Knowledge-Based Systems 222 (2021) 106986

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Discriminative active learning for domain adaptation

Fan Zhou [a], Changjian Shui [b], Shichun Yang [c], Bincheng Huang [d,e], Boyu Wang [f,g,*], Brahim Chaib-draa [a,**]

[a] Department of Computer Science and Software Engineering, Laval University, QC, Canada
[b] Department of Electrical and Computer Engineering, Laval University, QC, Canada
[c] School of Transportation Science and Engineering, Beihang University, Beijing, China
[d] Key Laboratory of Cognition and Intelligence Technology, China Electronics Technology Group Corporation, Beijing, China
[e] Information Academy, China Electronics Technology Group, Beijing, China
[f] Department of Computer Science, University of Western Ontario, ON, Canada
[g] Vector Institute, ON, Canada

ARTICLE INFO

Article history:
Received 25 November 2020
Received in revised form 15 February 2021
Accepted 21 March 2021
Available online 26 March 2021

Keywords:
Domain adaptation
Adversarial learning
Active learning

ABSTRACT

Domain Adaptation aiming to learn a transferable feature between different but related domains has been well investigated and has shown excellent empirical performances. Previous works mainly focused on matching the marginal feature distributions using the adversarial training methods while assuming the conditional relations between the source and target domain remained unchanged, *i.e.*, ignoring the conditional shift problem. However, recent works have shown that such a conditional shift problem exists and can hinder the adaptation process. To address this issue, we have to leverage labeled data from the target domain, but collecting labeled data can be quite expensive and time-consuming. To this end, we introduce a discriminative active learning approach for domain adaptation to reduce the efforts of data annotation. Specifically, we propose three-stage active adversarial training of neural networks: invariant feature space learning (first stage), uncertainty and diversity criteria and their trade-off for query strategy (second stage) and re-training with queried target labels (third stage). Empirical comparisons with existing domain adaptation methods using four benchmark datasets demonstrate the effectiveness of the proposed approach. Furthermore, by comparing different query strategies, we could demonstrate the benefits of our proposed method.

1. Introduction

In general machine learning tasks, we usually assume the datasets, where the hypothesis was trained and tested, are from the same distribution. However, this assumption, in general, is not realistic in many practical scenarios. For example, appearance shifts caused by illumination, seasonal, or weather changes are significant challenges for computer vision-based systems [1]. A vision system trained on one dataset but deployed on another may suffer from rapid performance drop [2,3]. More severely, to train a high-performance vision system requires a large amount of labeled data, and getting such labels may be expensive. Data hungry has become a major limitation for modern machine learning problems. One approach to deal with this issue is *Domain Adaptation* (DA), which aims to improve the learning performance of a target domain by leveraging the unlabeled data in the target domain as well as the labeled data from a different but related domain (source domain). Previous works have theoretically analyzed the learning guarantees of DA [4–6] and have reported some empirical applications in natural language processing [7–9] and computer vision [10,11]. Previous survey papers [12,13] showed large amounts of applications of DA in computer vision tasks.

Most recent DA advancements [5,14,15] are mostly based on the basic *Covariate Shift* assumption [16] that the marginal distributions of source and target domain change ($\mathbb{P}_S(\mathbf{x}) \neq \mathbb{P}_T(\mathbf{x})$) while the conditional distribution (predictive relation) is preserved ($\mathbb{P}_S(y|\mathbf{x}) = \mathbb{P}_T(y|\mathbf{x})$) during the adaptation process. However, some recent works have revealed that this assumption may not hold, and in this case, one may still need some labeled data from the target domain in order to successfully transfer information from one domain to another. Specifically, Zhao et al. [6] discussed the conditional shift problem showing that such a problem exists and can hinder the adaptation process. They proved that the risk on target domain is controlled by the source risk,

* Corresponding author at: Department of Computer Science, University of Western Ontario, ON, Canada.
** Corresponding author at: Department of Computer Science and Software Engineering, Laval University, QC, Canada.
E-mail addresses: bwang@csd.uwo.ca (B. Wang), brahim.chaib-draa@ift.ulaval.ca (B. Chaib-draa).

https://doi.org/10.1016/j.knosys.2021.106986
0950-7051/

the marginal distribution divergence, and disagreement between the *two labeling distributions*:

$$\epsilon_{\mathcal{T}}(h) \leq \epsilon_{\mathcal{S}}(h) + d_{\hat{\mathcal{H}}}(\mathcal{D}_{\mathcal{S}}, \mathcal{D}_{\mathcal{T}}) + \underbrace{\min\{\mathbb{E}_{\mathcal{D}_{\mathcal{S}}}\left[|f_{\mathcal{S}} - f_{\mathcal{T}}|\right], \mathbb{E}_{\mathcal{D}_{\mathcal{T}}}\left[|f_{\mathcal{S}} - f_{\mathcal{T}}|\right]\}}_{\text{Impossible to measure in unsupervised DA}} \quad (1)$$

Here $\epsilon_{\mathcal{T}}(h)$, $\epsilon_{\mathcal{S}}(h)$ and f refer to target risk, source risk and labeling function, respectively. We will formally define the notations in Section 3. In a typical *unsupervised DA* setting, it is not possible to measure the third term in Eq. (1). One possible way to measure this term is to query some data labels from target domain so that the learner can learn the conditional relations in the target domain. However, the label annotations usually is expensive. Notice that the convergence rate at the disagreement term would generally be $\mathcal{O}(1/\sqrt{N_t})$ [17] with *slow* convergence behavior if the label is *i.i.d.* sampled from the target set with size N_t, which is far sufficient to minimize the last term.

To alleviate such difficulties, one can use *Active Learning* (AL) [18] technique for DA so that the learner can reduce the cost of acquiring labels by requesting labeling from the oracle. AL only tries to query the labels of the most informative examples, and has been shown, in some optimal cases, to achieve **exponentially-lower label-complexity** (number of queried labels) than passive learning [19]. From this perspective, we tried to break the general *i.i.d.* sampling with limited information in the target domain (*a.k.a* semi-supervised domain adaptation approach). Most previous active learning approaches were rooted in uncertainty-based approaches. Dasgupta [20] pointed out that only focusing on the uncertainty might lead to *sample bias*. To overcome such bias problems, we also need to consider the diversity in the query process. Recently, Sinha et al. [21] and Shui et al. [22] proposed adversarial training techniques to query the most informative features via a critic function, which could overcome the sample bias problems.

Aiming to address all the aforementioned issues, we proposed a three-stage discriminative active domain adaptation algorithm, which aims to actively query the most informative instances in the target domain to minimize the labeling disagreement term, under the same and small querying label budget.

In the first stage, we adopted the Wasserstein Distance-based adversarial training technique [5,15] for unsupervised DA through training a critic function for learning the domain invariant feature. The critic could also be used to discriminate the target domain features for active querying. In the second stage, we derived a sample-efficient and straightforward active query strategy based on the network structure, for sampling the most *informative* samples in the target domain by controlling *uncertainty* and *diversity* for selecting the target instances. Finally in the third stage, we deployed a re-weighting technique based on the prediction uncertainty for determining the importance of queried samples to retrain the network.

