optimization problem on our graph $G_k$. Let us consider first revisiting the problem statement of the budget constrained pair-selection task.

**Problem Statement**: Given a labeling budget, $B$, and a set of training image pairs from a camera network, we have to select an optimal subset of size at most $B$ from the training set for human annotation. The notion of 'optimal subset' is *incomplete*. In the introduction of this paper, we have already seen that transitive relations defined over associations between different persons (vertices) can be utilized for labeling effort reduction. Now we give that idea a concrete shape by making some specific observations in the context of our graph $G_k$.

- For any triangle in our graph, we have a total three edges from which we can select for manual labeling.
- We may always want to select positive edges as they will contribute more towards reducing manual labeling effort because transitive inference in our graph always requires at least one positive edge.
- Based on the examples mentioned in Section 1, if we have precise information about two edges in a triangle of our graph, and one of them is a positive edge then we can deterministically infer the label of the third edge. For this reason we must always want to constrain the number of edges chosen for manual labeling in a triangle be at most two in order to respect the budget.
- As we cannot foresee the actual labels, we have to choose that pair of edges from any triangle which will maximize the probability of getting at least one positive match.
- Also, note that any edge is a part of multiple triangles in our graph, so inference propagation can occur from different directions.

With these observations in mind, our optimization problem can be stated as follows:

- *Given a complete $k$-partite graph $G_k = (V, E)$ with non-negative edge weights and an integer $B$, choose a maximum-weight set $S$ of edges from $E$ such that $G' = (V, S)$ is triangle free and $|S| \leq B$.*

**Lemma 1.** *The decision problem corresponding to the above optimization problem is NP-hard for $k$-partite graphs with $k \geq 4$. (Proof: see supplementary material)*

### 3.3. An Equivalent Binary Integer Program

We can recast our combinatorial optimization as a binary integer programing problem as follows:

$$\underset{\substack{x^{i,j}_{k_1,k_2} \\ \forall (i,j) \in \delta(k_1,k_2) \\ \forall k_1,k_2 \in \{1,\ldots,k\} \ s.t \ k_1<k_2}}{\operatorname{argmax}} \left( \sum_{\substack{k_1,k_2=1 \\ k_1<k_2}}^{k} \sum_{i,j=1}^{n_{k_1},n_{k_2}} w^{i,j}_{k_1,k_2} x^{i,j}_{k_1,k_2} \right) \tag{5}$$

$$\text{subject to: } \sum_{\substack{k_1,k_2=1 \\ k_1<k_2}}^{k} \sum_{i,j=1}^{n_{k_1},n_{k_2}} x^{i,j}_{k_1,k_2} \leq B,$$

$$\forall (i,j) \in \delta(k_1,k_2) \ \forall k_1,k_2 \in \{1,2,...,k\} \ \ s.t \ k_1 < k_2 \tag{6}$$

$$x^{i,j}_{k_1,k_2} + x^{i,l}_{k_1,k_3} + x^{j,l}_{k_2,k_3} \leq 2, \forall (i,j) \in \delta(k_1,k_2)$$
$$k_1,k_2,k_3 \in \{1,2,...,k\} s.t. \ k_1 < k_2 < k_3 \tag{7}$$

$$x^{i,j}_{k_1,k_2} \in \{0,1\}, \forall (i,j) \in \delta(k_1,k_2),$$
$$\forall k_1,k_2 \in \{1,2,...,k\} \ s.t. \ k_1 < k_2 \tag{8}$$

where, Equation (5) represents the linear objective function, which aims to maximize the total weight of the chosen subgraph. $\delta(k_1,k_2)$ denotes the edge-set between camera $k_1$ and $k_2$. Equations (6)-(8) are the constraints we have to satisfy. In the above set of equations, $x^{i,j}_{k_1,k_2}$ denotes the edge between $i$-th person in camera $k_1$ and $j$-th person in camera $k_2$. $x^{i,j}_{k_1,k_2}$'s are defined over all possible values of $i, j, k_1$ and $k_2$ as described above and together all possible $x^{i,j}_{k_1,k_2}$'s form the decision variable set. $w^{i,j}_{k_1,k_2}$'s are the weights of the corresponding edges and $B$ is our labeling budget. The first constraint (6) dictates that we can select at most $B$ number of edges. Equation (7) constrain that the subgraph formed by the selected edges be triangle free. Equation (8) denotes that optimization variable be binary, where a $1$ would indicate that an edge is chosen for manual labeling and $0$ otherwise.

