

Cross-view Asymmetric Metric Learning for Unsupervised Person Re-identification

Hong-Xing Yu^{1,5}, Ancong Wu², and Wei-Shi Zheng^{1,3,4}

¹School of Data and Computer Science, Sun Yat-sen University, China

²School of Electronics and Information Technology, Sun Yat-sen University, China

³Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, China

⁴Collaborative Innovation Center of High Performance Computing, NUDT, China

⁵Guangdong Key Laboratory of Big Data Analysis and Processing, Guangzhou, China

xkoven@gmail.com, wuancong@mail2.sysu.edu.cn, wszheng@ieee.org

Abstract

While metric learning is important for Person re-identification (RE-ID), a significant problem in visual surveillance for cross-view pedestrian matching, existing metric models for RE-ID are mostly based on supervised learning that requires quantities of labeled samples in all pairs of camera views for training. However, this limits their scalabilities to realistic applications, in which a large amount of data over multiple disjoint camera views is available but not labelled. To overcome the problem, we propose unsupervised asymmetric metric learning for unsupervised RE-ID. Our model aims to learn an asymmetric metric, i.e., specific projection for each view, based on asymmetric clustering on cross-view person images. Our model finds a shared space where view-specific bias is alleviated and thus better matching performance can be achieved. Extensive experiments have been conducted on a baseline and five large-scale RE-ID datasets to demonstrate the effectiveness of the proposed model. Through the comparison, we show that our model works much more suitable for unsupervised RE-ID compared to classical unsupervised metric learning models. We also compare with existing unsupervised RE-ID methods, and our model outperforms them with notable margins. Specifically, we report the results on large-scale unlabelled RE-ID dataset, which is important but unfortunately less concerned in literatures.

1. Introduction

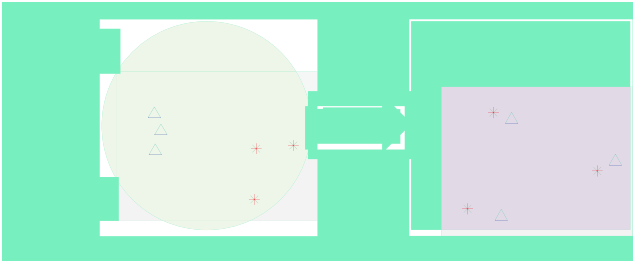
Person re-identification (RE-ID) is a challenging problem focusing on pedestrian matching and ranking across non-overlapping camera views. It remains an open problem

although it has received considerable exploration recently, in consideration of its potential significance in security applications, especially in the case of video surveillance. It has not been solved yet principally because of the dramatic intra-class variation and the high inter-class similarity. Existing attempts mainly focus on learning to extract robust and discriminative representations [33, 23, 19], and learning matching functions or metrics [38, 14, 18, 22, 19, 20, 26] in a supervised manner. Recently, deep learning has been adopted to RE-ID community [1, 32, 28, 27] and has gained promising results.

However, supervised strategies are intrinsically limited due to the requirement of manually labeled cross-view training data, which is very expensive [31]. In the context of RE-ID, the limitation is even pronounced because (1) manually labeling may not be reliable with a huge number of images to be checked across multiple camera views, and more importantly (2) the astronomical cost of time and money is prohibitive to label the overwhelming amount of data across disjoint camera views. Therefore, in reality supervised methods would be restricted when applied to a new scenario with a huge number of unlabeled data.

To directly make full use of the cheap and valuable unlabeled data, some existing efforts on exploring unsupervised strategies [8, 35, 29, 13, 21, 24, 30, 12] have been reported, but they are still not very satisfactory. One of the main reasons is that without the help of labeled data, it is rather difficult to model the dramatic variances across camera views, such as the variances of illumination and occlusion conditions. Such variances lead to view-specific interference/bias which can be very disturbing in finding what is more distinguishable in matching people across views (see Figure 1). In particular, existing unsupervised models treat the samples from different views in the same manner, and thus the effects of view-specific bias could be overlooked.

