Active Sampling Based on MMD for Model Adaptation

Qi Zhang[1,2], Donghai Guan[1,2(✉)], Weiwei Yuan[1,2], and Asad Masood Khattak[3]

[1] College of Computer Science and Technology,
Nanjing University of Aeronautics and Astronautics, Nanjing, China
{stonewell,dhguan,yuanweiwei}@nuaa.edu.cn
[2] Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing 210016, China
[3] College of Technological Innovation, Zayed University,
Dubai, United Arab Emirates
Asad.Khattak@zu.ac.ae

Abstract. In this paper, we demonstrate a method for transfer learning with minimal supervised information. Recently, researchers have proposed various algorithms to solve transfer learning problems, especially the unsupervised domain adaptation problem. They mainly focus on how to learn a good common representation and use it directly for downstream task. Unfortunately, they ignore the fact that this representation may not capture target-specific feature for target task well. In order to solve this problem, this paper attempts to capture target-specific feature by utilizing labeled data in target domain. Now it's a challenge that how to seek as little supervised information as possible to achieve good results. To overcome this challenge, we actively select instances for training and model adaptation based on MMD method. In this process, we try to label some valuable target data to capture target-specific feature and fine-tune the classifier networks. We choose a batch of data in target domain far from common representation space and having maximum entropy. The first requirement is helpful to learn a good representation for target domain and the second requirement tries to improve the classifier performance. Finally, we experiment with our method on several datasets which shows significant improvement and competitive advantage against common methods.

Keywords: Active sampling · Maximum mean discrepancy · Transfer learning · Characteristics · Uncertainty

1 Introduction

Recent years, deep learning have made great success in various applications across many fields. For example, there have been many CNN models such as AlexNet [12], GoogLeNet [21], ResNet [7] and so on, which have improved classification accuracy on images datasets. It's well known that training deep convolutional neural networks always need numerous labeled data. However, in many

X. B. Zhai et al. (Eds.): MLICOM 2019, LNICST 294, pp. 397–409, 2019.
https://doi.org/10.1007/978-3-030-32388-2_34

cases, there're not enough labelled data due to expensive cost and huge time consumption. There are two paradigms-transfer learning and active learning-which can effectively overcome that challenge.

Transfer learning or domain adaptation [16] tries to tackles this problem by transferring knowledge from a label-rich and similar domain (known as source domain) to a label-scarce domain (known as target domain). At present, researchers mainly devoted much attention to unsupervised domain adaptation assuming that there are sufficient labeled data in source domain and no labeled data in target domain. However, different domains have their own characteristics so that the model learned only from labeled source data can't generalize well to the target domain. It's easily to consider obtaining labeled target domain data to solve this problem. So, the following question is can we use as small amount of supervised information of target domain as possible while keeping the model's performance improving. That encourages us to combine transfer learning with active learning in CNN model.

Active learning [19] is one of effective paradigms to reduce the labeling cost. Its basic assumption is that different data has different amount of information. Therefore, people can query the most valuable instance to label and obtain considerable performance's improvement. Based on this idea, various criteria have been proposed to evaluate data's value. For instance [10], informativeness and representativeness are two frequently used criteria to choose data. [1,2] proposed an method that select data based on marginal distribution matching. When it comes to transfer learning, we know that there must exist distribution shift. So, many traditional active learning methods can't adapt to such case well while distribution based sampling strategy will be a good choice. What's more, our classifiers are typical CNN models. Hence, researchers should consider both network architecture and datasets shift when designing strategy for selecting data. Recently, there are several works [3,24] to combine active learning and deep neural networks. However, their methods can not be adapted to transfer learning models well and thus may lead to waste of annotated data by learning from scratch.

In common transfer learning process, initially, we can use pre-trained model as a backbone and then use labeled source data and unlabeled target data to update the parameters of the model. These two steps are so called unsupervised domain adaptation. But it's well known that the insight behind transfer learning or unsupervised domain adaptation is we can learn the common parts from similar domains. So, if the target domain is not much similar to source domain, the learned common representation may be unsatisfactory for target task. To address this problem, we attempt to provide as little supervised information in target domain as possible to capture the good representation beneficial to target domain task. In this paper, we propose to a new method based on maximum mean discrepancy (MMD) [6,9], which can effectively select data based on distribution to train models. The intuition behind the method is that we can choose data in target domain most dissimilar to source domain to capture the distinctive information in target task.

