Contrastive Adaptation Network for Unsupervised Domain Adaptation

Seri Lee
Computer Science and Engineering
Seoul National University
Seoul, Republic of Korea
sally20921@snu.ac.kr

***Abstract*—Unsupervised Domain Adaptation (UDA) makes predictions for the target domain data while manual annotations are only available in the source domain. Previous methods minimize the domain discrepancy neglecting the class information, which may lead to misalignment and poor generalization performance. To address this issue, this paper proposes Contrastive Adaptation Network (CAN) optimizing a new metric which explicitly models the intra-class domain discrepancy and the inter-class domain discrepancy. We design an alternating update strategy for training CAN in an end-to-end manner. Experiments on two real-world benchmarks Office-31 and VisDA-2017 demonstrate that CAN performs favorably against the state-of-the-art methods and produce more discriminative features.**

I. Introduction

Recent advancements in deep neural networks [**?**] have successfully improved a variety of learning problems. For supervised learning, however, massive labeled training data is still the key to learning an accurate deep model. Although abundant labels may be available for a few pre-specified domains, such as ImageNet, manual labels often turn out to be difficult or expensive to obtain for every ad-hoc target domain or task. The absence of in-domain labeled data hinders the application of data-fitting models in many real-world problems.

In the absence of labeled data from target domain, Unsupervised Domain Adaptation (UDA) methods have emerged to mitigate the domain shift in data distributions. It relates to unsupervised learning as it requires manual labels only from the source domain and zero labels from the target domain. Among the recent work on UDA, a seminal line of work proposed by Long et al. aims at minimizing the discrepancy between the source and target domain in the deep neural network, where the domain discrepancy is measured by Maximum Mean Discrepancy (MMD) and Joint MMD (JMMD). MMD and JMMD have proven effective in many computer vision problems and demonstrated the state-of-the-art results on several UDA benchmarks.

Despite the success of previous methods based on MMD and JMMD, most of them measure the domain discrepancy at the domain level, neglecting the class from which the samples are drawn. These class=agnostic approaches, hence, do not discriminate whether samples from two domains should be aligned according to their class labels. This can impair the adaptation performance due to the following reasons. First, samples of different classes may be aligned incorrectly, e.g. both MMD and JMMD can be minimized even when the target-domain samples are misaligned with the source-domain samples of a different class. Second, the learned decision boundary may generalize poorly for the target domain. There exist many suboptimal solutions near the decision boundary. These solutions may overfit the source data well but are less discriminative for the target.

To address the above issues, we introduce a new $ContrastiveDomainDiscrepancy(CDD)$ objective to enable class-aware UDA. We propose to minimize the intra-class discrepancy, *i.e.*, the domain discrepancy within the same class, and maximize the inter-class margin, *i.e.*, the domain discrepancy between different classes. Considering the toy example, CDD will draw closer the source and target samples of the same underlying class (*e.g.* the blue and red triangles), while pushing apartt the samples from different classes (*e.g.* the blue triangle and the red star).

Unfortunately, to estimate and optimize with CDD, we may not train a deep network out-of-the-box as we need to overcome the following two technical issues. First, we need labels from both domains to compute CDD, however, target labels are unknown in UDA. A straightforward way, of course, is to estimate the target labels by the network outputs during training. However, because the estimation can be noisy, we find it can harm the adaptation performance. Second, during the mini-batch training, for a class C, the mini-batch may only contain samples from one domain (source or target), rendering it infeasible to estimate the intra-class domain discrepancy of C. This can result in a less efficient adaptation. The above issues require special design of the network and the training paradigm.

In this paper, we propose Contrastive Adaptation Network (CAN) to facilitate the optimization with CDD. During training, in addition to minimizing the cross-entropy loss on labeled source data, CAN alternatively estimates the underlying label hypothesis of target samples through clustering, and adapts the feature representations according to the CDD metric. After clustering, the ambiguous target data (i.e. far from the cluster centers) and ambiguous classes (i.e. containing few target samples around the cluster centers) are zeroed out in estimating the CDD. Empirically we find that during training, an increasing amount of samples will be taken into

account. Such progressive learning can help CAN capture more accurate statistics of data distributions. Moreover, to facilitate the mini-batch training of CAN, we employ the class-aware sampling for both source and target domains, i.e. at each iteration, we sample data from both domains for each class within a randomly sampled class subset. Class-aware sampling can improve the training efficiency and the adaptation performance.