Corresponding author



achieved. On the other hand, as for the means to learn a metric or a transformation, existing unsupervised methods for RE-ID rarely consider clustering while we introduce an asymmetric metric clustering to characterize data in the learned space. While the methods proposed in [4, 2, 3] could also learn view-specific mappings, they are supervised methods and more importantly cannot be generalized to handle unsupervised RE-ID.

Apart from our model, there have been some clustering-based metric learning models [34, 25]. However, to our best knowledge, there is no such attempt in RE-ID community before. This is potentially because clustering is more susceptible to view-specific interference and thus data points from the same view are more inclined to be clustered together, instead of those of a specific person across views. Fortunately, by formulating asymmetric learning and further limiting the discrepancy between view-specific transforms, this problem can be alleviated in our model. Therefore, our model is essentially different from these models not only in formulation but also in that our model is able to better deal with cross-view matching problem by treating each view asymmetrically. We will discuss the differences between our model and the existing ones in detail in Sec. 4.3.

3. Methodology

3.1. Problem Formulation

Under a conventional RE-ID setting, suppose we have a surveillance camera network that consists of V cameras, from each of which we have collected N_p ($p = 1, \dots, V$) images and thus there are $N = N_1 + \dots + N_V$ images in total as training samples.

Let $\mathbf{X} = [x_1^1, \dots, x_{N_1}^1, \dots, x_1^V, \dots, x_{N_V}^V] \in \mathbb{R}^{M \times N}$ denote the training set, with each column x_i^p ($i = 1, \dots, N_p; p = 1, \dots, V$) corresponding to an M -dimensional representation of the i -th image from the p -th camera view. Our goal is to learn V mappings i.e., $\mathbf{U}^1, \dots, \mathbf{U}^V$, where $\mathbf{U}^p \in \mathbb{R}^{M \times T}$ ($p = 1, \dots, V$), corresponding to each camera view, and thus we can project the original representation x_i^p from the original space \mathbb{R}^M into a shared space \mathbb{R}^T in order to alleviate the view-specific interference.

3.2. Modelling

Now we are looking for some transformations to map our data into a shared space where we can better separate the images of one person from those of different persons. Naturally, this goal can be achieved by narrowing intra-class discrepancy and meanwhile pulling the centers of all classes away from each other. In an unsupervised scenario, however, we have no labeled data to tell our model how it can exactly distinguish one person from another who has a

confusingly similar appearance with him. Therefore, it is acceptable to relax the original idea: we focus on gathering similar person images together, and hence separating relatively dissimilar ones. Such goal can be modelled by minimizing an objective function like that of k -means clustering [10]:

$$\min_{\mathbf{U}} F_{\text{intra}} = \sum_{k=1}^K \sum_{i \in C_k} \|\mathbf{U}^T x_i - c_k\|^2, \quad (3)$$

where K is the number of clusters, c_k denotes the centroid of the k -th cluster and $C_k = \{i | \mathbf{U}^T x_i \in \text{cluster } k\}$.

However, clustering results may be affected extremely by view-specific bias when applied in cross-view problems. In the context of RE-ID, the feature distortion could be view-sensitive due to view-specific interference like different lighting conditions and occlusions [4]. Such interference might be disturbing or even dominating in searching the similar person images across views during clustering procedure. To address this cross-view problem, we learn specific projection for each view rather than a universal one to explicitly model the effect of view-specific interference and to alleviate it. Therefore, the idea can be further formulated by minimizing an objective function below:

$$\begin{aligned} \min_{\mathbf{U}^1, \dots, \mathbf{U}^V} F_{\text{intra}} &= \sum_{k=1}^K \sum_{i \in C_k} \|\mathbf{U}^{p^T} x_i^p - c_k\|^2 \\ \text{s.t.} \quad &\mathbf{U}^{p^T} \mathbf{P} \mathbf{U}^p = \mathbf{I} \quad (p = 1, \dots, V), \end{aligned} \quad (4)$$

where the notation is similar to Eq. (3), with p denotes the view index, $\mathbf{P} = \mathbf{X}^p \mathbf{X}^{p^T} / N_p + \mathbf{I}$ and \mathbf{I} represents the identity matrix which avoids singularity of the covariance matrix. The transformation \mathbf{U}^p that corresponds to each instance x_i^p is determined by the camera view which x_i^p comes from. The quasi-orthogonal constraints on \mathbf{U}^p ensure that the model will not simply give zero matrices. By combining the asymmetric metric learning, we actually realize an asymmetric metric clustering on RE-ID data across camera views.