Eventually, We perform experiments on several datasets. The results demonstrate that our approach can effectively learn distinctive representation for target task and significantly improve accuracy with lower labelling cost comparing with other method such as random sampling, entropy-based sampling and ADMA [11]. The main contribution of our paper are summarized as follows:

- We utilize MMD method which can identify data's distinctiveness for target task, and based on this we can choose the most valuable data for query.
- Our selecting strategy can adapt well to distribution shift scenario.
- We evaluate our approach on various datasets and achieve a satisfactory results.

The rest of this paper is organized as follows. In Sect. 2, we presents a brief review of related work. Section 3 introduces the background knowledge of MMD. Section 4 discusses the detailed components of our approach and the corresponding algorithm. Section 5 demonstrates the experimental results and the corresponding empirical analysis. Section 6 makes a conclusion of this paper.

2 Related Work

Domain adaptation is one of hot topics in transfer learning, especially unsupervised domain adaptation(UDA) which attracts many people's attention. Towards transfer learning paradigm, it aims to match distributions between source data and target data with smaller loss after feature transformation. To tackle this problem, the core is how to measure the difference or loss between source domain and target domain after feature transformation. There are about three ways to deal with it—discrepancy loss, reconstruction loss and adversarial loss. The first one [14,15,23] often utilize MMD criterion, MMD computes the norm of the difference between kernel mean embedding in two domains. The DDC method [23] shares common features in low level across different domains but adds adaptation layers in high level layers using MMD to minimize distance between two domains. The deep adaptation network (DAN) [14] uses MMD when task-specific layers embedded in a RKHS where the mean embedding of different domain distributions can be explicitly matched. But the method only considers the marginal distribution matching. To deal with this problem, joint adaptation networks (JAN) [15] was proposed to learn a transfer network by aligning joint distributions of multiple domain-specific layers across domains based on a joint MMD criterion. The second [5] proposed an auto-encoder based framework for domain adaptation by simultaneously minimizing the reconstruction loss of the auto-encoder and the classification error. The last [4,8,22] using adversarial method currently is the mainstream, which has chosen an adversarial loss to minimize domain distribution shift distance. In this method, the network adds a module discriminator that discriminates the learned representation coming from target data or source data. If the discriminator can't distinguish well, then we can admit that source domain and target domain are aligning. For instance, the

DANN [4] introduces a domain classifier with binary labels to distinguish the source domain from target domain to learn invariant representation.

Though there are have been extensive prior work on UDA, some researchers have noted that if we can learn the difference or distinctiveness for target task, then the performance will make a step forward. Prior mainstream domain adaptation approaches tied weights of source and target domain on the model. Such as DANN [4,20], source domain and target domain learned the feature representation through the same convolutional layers that means both of them learn the representation by the same way. In intuition, target domain dataset learning representation in such way may lose its distinctive information. To reform such network architecture, the ADDA [22] method designs a new adversarial training networks that source domain and target domain have their own mapping networks and share common label classifier. Through different convolutional layers, the extracted feature can maintain domain-specific information. [18] introduce a new approach that attempts to align distributions of source and target by utilizing the task-specific decision boundaries. It firstly maximizes the discrepancy between two classifiers' output to detect target samples far from the support of source, then feature generator learns to generate target features near the support to minimize the discrepancy. Such two methods above don't utilize any supervised information, so we can't be sure of the quality of learned representation. But the following tries to use as little supervised information as possible to achieve significant improvement in performance.

[11] proposed ADMA algorithm that iteratively selected data according to its distinctiveness and uncertainty. Its main contribution is introducing a novel criterion *distinctiveness* to measure the ability of an instance on improving the representation quality of the neural network for target task. ADMA aims to use a few labeled data to update the parameters of pre-trained model but in some extend, it doesn't care much about domain distribution shift. [1] is one of classical work about combining transfer learning and active learning, which selects data and adjusts weights simultaneously. But noticing that transfer learning based on endowing instance with weight is not the best method. What's more, the method is built on the shallow model which can not model for complex situation well. Our approach adopts part of this paper's basic idea and designs a novel criterion which can be used in deep CNN models.