To summarize, our contributions are two-folds: **(1)** we theoretically analyzed the conditional shift problem in domain adaptation using the Wasserstein distance and provide an active query strategy to migrate the disagreement term between the source and target domain; **(2)** based on the theoretical analysis, we then proposed the active query strategy based on the Wasserstein critic and model classifier without requiring extra computations.

We then implemented extensive experiments on four benchmark datasets. The empirical results showed that our proposed algorithm could improve the classification accuracy with a small query budget. When the query budget is small, the proposed approach can have better performance than its *i.i.d* (random) selection counterparts (reported in Table 5). Furthermore, the comparison with other query strategy based DA baselines also demonstrates the effectiveness of our algorithm.

2. Related works

Our work is most related to DA and AL.

Domain adaptation. Domain Adaptation (DA) aims to learn a domain-invariant feature, which can be transferred from a source domain to another. A large number of efforts have been addressed toward DA [13]. Ben-David et al. [4] first theoretically analyzed the theoretical guarantees with the notion of $\mathcal{H}$-divergence. Then, the domain adversarial training method was proposed by Ganin et al. [14] where the gradient reverse method was proposed. Inspired by As stated before, many of the previous advancements [4, 14,15,23] were based on the assumption that the conditional relations remain unchanged during the adaptation process. Some recent works proposed to tackle the conditional shifts problem. Long et al. [24] proposed to extract the cross-covariance between the source and target feature representations, and also measure the conditional entropy as an uncertainty measure to control the transferability. Wen et al. [11] proposed the Bayesian Neural Network with entropy and variable uncertainty measures to jointly match the marginal distribution ($\mathbb{P}(\mathbf{x})$) and conditional distribution ($\mathbb{P}(y|\mathbf{x})$).

Active learning. AL has been widely investigated by academia in the context of theory or applications. Recently, Sinha et al. [21] proposed a variational autoencoder based adversarial approach to query the informative unlabeled feature from the labeled ones and Gissin and Shalev-Shwartz [25] proposed discriminative active learning. Shui et al. [22] extended and adopted a critic network for querying the diverse features. Those above usually assumes that labeled and unlabeled data are from same distribution. Few works were proposed to implement active learning for enhancing domain adaptation *i.e.*, two or more distributions. Active Learning may also suffer from the same conditional shift problem due to the covariate shift assumption. For example, Yan et al. [26] investigate the learning when active learning starts from a pre-collected batch data, the learner suffers from covariate shift. Their active querying from the unlabeled data distribution is similar to querying from the target dataset in our context.

Active learning for domain adaptation. Persello and Bruzzone [27] proposed a two-direction AL algorithm for DA: query the most informative from the target domain and remove the most strange features out of the source domain. Wang et al. [28] proposed the active transfer technique for the model shift problem while assuming the shifts are smooth and implemented conditional distribution matching algorithm and off-set algorithm to modeling the source and target tasks via comparing the Gaussian Distributions. Zhang et al. [29] proposed a distribution correction algorithm over kernel embeddings to handle the target shift. The last two methods held on the assumption that there existed an affine transformation of conditional distribution from the source to target. Su et al. [30] proposed an active learning method using $\mathcal{H}$ divergence and rooting in the importance sampling technique to query the target instances. However, the importance sampling, query strategy they adopted, assumed that $\text{supp}(\mathcal{T}) \subseteq \text{supp}(\mathcal{S})$, may not hold in many DA settings. Furthermore, our method is based on the Wasserstein adversarial training method which could predict and constant critic score. On the contrary, the domain adversarial neural network (DANN) [14] training method, which is adopted by the AADA method, predicts the domain label under a binary classification mode to distinguish the instances from source or target domain, which restricts the power of active training.

3. Problem Setup

Notations and basic definitions. Follow [5], we consider a classification task, denote $\mathcal{X}$ and $\mathcal{Y}$ as the input and output space. A learning algorithm is then provided with a *labeled source dataset* $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m_s}$ consisting of m_s examples drawn *i.i.d.* from $\mathcal{S}_{\mathbf{x}\times y} \sim \mathcal{D}_s$ and an unlabeled target dataset $T = \{\mathbf{x}_j\}_{j=1}^{m_t}$ consisting of m_t examples drawn *i.i.d.* from $\mathcal{T}_{\mathbf{x}}$, where $\mathcal{S}_{\mathbf{x}\times\mathbf{y}}$ is the joint distribution on $\mathbf{x}\times y$ and $\mathcal{T}_{\mathbf{x}}$ is the marginal target distribution on $\mathbf{x}$, respectively. The expected source and target risk of $h \in \mathcal{H}$ over $\mathcal{S}$ (respectively, $\mathcal{T}$), are the probabilities that h errs on the entire distribution $\mathcal{D}_S$ (respectively, $\mathcal{D}_T$): $\epsilon_{\mathcal{S}}(h) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{S}}\mathcal{L}(h(\mathbf{x}, y))$ and $\epsilon_{\mathcal{T}}(h) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{T}}\mathcal{L}(h(\mathbf{x}, y))$, where $\mathcal{L}(\cdot)$ is the loss function. The goal of DA is to build a classifier $h \in \mathcal{H} : \mathcal{X} \to \mathcal{Y}$ training on source domain with a low *target risk* $\epsilon_{\mathcal{T}}(h)$.

3.1. Optimal transport and Wasserstein Distance

Optimal Transport (OT) theory [31] and Wasserstein Distance based adversarial training [5,15] were recently widely investigated in machine learning [22,32–35] especially in the domain adaptation area [15,36]. We adopted the OT and Wasserstein adversarial training, which is implemented to align the feature distribution for the unsupervised domain adaptation stage (first stage). OT could constrain labeled source samples from the same category to keep close with each other during the transportation process [36], which could help to alleviate the semantic misalignment problem during the adversarial training process [34]. Besides, compared with some other information theoretical metrics, such as KL divergence, which is not capable to measure the inherent geometric relations among the different domains [32], OT is capable to exactly measure their corresponding geometry properties of each domain. Furthermore, compared with $\mathcal{H}$ divergence [4], Wasserstein distance has better gradient property [32] and has promising generalization bound [5].