Intuitively, if we minimize this objective function directly, \mathbf{U}^p will largely depend on the data distribution from the p -th view. Now that there is specific bias on each view, any \mathbf{U}^p and \mathbf{U}^q could be arbitrarily different. This result is very natural, but large inconsistencies among the learned transformations are not what we exactly expect, because the transformations are with respect to person images from different views: they are inherently correlated and homogeneous. More critically, largely different projection basis pairs would fail to capture the discriminative nature of cross-view images, producing an even worse matching result.

Hence, to strike a balance between the ability to capture discriminative nature and the capability to alleviate view-specific bias, we embed a cross-view consistency regularization term into our objective function. And then, in con-

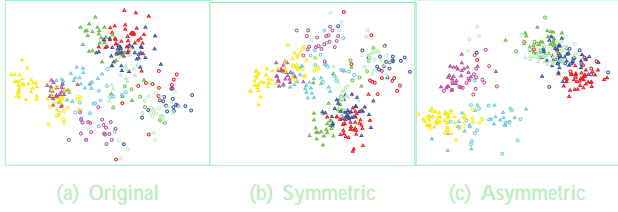


Figure 2. Illustration of how symmetric and asymmetric metric clustering structure data using our method for the unsupervised RE-ID problem. The samples are from the SYSU dataset [4]. We performed PCA for visualization. One shape (triangle or circle) stands for samples from one view, while one color indicates samples of one person. (a) Original distribution (b) distribution in the common space learned by symmetric metric clustering (c) distribution in the shared space learned by asymmetric metric clustering. (Best viewed in color)

sideration of better tractability, we divide the intra-class term by its scale N , so that the regulating parameter would not be sensitive to the number of training samples. Thus, our optimization task becomes

$$\begin{aligned} \min_{\mathbf{U}^1, \dots, \mathbf{U}^V} F_{\text{obj}} &= \frac{1}{N} F_{\text{intra}} + F_{\text{consistency}} \\ &= \frac{1}{N} \sum_{k=1}^K \sum_{i \in C_k} \|\mathbf{U}^{p^T} \mathbf{x}_i^p - \mathbf{c}_k\|^2 + \sum_{p=q}^V \|\mathbf{U}^p - \mathbf{U}^q\|_F^2 \\ \text{s.t. } \mathbf{U}^{p^T} \mathbf{U}^p &= \mathbf{I} \quad (p = 1, \dots, V), \end{aligned} \quad (5)$$

where $\|\cdot\|_F$ is the cross-view regularizer and $\|\cdot\|_F$ denotes the Frobenius norm of a matrix. We call the above model the Clustering-based Asymmetric Metric Learning (CAMEL).

To illustrate the differences between symmetric and asymmetric metric clustering in structuring data in the RE-ID problem, we further show the data distributions in Figure 2. We can observe from Figure 2 that the view-specific bias is obvious in the original space: triangles in the upper left and circles in the lower right. In the common space learned by symmetric metric clustering, the bias is still obvious. In contrast, in the shared space learned by asymmetric metric clustering, the bias is alleviated and thus the data is better characterized according to the identities of the persons, i.e., samples of one person (one color) gather together into a cluster.²

3.3. Optimization

For convenience, we denote $\mathbf{y}_i = \mathbf{U}^{p^T} \mathbf{x}_i^p$. Then we have $\mathbf{Y} \in \mathbb{R}^{T \times N}$, where each column \mathbf{y}_i corresponds to the projected new representation of that from \mathbf{X} . For optimization, we rewrite our objective function in a more compact form.

²More distribution illustrations for gradual stages of CAMEL can be found in the supplementary.