3 Preliminary

3.1 Kernel Embedding of Probability Distributions

Given any positive definite kernel function $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{R}$, there exists a unique reproducing kernel Hilbert space(RKHS) $\mathcal{H}$ which is a function space. Let $f : \mathcal{X} \rightarrow \mathbb{R}$ where evaluation can be written as an inner product, specifically, $f(x) = <f, k(\cdot, x)>_{\mathcal{H}}$, for all $f \in \mathcal{H}, x \in \mathcal{X}$. Furthermore, if a probability distribution P was given, then its kernel mean embedding into $\mathcal{H}$ is defined as:

$$\mu_P \triangleq E_P[f(x)] = \int_{\mathcal{X}} f(x)\, dP(x)$$

Considering a dataset $X = x_1, \cdots, x_n$ drawn from P(x), then its empirical kernel mean embedding is

$$\widehat{\mu_P} = \frac{1}{n} \sum_{i=1}^{n} f(x_i)$$

3.2 Maximum Mean Discrepancy(MMD)

Given observations $X_s = \{x_1^s, \cdots, x_{n_s}^s\}$ and $X_t = \{x_1^t, \cdots, x_{n_t}^t\}$ drawn from distributions P(X) and Q(X) respectively. Maximum Mean Discrepancy(MMD) used as a test statistic in a two-sample test which rejects or accepts the null hypothesis $P = Q$. The basic idea behind MMD is that if two distributions are identical, all of the statistics are the same. We define the MMD and its empirical estimation as:

$$MMD[P, Q] = \sup_{f \in \mathcal{H}} \|E_{x^s \sim P}[f(x^s)] - E_{x^t \sim Q}[f(x^t)]\|_{\mathcal{H}}^2 \tag{1}$$

$$MMD[X_s, X_t] = \sup_{f \in \mathcal{H}} \|\frac{1}{n_s} \sum_{i=1}^{n_s} f(x_i) - \frac{1}{n_t} \sum_{j=1}^{n_t} f(x_j)\|_{\mathcal{H}}^2 \tag{2}$$

where $\mathcal{H}$ is an universal RKHS that is rich enough to distinguish two distributions. From the formula, we can see the MMD is defined as the squared distance between the mean embedding. [6] gave the theoretical result that $P = Q$ if and only if $MMD[P, Q] = 0$. Formula (2) is an unbiased estimate of formula (1). In practice, we can extend the formula (2) as the following result:

$$MMD[P, Q] = \frac{1}{n_s^2} \sum_{i=1}^{n_s} \sum_{j=1}^{n_s} k(x_i^s, x_j^s) + \frac{1}{n_t^2} \sum_{i=1}^{n_t} \sum_{j=1}^{n_t} k(x_i^t, x_j^t) - \frac{2}{n_s n_t} \sum_{i=1}^{n_s} \sum_{j=1}^{n_t} k(x_i^s, x_j^t) \tag{3}$$

where k is a kernel function in $\mathcal{H}$.

4 The Proposed Approach and Criteria

Let $X^t = \{x_i^t\}_{i=1}^{n_t}$ denotes the unlabelled target domain with n_t instances and the model in iteration t will be denoted as $\mathcal{M}_t$. In this paper, we perform batch-mode active learning. At each iteration, we will select a small batch of instances (batchsize = b) $Q = \{x_i^t\}_{i=1}^{b}$ for querying their labels. We will use each batch of data to retrain the current model to update the parameters. To improve the neural network's performance with less cost, we must consider two points. First one is the instance's contribution for learning target-specific feature. Second point is the instance's contribution for learning classifier's distinctiveness.

Comparing with conventional methods of active learning strategy, we should consider distribution shift. In many UDA approaches, we can align the distributions between source domain and target domain transformed or mapped by neural networks. But it's worth noting that specific features for target task are

ignored by many UDA algorithms. Furthermore, without relational labeled data, it's hard to capture the specific features. In order to be from good to better, we query as few instances as possible for labeling. In each iteration, we aim to select a small batch of data and we hope there exists vast difference between such selected instances and source domain instances. Because, the intuition tells us that vast difference may promote to learn specific features by such instances in target domain. MMD is a powerful tool to measure the distance between two different distributions. We can evaluate the instances' contribution to specific features by this tool. Simultaneously, the uncertainty of instance should is necessary to consider. Combining such two criteria, we can evaluate each batch of instances' value well. The following is concrete formula for such two criteria. For the sake of writing conveniently, the symbol x_i^s, x_j^t are not the original data in domains but the features extracted by convolutional layers.