We follow Redko et al. [5] and define $c : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$ as the cost function for transporting one unit of mass $\mathbf{x}$ to $\mathbf{x}'$, then Wasserstein Distance could be computed by

$$W_p^p(\mathcal{D}_i, \mathcal{D}_j) = \inf_{\gamma\in\Pi(\mathcal{D}_i,\mathcal{D}_j)} \int_{\mathcal{X}\times\mathcal{X}} c(\mathbf{x}, \mathbf{x}')^p d\gamma(\mathbf{x}, \mathbf{x}')$$

where $\Pi(\mathcal{D}_i, \mathcal{D}_j)$ is the joint probability measures on $\mathcal{X}\times\mathcal{X}$ with marginals $\mathcal{D}_i$ and $\mathcal{D}_j$ referring to all the possible coupling functions. Throughout this paper, we shall use Wasserstein-1 distance only ($p = 1$). According to *Kantorovich–Rubinstein* theorem, let f be a Lipschitz-continuous function $\|f\|_L < 1$, we have

$$W_1(\mathcal{D}_i, \mathcal{D}_j) = \sup_{\|f\|_L<1} \mathbb{E}_{x\in\mathcal{D}_i} f(x) - \mathbb{E}_{x'\in\mathcal{D}_j} f(x') \tag{2}$$

In practice, we could implement a deep neural network to approximate function f. Then, computing the sup in Eq. (2) is to find out the maximum of $W-1$ distance by arg max operator through the general deep neural network optimizer (*e.g.* SGD [37] or Adam [38] optimizer). This allows us to compute the Wasserstein distance efficiently and the complexity $w.r.t$ $f(x)$ is only $\mathcal{O}(n+m)$.

3.2. Conditional shift and error bound

As stated before, traditional DA researches [4,14,23] typically assumed that the conditional relationships remain unchanged during the adaptation process. From a probabilistic perspective, the general learning process of most previous DA approaches is to learn the joint distribution of the target domain $\mathbb{P}_T(\mathbf{x}, y)$ through source domain joint distribution $\mathbb{P}_S(\mathbf{x}, y)$. Note that $\mathbb{P}_T(\mathbf{x}, y) = \mathbb{P}_T(\mathbf{x}|y)\mathbb{P}_T(\mathbf{x})$, to guarantee a successful transfer from source domain S to target domain T, the underlying assumption is $\mathbb{P}_S(y|\mathbf{x}) \approx \mathbb{P}_T(y|\mathbf{x})$. Recently, Wen et al. [11] showed that such condition is not sufficiently hold.

For the conditional shift situation, $\mathbb{P}_S(y|\mathbf{x}) \neq \mathbb{P}_T(y|\mathbf{x})$, Zhao et al. [6] theoretically showed that such a conditional shift problem exists in many situations and that typically if we only try to minimize the source error together with the domain distances, the target error might increase, which shall hinder the adaptation process. Their analysis was based on $\hat{\mathcal{H}}$ divergence, which is somehow hard to compute in deep learning based methods. In order to be coherent with our proposed work, we now present it using Wasserstein Distance with the following Theorem 1.

Theorem 1. *Let $\langle\mathcal{D}_S, f_S\rangle$ and $\langle\mathcal{D}_T, f_T\rangle$ be the source and target distributions and corresponding labeling function, if the hypothesis h is 1-Lipschitz and the loss function is $0-1$ loss, then we have*

$$\epsilon_{\mathcal{T}}(h) \le \epsilon_{\mathcal{S}}(h) + 2W_1(\mathcal{D}_s, \mathcal{D}_t) + \mathbb{E}_{\mathcal{D}_S}\left[\left|f_S - f_T\right|\right] \tag{3}$$

The proof is illustrated in the supplementary materials. This theorem showed that error on the target domain is decided by source domain error, Wasserstein Distance between source and target, and the conditional distribution on both source and target domains. Here the third term is not measurable in the unsupervised domain adaptation setting. If the conditional distribution changes during the adaptation process, then the target error may diverge [6]. One direct approach to reduce the disagreement between f_S and f_T is to partially acquire the labeling function f_T, *i.e.*, the labels in the target domain.

Besides, the Wasserstein distance between the source and target distribution (second term in Eq. (3)), is measured by total transportation cost between the source and target domain. Denote $\mathcal{D}_U$ and $\mathcal{D}_L$ by the corresponding distributions of unlabeled and labeled datasets, then the Wasserstein distance is denoted by:

$$W_1(\mathcal{D}_U, \mathcal{D}_L) = \inf_{\gamma\in\Pi(\mathcal{U},\mathcal{L})} \int_{\mathcal{X}\times\mathcal{X}} c(\mathbf{x}_l, \mathbf{x}_u) d\gamma(\mathbf{x}_l, \mathbf{x}_u)$$

Intuitively, if we can query some instances in the target domain $\mathcal{T}$ ($\mathcal{D}_U$) and move them from target into the source domain $\mathcal{S}$ ($\mathcal{D}_L$), we can reduce the total transportation cost between the two domains, *i.e.*, the Wasserstein distance between the two domains.

Based on this, to minimize the RHS of Eq. (3) is equivalent to train a learner $h \in \mathcal{H}$ that: (1) minimize the source error; (2) train a critic to estimate the empirical Wasserstein Distance between the source and target domain and approximately find a feature extractor that can minimize the total transportation cost between the source and target domain in an adversarial way with the critic; (3) can query the labeling information in the target domain so that to minimize the disagreement of labeling function between the source and target domain *i.e.*, the third term of Eq. (3).

To this end, we argue that if the learner can actively query labeling information in the target domain, then, it can partially get the conditional information in the target domain. With the minority of labeled target instances in hand, it can learn to jointly minimize the error both on the source and target domain. Furthermore, to *i.i.d.* query the label is somehow slow. In order to reduce the annotation expense, we may expect the learner to query some informative instances using an active learning strategy. Also, if the queried instances in the target domain are informative enough, they will have a better representative property on the target domain. Then, the learner can have better generalization performance on the target domain.

We then could prove that the error after such active selection could be bounded by the following theorem,

Theorem 2. *Assume the learner has a budget β of total target samples to query the oracle for ground truth label. Let $\mathcal{X}_s$ and $\mathcal{X}_t$ be two sample sets with size m_s and m_t drawn i.i.d. from $\mathcal{D}_S$ and $\mathcal{D}_T$ respectively. Let $\hat{\mathcal{D}}_S = \frac{1}{m_s}\sum_{i=1}^{m_s}\Delta_{x_i}^s$ and $\hat{\mathcal{D}}_T = \frac{1}{m_t}\sum_{i=1}^{m_t}\Delta_{x_i}^t$ be the associated empirical measure. Then $\forall d' \geq d$ and $\lambda' < \lambda$ there exists some constant N_0 depending on d' such that for any $\delta > 0$ and $\min(m_s, m_t) \geq N_0 \max(\delta^{-(d'+2),1})$ with probability at least $1-\delta$ for all hypothesis $h \in \mathcal{H}$ the following holds,*

$$\epsilon_t(h) \leq \epsilon_s(h) + 2W_1(\hat{\mathcal{D}}_s, \hat{\mathcal{D}}_t) + \mathbb{E}_{\mathcal{D}_S}\left[\left|f_S - f_T\right|\right] + 2\sqrt{2\log(\frac{1}{\delta})/\lambda'}\left(\sqrt{\frac{1}{N_s+\beta N_t}} + \sqrt{\frac{1}{N_t-\beta N_t}}\right) \tag{4}$$

Take those above into consideration, we can formally propose the discriminative active domain adaptation method.

4. Active discriminative domain adaptation

The learning process mainly consists of three main stages. The three-stage scheme is designed to firstly implement adversarial training to learn domain invariant features through OT. The second stage is to actively query the most informative instances on the invariant feature space. Finally, those informative instances could be used for retraining the network to reinforce the importance of the target features. We will introduce them in details.