The first term can be rewritten as follow [6]:

$$\frac{1}{N} \sum_{k=1}^K \sum_{i \in C_k} \|\mathbf{y}_i - \mathbf{c}_k\|^2 = \frac{1}{N} [\text{Tr}(\mathbf{Y}^T \mathbf{Y}) - \text{Tr}(\mathbf{H}^T \mathbf{Y}^T \mathbf{Y} \mathbf{H})], \quad (6)$$

where

$$\mathbf{H} = [\mathbf{h}_1, \dots, \mathbf{h}_K], \quad \mathbf{h}_k^T \mathbf{h}_l = \begin{cases} 0 & k \neq l \\ 1 & k = l \end{cases} \quad (7)$$

$$\mathbf{h}_k = [0, \dots, 0, 1, \dots, 1, 0, \dots, 0, 1, \dots]^T / \sqrt{n_k} \quad (8)$$

is an indicator vector with the i -th entry corresponding to the instance \mathbf{y}_i , indicating that \mathbf{y}_i is in the k -th cluster if the corresponding entry does not equal zero. Then we construct

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^1 & \dots & \mathbf{x}_{N_1}^1 & 0 & \dots & 0 & \dots & 0 \\ 0 & \dots & 0 & \mathbf{x}_1^2 & \dots & \mathbf{x}_{N_2}^2 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & 0 & \dots & 0 & \dots & \mathbf{x}_{N_V}^V \end{bmatrix} \quad (9)$$

$$\mathbf{U} = [\mathbf{U}^1, \dots, \mathbf{U}^V]^T, \quad (10)$$

so that

$$\mathbf{Y} = \mathbf{U}^T \mathbf{X}, \quad (11)$$

and thus Eq. (6) becomes

$$\begin{aligned} \frac{1}{N} \sum_{k=1}^K \sum_{i \in C_k} \|\mathbf{y}_i - \mathbf{c}_k\|^2 &= \frac{1}{N} \text{Tr}(\mathbf{X}^T \mathbf{U} \mathbf{U}^T \mathbf{X}) - \frac{1}{N} \text{Tr}(\mathbf{H}^T \mathbf{X}^T \mathbf{U} \mathbf{U}^T \mathbf{X} \mathbf{H}). \end{aligned} \quad (12)$$

As for the second term, we can also rewrite it as follow:

$$\sum_{p=q}^V \|\mathbf{U}^p - \mathbf{U}^q\|_F^2 = \text{Tr}(\mathbf{U}^T \mathbf{D} \mathbf{U}), \quad (13)$$

where

$$\mathbf{D} = \begin{bmatrix} (V-1)\mathbf{I} & -\mathbf{I} & -\mathbf{I} & \dots & -\mathbf{I} \\ -\mathbf{I} & (V-1)\mathbf{I} & -\mathbf{I} & \dots & -\mathbf{I} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\mathbf{I} & -\mathbf{I} & -\mathbf{I} & \dots & (V-1)\mathbf{I} \end{bmatrix}. \quad (14)$$

Then, it is reasonable to relax the constraints

$$\mathbf{U}^{p^T} \mathbf{U}^p = \mathbf{I} \quad (p = 1, \dots, V) \quad (15)$$

to

$$\sum_{p=1}^V \mathbf{U}^{p^T} \mathbf{U}^p = \mathbf{U}^T \mathbf{U} = \mathbf{V} \mathbf{I}, \quad (16)$$

where $\mathbf{V} = \text{diag}(1, \dots, 1)$ because what we expect is to prevent each \mathbf{U}^p from shrinking to a zero matrix. The relaxed version of constraints is able to satisfy our need, and it bypasses trivial computations.

