Robust Target Training for Multi-Source Domain Adaptation

Zhongyang Deng, Da Li, Yi-Zhe Song, Tao Xiang

Email: z.deng@surrey.ac.uk

Code: https://github.com/Zhongyang-Deng/BORT2

Multi-Source Domain Adaptation

**Training Set**
- Cartoon
- Photo
- Sketch
- Art
  - labeled source domains
  - unlabeled target domain

**Test Set**
- Art
  - source features
  - target features

- Goal
  - transfer knowledge
- Challenge
  - domain shift
- Scenario
  - labeled source domains
  - unlabeled target domain

**Motivation**
- To reduce the domain shift, most existing methods try to align feature distributions across domains.

**Proposed Method**
- **Motivation:** A second step training on target domain to alleviate source-domain bias
- **Key:** Train noise-robust model and improve the quality of pseudo-labels

**Comparison to baseline models**

- Vanilla
- Target Re-training
- BORT2

**Comparison to the state of the art**

- Results on PACS
- Results on DomainNet

**Conclusions**

- BORT2 extracts more discriminative features

**Experiments**

- A naive second step training on the target domain using pseudo-labels can improve the performance of existing MSDA methods.
- BORT2 further improve the performance.

**Proposed Method**

- **Step 1:** Train labeling function on both source and target domains
- **Step 2:** Train noise-robust model only on the pseudo-labeled target domain

**Bi-level Optimization based Robust Target Training (BORT2)**

**Step 1** can be trained using any existing multi-source domain adaptation (MSDA) methods
- E.g., DANN, MSDA, DRT.

**Step 2** is the **bi-level optimization** built on feature uncertainty estimation
- A stochastic CNN layer in the noise-robust target model is used to model each target instance feature as a Gaussian distribution
- The variance of the Gaussian measure the label uncertainty as per [1]
- In the inner loop, the feature uncertainty can help the cross-entropy loss to identify low-quality pseudo-labels
- Low-quality pseudo-labels are downplayed to train a noise-robust model
- The outer loop treat the labeling function as hyper network, which is optimized to minimize feature uncertainty loss using bi-level optimization
- Low feature uncertainty usually implies higher probability of pseudo-labels.