Computer Vision

Mid-Evaluations Presentation

Team Members

- Anvesh Chaturvedi **201610**94
- Samyak Jain **201610**83
- Yudhik Agrawal **201610**93

Introduction

Our project deals with **Efficient Learning of Domain-invariant Image Representations.** We *propose an algorithm* that ***learns representations*** which ***explicitly compensate for domain mismatch*** and which can be *efficiently realized as linear classifiers*. The *ideal image representations does not only depend on the task but also on the domain*. It has been observed that a ***significant degradation in the performance*** of state-of-the-art image classifiers *when input feature distributions change* ***due to different image sensors and noise conditions***, ***pose changes***, a shift from commercial to consumer video, and, more generally, training datasets biased by the way in which they were collected.

Variations In Images in Different Domains

Goal

The ***main goal*** of this project is to ***optimize both the transformation and classifier parameters jointly***, and *introduce an efficient cost function* based on *misclassification loss*. The method will ***combine several features*** previously unavailable in a single algorithm: *multi-class adaptation through representation learning*, *ability to map across heterogeneous feature spaces*, and *scalability to large datasets*.

Input - Desired Output Domain Transformation

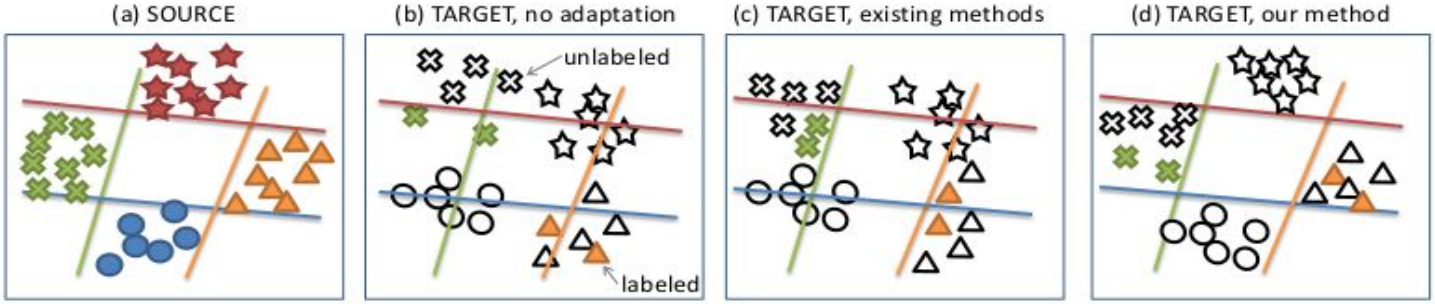
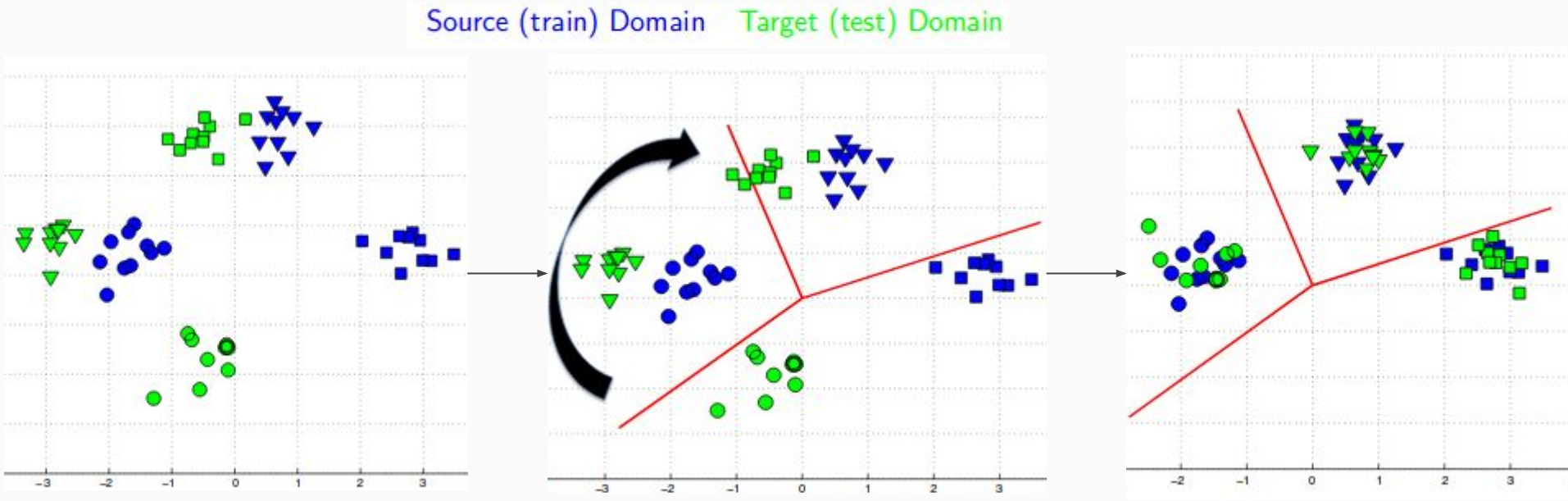


Figure 1: (a) Linear classifiers (shown as decision boundaries) learned for a four-class problem on a fully labeled source domain. (b) Problem: classifiers learned on the source domain do not fit the target domain points shown here due to a change in feature distribution. (c) Existing SVM-based methods only adapt the features of classes with labels (crosses and triangles). (d) Our method adapts all points, including those from classes without labels, by transforming all target features to a new domain-invariant representation.