4.1 Characteristics

To describe the instance's ability of learning target-specific feature representation, we introduce the *characteristics* as an index.

$$\| \frac{1}{n_s} \sum_{i=1}^{n_s} \phi(x_i^s) - \frac{1}{b} \sum_{j=1}^{n_t} \alpha_j \phi(x_j^t) \|_{\mathcal{H}}^2, \qquad \alpha_j \in \{0, 1\}$$

This formula compute distance between a batch of b instances and source domain data. If we minimize the index that means we hope to find most similar instances to source domain data. It will be beneficial for learning the common representation. In this paper, we solve the problem that we have been learned a good common representation and aim to capture the specific features for target task. So, we can maximize *characteristics* to find instances in target domain dissimilar to source domain distribution. Actually, the maximal value is hard or impossible to find. In practical computation, we will use approximate point-wise computational method to find such b instances. The detailed explanation will be occurred in the next content.

4.2 Uncertainty

To combine with conventional active sampling strategy, we consider the uncertainty based method. Uncertainty is a commonly used criterion to evaluate how uncertain the prediction of the model for a given instance. In this paper, we adopt the maximum information entropy as our evaluation criterion. Of course, you can use any other methods such as margin-based or low-confidence approaches. Assume that there are $|\mathcal{Y}|$ classes for target task. Then the information entropy can be written as:

$$H(x) = -\sum_{i=1}^{|\mathcal{Y}|} p_i log(p_i)$$

where the p_i is the corresponding probability of class i.

4.3 Combination

Now combining such two criteria, we can get the following objective function:

$$\begin{aligned} \max_{\alpha} \quad & \| \frac{1}{n_s} \sum_{i=1}^{n_s} \phi(x_i^s) - \frac{1}{b} \sum_{j=1}^{n_t} \alpha_j \phi(x_j^t) \|_{\mathcal{H}}^2 + \sum_{j=1}^{n_t} \alpha_j H(x_j^t) \\ & \text{s.t.} \quad \sum_{i=1}^{n_t} \alpha_i = b \\ & \alpha_i \in \{0,1\} \end{aligned} \tag{4}$$

where $H(x_j^t)$ is the corresponding entropy of x_j^t towards the classifier. It's easy to extend the objective function as the following form:

$$\begin{aligned} \max_{\alpha} \quad & \frac{1}{n_s^2} \sum_{i=1}^{n_s} \sum_{j=1}^{n_s} k(x_i^s, x_j^s) + \frac{1}{b^2} \sum_{i=1}^{n_t} \sum_{j=1}^{n_t} \alpha_i \alpha_j k(x_i^t, x_j^t) \\ & - \frac{2}{n_s b} \sum_{i=1}^{n_s} \sum_{j=1}^{n_t} \alpha_j k(x_i^s, x_j^t) + \sum_{j=1}^{n_t} \alpha_j H(x_i^t) \\ & \text{s.t.} \quad \sum_{i=1}^{n_t} \alpha_i = b \\ & \alpha_i \in \{0,1\} \end{aligned} \tag{5}$$

$\alpha = (\alpha_i, \cdots, \alpha_{n_t})^T$. Dropping the constant term, formula (5) can be rewritten as :

$$\begin{aligned} \max_{\alpha} \quad & \frac{1}{b^2} \alpha^T K_{t,t}\, \alpha - \frac{2}{n_s b} \mathbb{1}_{n_s}^T K_{s,t} \alpha + H(X^t)^T \alpha \\ & \text{s.t.} \quad \mathbb{1}_{n_t}^T \alpha_i = b \\ & \alpha_i \in \{0,1\} \end{aligned} \tag{6}$$