### 3.4. Polynomial Time Approx.-Optimal Algorithms

In case of smaller datasets, we can easily solve our optimization problem using traditional integer programing algorithms, such as cutting plane methods [16], Branch and Cut [21] etc. These methods always provide globally optimal solutions. However, as they are exponential time algorithms, we cannot employ them for larger datasets. In order to tackle this challenge, we propose two polynomial time algorithms which drastically improve scalability.

**Algorithm 1.** This algorithm is motivated by the observation that if we make any cut on the vertex set of a graph, the set of cut crossing edges induces a triangle free subgraph. So if we can make a cut which maximizes the total weight of edges crossing the cut, then we may construct a approximately-optimal solution using those edges. In graph theory, the max-cut problem is well studied where the objective is to find such max-weight cut. As max-cut is also an NP-hard [19] problem, there is no known efficient algorithm for it. However there exists a deterministic $1/2$-approximation algorithm for max-cut [7, 22]. Our first algorithm uses this $1/2$-max cut to achieve $1/2$ approximation for our problem.

After initialization steps, Max-Cut Select algorithm constructs the subgraph $G'$ using the top $B$ heaviest edges in $E$. Then it employs the deterministic $1/2$-max cut algorithm on $G'$ to generate a cut $(S, V \setminus S)$. Finally the algorithm selects the set $T$ of edges which crosses the cut $(S, V \setminus S)$ and returns it. Below we prove that Max-Cut Select is a $1/2$-approximation algorithm.

**Algorithm 1:** Max-Cut based edge selection

```
Max-Cut Select $(G, B)$
Input  : An edge weighted graph $G$ and budget $B$
Output : $T$, a subset of edges in $G$
$E' \leftarrow B$ heaviest edges in $G.E$
$V \leftarrow G.V$
$G' \leftarrow (V, E')$
$S \leftarrow 1/2$-Max-Cut$(G')$
$T \leftarrow$ Edge set crossing the cut $(S, V \setminus S)$
return $T$
```

**Lemma 2.** *Algorithm 1 is a $1/2$-approximation algorithm for the budget constrained triangle free subgraph weight maximization problem.*

*Proof.* Let, for a given graph $G = (V, E)$ and a budget $B$, $OPT$ be the weight of the optimal solution to our problem.The algorithm returns the edge-set $T$ as a solution. We prove our lemma by showing:

1. $weight(T) \geq OPT/2$, and
2. $T$ induced subgraph (let us denote this subgraph by $G_T$) is a triangle free subgraph of $G$.

Note that $E'$ is defined as the set of the $B$ heaviest edges in $E$. So, we have,

$$OPT \leq weight(E') \tag{9}$$

Now, $T$ is the set of cut-crossing edges which we obtained after applying the deterministic $1/2$-max-cut algorithm on the $E'$ induced subgraph, $G'$. So $G_T$ is a bipartite subgraph of $G'$ and this implies $G_T$ is a bipartite subgraph of $G$. This proves our second claim that $G_T$ is triangle free. Also, from the property of $1/2$-max cut we have,

$$weight(T) \geq weight(E')/2. \tag{10}$$

Then by Equation (9) and (10), we get $weight(T) \geq OPT/2$, which proves our first claim. □

In Algorithm 1, we have used the deterministic $1/2$-approx. algorithm for the subroutine '$1/2$-Max-Cut', which cuts at least $1/2$ of the total edge weights.

**Algorithm 2.** Often in practice, simple greedy heuristics give better performance as compared to other theoretically superior algorithms. This perspective has motivated us to explore greedy strategies for our problem resulting the 'Greedy-Select' algorithm. Greedy-Select begins with an empty set $T$ and iterates over the edges in decreasing weight order. In each iteration the algorithm adds the current edge to the set $T$ if the current edge does not form any triangle with the existing edges in $T$. The algorithm terminates either when we have collected $B$ number of edges in set $T$ or we have iterated over all the edges in the graph.