4.1. Stage 1: Domain adversarial training via optimal transport

For the first stage, we adopt *Wasserstein Distance Guided Representation Learning* [15] method for adversarial training. The network receives a pair of instances from the source and target domain. Denoted by F and C the feature extractor and classifier, parameterized by θ_f and by θ_c, respectively. The feature extractor is trained to learn invariant features, and the classifier is expected to learn the conditional prediction relations $\mathbb{P}(Y|\mathbf{X})$ for predicting the instances from both source and target domain correctly. For the classification loss, we employ the traditional cross-entropy loss: $\mathcal{L}_{cls} = -\sum_{i=1}^{m} y_i \log(\mathbb{P}(C(F(\mathbf{x}_i))))$.

Then, there follows the domain critic network D, parameterized by θ_d. It estimates the empirical Wasserstein Distance between the source and target domain through a pair of batched instances $\mathcal{X}_S$ and $\mathcal{X}_T$,

$$W_1(\mathcal{X}_S, \mathcal{X}_T) = \frac{1}{n_s}\sum_{\mathbf{x}_s \in \mathcal{X}_S} D(F(\mathbf{x}_s)) - \frac{1}{n_t}\sum_{\mathbf{x}_t \in \mathcal{X}_T} D(F(\mathbf{x}_t)) \tag{5}$$

The feature extractor F is then trained to minimize the estimated Wasserstein Distance in an adversarial manner with the critic D. Then, goal of first stage training is described by

$$\min_{\theta_f, \theta_c} \max_{\theta_d} \mathcal{L}_{cls} + \lambda_w(W_1(\mathcal{X}_S, \mathcal{X}_T) - \mathcal{L}_{grad}) \tag{6}$$

where λ_w is a trade-off coefficient and $\mathcal{L}_{grad}$ is the gradient penalty term suggested by Gulrajani et al. [39]. When computing the gradient of such loss function, we use the gradient penalty method suggested in Gulrajani et al. [39] which can help to prevent gradient vanishing or exploding problems caused by weight clipping.

$$\mathcal{L}_{grad} = (\|\nabla_{F(X)} D(F(x))\|_2 - 1)^2 \tag{7}$$

The source and target features (marginal distributions) could be aligned via such an adversarial training process (Eq. (6)). Then, based on this aligned marginal distribution, we can implement the active strategy to query the most informative target instances

4.2. Stage 2: Active query with wasserstein critic

For the second stage, we hope the active leaner can find out the most informative features among the unlabeled target so that it could leverage from the labeling information of the target domain. The informative features, intuitively, are the *ones most different from what the learner has already known*. Intuitively, the hardest instances to adapt are those with least confidence (*i.e.* the most uncertain ones) to predict based on current classifier. As pointed out in previous work [20], only focus on the uncertainty might lead to the *sampling bias*. In order to reduce the sampling bias, the active learner shall also search some target samples with high diversity. We therefore find the most informative target samples holding both uncertainty and diversity properties.

Prediction uncertainty. The conditional prediction $\mathbb{P}_T(Y|\mathbf{X})$ is learned by the classification network. To measure the uncertainty, we adopt the entropy measure to quantify the uncertain of the classifier. The uncertainty entropy measure over an instance $\mathbf{x}_t$ is denoted by

$$\mathcal{U}(y_t|\mathbf{x}_t) = \mathcal{H}(\hat{\mathbb{P}}(y_t|\mathbf{x}_t)) \tag{8}$$

where $\mathcal{H}(\cdot)$ is the information entropy measure, $\hat{\mathbb{P}}(y_t|\mathbf{x}_t)$ is the output of classification network $\hat{\mathbb{P}}(y_t|\mathbf{x}_t) = C(F(\mathbf{x}_t))$.

Diversity by critic function. If the some instances, in terms of distribution distance measures, are very far from the unknown labeled ones, then they should contain most informative and diverse features from the known labeled ones. Recall that in the first stage, we match the marginal distribution between the source and target domain to achieve a domain invariant feature space with Wasserstein Distance. Then, for the target domain instances, the one with highest critic score is the one that have the highest transportation cost.

Sinha et al. [21] and Shui et al. [22] showed that such critic term $D(F(\cdot)) : \mathcal{X} \to [0, 1]$ indicates the diversity in the query process. Then, we can leverage from the trained Wasserstein Critic network to evaluate and find out the most informative (diverse) target features on the invariant feature space. That is, measuring the diversity of target instances via critic score. Consider the critic output of a target instance $\mathbf{x}_t$, if $D(F(\mathbf{x}_t)) \to 1$, then $\mathbf{x}_t$ is far, $w.r.t.$ Wasserstein Distance, from the source domain images and if $D(F(\mathbf{x}_t)) \to 0$, then $\mathbf{x}_t$ is near to the source images.

Based on those above, if we hope to find out the most informative (uncertain and diverse) instances in the target domain, then we should query by controlling two terms:

- uncertainty score $\mathcal{U} = \mathcal{H}(\hat{\mathbb{P}}(y_t|\mathbf{x}_t))$ defined by Eq. (8), which indicates the uncertainty of the classifier to predict a label $y_t^?$ given the instance $\mathbf{x}_t$ in the target domain
- critic score $D(F(\mathbf{x}_t))$ by the Wasserstein critic function, which indicates the diversity of the unlabeled target instance compared with the source labeled ones.

Then, we shall have the following objective

$$\text{argmax}_{x_t \in \mathcal{X}_t} \mathcal{U}(y_t^?|\mathbf{x}_t) - \lambda_{div} D(F(\mathbf{x}_t)) \tag{9}$$

where λ_{div} is a coefficient to regularize the Wasserstein critic term. So, for a query budget β and m_t of target set instances, the query process could be described as: *looking for $m_q = \beta m_t$ instances by solving Eq. (9) and query the labels of those m_q instance from the oracle*. Denote the queried set by $Q = \{(\mathbf{x}_1^q, y_1^q), \ldots, (\mathbf{x}_{m_q}^q, y_{m_q}^q)\}$. Then, uniting such small batch instances

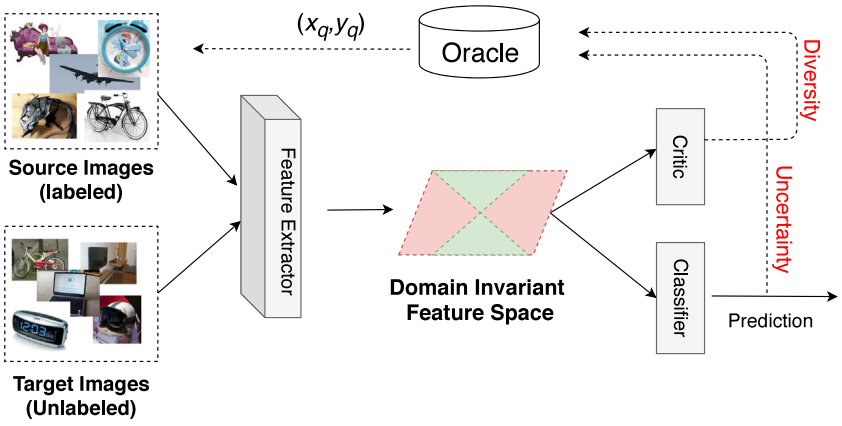

Fig. 1. Ac-DA workflow: feature extractor are trained to learn a domain invariant feature space together with the critic. The learner selects the informative instances by measuring *uncertainty* and *diversity* based on critic and classifier outputs.