The formula (6) is similar to convex quadratic programming, unfortunately, it can't be sufficient for QP's conditions. The constraints make the problem be an integer programming problem. Furthermore, the key point is the objective function is expecting for a maximum value and $K_{t,t}$ is a positive matrix. So it's not a conventional convex optimization problem. And if we relax the constraint $\alpha_i \in \{0,1\}$ to be a linear inequation $\alpha_i \in [0,1]$, then the maximal value is impossible or reaches at boundary. To escape of this dilemma, we propose a approximate method to find a suboptimal solution. Looking back to the formula (5), the first term is a constant and the second term measures the similarity of data pairs in selected dataset Q, the next term measures the similarity of instances in Q with data in X^s and the last term evaluates the entropy. So towards instances in Q, if each of them is much similar with other data in Q and dissimilar with data in X^s simultaneously has a large information entropy then such batch of b instances are good enough to label. Actually, the second term's computation is unbearable because it has to compute $\mathrm{C}_b^2 \mathrm{C}_{n_t}^b$ of times. So, we

can drop this term and consider the third term that means selecting b instances dissimilar with source domain data then we have a larger possibility to make function (5)'s value larger. So each data's *characteristics* in target domain can be reduced to:

$$characteristics(x_i^t) = -\sum_{i=1}^{n_s} k(x_i^s, x_j^t)$$

Actually, without considering the constant term, the definition is equal to:

$$\| \frac{1}{n_s} \sum_{i=1}^{n_s} \phi(x_i^s) - \phi(x_j^t) \|_{\mathcal{H}}^2$$

4.4 Practical Computation

We define each data's score as:

$$\mathcal{S}(x) = \lambda \cdot characteristics(x) + (1 - \lambda) uncertainty(x)$$

Here, we introduce a balanced factor λ which is relational with iterations. As discussed before, *characteristics* measures the ability of capturing specific features for target task and *uncertainty* measures the ability of improving the classifier's performance. At the start of iterations, the key point is adapting the feature extractor—convolutional layers and assures features extracted can be beneficial to target task. But with the progress of iterations, such feature extractor is good enough to extract features for target task. If the selected data is still to change the feature extractor then classifier' performance will be degraded. At this time, *uncertainty* need more attention. Hence, the dynamic trade-off is necessary.

Algorithm: AL_MMD

Input:
X^t: Unlabeled target dataset
$\mathcal{M}_t$: the model in iteration t
$\mathcal{M}_0$: the initial model trained by domain adaptation methods
Initialization:
Use DANN algorithm to get the initial model $\mathcal{M}_0$
While $t < iterations$:
  For each instance in $x \in X^t$
    compute $x's$ transformation after convolutional layers $x = conv(x)$
    compute $characteristics(x) = -\sum_{i=1}^{n_s} k(x_i^s, x)$
    compute $uncertainty(x) = -\sum_{i=1}^{|\mathcal{Y}|} p_i log(p_i)$
    compute $\mathcal{S}(x) = \lambda \cdot characteristics(x) + (1 - \lambda) uncertainty(x)$
  End For
Select top b largest $\mathcal{S}(x)$ in target domain for Q
Query such b instances' labels and remove Q from X^t
Fine-tune the model $\mathcal{M}_t$ with Q
End While

5 Experiments and Results

We perform our proposed approach on two popular image datasets comparing with maximum entropy strategy, random sampling strategy. Towards active learning, we know a good initial model $\mathcal{M}_0$ is necessary. Models trained by various UDA algorithms can be qualified for this task. In this paper, we choose DANN [4] to train our initial model. Based on it, we use AL_MMD to select instances with larger *characteristics* and *uncertainty* to fine-tune the initial model. The following are two different datasets and corresponding results.

5.1 Datasets

MNIST and MNIST-M. Our first experiment deals with the MNIST dataset [13] (as source domain). MNIST-M (as target domain) is a dataset that MNIST blends digits with color photos.

Office-31 [17] is a standard dataset for domain adaptation, which consists of 3 domains *Amazon*(A), *Webcam*(W), and *Dslr*(D). Each contains images from amazon.com, or office environment images taken with varying lighting and pose changes using a webcam or a dslr camera, respectively. And it includes 4652 images with 31 classes. In this dataset, we do two transfer experiments *Amazon* → *Dslr* and *Dslr* → *Webcam*.