We validate our method on two public UDA benchmarks: Office-31 and VisDA-2017. The experimetal results show that our method performs favorably against the state-of-the-art UDA approaches, i.e. we achieve the bets-published result on the Office-31 benchmark and very competitive result on the challenging VisDA-2017 benchmark. Ablation studies are presented to verify the contribution of each key component in our framework.

In a nutshell, our contributions are as follows,

- We introduce a new discrepanc metric Contrastive Domain Discrepancy (CDD) to perform class-aware alignment for unsupervised domain adaptataion.
- We propose a network Contrastive Adaptation Network to facilitate the end-to-end training with CDD.
- Our method achieves the best-published result on the Office-31 benchmark and competitive performance compared to the state-of-the-art on the challenging VisDA-2017 benchmark.

II. Related Work

Class-agnostic domain alignment. A common practice for UDA is to minimize the discrepancy between domains to obtain domain-invariant features. For example, Tzeng et al. proposed a kind of domain confusion loss to encourage the network to learn both semantically meaningful and domain invariant representations. Long et al. proposed DAN and JAN to minimize the MMD and Joint MMD distance across domains respectively, over domain-specific layers. Ganin et al. enabled the network to learn domain invariant representations in adversarial way by back-propagating the reverse gradients of the domain classifier. Unlike these domain-discrepancy minimization methods, our method performs class-aware domain alignment.

Discriminative domain-invariant feature learning. Some previous works pay efforts to learn more discriminative features while performing domain alignment. Adversarial Dropout Regularization (ADR) and Maximum Classifier Discrepancy (MCD) were proposed to train a deep neural network in adversarial away to avoid generating non-discriminative features lying in the region near the decision boundary. Similar to us, Long et al. and Pei et al. take the class information into account while measuring the domain discrepancy. However, our method differs from theirs mainly in two aspects. Firstly, we explicitly model two types of domain discrepancy, i.e. the intra-class domain discrepancy and the inter-class domain discrepancy. The inter-class domain discrepancy, which has been ignored by most previous methods, is proved to be beneficial for enhancing the model adaptation performance. Secondly, in the context of deep neural networks, we treat the training process as an alternative optimization over target label hypothesis and features.

Intra-class compactness and inter-class separability modeling. This paper is also related to the work that explicitly models the intra-class compactness and the inter-class separability e.g. the contrastive loss and the triplet loss. These methods have been used in various applications, e.g. face recognition, person re-identification, etc. Different from these methods designed for a single domain, our work focuses on adaptation arocss domains.

III. Methodology

Unsupervised Domain Adaptation (UDA) aims at improving the model's generalization performance on target domain by mitigating the domain shift in data distribution of the source and target domain. Formally, given a set of source domain samples $S = \{(s_1^s, y_1^s), \ldots, (x_N^s, y_N^s)\}$, and target domain samples $T = \{x_1^t, \ldots, x_{N_t}^t\}$, x^s, x^t represent the input data, and $y^s \in \{0, 1, \ldots, M-1\}$ denote the source data label of M classes. The target data label $y^t \in \{0, 1, \ldots, M-1\}$ is unknown. Thus, in UDA, we are interested in training a network using labeled source domain data S and unlabeled target domain data T to make accurate predictions $\{\hat{y}^t\}$ on T.

We discuss our method in the context of deep neural networks. In deep neural networks, a sample owns hierarchical features/representations denoted by the activations of each layer $l \in L$. In the following, we use $\phi_l(x)$ to denote the outputs of layer l in a deep neural network Φ_θ for the input x, where $\phi(\cdot)$ denotes the mapping defined by the deep neural network from the input to a specific layer.