**Algorithm 2:** Greedy algorithm for edge selection

```
1  Greedy-Select ($G, B$)
   Input  : An edge weighted graph $G$ and budget $B$
   Output : $T$, a subset of edges in $G$
2  $T \leftarrow \varnothing$
3  $Q \leftarrow G.E$
4  while $|T| \leq B$ and $|Q| \geq 1$ do
5      $(u, v) \leftarrow$ Extract-Max($Q$)
6      if $(u, v)$ doesn't create any triangle with the
         existing edges in $T$ then
7          $T \leftarrow T \cup (u, v)$
8      end
9  end
10 return $T$
```

**Time Complexity:** Algorithm 1 first selects $B$ heaviest edges of the graph. With sorting, this selection can be done in $\Theta(|E| \log |E|)$ time. All the other operations including the $1/2$-max cut can be done in $O(|E|)$ time. So, any sorting based implementation of Algorithm 1 will take $\Theta(|E| \log |E|)$ time. Using linear time selection algorithm instead of sorting, this can be further reduced to $O(|E|)$.

The outer loop in Algorithm 2 runs at most $|E|$ times which takes $O(|E|)$ time. Within the loop, triangle free checking (in line 6) can be done naively in $O(|E|)$ time and it is the most expensive step. So, any basic implementation of Algorithm 2 will take $O(|E|^2)$ time. This complexity can be reduced to $O(|E|^{3/2} \log |V|)$ using the merge step of merge-sort algorithm for triangle checking or even better to $O(|E|^{3/2})$ using hash tables along with 'merge' protocol.

# 4. Experimental Results

**Datasets**: To substantiate our proposed algorithm, we conduct extensive experiments on three publicly available benchmark datasets, namely WARD [18], RAID [5] and Market-1501 [36].

**Metric learning model**: In this work, we use KISS metric learning method [11] for our experiments. The reason we choose KISSME is twofold, first it is an incredibly fast method and secondly, it is still a top performing method on several datasets [9].

**Two Stage Edge Selection.** Given a budget of $B$, we use a portion of the budget $pB(0 < p < 1)$ [we used $p = 0.7$ for experiments] to select triangle free edges using our optimization problem. However, in cases where the selected edges in a triangle are both negative matches, we cannot infer about the label of the third edge and we may want to gather information about it. For this reason, after first stage of triangle free selection, we employ a greedy top selection mechanism to exhaust the rest of the budget.

**Feature representation**: To represent each person node in the graph we use 29600 dimensional LOMO features [13]. For metric learning, we project the features into 100-dimensional space using PCA.

**Performance measures**: We use Cumulative Matching Curves (CMC) to demonstrate recognition performance at a given budget. Also, for each dataset we provide labeling effort vs. recognition performance plots trade-off between the two. For each dataset, we compare the computation costs associated with the proposed approaches. We also provide the percentage of total labels and positive labels obtained, as defined below, using only $B$ manual labels.

$$\text{Total Labels in \%} = \frac{\text{\# Inferred labels + \# Manual labels}}{\text{\# Total pairs}} * 100$$

$$\text{+ve Labels in \%} = \frac{\text{\# +ve pairs in (Manual + Inferred) labels}}{\text{\# +ve pairs in the dataset}} * 100$$

**Baseline**: In this work, we use top-$B$ edge selection as the baseline strategy. For all our experiments we compare our method against this baseline.

**Similarity score computation**: We use euclidean metric as our distance function. In any on-line setting, similarity scores at any time instant can be computed using the learned metric from the previous instance.

## 4.1. WARD

WARD [18] has a total 4786 number of images of 70 people. All the images were captured by three non-overlapping cameras. Large variation of illumination poses the main challenge for this dataset. Following the protocols in existing literature [5], we use 35 persons for training and 35 for testing set. We consider two different setup for experiments as described below:

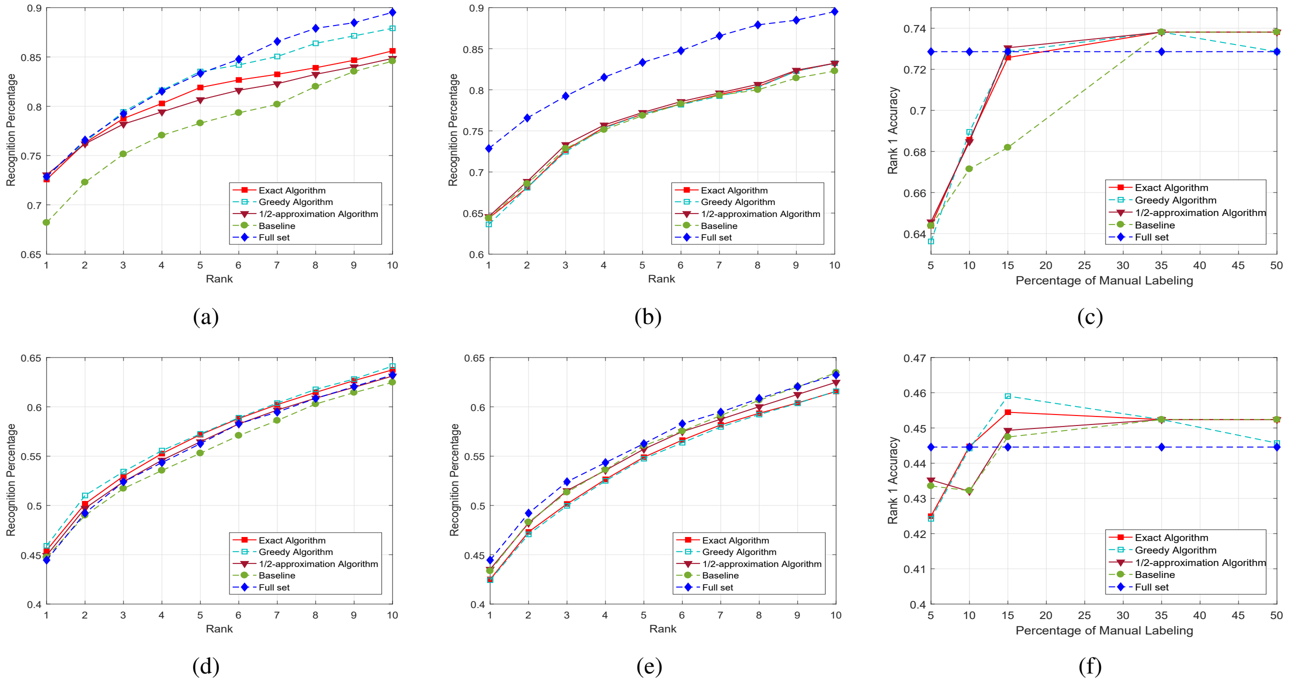


Figure 3: This figure presents the comparisons of the proposed approach with baselines on the WARD dataset. (a)-(c) are for Configuration 1 and (d)-(f) for Configuration2. (a,d) are CMC curvers with 15% manual labeling and (b,e) are CMC curves with 5% manual labeling. (c,f) presents the plot for manual labeling effort vs. Rank-1 accuracy.

1. All the 35 persons are available to each camera. We denote this setup as Configuration 1.
2. We remove randomly 4 persons from each camera and the use the rest for training. This configuration emulates more realistic surveillance scenario where each person may not get captured by each camera. This setup will be referred as Configuration 2.

We present experimental results for Config. 1 in Figure 3. The CMC curve in Figure 3a demonstrates that with only 15% manual labeling all the three variants of our proposed scheme achieve similar rank-1 accuracy as the full set. The greedy algorithm maintains this performance level for higher ranks as well. Recognition accuracy slightly deteriorates at rank 3 and beyond, in case of Exact method and $1/2$-approximation algorithm. Almost at all rank instances, the baseline method gives significantly worse performance compared to the proposed methods. Figure 3b shows performance of the selection strategies with 5% labeling budget. No methods achieve full set accuracy with this amount of label. We demonstrate percentage of manual labeling vs Rank-1 accuracy plot for Config. 1 in Figure 3c. This plot shows that the baseline cannot reach full set recognition performance with 32% manual labeling, whereas, the proposed methods reach the same by only 15% manual labeling.

For Config. 2, with 15% annotations, all the proposed approaches achieve (see Figure 3d) similar performance similar to the full set. Figure 3e shows CMC curves for 5% labeling. Even with this tiny amount of labeling, the proposed methods performs as good as the full set. The baseline also performs competitively.