By now we can rewrite our optimization task as follow:

$$\begin{aligned} \min_{\mathbf{U}} F_{\text{obj}} = & \frac{1}{N} \text{Tr}(\mathbf{X}^T \mathbf{U} \mathbf{U}^T \mathbf{X}) + \text{Tr}(\mathbf{U}^T \mathbf{D} \mathbf{U}) \\ & - \frac{1}{N} \text{Tr}(\mathbf{H}^T \mathbf{X}^T \mathbf{U} \mathbf{U}^T \mathbf{X} \mathbf{H}) \\ \text{s.t. } & \mathbf{U}^T \mathbf{U} = \mathbf{V} \mathbf{I}. \end{aligned} \quad (17)$$

It is easy to realize from Eq. (5) that our objective function is highly non-linear and non-convex. Fortunately, in the form of Eq. (17) we can find that once \mathbf{H} is fixed, Lagrange's method can be applied to our optimization task. And again from Eq. (5), it is exactly the objective of k-means clustering once \mathbf{U} is fixed [10]. Thus, we can adopt an alternating algorithm to solve the optimization problem. Fix \mathbf{H} and optimize \mathbf{U} . Now we see how we optimize \mathbf{U} . After fixing \mathbf{H} and applying the method of Lagrange multiplier, our optimization task (17) is transformed into an eigen-decomposition problem as follow:

$$\mathbf{G} \mathbf{u} = \lambda \mathbf{u}, \quad (18)$$

where λ is the Lagrange multiplier (and also is the eigen-value here) and

$$\mathbf{G} = -\frac{1}{N} (\mathbf{D} + \frac{1}{N} \mathbf{X} \mathbf{X}^T - \frac{1}{N} \mathbf{X} \mathbf{H} \mathbf{H}^T \mathbf{X}^T). \quad (19)$$

Then, \mathbf{U} can be obtained by solving this eigen-decomposition problem.

Fix \mathbf{U} and optimize \mathbf{H} . As for the optimization of \mathbf{H} , we can simply fix \mathbf{U} and conduct k-means clustering in the learned space. Each column of \mathbf{H} , \mathbf{h}_k , is thus constructed according to the clustering result.

Based on the analysis above, we can now propose the main algorithm of CAMEL in Algorithm 1. We set maximum iteration to 100. After obtaining \mathbf{U} , we decompose it back into $\{\mathbf{U}^1, \dots, \mathbf{U}^V\}$. The algorithm is guaranteed to convergence, as given in the following proposition:

Proposition 1. In Algorithm 1, F_{obj} is guaranteed to convergence.

Proof. In each iteration, when \mathbf{U} is fixed, if \mathbf{H} is the local minimizer, k-means remains \mathbf{H} unchanged, otherwise it seeks the local minimizer. When \mathbf{H} is fixed, \mathbf{U} has a closed-form solution which is the global minimizer. Therefore, the F_{obj} decreases step by step. As $F_{\text{obj}} \geq 0$ has a lower bound 0, it is guaranteed to convergence. \square

4. Experiments

4.1. Datasets

Since unsupervised models are more meaningful when the scale of problem is larger, our experiments were conducted on relatively big datasets except VIPeR [9] which

Algorithm 1: CAMEL

Input : $\mathbf{X}, \mathbf{K}, \epsilon = 10^{-8}$
Output: \mathbf{U}

- 1 Conduct k-means clustering with respect to each column of \mathbf{X} to initialize \mathbf{H} according to Eq. (7) and (8).
- 2 Fix \mathbf{H} and solve the eigen-decomposition problem described by Eq. (18) and (19) to construct \mathbf{U} .
- 3 while decrement of $F_{\text{obj}} > \epsilon$ & maximum iteration unreached do
 - Construct \mathbf{Y} according to Eq. (11).
 - Fix \mathbf{U} and conduct k-means clustering with respect to each column of \mathbf{Y} to update \mathbf{H} according to Eq. (7) and (8).
 - Fix \mathbf{H} and solve the eigen-decomposition problem described by Eq. (18) and (19) to update \mathbf{U} .
- 4 end

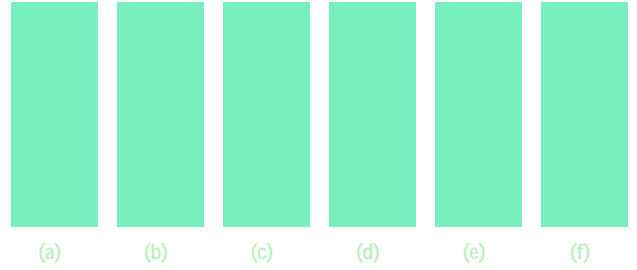


Figure 3. Samples of the datasets. Every two images in a column are from one identity across two disjoint camera views. (a) VIPeR (b) CUHK01 (c) CUHK03 (d) SYSU (e) Market (f) ExMarket. (Best viewed in color)

Dataset	VIPeR	CUHK01	CUHK03	SYSU	Market	ExMarket
# Samples	1,264	3,884	13,164	24,448	32,668	236,696
# Views	2	2	6	2	6	6

Table 1. Overview of dataset scales. “#” means “the number of”.

is small but widely used. Various degrees of view-specific bias can be observed in all these datasets (see Figure 3).