Input - Desired Output Domain Transformation



Algorithm

Cost Function

$$J(W,\theta_k,b_k) = \frac{1}{2}||W||_F^2 + \sum_{k=1}^{K}\left[\frac{1}{2}||\theta_k||_2^2 + C_S\sum_{i=1}^{n_S}\mathcal{L}\left(y_i^s,\begin{bmatrix}x_i^s\\1\end{bmatrix},\begin{bmatrix}\theta_k\\b_k\end{bmatrix}\right) + C_T\sum_{i=1}^{n_T}\mathcal{L}\left(y_i^t, W\cdot\begin{bmatrix}x_i^t\\1\end{bmatrix},\begin{bmatrix}\theta_k\\b_k\end{bmatrix}\right)\right]$$

where, the constant C_S penalizes the ***source classification error*** and C_T penalizes the ***target adaptation error.***

Algorithm

To solve the above ***optimization problem*** we perform **coordinate descent** on ***W and (θ, b).***

1. Set iteration *j = 0*, $W^{(j)} = 0$.

2. Solve the sub-problem $(\theta^{(j+1)}_k, b^{(j+1)}_k) = \text{argmin}_{\theta_k, b_k} J(W^{(j)}, \theta_k, b_k)$ by solving:

$$\min_{\theta,b} \sum_{k=1}^{K} \left[\frac{1}{2}||\theta_k||_2^2 + C_S \sum_{i=1}^{n_S} \mathcal{L}\left(y_i^s, \begin{bmatrix} x_i^s \\ 1 \end{bmatrix}, \begin{bmatrix} \theta_k; \\ b_k \end{bmatrix}\right) + C_T \sum_{i=1}^{n_T} \mathcal{L}\left(y_i^t, W^{(j)} \cdot \begin{bmatrix} x_i^t \\ 1 \end{bmatrix}, \begin{bmatrix} \theta_k \\ b_k \end{bmatrix}\right) \right]$$

3. Solve the subproblem $W^{(j+1)} = \text{argmin}_W J(W, \theta^{(j+1)}, b^{(j+1)})$ by solving the equation below and *increment* **j**.

$$\min_W \quad \frac{1}{2}||W||_F^2 + C_T \sum_{k=1}^{K} \sum_{i=1}^{n_T} \mathcal{L}\left(y_i^t, W \cdot \begin{bmatrix} x_i^t \\ 1 \end{bmatrix}, \begin{bmatrix} \theta_k^{(j+1)} \\ b_k^{(j+1)} \end{bmatrix}\right)$$

4. Iterate *steps 2 & 3* ***until convergence***.

Explanation

Previous work that has been followed adapts linear SVMs, *learning a perturbation of the source hyper- plane* by *minimizing the classification error* on labeled target examples for each binary task. The proposed algorithm is called **Max-Margin Domain Transform** (MMDT). It *uses an asymmetric transform* **W** to ***map target features x*** to a ***new representation Wx*** maximally aligned with the source and learning the transform jointly on all categories for which target labels are available. It provides a way to *adapt max-margin classifiers in a multi-class* manner, by ***learning a shared component of the domain shift*** as captured by the ***feature transformation W*** . Moreover, it can be optimised in linear space, making it a feasible solution for problem settings with a large amount of training data.

Current Progress

- Explored **Office + Caltech** *Dataset*. Four domains are included: *Caltech (C), Amazon(A), Webcam(W) and DSLR(D)*. In fact, this dataset is constructed from two datasets: **Office-31** (*which contains 31 classes of A, W and D*) and **Caltech-256** (*which contains 256 classes of C*). There are just ***10 common classes*** in both, so the **Office+Caltech dataset** is formed.
- Trained an ***SVM based classifier*** on this dataset and *compared the results the proposed MMDT algorithm*.
- We ***tried multiple combination of source*** and ***target domains*** - **Amazon, WebCam, DSLR and Caltech**. We fed them into the ***svm based classifier*** and compared the results.

Results

The dataset we used is **Office+Caltech 256**. The Office dataset has images taken from three different sources namely - Amazon (A) , WebCam (W) and DSLR (D). The Office and Caltech (C) dataset has 10 common classes and we have ***computed our accuracies on these classes by varying the Source (S) and Target (T)*** and compared our results with the results of the proposed MMDT algorithm.

S - T	MMDT (*Accuracy*)	SVM Based Classifier (*Accuracy*)
A - W	**64.6**	29.67
A - D	**56.7**	29.52
W - D	**67.0**	49.96
D - W	**74.1**	55.96
D - C	**34.1**	21.59
C - A	**49.4**	30.34
C - D	**56.5**	32.40

Milestones Left

- Implementing the ***asymmetricTransformWithSVM*** procedure so that we can ***train*** *both* ***classification*** *and* ***transformation parameters***.
- *Improving the performance* of implemented algorithms.
- *Extensive testing and observation* of results obtained using the implemented algorithm.

Thanks