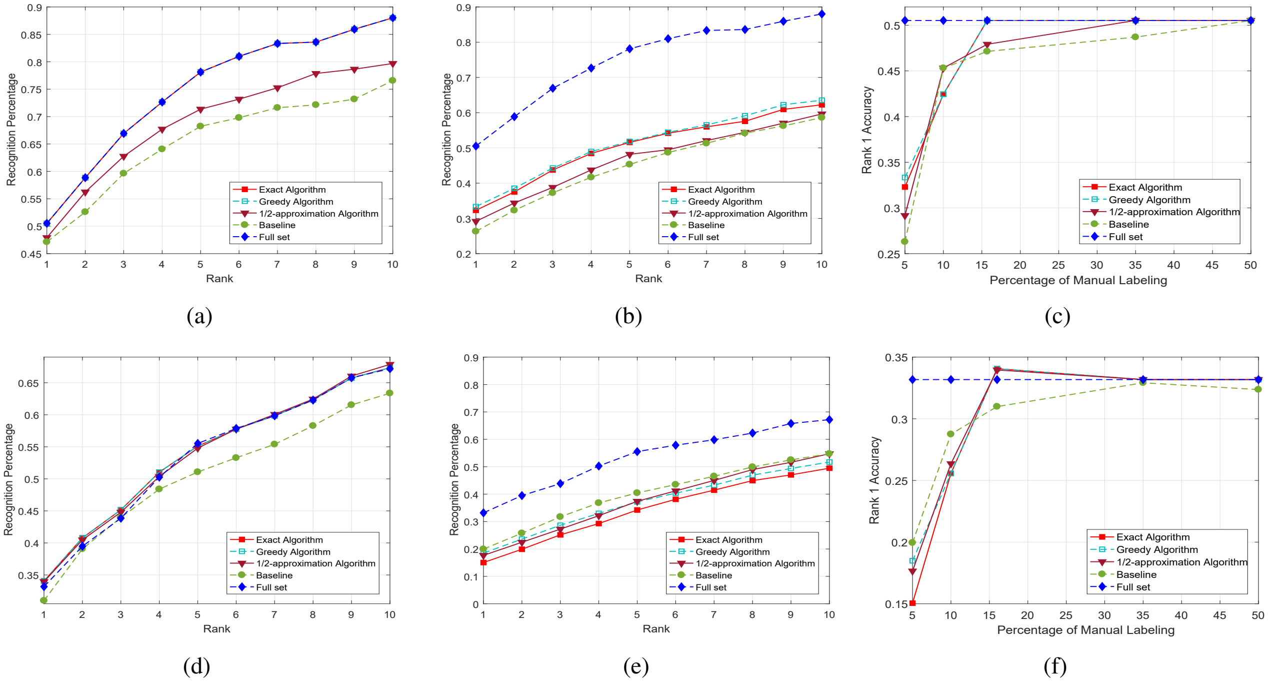
Table 1: Comparison of computation time requirements for WARD dataset. WARD-1 and 2 denotes the Configuration 1 and 2 respectively. Here, $NV$ stands for number of optimization variables and $B$ is our budget.

| Algorithm | WARD-1 ($NV = 3675, B = 15\%$) | WARD-2 ($NV = 2883, B = 15\%$) |
|---|---|---|
| Exact | 0.63 | 0.24 |
| $1/2$-Approx. | 0.017 | 0.012 |
| Greedy | 0.054 | 0.037 |

Table 1 compares computation time of the proposed approaches for the two different configurations as we described above. As we can see from the table, the $1/2$-approximation algorithm takes the least amount of time, among the all three algorithms. The problem sizes are of order $10^3$, so solving the BIP also does not demand excessive time.

## 4.2. RAID

Re-identification Across Indoor-outdoor Dataset (RAID) is a wide-area camera network dataset. RAID contains 6920 bounding boxes of 43 subjects. For our experiments, we use 41 identities which are common to each camera. We partition the dataset into 25-16 split for training and testing purpose. We consider this split to effectively create the two configurations similar to the experiments for WARD dataset, i.e., again we consider two different scenarios to demonstrate the performance of our framework under varying setup.The first configuration of our experimental setup assumes all 25



(a)

(b)

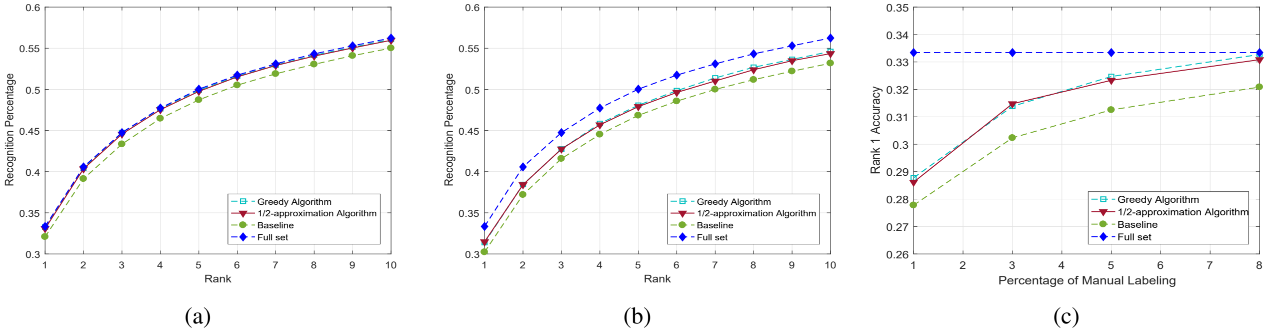
(c)