The VIPeR dataset contains 632 identities, with two images captured from two camera views of each identity.

The CUHK01 dataset [16] contains 3,884 images of 971 identities captured from two disjoint views. There are two images of every identity from each view.

The CUHK03 dataset [17] contains 13,164 images of 1,360 pedestrians captured from six surveillance camera views. Besides hand-cropped images, samples detected by a state-of-the-art pedestrian detector are provided.

The SYSU dataset [4] includes 24,448 RGB images of 502 persons under two surveillance cameras. One camera view mainly captured the frontal or back views of persons, while the other observed mostly the side views.

The Market-1501 dataset [37] (Market) contains 32,668 images of 1,501 pedestrians, each of which was captured by at most six cameras. All of the images were cropped by a pedestrian detector. There are some bad-detected samples

in this datasets as distractors as well.

The ExMarket dataset³. In order to evaluate unsupervised RE-ID methods on even larger scale, which is more realistic, we further combined the MARS dataset [36] with Market. MARS is a video-based RE-ID dataset which contains 20,715 tracklets of 1,261 pedestrians. All the identities from MARS are of a subset of those from Market. We then took 20% frames (each one in every five successive frames) from the tracklets and combined them with Market to obtain an extended version of Market (ExMarket). The imbalance between the numbers of samples from the 1,261 persons and other 240 persons makes this dataset more challenging and realistic. There are 236,696 images in ExMarket in total, and 112,351 images of them are of training set. A brief overview of the dataset scales can be found in Table 1.

4.2. Settings

Experimental protocols: A widely adopted protocol was followed on VIPeR in our experiments [19], i.e., randomly dividing the 632 pairs of images into two halves, one of which was used as training set and the other as testing set. This procedure was repeated 10 times to offer average performance. Only single-shot experiments were conducted.

The experimental protocol for CUHK01 was the same as that in [19]. We randomly selected 485 persons as training set and the other 486 ones as testing set. The evaluating procedure was repeated 10 times. Both multi-shot and single-shot settings were conducted.

The CUHK03 dataset was provided together with its recommended evaluating protocol [17]. We followed the provided protocol, where images of 1,160 persons were chosen as training set, images of another 100 persons as validation set and the remainders as testing set. This procedure was repeated 20 times. In our experiments, detected samples were adopted since they are closer to real-world settings. Both multi-shot and single-shot experiments were conducted.

As for the SYSU dataset, we randomly picked 251 pedestrians' images as training set and the others as testing set. In the testing stage, we basically followed the protocol as in [4]. That is, we randomly chose one and three images of each pedestrian as gallery for single-shot and multi-shot experiments, respectively. We repeated the testing procedure by 10 times.

Market is somewhat different from others. The evaluation protocol was also provided along with the data [37]. Since the images of one person came from at most six views, single-shot experiments were not suitable. Instead, multi-shot experiments were conducted and both cumulative matching characteristic (CMC) and mean average precision (MAP) were adopted for evaluation [37]. The protocol of ExMarket was identical to that of Market since the

identities were completely the same as we mentioned above.

Data representation: In our experiments we used the deep-learning-based JSTL feature proposed in [32]. We implemented it using the 56-layer ResNet [11], which produced 64-D features. The original JSTL was adopted to our implementation to extract features on SYSU, Market and ExMarket. Note that the training set of the original JSTL contained VIPeR, CUHK01 and CUHK03, violating the unsupervised setting. So we trained a new JSTL model without VIPeR in its training set to extract features on VIPeR. The similar procedures were done for CUHK01 and CUHK03.