Algorithm 1 The Active Discriminative Domain Adaptation

Input: Source and target domain input S, T; Query budget β
Parameter: Feature extractor θ_f; Classifier θ_c; Critic θ_d
Output: Optimized $\theta_f^\star$, $\theta_c^\star$, $\theta_d^\star$

while Domain level adaptation not finish **do**
  Sample batches $(\mathbf{x}_s, y_s) \sim S$, $\mathbf{x}_t \sim T$
  Train the network based on Eq. (6) until converge
end while
if Query budget is not empty **then**
  Select the target instances $\{\mathbf{x}_1^q, \ldots, \mathbf{x}_{m_q}^q\}$ according to Eq. (9) and query the label $\{y_1^q, \ldots, y_{m_q}^q\}$ from oracle.
else
  Update the dataset $Q = \{(\mathbf{x}_1^q, y_1^q), \ldots, (\mathbf{x}_{m_q}^q, y_{m_q}^q)\}$, $S' = S \cup Q$, $T' = T/Q$.
end if
Compute the uncertainty vector $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_C]_{j=1}^C$ with Eq. (10)
Train the network on new labeled and unlabeled dataset via domain adaptation techniques with Eq. (11).
return solution

with the source domain and removing them from the target domain. The source and target datasets shall be updated as: $S' = S \cup Q$, $T' = T/Q$. We illustrate a general query workflow in Fig. 1.

4.3. Stage 3: DA training with new dataset

The goal of our proposed method is to leverage the most informative instances in the target domain to reinforce the adaptation process. General adversarial training methods for domain adaptation usually assign each instance with the same importance weight. In order to enforce the uncertainty information to the classifier, we hope to give higher weights to the instances with higher uncertainty scores during the supervised classification process.

Denote by a set of m_q queried instances $\{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^{m_q}$, we shall re-weight the importance of each instance classes based on their uncertainty score. Denote by uncertainty vector $\boldsymbol{\alpha} = [\alpha_1 \ldots \alpha_j \ldots \alpha_C]_{j=1}^C$ over all C classes. For each class j, the weight is computed by,

$$\alpha_j = \frac{N_j \cdot \mathcal{U}(y_j|\mathbf{x})}{\sum_{i=1}^{m_q} \mathcal{U}(y^{(i)}|\mathbf{x})} \tag{10}$$

where N_j is the number of instances with label y_j, $\mathcal{U}(\cdot)$ is the uncertainty score defined in Eq. (8).

For a batch of queried instances, the weighted crossentropy loss could be computed by

$$\mathcal{L}_w^q = \alpha_j(-y_j \log(\sum_{j=1}^{C} \exp(\mathbb{P}(y_j|\mathbf{x}))))$$

Then, objective function for the third stage is,

$$\min_{\theta_f, \theta_c} \max_{\theta_d} \mathcal{L}_w^q + \mathcal{L}_{cls} + \lambda_w(W_1(\mathcal{X}_S', \mathcal{X}_T') - \mathcal{L}_{grad}) \tag{11}$$

where $\mathcal{X}_S'$ and $\mathcal{X}_T'$ are sampled from the updated source and target datasets, $\mathcal{L}_{cls}$ is the classification loss on the original source set and $\mathcal{L}_w^q$ is the weighted loss for the query set. Finally, we illustrate our Active Discriminative Domain Adaptation (Ac-DA) algorithm in Algorithm 1

5. Experiments and results

In the experiments part, we aim to demonstrate the following aspects: Firstly, we would like to show that even randomly select a few amount of labeled target data can improve the performance compared with the unsupervised counterparts. Secondly, we show that compared with the semi-supervised method, *i.e.*, the random (*i.i.d.*) selection, the active learning query strategy could have more benefits. Thirdly, in order to confirm the effectiveness of our method comparing with another active domain adaptation method, and also other query strategies with certain query budget for further analysis.

For the comparison with unsupervised DA methods, we evaluate the performance of the proposed algorithm on four benchmark datasets and compared with some other approaches: Wasserstein Distance Guided Domain Adaptation (WDGRL [15]), Domain Adversarial Neural Networks (DANN [14]), Adversarial Discriminative Domain Adaptation (ADDA [23]) and Conditional Adversarial Domain Adaptation (CDAN [24]). In order to show the benefits of active query method, we also compare the results with random selection process when the query budget is the same. To confirm the effectiveness of our query method, we also compare our method with different query strategies. All experiments are programmed by *Pytorch*.

Table 1
Classification accuracy (%) on **digits datasets** with different adaptation tasks. The last two line are our method, Random refers to randomly query some instance while Ac-DA is the proposed approach. Both two methods are restrict to 10% query budget.

Method	M → MM	M → U	U → M	avg.
LeNet5	56.1	67.4	65.3	60.3
DANN	74.2	77.1	73.2	74.6
WDGRL	80.3	81.1	74.2	76.2
ADDA	78.9	83.5	82.3	81.5
Rand.	92.4	**95.7**	95.8	94.7
Ac-DA	**95.4**	95.5	**96.5**	**95.6**

Table 2
Classification accuracy (%) on **Office-31** dataset with different adaptation settings with 10% query budget.

Method	A → W	A → D	D → A	W → A	avg.
ResNet50	68.6	69.3	61.1	60.7	64.9
DAN	80.5	78.6	63.6	60.7	62.7
DANN	81.3	79.2	68.2	67.4	74.0
WGDRL	79.2	80.2	69.3	69.1	74.5
Rand.	86.1	85.6	76.3	78.1	81.6
Ac-DA	**86.6**	**87.7**	**78.5**	**80.2**	**83.3**

5.1. Datasets and implementations

We test our proposed algorithm on four benchmark datasets.

Digits Datasets: We test our algorithm on digits datasets with the experiments setting : USPS (U)↔ MNIST (M) and MNIST → MNIST-M (MM). For USPS we resize the images to size 28 × 28. We train the network using training sets with size: MNIST/MNIST-M(60k), USPS(7,291) and testing sets with size: MNIST/MNIST-M (10k), USPS(2,007).

Office-31 dataset is a standard benchmark for domain adaptation evaluations. It contains three different domains: Amazon (A), Dslr (D) and WebCam (W), with 31 categories in each domain. We report the average results in Table 2.

Office Home dataset: is more challenging than Office-31, contains four different domains: *Art* (Ar), *Clipart* (Cl), *Prodcut* (Pr) and *Real World* (Rw), with 65 categories in each domain. We report the average results in Table 3.