(d)

(e)

(f)

Figure 4: This figure presents the comparisons of the proposed approach with baselines on the RAID dataset. (a)-(c) are for Configuration 1 and (d)-(f) for Configuration 2. (a) and (d) are CMC curves with 15.7% and 16% manual labeling respectively. (b,e) are CMC curves with 5% manual labeling. (c,f) presents the plot for manual labeling effort vs. Rank-1 accuracy.



(a)

(b)

(c)

Figure 5: This figure presents the comparisons of the proposed approach with baselines on the Market 1501 dataset using Configuration 2. (a) and (b) are CMC curves with 8% and 3% manual labeling respectively. (c) presents the plot for manual labeling effort vs. Rank-1 accuracy.

persons are present in the field of view of every camera,while the second one again considers a more generalized setting where we remove four persons from each camera randomly and use the rest 21 persons for metric learning.

We demonstrate the experimental results for Configuration-1 in Figure 4. The key observations are as follows: With only 15.7% labeled pairs,followed by label propagation [see 4.4], both our exact and greedy algorithms achieve full set rank-1 accuracy which is 50.50% [see Figure 4a]. Their performance level matches the full set recognition rates at other ranks as well. In comparison, the 1/2-approximation algorithm gives relatively poor rank-1 recognition rate, and performance degrades with increasing rank values. The baseline method performs worse than Greedy and Exact algorithms, but provides similar rank-1 results as 1/2-approx. algorithm. Figure 4b presents the CMC curves of different approaches with 5% labeling. Though the proposed method could not achieve full set recognition performance with this meager amount of labels, it still manages to perform within around 70% of the full set performance @ rank-1. At this labeling budget, the Exact and Greedy schemes achieve nearly similar re-id rate throughout all the ranks and compared to them 1/2-approx gives slightly poor performance. The baseline method starts with an accuracy gap of 24.05% from the full set and gives similar poor performance for higher rank retrievals as well. We show the trade off between labeling effort and rank-1 accuracy in Figure 4c which highlights the fact that our

Table 2: Comparison of time (in seconds) requirements for RAID dataset. Notations are same as Table 1

| Algorithm | RAID-1 ($NV = 3750, B = 15\%$) | RAID-2 ($NV = 2646, B = 15\%$) |
|---|---|---|
| Exact | 0.54 | 0.27 |
| $1/2$-Approx. | 0.014 | 0.013 |
| Greedy | 0.055 | 0.038 |

framework is capable of obtaining almost same level of performance as the full set using only about 15% of labeled pairs.

Config. 2 is a far more challenging scenario compared to the first one. Naturally, in this setting the full set rank-1 accuracy itself drops below 35%. With only 16% labels all the proposed algorithms achieve this accuracy, which can be seen from Figure 4d. Our baseline achieves around 30.5% rank-1 accuracy and attains 63% test recognition rate at rank 10 while maintaining a considerable gap with the proposed methods throughout this range. Figure 4e shows the CMC curve with 5% manual labels. In this setting, performance of Exact algorithm gets surpassed by all the other methods and Baseline algorithm gives better performance compared to all the proposed methods.Figure 4f presents labeling budget vs rank-1 accuracy plots for Config. 2.In this dataset for Config. 2, 5-10% labeling is an extreme case studied to understand the performance degradation of our approach; from about 15% labeling, followed by label propagation[see 4.4], our method performs better than competing ones. Table 2 compares running time of the different approaches for RAID.

### 4.3. Market 1501

Market 1501 [36] is one of the biggest person re-id datasets available today. It has 32,668 images of 1501 persons taken from six cameras. We use the train-test split given in the dataset. Apart from large variations in pose and illuminations, the size of the dataset itself introduces a new level of computational challenge. For Market, the optimization problem we consider , has more than 4.3 millions variables. This is a staggeringly large optimization problem. Naturally, the problem gets intractable to be solved by any exact method. So for this dataset, we do not report any results using the Exact method. Also, we do not construct additional experimental settings as we did for WARD and RAID because there are many persons available in the dataset who are captured by only a subset of the total number of cameras.