A. Maximum Mean Discrepancy Revisit

In Maximum Mean Discrepancy (MMD), $\{x_i^s\}$ and $\{x_i^t\}$ are *i.i.d.* sampled from the marginal distributions $P(X^s)$ and $Q(X^t)$ respectively. Based on the observed samples, MMD performs a kernel two-sample test to determine whether to accept the null hypothesis $P = Q$ or not. MMD is motivated by the fact that if two distributions are identical, all of their statistics should be the same. Formally, MMD defines the difference between two distributions with their mean embeddings in the reproducing kernel Hilbert space (RKHS), *i.e.*

$$D_{\mathcal{H}}(P, Q) \triangleq \sup_{f \sim \mathcal{H}} (\mathbb{E}_{X^s}[f(X^s)] - \mathbb{E}_{X^t}[f(X^t)])_{\mathcal{H}} \quad (1)$$

where $\mathcal{H}$ is class of functions. In practice, for a layer l, the squared value of MMD is estimated with the empirical kernel mean embeddings

$$\begin{aligned} \hat{D}_l^{mmd} = & \frac{1}{n_s^2} \sum_{i=1}^{n_s} \sum_{j=1}^{n_s} k_l(\phi_l(x_i^s), \phi_l(x_j^s)) \\ & + \frac{1}{n_t^2} \sum_{i=1}^{n_t} \sum_{j=1}^{n_t} k_l(\phi_l(x_i^t) \phi_l(x_j^t)) \\ & - \frac{2}{n_s n_t} \sum_{n=1}^{n_s} \sum_{j=1}^{n_t} k_l(\phi_l(x_i^s) \phi_l(x_j^t)) \end{aligned} \quad (2)$$

wher $x^s \in S' \subset S$, $x^t \in T' \subset T$, $n_s = |S'|$, $n_t = |T'|$. The S' and T' represent the mini-batch source and target data sampled from S and T respectively. And k_l denotes the kernel selected for the l-th layer of deep neural network.

B. Contrastive Domain Discrepancy

We propose to explicitly take the class information into account and measure the intra-class and inter-class discrepancy across domains. The intra-class domain discrepancy is minimized to compact the feature representations of samples within a class, whereas the inter-class domain discrepancy is maximized to push the representations of each other further away from the decision boundary. The intra-class and inter-class discrepancies are jointly optimized to improve the adaptation performance.

The proposed Contrastive Domain Discrepancy (CDD) is established on the difference between conditional data distributions across domains. Without any constraint on the type (*e.g.* marginal or conditional) of data distributions, MMD is convenient to measure such difference between $P(\phi(X^s)|Y^s)$ and $Q(\phi(X^t)|Y^t)$, *i.e.* $D_{\mathcal{H}}(P,Q) \triangleq sup_{f\sim\mathcal{H}}(\mathbb{E}_{X^s}(f(\phi(X^s))|Y^s) - \mathbb{E}_{X^t}(f(\phi(X^t))|Y^t))_{\mathcal{H}}$.

Supposing

$$\mu_{cc'}(y,y') = \begin{cases} 1 & \text{if } y = c, y' = c' \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

, for two classes c_1, c_2 (which can be same or different), the kernel mean embedding estimation for squared $D_{\mathcal{H}}(P,Q)$ is

$$\hat{D}^{c_1c_2}(\hat{y}_1^t, \hat{y}_2^t, \ldots, \hat{y}_{n_t}^t, \phi) = e_1 + e + 2 - 2e_3 \tag{4}$$

where

$$\begin{aligned} e_1 &= \sum_{i=1}^{n_s}\sum_{j=1}^{n_s} \frac{\mu_{c_1c_1}(y_i^s, y_j^s)k(\phi(x_i^s), \phi(x_j^s))}{\sum_{i=1}^{n_s}\sum_{j=1}^{n_s}\mu_{c_1c_1}(y_i^s, y_j^s)} \\ e_2 &= \sum_{i=1}^{n_t}\sum_{j=1}^{n_t} \frac{\mu_{c_2c_2}(y_i^t, y_j^t)k(\phi(x_i^t), \phi(x_j^s))}{\sum_{i=1}^{n_t}\sum_{j=1}^{n_t}\mu_{c_2c_2}(\hat{y}_i^t, \hat{y}_j^t)} \\ e_3 &= \sum_{i=1}^{n_s}\sum_{j=1}^{n_t} \frac{\mu_{c_1c_2}(y_i^s, y_j^t)k(\phi(x_i^s), \phi(x_j^t))}{\sum_{i=1}^{n_s}\sum_{i=1}^{n_t}\mu_{c_1c_2}(y_i^s, \hat{y}_j^t)} \end{aligned} \tag{5}$$