Image-CLEF 2014 dataset contains three domains, which are *Caltech-256*(C), *ILSVRC-2012*(I), and PascalVOC-2012(P), with 12 common shared categories. We report the average results in Table 4

For digits datasets, we do not apply any data-augmentation. For Office-31, Office-Home and Image-CLEF datasets, we apply the following pre-processing pipeline: (1) for training set, firstly resize the image to 256 × 256 then, apply *RandomCrop* downgrade the size to 224 × 224, after that, apply the random flipping strategy; (2) for testing set, resize the images to 256 × 256 then use *CenterCrop* to size 224 × 224.

CNN architecture and implementations. For digits experiments, we adopt *LeNet-5* as feature extractor and trained from scratch. For the rest three real-world datasets, we implement ImageNet pretrained *ResNet-50* as feature extractor. For the digits experiments, we train the network with mini-batch size 64 and for the rest three datasets with mini-batch size 16. We adopt Adam optimizer for training the network. For stable training, we set $\lambda_w = \frac{2}{1+\exp(-10 \cdot p)} - 1$, and p is the training progress. Also, we empirically set $\lambda_{div} = 10$. To avoid over-training, we also adopt early-stopping technique.

5.2. Results and analysis

We illustrate the T-SNE visualization comparison of non-adaptation setting and our proposed approach Ac-DA (see Fig. 2). We can observe that our proposed method has a good alignment performance. We report the average results of our proposed algorithm and baselines using our data pre-processing pipeline on Digits, Office-31, Office-Home and Image-CLEF datasets in Tables 1–4, respectively. In order to show the effectiveness of active query strategy, for a given budget, we also implemented *random (i.i.d.) selection* method to query the labels for compassion. The name of such implementations are denoted by *rand.* and *Ac-DA* in each table. In Table 5, we also compared the performances under different budget.

Value of target labels. From the tests results on the four benchmark datasets, we could observe that the to randomly select some instances in the target domain could benefit the classification performance on the target domain. Our method is rooted in WDGRL, comparing accuracy performance between the *random selection* with WDGRL we could observe obvious improvements on the benchmark datasets, which confirms the usefulness of label information for adaptation. Also, for each adaptation task on every dataset, we can observe that the proposed Ac-DA algorithm outperforms the random selection method in almost all the tasks. This also confirms that active query can outperform *i.i.d.* selection.

Effectiveness of active query. We then compared the performance between active query and random selection. We implement the experiments with different query budgets (with 5%, 10% and 15%). The average accuracy on the different datasets is reported in Table 5. We can observe that the accuracy will increase as the query budget increases. Also, with the same query budget, we compare the accuracy of active query and random selection. We can observe that the active query method can outperform the random query method with query budget 5% and 10%. That is, *with a smaller query budget, the active query strategy can have better performance than random selection*. This confirms the effectiveness of the active query strategy. When the query budget goes to 15%, the differences between the random selection and active query become smaller. One interpolation is that as the query budget increase, more instances in the target domain will be labeled, and those most informative ones will be covered with high probability. When the query budget is relatively small, the active strategy can exactly look for the most informative instances rather than uniformly (random) selecting some instances.

Comparison with different query strategies. In order to evaluate the effectiveness of our method, we compare our method with the active domain adaptation baselines. To the best of our knowledge, Adversarial Active Domain Adaptation (AADA) [30] is the only similar baseline to our proposed method. AADA is based on the DANN [14] as the adversarial training mode, where the invariant features are learned by fooling the domain discriminator D by a binary classification to predict the instances are from source or target domain. Upon the submission of our work, the official code release of AADA has not yet been published. We reproduced the baseline by religiously following the original implementation while made some adaptations to our setting for a fairer comparison. The original AADA implementation selected certain target instances and retrained the model under a few-shot mode with several query rounds. For a fairer comparison with our proposed method, we reproduced the AADA with the similar setting with ours by selecting a certain ratio of instances when the first stage of adversarial training is stable. The reproduced AADA model was based on the DANN implementations. We follow Su et al. [30] to construct the uncertainty cue and diversity cue implementation by scoring the instances with query strategy:

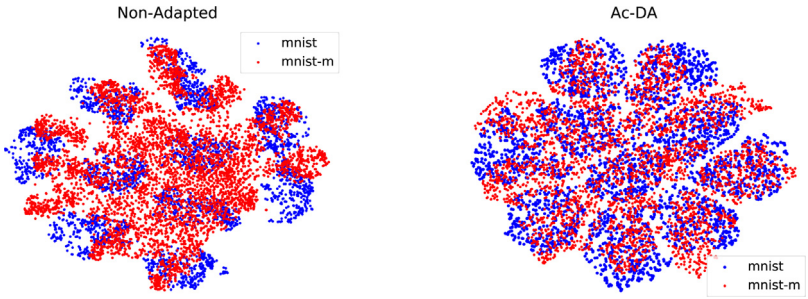

Fig. 2. T-SNE visualization between our proposed Active Discriminative Domain Adaptation (right, with 5% query budget) and non-adapted setting (left) for MNIST → MNIST-M adaptation task.

Table 3
Classification accuracy (%) on **Office Home** dataset with different adaptation settings with query budget 10%.

Method	Ar → Cl	Ar → Pr	Ar → Rw	Cl → Ar	Cl → Pr	Cl → Rw	Pr → Ar	Pr → Cl	Pr → Rw	Rw → Ar	Rw → Cl	Rw → Pr	avg.
ResNet50	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
DANN	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
WGDRL	42.6	57.9	69.3	47.3	59.5	63.4	46.2	41.3	67.4	62.4	52.8	74.9	57.1
CDAN	49.0	69.2	74.5	54.4	66.0	68.4	55.6	48.3	75.9	68.4	55.4	80.5	63.8
Rand.	56.9	76.4	76.3	**61.7**	78.1	73.3	57.8	**56.9**	74.2	68.5	60.3	83.2	68.6
Ac-DA	56.8	80.3	80.8	67.2	80.0	78.4	64.8	57.5	80.1	75.9	62.8	88.7	**72.7**

Table 4
Classification accuracy (%) on **Image-CLEF dataset** with different adaptation tasks under 10% query budget.

Method	C → I	C→P	I → P	I → C	P → C	P → I	avg.
ResNet50	76.4	62.5	73.2	89.3	90.3	79.8	78.5
DANN	84.8	72.6	73.8	92.8	91.5	81.9	82.9
WDGRL	82.3	70.8	73.9	90.7	91.3	85.4	82.4
Rand.	89.8	75.0	78.2	94.4	**94.9**	89.9	87.1
Ac-DA	**91.1**	**76.3**	**80.8**	**96.7**	94.7	**94.2**	**88.9**

Table 5
Comparison of different query budgets (5%, 10%, 15%) on three datasets. For each query budget, we report the improvements by applying the active query strategy comparing with the random query strategy in the parentheses.

budget	Digits		Office-Home		Image-CLEF	
	Rand.	Ac-DA	Rand.	Ac-DA	Rand.	Ac-DA
5%	91.6	92.9(+1.3)	62.4	65.6(+3.2)	82.2	84.9(+2.7)
10%	94.7	95.6(+0.9)	68.6	72.7(+4.1)	87.1	88.9(+1.8)
15%	96.2	96.9(+0.7)	73.9	75.8(+1.9)	89.8	90.4(+0.6)

$score(x) = \mathcal{H}(\hat{y})w(z)$, where $\hat{y} = C(F(x))$ is the model prediction and $\mathcal{H}(\cdot)$ is the information entropy, while $w(x) = (1-D(z))/D(z)$, where $z = F(x)$ is the extracted feature, involved the diversity cue. We compare the performance of re-implemented AADA and our method on the Office-home dataset with different query budgets and illustrate the performance of each adaptation task average accuracy in Table 6.