Figure 5a demonstrates re-id performance with 8% labels. As we can observe from this plot, both of our approaches achieve full set accuracy with this amount of labeling. While with 3% labels, performance of the proposed approaches slightly degrades [see Figure 5b]. In Figure 5c, we provide the manual labeling percentage vs rank-1 accuracy graph. From all these three graphs, it can be easily observed that the proposed approach performs better than the baseline across all the conducted experiments on Market dataset.

We compare run times for Market in Table 3. It can be clearly seen that the $1/2$-approximation algorithm can be significantly faster than the greedy method. However, a point to be considered is efficient implementation of these algorithms may further improve their computational performance.

Table 3: Comparison of time requirements (in seconds) for Market 1501 dataset. Notations are same as Table 1

| Algorithm | Market ($B = 8\%$) | Market ($B = 3\%$) |
|---|---|---|
| $1/2$-Approx. | 10.59 | 4 |
| Greedy | 2100 | 595 |

### 4.4. Label Gains

Table 4 and 5 presents the total and positive labels acquired after manual labeling followed by transitive inference. Table 4 demonstrates the label gains for the budgets used in the above experiments, while Table 5 presents the amount of manual labeling required to achieve ”near-100%” positive labels. As can be seen from the tables, the proposed approach possess the ability to recover most of the labels using a meager amount of manual annotations.

Table 4: Comparison of Total Labels (Positive Labels)

| Algo. | WARD<br>15% Manual | RAID<br>15.7% Manual | Market<br>8% Manual |
|---|---|---|---|
| Exact | 43.5 (90.5) | **91.9** (**97.3**) | - |
| Greedy | **50.9** (**92.4**) | **91.9** (**97.3**) | **81.8** (**94.9**) |
| $1/2$-apx. | 42.7 (89.5) | 80.1 (94.7) | 72.7 (92.9) |
| Baseline | 35.3 (81.1) | 52.0 (82.0) | 35.9 (79.2) |

Table 5: This table presents the manual labeling required to achieve near 100% Positive Labels. Values in format Total Labels (Positive Labels).

| Algo. | WARD<br>23% Manual | RAID<br>17% Manual | Market<br>20% Manual |
|---|---|---|---|
| Exact | 85.5 (96.2) | 99.0 (100) | - |
| Greedy | 86.7 (96.2) | 99.0 (100) | 88.1 (96.6) |
| $1/2$-apx. | 80.7 (96.2) | 99.0 (100) | 82.5 (95.0) |

## 5. Conclusions

In this work, we addressed the problem of labeling reduction for person re-identification in camera networks. In pursuit of this goal, we first formulated our problem as a combinatorial optimization on $k$-partite graph. The decision version of our optimization problem is NP-complete. So to make our approach scalable, we propose two polynomial time sub-optimal algorithms. One of the proposed algorithm is $1/2$-approximation algorithm. We validated our framework by conducting experiments on three benchmark datasets and the results clearly demonstrated the efficacy of our approach. Future works can be targeted towards developing algorithms with better optimality guarantees.

**Acknowledgment.** This work was partially supported by NSF grants IIS-1316934, IIS-1619463 & CNS-1544969. We thank Victor Hill for setting up the computing infrastructure.

# References

[1] S. Bai, X. Bai, and Q. Tian. Scalable person re-identification on supervised smoothed manifold. *arXiv preprint arXiv:1703.08359*, 2017.

[2] S. Bak, E. Corvee, F. Bremond, and M. Thonnat. Multiple-shot human re-identification by mean riemannian covariance grid. In *AVSS*, 2011.

[3] A. Chakraborty, A. Das, and A. K. Roy-Chowdhury. Network consistent data association. *TPAMI*, 2016.

[4] D. S. Cheng, M. Cristani, M. Stoppa, L. Bazzani, and V. Murino. Custom pictorial structures for re-identification. In *BMVC*, 2011.

[5] A. Das, A. Chakraborty, and A. K. Roy-Chowdhury. Consistent re-identification in a camera network. In *ECCV*, 2014.

[6] A. Das, R. Panda, and A. Roy-Chowdhury. Active image pair selection for continuous person re-identification. In *ICIP*, 2015.

[7] T. F. Gonzalez. *Handbook of approximation algorithms and metaheuristics*. CRC Press, 2007.

[8] W. Hu, T. Tan, L. Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *TSMC- Part C*, 2004.