Note that Eq. (4) defines two kinds of class-aware domain discrepancy, 1) when $c_1 = c_2 = c$, it measures intra-class domain discrepancy, 2) when $c_1 \neq c_2$, it becomes the inter-class domain discrepancy. To compute the mask $\mu_{c_2c_2}(\hat{y}_i^t, \hat{y}_j^t)$ and $\mu_{c_1c_2}(y_i^s, \hat{y}_j^t)$, we need to estimate target label $\{\hat{y}_i^t\}$.

Based on the above definitions, the CDD is calculated as (The $\hat{y}_1^t, \hat{y}_2^t, \ldots, \hat{y}_{n_t}^t$ is abbreviated as $\hat{y}_{1:n_t}^t$)

$$\begin{aligned} \hat{D}^{cdd} = &\underbrace{\frac{1}{M}\sum_{c=1}^{M}\hat{D}^{cc}(\hat{y}_{1:n_t}^t, \phi)}_{\text{intra}} \\ &\underbrace{- \frac{1}{M(M-1)}\sum_{c=1}^{M}\sum_{\substack{c'=1 \\ c'\neq c}}^{M}\hat{D}^{cc'}(\hat{y}_{1:n_t}^t, \phi)}_{\text{inter}} \end{aligned} \tag{6}$$

where the intra- and inter-class domain discrepancies will be optimized in the opposite direction.

Note although the estimation of the labels $\{\hat{y}_i^t\}$ can be noisy, the CDD (which is established on MMD) in itself is robust to the noise to an extent. Because MMD is determined by the mean embeddings of distributions in the RKHS, the sufficient statistics is less likely to be severely affected by the label noise, especially when the amount of data is large.

C. Contrastive Aaptation Network

Deep convolutional neural networks (CNNs) is able to learn more transferable features than shallow methods. However, the discrepancy still exists for domain-specific layers. Specifically, the convolutional layers extracting general features are more transferable, while the fully-connected (FC) layers which exhibit abstract and domain-specific features should be adapted.

In this paper, we start from ImageNet pretrained networks, *e.g.* ResNet, and replace the last FC layer with task-specific ones. We follow the general practice that minimizes the domain discrepancy of last FC layers and fine-tunes the convolutional layers through back-propagation. Then our proposed CDD can be readily incorporated into the objective as an adaptation module over the activations of FC layers. We name our network Contrastive Adaptation Network (CAN).

The overall objective. In a deep CNN, we need to minnimize CDD over multiple FC layers, *i.e.* minimizing

$$\hat{D}_L^{cdd} = \sum_{l=1}^{L}\hat{D}_l^{cdd} \tag{7}$$

. Besides, we train the network with labeled source data through minimizing the cross-entropy loss,

$$l^{ce} = -\frac{1}{{n'}_s}\sum_{i'=1}^{{n'}_s}\log P_\theta(y_{i'}^s|x_{i'}^s) \tag{8}$$

where $y^s \in \{0, 1, \ldots, M-1\}$ is the ground-truth label of sample x^s. $P_\theta(y|x)$ denotes the predicted probability of label y with the network parameterized by θ, given input x.

Therefore, the overall objective can be formulated as

$$\min_\theta l = l^{ce} + \beta\hat{D}_L^{cdd} \tag{9}$$

where β is the weight of the discrepancy penalty term. Through minimizing $\hat{D}_L^{cdd}$, the intra-class domain discrepancy is minimized and the inter-class domain discrepancy is maximized to peform class-aware domain alignment.

Note that we independently sample the labeled source data to minimize the cross-entropy loss l^{ce} and those to estimate the CDD $\hat{D}_L^{cdd}$. In this way, we are able to desgin more efficient sampling strategy to facilitate the mini-batch stochastic optimization with CDD, while not disturbing the conventional optimization cross-entropy loss on labeled source data.

D. Optimizing CAN