In order to evaluate the effectiveness of our query strategy, we compare the performance of the active domain adaptation algorithm with different query strategies,*i.e.*, we implement some baselines by replacing the query strategy in Eq. (9) with the following query strategies:

- Random sampling (Rand.): randomly select potential instances from the target domain.
- Least confidence (Lst. Conf.) [40]: select the instances with least confidence over the classifier.
- Smallest Margin (Marg.) [41]: select the instances via a defined margin.
- Maximum-Entropy sampling (Ent.) [42]: selecting the instances with the maximum entropy, *i.e.*, the most uncertain ones.
- K-Median (K-Ms.) [43]: choosing the points to be labeled as the cluster centers of K-Median algorithm

We evaluate the empirical results of the Wasserstein adversarial training with different query strategies and report the overall average of all tasks in Table 7. We then report the empirical results on different adaptation tasks in Fig. 3 by choosing random selection as baseline (set as 0) and show the differences. From the empirical results, we could observe that our method could always outperform the baseline query strategies under different query budgets *w.r.t.* the averaged accuracy. The most diverse performance occurs on the task Ar ↔ Cl. This may due to the features from these two domains look similar to each other. When querying some diverse and uncertain features, it may find some uncommon features which may hurt the learning performance. Besides, we could also observe that the performance of entropy sampling diverse a lot. Since entropy sampling means the learner only selects the instances with the most uncertainty, this may lead the learner to find some strange features, which may hurt the learning performance. Generally, our method could have a better averaged performance over all the query strategies under different query budgets.

6. Conclusion

We proposed a three-stage discriminative active algorithm to improve the domain adaptation performance. The first stage adopted general domain adversarial training. In the second stage, we proposed an end-to-end query strategy combining *uncertainty* and *diversity* criteria to find out the most informative features in the target domain. Finally, in the third stage, we deployed a re-weighting technique based on the prediction uncertainty for determining the importance of the queried samples to retrain the network. The empirical results confirmed the effectiveness of our active domain adaptation algorithm especially when the query budget is small.

Fig. 3. Comparison of different query strategies. We take random selection (set the baseline accuracy to 0) as the baseline and report the relative accuracy difference with different query strategies (1% ∼ 19%).

Table 6
Comparison of our method and the re-implemented AADA with different query budget.

budget	methods	Ar → Cl	Ar → Pr	Ar → Rw	Cl → Ar	Cl → Pr	Cl → Rw	Pr → Ar	Pr → Cl	Pr → Rw	Rw → Ar	Rw → Cl	Rw → Pr	avg.
5%	AADA	41.9	70.9	75.7	58.3	72.6	67.1	56.7	51.5	77.0	71.3	55.8	80.7	65.0
	Ours	40.0	71.5	76.3	62.0	72.8	68.0	56.7	52.1	77.6	72.2	56.2	80.5	**65.5**
10%	AADA	57.8	78.9	78.9	65.9	78.1	77.2	64.1	56.9	79.7	74.7	62.2	88.4	71.9
	Ours	56.8	80.3	80.8	67.2	80.0	78.4	64.8	57.5	80.1	76.0	62.8	88.7	**72.7**
15%	AADA	65.8	81.9	83.5	71.9	82.9	80.9	71.2	65.3	84.3	79.4	69.7	91.1	77.3
	Ours	66.0	84.1	84.8	71.1	83.1	81.8	71.8	64.8	84.9	80.1	69.2	91.3	**77.8**
20%	AADA	68.2	87.6	87.1	73.8	86.1	81.9	72.6	68.2	86.5	82.5	71.8	92.8	79.9
	Ours	68.9	87.4	87.4	74.7	87.2	83.0	73.4	69.1	87.0	83.1	72.6	93.4	**80.6**

Table 7
Averaged performance of different query strategies on Office home dataset with different query budget (from 1% to 19% of the total instances).

	1%	3%	5%	7%	9%	11%	13%	15%	17%	19%
Rand.	57.76	62.61	63.75	67.39	69.99	71.67	72.81	74.42	75.80	76.68
K-Ms.	57.06	62.94	66.13	68.59	70.81	72.13	74.10	75.60	76.60	77.78
Lst-Conf.	55.70	60.80	63.95	67.18	70.33	73.13	74.18	75.92	77.52	78.71
Marg.	58.20	63.14	67.22	69.78	72.05	73.89	75.90	77.17	78.50	79.91
Ent.	56.92	60.68	63.75	66.03	70.50	72.00	72.97	74.85	75.97	77.69
Ours	**59.27**	**64.01**	**67.61**	**69.98**	**72.29**	**74.43**	**75.84**	**77.80**	**79.06**	**80.40**

CRediT authorship contribution statement

Fan Zhou: Conceptualization, Methodology, Software, Validation, Formal analysis, Writing - original draft. **Changjian Shui:** Writing - review & editing. **Shichun Yang:** Resources, Writing - review & editing. **Bincheng Huang:** Writing - review & editing. **Boyu Wang:** Supervision, Writing - review & editing. **Brahim Chaib-draa:** Supervision, Writing - review & editing, Funding acquisition, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work has been partially supported by Natural Sciences and Engineering Research Council of Canada (NSERC). Fan Zhou is supported by China Scholarship Council. Boyu Wang is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), Discovery Grants Program.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.knosys.2021.106986.