Parameters: We set λ , the cross-view consistency regularizer, to 0.01. We also evaluated the situation when λ goes to infinite, i.e., the symmetric version of our model in Sec. 4.4, to show how important the asymmetric modelling is.

Regarding the parameter T which is the feature dimension after the transformation learned by CAMEL, we set T equal to original feature dimension i.e., 64, for simplicity. In our experiments, we found that CAMEL can align data distributions across camera views even without performing any further dimension reduction. This may be due to the fact that, unlike conventional subspace learning models, the transformations learned by CAMEL are view-specific for different camera views and always non-orthogonal. Hence, the learned view-specific transformations can already reduce the discrepancy between the data distributions of different camera views.

As for K , we found that our model was not sensitive to K when $N \gg K$ and K was not too small (see Sec. 4.4), so we set $K = 500$. These parameters were fixed for all datasets.

4.3. Comparison

Unsupervised models are more significant when applied on larger datasets. In order to make comprehensive and fair comparisons, in this section we compare CAMEL with the most comparable unsupervised models on six datasets with their scale orders varying from hundreds to hundreds of thousands. We show the comparative results measured by the rank-1 accuracies of CMC and MAP (%) in Table 2.

Comparison to Related Unsupervised RE-ID Models. In this subsection we compare CAMEL with the sparse dictionary learning model (denoted as Dic) [13], sparse representation learning model ISR [21], kernel subspace learning model RKSL [30] and sparse auto-encoder (SAE) [15, 5]. We tried several sets of parameters for them, and report the best ones. We also adopt the Euclidean distance which is adopted in the original JSTL paper [32] as a baseline (denoted as JSTL).

From Table 2 we can observe that CAMEL outperforms other models on all the datasets on both settings. In addition, we can further see from Figure 4 that CAMEL outperforms other models at any rank. One of the main reasons

³Demo code for the model and the ExMarket dataset can be found on <http://i see. sysu. edu. cn/project/CAMEL. html>.

Dataset	VIPeR	CUHK01	CUHK03	SYSU	Market	ExMarket
Setting	SS	SS/MS	SS/MS	SS/MS	MS	MS
Dic [13]	29.9	49.3/52.9	27.4/36.5	21.3/28.6	50.2(22.7)	52.2(21.2)
ISR [21]	27.5	53.2/55.7	31.1/38.5	23.2/33.8	40.3(14.3)	-
RKSL [30]	25.8	45.4/50.1	25.8/34.8	17.6/23.0	34.0(11.0)	-
SAE [15]	20.7	45.3/49.9	21.2/30.5	18.0/24.2	42.4(16.2)	44.0(15.1)
JSTL [32]	25.7	46.3/50.6	24.7/33.2	19.9/25.6	44.7(18.4)	46.4(16.7)
AML [34]	23.1	46.8/51.1	22.2/31.4	20.9/26.4	44.7(18.4)	46.2(16.2)
UsNCA [25]	24.3	47.0/51.7	19.8/29.6	21.1/27.2	45.2(18.9)	-
CAMEL	30.9	57.3/61.9	31.9/39.4	30.8/36.8	54.5(26.3)	55.9(23.9)

Table 2. Comparative results of unsupervised models on the six datasets, measured by rank-1 accuracies and MAP (%). “-” means prohibitive time consumption due to time complexities of the models. “SS” represents single-shot setting and “MS” represents multi-shot setting. For Market and ExMarket, MAP is also provided in the parentheses due to the requirement in the protocol [37]. Such a format is also applied in the following tables.

is that the view-specific interference is noticeable in these datasets. For example, we can see in Figure 3(b) that on CUHK01, the changes of illumination are extremely severe and even human beings may have difficulties in recognizing the identities in those images across views. This impedes other symmetric models from achieving higher accuracies, because they potentially hold an assumption that the invariant and discriminative information can be retained and exploited through a universal transformation for all views. But CAMEL relaxes this assumption by learning an asymmetric metric and then can outperform other models significantly. In Sec. 4.4 we will see the performance of CAMEL would drop much when it degrades to a symmetric model.

