Summary of Universal Domain Adaptation through Self-Supervision

Ashwin Ramachandran, Akshat Kumar

October 20, 2023

1 Introduction

The paper addresses the challenge of universal domain adaptation, where the objective is to adapt a deep learning model to perform well in a target domain without prior knowledge of the category shift. This task involves labeled source data (with known categories) and unlabeled target data containing both known and potentially unknown categories. The goal is to classify the target samples into either known categories or as "unknown" categories. The model is trained on a combination of source and target data and evaluated on the target domain.

2 Handling category shifts

The approach focuses on avoiding complete alignment between source and target distributions, as forcing such alignment can lead to misalignment issues. Instead, the goal is to extract well-clustered target features while maintaining a flexible alignment with source classes and the ability to reject "unknown" points.

2.1 Techniques

The techniques below collectively constitute the DANCE framework, which is designed to address universal domain adaptation challenges without prior knowledge of category shifts.

2.2 Neighborhood Clustering (NC)

2.2.1 Description

Neighborhood Clustering (NC) is a self-supervision technique used to cluster target samples in a target domain. It aligns each target sample with either a "known" class prototype from the source domain or its nearest neighbor in the target domain. This approach helps the model learn a discriminative metric for semantic similarity, independent of known class labels. The primary goal is to create well-clustered features, which benefits the extraction of discriminative features, even for "unknown" samples, making it suitable for Universal Domain Adaptation.

2.2.2 Approach

The key idea is to minimize the entropy of each target point's similarity distribution to other target samples and prototypes. This eliminates the need to specify the number of clusters in the target domain, making it adaptable for Universal Domain Adaptation.

2.2.3 Implementation Details

- Similarity between target samples and prototypes is calculated for each mini-batch of target features. - A memory bank (V) stores all target features, and a combined matrix (F) stores target features and prototype weight vectors. - A memory bank is used to account for target samples not present in the mini-batch. - In each iteration, the memory bank (V) is updated with features from the current mini-batch. - Memory bank entries are updated for target samples present in the mini-batch. - Probability of one feature being a neighbor of another is calculated using an exponential function with a temperature parameter (τ). - The entropy loss (L_{nc}) minimizes the divergence between calculated probabilities and their logarithms, aligning target samples with either target neighbors or prototypes based on proximity.

$$L_{nc} = -\frac{1}{|B_t|} \sum_{i \in B_t} \sum_{j \in B_t, j \neq i} \frac{\exp(F_j^T f_i/\tau)}{\sum_{k=1, k\neq i}^{N_t+K} \exp(F_k^T f_i/\tau)} \log \left(\frac{\exp(F_j^T f_i/\tau)}{\sum_{k=1, k\neq i}^{N_t+K} \exp(F_k^T f_i/\tau)} \right) \tag{1}$$

This approach promotes feature clustering in the target domain, enabling the extraction of discriminative features, even for "unknown" categories, without the need to pre-specify the number of clusters. Further parameter analysis is provided in the paper's supplemental material.

3 Entropy Separation Loss (ES)

The Entropy Separation Loss (ES) is introduced to align target samples with "known" source categories or classify them as "unknown." It operates on the entropy of the output from the "known" category classifier, forcing it to be either low (indicating the sample belongs to a "known" class) or high (indicating the sample is far from any "known" class).

3.1 Motivation

"Unknown" target samples typically exhibit higher entropy in the source classifier's output compared to "known" target samples due to the lack of common features with "known" source classes. This motivates the need for a boundary in entropy space to distinguish between "known" and "unknown" points.

3.2 Approach

The goal is to create a boundary in entropy space, denoted as ρ, representing the threshold between "known" and "unknown" points. The distance between the entropy and the threshold boundary, ρ, is defined as $|H(p) - \rho|$, where p is the classification output for a target sample. Maximizing this distance keeps $H(p)$ far from ρ. The value for ρ is defined as $\rho = \frac{\log(K)}{2}$, where K is the number of source classes.

To address potential ambiguity, a confidence threshold parameter m is introduced. The final form of the loss, Les, is defined as follows:

$$Les = \frac{1}{|Bt|} \sum_{i \in Bt} (-|H(p_i) - \rho|) \cdot \mathbf{1}(|H(p_i) - \rho| > m)$$

The introduction of the confidence threshold m allows the separation loss to be applied only to confident samples. When $|H(p_i) - \rho|$ is sufficiently large, it indicates that the network is confident about categorizing a sample as "known" or "unknown," and it is trained to keep the sample far from the threshold value ρ.

The Entropy Separation Loss helps distinguish between "known" and "unknown" target samples, contributing to the goal of Universal Domain Adaptation.

3.3 Final Objective

The final objective is formulated as follows:

$$L = L_{cls} + \lambda(L_{nc} + L_{es}) \tag{2}$$

Where: - L_{cls} denotes the cross-entropy loss on source samples. - The loss on source and target samples is calculated using different mini-batches to facilitate domain-specific batch normalization.

In an effort to reduce the number of hyperparameters, the paper employs the same weighting hyperparameter, λ, for both L_{nc} and L_{es}.

This approach helps to achieve better alignment between source and target domains while maintaining the balance between various loss components.