[9] S. Karanam, M. Gou, Z. Wu, A. Rates-Borras, O. Camps, and R. J. Radke. A systematic evaluation and benchmark for person re-identification: Features, metrics, and datasets. *arXiv preprint arXiv:1605.09653*, 2016.

[10] E. Kodirov, T. Xiang, Z. Fu, and S. Gong. Person re-identification by unsupervised\ell _1 graph learning. In *ECCV*, 2016.

[11] M. Koestinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof. Large scale metric learning from equivalence constraints. In *CVPR*, 2012.

[12] W. Li, X. Zhu, and S. Gong. Person re-identification by deep joint learning of multi-loss classification. *arXiv preprint arXiv:1705.04724*, 2017.

[13] S. Liao, Y. Hu, X. Zhu, and S. Z. Li. Person re-identification by local maximal occurrence representation and metric learning. In *CVPR*, 2015.

[14] J. Lin, L. Ren, J. Lu, J. Feng, and J. Zhou. Consistent-aware deep learning for person re-identification in a camera network. In *CVPR*, 2017.

[15] X. Ma, X. Zhu, S. Gong, X. Xie, J. Hu, K.-M. Lam, and Y. Zhong. Person re-identification by unsupervised video matching. *Pattern Recognition*, 2017.

[16] H. Marchand, A. Martin, R. Weismantel, and L. Wolsey. Cutting planes in integer and mixed integer programming. *Discrete Applied Mathematics*, 2002.

[17] N. Martinel, A. Das, C. Micheloni, and A. K. Roy-Chowdhury. Temporal model adaptation for person re-identification. In *ECCV*, 2016.

[18] N. Martinel and C. Micheloni. Re-identify people in wide area camera network. In *CVPRW*, 2012.

[19] R. G. Michael and S. J. David. Computers and intractability: a guide to the theory of np-completeness. *WH Free. Co., San Fr*, 1979.

[20] A. Mignon and F. Jurie. Pcca: A new approach for distance learning from sparse pairwise constraints. In *CVPR*, 2012.

[21] J. E. Mitchell. Branch-and-cut algorithms for combinatorial optimization problems. *Handbook of applied optimization*, 2002.

[22] M. Mitzenmacher and E. Upfal. *Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis*. Cambridge university press, 2017.

[23] B. T. Morris and M. M. Trivedi. A survey of vision-based trajectory learning and analysis for surveillance. *TCSVT*, 2008.

[24] F. Nan and V. Saligrama. Adaptive classification for prediction under a budget. *arXiv preprint arXiv:1705.10194*, 2017.

[25] F. Nan, J. Wang, and V. Saligrama. Pruning random forests for prediction on a budget. In *NIPS*, 2016.

[26] S. Pedagadi, J. Orwell, S. Velastin, and B. Boghossian. Local fisher discriminant analysis for pedestrian re-identification. In *CVPR*, 2013.

[27] A. K. Roy-Chowdhury and B. Song. Camera networks: The acquisition and analysis of videos over wide areas. *Synthesis Lectures on Computer Vision*, 2012.

[28] C. Su, S. Zhang, J. Xing, W. Gao, and Q. Tian. Deep attributes driven multi-camera person re-identification. In *ECCV*, 2016.

[29] Y.-C. Su and K. Grauman. Leaving some stones unturned: dynamic feature prioritization for activity detection in streaming video. In *ECCV*, 2016.

[30] S. Vijayanarasimhan, P. Jain, and K. Grauman. Far-sighted active learning on a budget for image and video recognition. In *CVPR*, 2010.

[31] H. Wang, S. Gong, and T. Xiang. Highly efficient regression for scalable person re-identification. *arXiv preprint arXiv:1612.01341*, 2016.

[32] H. Wang, S. Gong, X. Zhu, and T. Xiang. Human-in-the-loop person re-identification. In *ECCV*, 2016.

[33] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *JMLR*, 2009.

[34] L. Zhang, T. Xiang, and S. Gong. Learning a discriminative null space for person re-identification. In *CVPR*, 2016.

[35] R. Zhao, W. Ouyang, and X. Wang. Unsupervised salience learning for person re-identification. In *CVPR*, 2013.

[36] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian. Scalable person re-identification: A benchmark. In *ICCV*, 2015.

[37] L. Zheng, Y. Yang, and A. G. Hauptmann. Person re-identification: Past, present and future. *arXiv preprint arXiv:1610.02984*, 2016.