Comparison to Clustering-based Metric Learning Models. In this subsection we compare CAMEL with a typical model AML [34] and a recently proposed model UsNCA [25]. We can see from Fig. 4 and Table 2 that compared to them, CAMEL achieves noticeable improvements on all the six datasets. One of the major reasons is that they do not consider the view-specific bias which can be very disturbing in clustering, making them unsuitable for RE-ID problem. In comparison, CAMEL alleviates such disturbances by asymmetrically modelling. This factor contributes to the much better performance of CAMEL.

Comparison to the State-of-the-Art. In the last subsections, we compared with existing unsupervised RE-ID methods using the same features. In this part, we also compare with the results reported in literatures. Note that most existing unsupervised RE-ID methods have not been evaluated on large datasets like CUHK03, SYSU, or Market, so Table 3 only reports the comparative results on VIPeR and CUHK01. We additionally compared existing unsupervised RE-ID models, including the hand-craft-feature-based SDALF [8] and CPS [7], the transfer-learning-based UDML [24], graph-learning-based model (denoted as GL) [12], and local-saliency-learning-based GTS [29] and SDC [35]. We

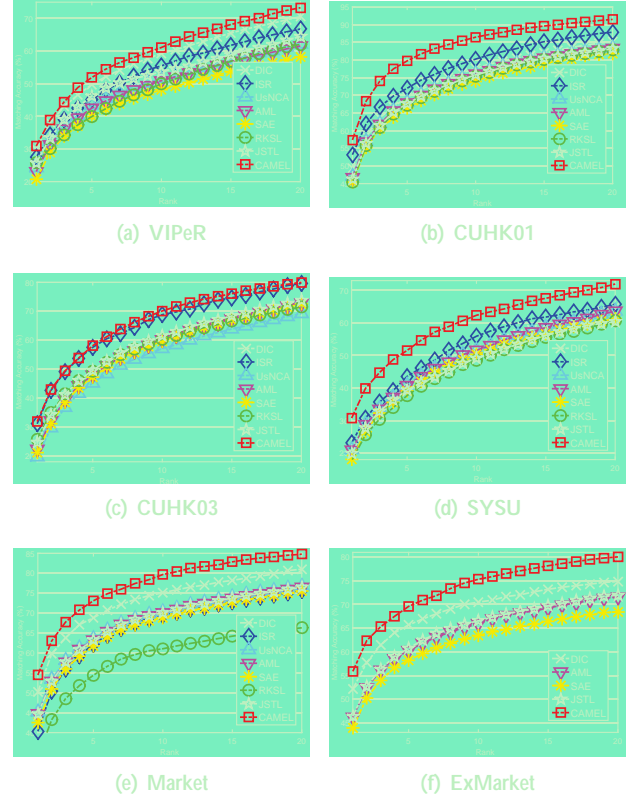


Figure 4. CMC curves. For CUHK01, CUHK03 and SYSU, we take the results under single-shot setting as examples. Similar patterns can be observed on multi-shot setting.

Model	SDALF [8]	CPS [7]	UDML [24]	GL [12]	GTS [29]	SDC [35]	CAMEL
VIPeR	19.9	22.0	31.5	33.5	25.2	25.8	30.9
CUHK01	9.9	-	27.1	41.0	-	26.6	57.3

Table 3. Results compared to the state-of-the-art reported in literatures, measured by rank-1 accuracies (%). “-” means no reported result.

can observe from Table 3 that our model CAMEL can outperform the state-of-the-art by large margins on CUHK01. **Comparison to Supervised Models.** Finally, in order to see how well CAMEL can approximate the performance of supervised RE-ID, we additionally compare CAMEL with its supervised version (denoted as CAMEL_s) which is easily derived by substituting the clustering results by true labels, and three standard supervised models, including the widely used KISSME [14], XQDA [19], the asymmetric distance model CVDCA [4]. The results are shown in Table 4. We can see that CAMEL_s outperforms CAMEL by various degrees, indicating that label information can further improve CAMEL’s performance. Also from Table 4, we notice that CAMEL can be comparable to other standard supervised models on some datasets like CUHK01, and even outper-

