Landmarks-based Kernelized Subspace Alignment for Unsupervised Domain Adaptation

Rahaf Aljundi[1], Rémi Emonet[1], Damien Muselet[1], Marc Sebban[1]
[1]LaHC, UMR CNRS 5516, Université Jean-Monnet, F-42000, Saint-Etienne, France.

Domain adaptation (DA) has gained a lot of success in the recent years in computer vision to deal with situations where the learning process has to transfer knowledge from a source to a target domain. In this paper, we introduce a novel unsupervised DA approach based on both subspace alignment and selection of landmarks similarly distributed between the two domains. Those landmarks are selected so as to reduce the discrepancy between the domains and then are used to non linearly project the data in the same space where an efficient subspace alignment (in closed-form) is performed. We carry out a large experimental comparison in visual domain adaptation showing that our new method outperforms the most recent unsupervised DA approaches.

Task and Approach

Source S and target T points are supposed to be respectively drawn from a source distribution D_S and a target distribution D_T. Domain adaptation supposes that the source and target distributions are not identical but that if we have a set of labels L_S for the source examples, they can be used to learn a classifier that is suitable for the target domain (Figure 1).

Figure 1: Example of distribution shift between images from two datasets. First row, some bike helmets from the Amazon subset and second row from the webcam subset. These 2 subsets are from the Office dataset.

In this context, subspace alignment-based DA methods have attracted a lot of interest [1]. In [1], Fernando et al. optimize a single linear mapping function that directly aligns the source and target subspaces. This new method has shown to be not only better than the state of the art but also computable in closed form. However, it assumes that the shift between the two distributions can be corrected by a linear transformation, that is a strong assumption that can be challenged in many real world applications. Moreover, it assumes that all source and target examples are necessary to proceed to the adaptation, while in most of the cases only a subset of source data are distributed similarly to the target domain and vice versa.

To overcome these drawbacks, our approach combines two simple ideas: First, it projects both source and target examples in a common subspace w.r.t. some well selected landmarks. Then, it performs a subspace alignment between the two domains. After selecting landmarks among $S \cup T$, all points in S and T are projected using a Gaussian kernel on the selected landmarks, leading to new representations K_S and K_T for the source and target points. The new representation is then used to compute a mapping using a subspace alignment approach. Compared to [1], our two-step approach remains fast and easy to implement while improving the accuracy by capturing non-linearity. Algorithm 1 gives the complete pseudo code.

Landmark selection Algorithm 2 sums up the landmark selection process. Each point from $S \cup T$ is considered as a candidate landmark. For each candidate c, we consider multiple scales s. If, for any of these scale, the overlap between the source and target distributions KV_S and KV_T w.r.t. c is greater than a threshold th, the candidate is promoted as a landmark.

Algorithm 1 LSSA: **L**andmarks **S**election-based **S**ubspace **A**lignment and classification.

Require: Source data S, Target data T, Source labels L_S, Threshold th, Subspace dimension d.
Ensure: L_T are the predicted labels for the points in T.

$A \leftarrow choose\_landmarks(S, T, th)$
$\sigma \leftarrow median\_distance(S \cup T)$
$K_S \leftarrow project\_using\_kernel(S, A, \sigma)$
$K_T \leftarrow project\_using\_kernel(T, A, \sigma)$
$X_S \leftarrow PCA(K_S, d)$; $X_T \leftarrow PCA(K_T, d)$
$M \leftarrow X_S' X_T$; $P_S \leftarrow K_S X_S M$; $P_T \leftarrow K_T X_T$
$classifier \leftarrow learn\_classifier(P_S, L_S)$
$L_T \leftarrow classifier(P_T)$

Algorithm 2 *choose_landmarks* used in Algorithm 1.

Require: Source data S, Target data T, Threshold th.
Ensure: A contains the selected landmarks.

$A \leftarrow \{\}$; $distances \leftarrow \{\|a-b\|, (a,b) \in (S \cup T)^2\}$
for c in $S \cup T$ **do**
  for s in $percentiles(distances)$ **do**
    $KV_S \leftarrow \{exp(-\|c-p\|^2/2s^2), p \in S\}$
    $KV_T \leftarrow \{exp(-\|c-p\|^2/2s^2), p \in T\}$
    if $overlap(KV_S, KV_T) > th$ **then** $A = A \cup \{c\}$
    end if
  end for
end for

Kernel Projection and Subspace Alignment Once the set of landmarks A has been selected, we use a Gaussian kernel to achieve a non-linear mapping of all the points into a common space defined by these landmarks. The subspaces from the source and the target domains are then aligned using a linear transformation as in [1]. 1) Each point p from $S \cup T$ is projected onto each landmark $\in A$ using a Gaussian kernel w.r.t. a standard deviation σ set to the median distance between any pair of points drawn randomly from $S \cup T$. 2) PCA is applied on each domain separately to extract the d eigenvectors having the largest eigenvalues and we compute the best alignment transformation M that transforms points from the source eigenspace to the target eigenspace.

Comparison with the state of the art unsupervised DA approaches

Table 1 gives a summary of the accuracy, averaged over 9 adaptation tasks. Our method improves the accuracy by 2% over the best existing method. More details can be found in the paper or at `http://home.heeere.com/publi-2015-cvpr.html`.

[1] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *ICCV*, 2013.

Table 1: Six unsupervised DA methods are compared. **NA**: No Adaptation; **KPCA+SA**: two independent KPCA are performed on the source and target data, then a subspace alignment is applied; **GFK**: Geodesic Flow Kernel; **SA**: one step Subspace Alignment; **TJM**: Joint Matching Transfer; **LSSA**: our approach.

Method	NA	KPCA+SA	GFK	SA	TJM	LSSA
Avg	42.2	8,7	45.9	49.3	50.5	52.6

This is an extended abstract. The full paper is available at the Computer Vision Foundation webpage.