Term Papers 2

Adarsh Shah, Sasanka Sekhar Sahu

1 Domain Adaptation for Structured Output via Discriminative Patch Representations

1.1 Introduction

Supervised semantic segmentation requires per-pixel annotations while training. Also, per-pixel annotations are required while fine-tuning to domains different from training. A novel domain adaptation technique is introduced to address this issue.

1.2 Idea

- Learn discriminative features of patches of images in the source domain by discovering patchwise output distribution to form clustered space.
- Using the features discovered as guidance, adversarially push the features obtained in target domain to the clustered features in source domain.

1.3 Description

- Given Images $I_s, I_t \in \mathcal{R}^{H \times W \times 3}$ from source and target domain with per-pixel annotations for Y_s, learn a model G that works on both domains.
- Patch Level Alignment: The features of image patches belonging to the same semantic region should be clustered in the clustered space. Each such cluster is called Patch mode.
  - Patch Mode Discovery:
    - Extract 2×2 image patches from the source domain and generate a label histogram. Hence, each patch is represented by a $2 \times 2 \times C$ vector.
    - Each patch is mapped to ground truth label Y_s as $\Gamma(Y_s)$
    - $F_s = H(G(I_s)) \in (0,1)^{(}U \times V \times K)$ where G is the model/feature extractor and H is the classification module.
    - The objective becomes

    $$\mathcal{L}_d(F_s, \Gamma(Y_s); G, H) = -\Sigma_{u,v}\Sigma_{k \in K}\Gamma(Y_s)^{(u,v,k)}log(F_s^{(u,v,k)})$$
  - Adversarial Alignment
    - $F_t = H(G(T_t))$. Here, I_t is the image from target domain
    - D is the discriminator to classify between source and target domains

    $$\mathcal{L}_{adv}(F_s, \Gamma(Y_s); G, H, D) = \Sigma_{u,v}\mathcal{E}[log(D(F_s^{(u,v,1)}))] + \mathcal{E}[log(1 - D(F_t^{(u,v,1)}))]$$
- The final learning objective becomes

  $$min_{G,H} max_D \mathcal{L}_s(G) + \lambda_d \mathcal{L}_d(G,H) + \lambda_{adv}\mathcal{L}_{adv}(G,H,D)$$

  where $\mathcal{L}_s$ is the supervised cross-entropy loss objective for semantic segmentation.

1.4 References

- `https://openaccess.thecvf.com/content_ICCV_2019/papers/Tsai_Domain_Adaptation_for_Structured_Output_via_Discriminative_Patch_Representations_ICCV_2019_paper.pdf`

2 Maximum Classifier Discrepancy for Unsupervised Domain Adaptation

2.1 Main Idea

A novel approach that attempts to align distributions of source and target by utilizing the task-specific decision boundaries and maximizing the discrepancy between two classifiers' outputs to detect target samples that are far from the support of the source. A feature generator adversarially learns to generate target features near the support to minimize the discrepancy.

2.2 Proposed Method

We have access to a labeled source image $\mathbf{x}_s$ and a corresponding label $\mathbf{y}_s$ drawn from labeled source images $\{\mathbf{X}_s, \mathbf{Y}_s\}$, as well as an unlabeled target image $\mathbf{y}_t$ drawn from unlabeled target images $\mathbf{X}_t$. We train a feature generator network G, which takes inputs $\mathbf{x}_s$ or $\mathbf{x}_t$, and classifier networks F_1 and F_2, which take features from G. F_1 and F_2 classify them into K classes - they output a K-dimensional logits vector by applying the softmax function: $p_1(\mathbf{y}|\mathbf{x})$ & $p_2(\mathbf{y}|\mathbf{x})$.

We assume that the two classifiers can classify source samples correctly but are initialized differently to obtain different classifiers from the beginning of training. The goal is to align source and target features by utilizing the task-specific classifiers as a discriminator. The target samples outside the support of the source are likely to be classified differently by the two classifiers. This region is called the $discrepancy\ region$. If we train the generator to minimize the discrepancy between the two classifiers, the generator will avoid generating target features outside the support of the source.

In order to effectively detect target samples outside the support of the source, we propose to train discriminators (F_1 and F_2) to maximize the discrepancy given target features. Without this step, the two classifiers can be very similar ones and cannot detect target samples outside the support of the source. We then, in an adversarial setting, train the generator to fool the discriminator by minimizing the discrepancy.

Discrepancy Loss. The absolute values of the difference between the two classifiers' probabilistic outputs is taken as the discrepancy loss:

$$d(p_1, p_2) = \frac{1}{K}\sum_{k=1}^{K} |p_{1k} = p_{2k}|$$

Training steps. First, we train both classifiers and generator to classify the source samples correctly by minimizing the softmax cross entropy objective as below.

$$\mathbf{min}_{G,F_1,F_2}\mathcal{L}(\mathbf{X}_s, \mathbf{Y}_s)$$

$$\mathcal{L}(\mathbf{X}_s, \mathbf{Y}_s) = \mathbb{E}_{(\mathbf{x}_s, y_s)\sim(\mathbf{X}_s, Y_s)}\sum_{k=1}^{K}\mathbb{1}_{[k=y_s]}\log p(\mathbf{y}|\mathbf{x}_s)$$

Then we train the classifiers (F_1, F_2) as a discriminator for a fixed generator (G). We add a classification loss on the source samples and use the same number of source and target samples to update the model. The objective is as follows:

$$\mathbf{min}_{F_1,F_2}\mathcal{L}(\mathbf{X}_s, \mathbf{Y}_s) - \mathcal{L}_{adv}(\mathbf{X}_t)$$

$$\mathcal{L}_{adv} = \mathbb{E}_{\mathbf{x}_t\sim\mathbf{X}_t}[d(p_1(\mathbf{y}|\mathbf{x}), p_2(\mathbf{y}|\mathbf{x}))]$$

We train the generator to minimize the discrepancy for fixed classifiers:

$$\mathbf{min}_{G}\mathcal{L}_{adv}(\mathbf{X}_t)$$

2.3 Reference

1. Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum Classifier Discrepancy for Unsupervised Domain Adaptation. `https://arxiv.org/pdf/1712.02560.pdf`.