References

[1] M. Wulfmeier, A. Bewley, I. Posner, Addressing appearance change in outdoor robotics with adversarial domain adaptation, in: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2017, pp. 1551–1558.
[2] B. Wang, J. Mendez, M. Cai, E. Eaton, Transfer learning via minimizing the performance gap between domains, Advances in Neural Information Processing Systems 32 (2019).
[3] K. You, M. Long, Z. Cao, J. Wang, M.I. Jordan, Universal domain adaptation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 2720–2729.
[4] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, J.W. Vaughan, A theory of learning from different domains, Mach. Learn. 79 (1–2) (2010) 151–175.
[5] I. Redko, A. Habrard, M. Sebban, Theoretical analysis of domain adaptation with optimal transport, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2017, pp. 737–753.
[6] H. Zhao, R.T. Des Combes, K. Zhang, G. Gordon, On learning invariant representations for domain adaptation, in: International Conference on Machine Learning, 2019, pp. 7523–7532.
[7] J. Jiang, C. Zhai, Instance weighting for domain adaptation in NLP, in: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, 2007, pp. 264–271.
[8] X. Glorot, A. Bordes, Y. Bengio, Domain adaptation for large-scale sentiment classification: A deep learning approach, in: Proceedings of the 28th International Conference on Machine Learning, ICML-11, 2011, pp. 513–520.
[9] R. Wang, M. Utiyama, L. Liu, K. Chen, E. Sumita, Instance weighting for neural machine translation domain adaptation, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 1482–1488.
[10] J. Wen, R. Liu, N. Zheng, Q. Zheng, Z. Gong, J. Yuan, Exploiting local feature patterns for unsupervised domain adaptation, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 5401–5408.
[11] J. Wen, N. Zheng, J. Yuan, Z. Gong, C. Chen, Bayesian Uncertainty matching for unsupervised domain adaptation, 2019, arXiv preprint arXiv:1906.09693.
[12] V.M. Patel, R. Gopalan, R. Li, R. Chellappa, Visual domain adaptation: A survey of recent advances, IEEE Signal Process. Mag. 32 (3) (2015) 53–69.
[13] M. Wang, W. Deng, Deep visual domain adaptation: A survey, Neurocomputing 312 (2018) 135–153.
[14] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, V. Lempitsky, Domain-adversarial training of neural networks, J. Mach. Learn. Res. 17 (1) (2016) 2030–2096.
[15] J. Shen, Y. Qu, W. Zhang, Y. Yu, Wasserstein distance guided representation learning for domain adaptation, in: AAAI Conference on Artificial Intelligence, 2018.
[16] I. Redko, E. Morvant, A. Habrard, M. Sebban, Y. Bennani, Advances in Domain Adaptation Theory, Elsevier, 2019.
[17] M. Mohri, A. Rostamizadeh, A. Talwalkar, Foundations of Machine Learning, MIT press, 2018.
[18] B. Settles, Active Learning Literature Survey, Technical Report, University of Wisconsin-Madison Department of Computer Sciences, 2009.
[19] D. Cohn, L. Atlas, R. Ladner, Improving generalization with active learning, Mach. Learn. 15 (2) (1994) 201–221.
[20] S. Dasgupta, Two faces of active learning, Theoret. Comput. Sci. 412 (19) (2011) 1767–1781.
[21] S. Sinha, S. Ebrahimi, T. Darrell, Variational adversarial active learning, 2019, arXiv preprint arXiv:1904.00370.
[22] C. Shui, F. Zhou, C. Gagné, B. Wang, Deep active learning: Unified and principled method for query and training, in: S. Chiappa, R. Calandra (Eds.), Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics, in: Proceedings of Machine Learning Research, vol. 108, PMLR, Online, 2020, pp. 1308–1318, URL: http://proceedings.mlr.press/v108/shui20a.html.
[23] E. Tzeng, J. Hoffman, K. Saenko, T. Darrell, Adversarial discriminative domain adaptation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7167–7176.
[24] M. Long, Z. Cao, J. Wang, M.I. Jordan, Conditional adversarial domain adaptation, in: Advances in Neural Information Processing Systems, 2018, pp. 1640–1650.
[25] D. Gissin, S. Shalev-Shwartz, Discriminative active learning, 2019, arXiv preprint arXiv:1907.06347.
[26] S. Yan, K. Chaudhuri, T. Javidi, Active learning with logged data, in: J. Dy, A. Krause (Eds.), Proceedings of the 35th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 80, PMLR, Stockholmsmässan, Stockholm Sweden, 2018, pp. 5521–5530, URL: http://proceedings.mlr.press/v80/yan18a.html.
[27] C. Persello, L. Bruzzone, Active learning for domain adaptation in the supervised classification of remote sensing images, IEEE Trans. Geosci. Remote Sens. 50 (11) (2012) 4468–4483.
[28] X. Wang, T.-K. Huang, J. Schneider, Active transfer learning under model shift, in: E.P. Xing, T. Jebara (Eds.), Proceedings of the 31st International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 32, (2) PMLR, Bejing, China, 2014, pp. 1305–1313, URL: http://proceedings.mlr.press/v32/wangi14.html.
[29] K. Zhang, B. Schölkopf, K. Muandet, Z. Wang, Domain adaptation under target and conditional shift, in: International Conference on Machine Learning, 2013, pp. 819–827.
[30] J.-C. Su, Y.-H. Tsai, K. Sohn, B. Liu, S. Maji, M. Chandraker, Active adversarial domain adaptation, in: The IEEE Winter Conference on Applications of Computer Vision, 2020, pp. 739–748.
[31] C. Villani, Optimal Transport: Old and New, vol. 338, Springer Science & Business Media, 2008.
[32] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein gan, 2017, arXiv preprint arXiv:1701.07875.

[33] C. Shui, M. Abbasi, L.-É. Robitaille, B. Wang, C. Gagné, A principled approach for learning task similarity in multitask learning, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, International Joint Conferences on Artificial Intelligence Organization, 2019, pp. 3446–3452, http://dx.doi.org/10.24963/ijcai.2019/478, https://doi.org/10.24963/ijcai.2019/478.

[34] F. Zhou, Z. Jiang, C. Shui, B. Wang, B. Chaib-draa, Domain generalization via optimal transport with metric similarity learning, Neurocomputing (ISSN: 0925-2312) (2021) http://dx.doi.org/10.1016/j.neucom.2020.09.091, URL: https://www.sciencedirect.com/science/article/pii/S0925231221002009.

[35] F. Zhou, C. Shui, M. Abbasi, L.-É. Robitaille, B. Wang, C. Gagné, Task similarity estimation through adversarial multitask neural network, IEEE Trans. Neural Netw. Learn. Syst. 32 (2) (2021) 466–480, http://dx.doi.org/10.1109/TNNLS.2020.3028022.

[36] N. Courty, R. Flamary, D. Tuia, A. Rakotomamonjy, Optimal transport for domain adaptation, IEEE Trans. Pattern Anal. Mach. Intell. 39 (9) (2016) 1853–1865.

[37] L. Bottou, Large-scale machine learning with stochastic gradient descent, in: Proceedings of COMPSTAT'2010, Springer, 2010, pp. 177–186.

[38] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.

[39] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A.C. Courville, Improved training of wasserstein gans, in: Advances in Neural Information Processing Systems, 2017, pp. 5767–5777.

[40] A. Culotta, A. McCallum, Reducing labeling effort for structured prediction tasks, in: AAAI Conference on Artificial Intelligence, 2005.

[41] T. Scheffer, S. Wrobel, Active learning of partially hidden markov models, in: Proceedings of the ECML/PKDD Workshop on Instance Selection, Citeseer, 2001.

[42] B. Settles, Active learning, Synth. Lect. Artif. Intell. Mach. Learn. 6 (1) (2012) 1–114.

[43] O. Sener, S. Savarese, Active learning for convolutional neural networks: A core-set approach, in: International Conference on Learning Representations, 2018, URL: https://openreview.net/forum?id=H1aIuk-RW.