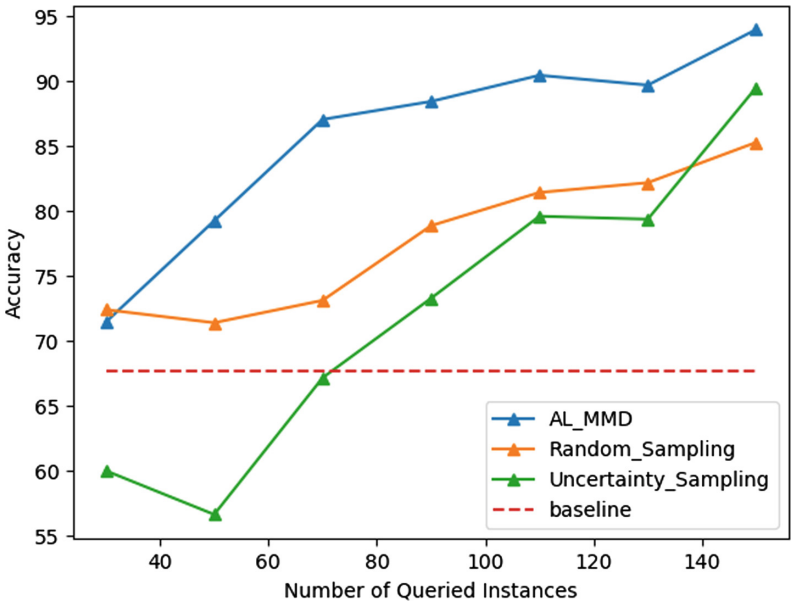

Fig. 1. Dslr → Webcam.

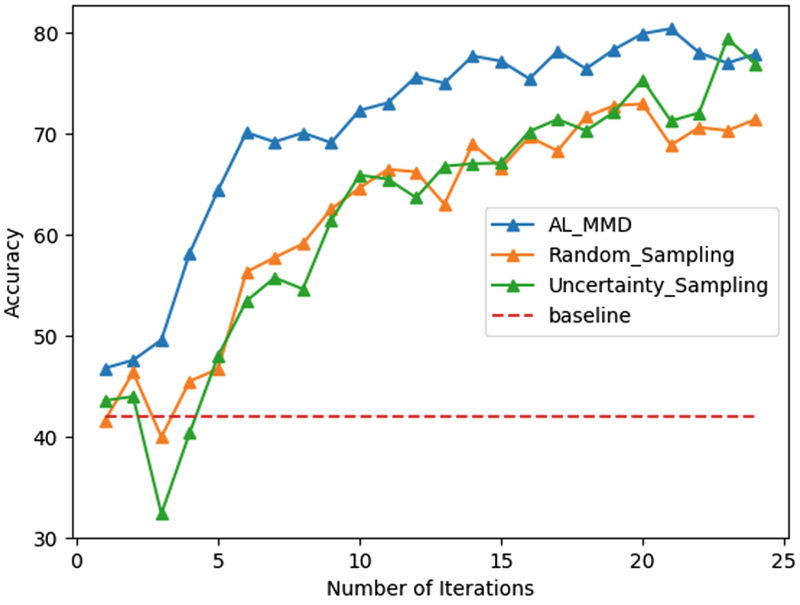

Fig. 2. MNIST → MNIST-M

5.2 Results

In MNIST → MNIST-M transfer experiment, we use LeNet as our backbone. In Office-31 transfer experiment, we use pre-trained AlexNet as our backbone. Then we use DANN algorithm to update weights to get initial model. Eventually, we compare our method with random sampling and entropy-based sampling. Table 1 and Fig. 1 were the results in MNIST dataset. Table 2 and Fig. 2 were the results in Office-31. The results show that our methods have achieved superior performance. From the figure, you will find that at the start of iterations, results of entropy_sampling and random_sampling are so volatile. On the contrary, our method is stable comparing these two methods. According to our method' idea, we aim to learn good representation for target task so that we need less data to modify the parameters. But towards another two methods, if the learned

Table 1. Accuracy on MNIST → MNIST_M(%)

Methods	Number of queried instances							
	20	50	80	110	140	170	200	230
AL_MMD	**47.600**	**64.400**	**70.100**	**73.066**	**77.714**	**78.177**	**79.920**	77.000
Random_sampling	46.400	46.720	59.120	66.480	68.960	68.320	72.960	70.320
Entropy_sampling	44.000	48.000	54.640	65.520	67.043	71.440	75.360	**79.440**

Table 2. Accuracy on Dslr → Webcam(%)

Methods	Number of queried instances						
	30	50	70	90	110	130	150
AL_MMD	71.446	**79.245**	**87.044**	**88.0427**	**90.440**	**89.685**	**93.962**
Random_sampling	**72.410**	71.404	73.123	78.867	81.425	82.180	85.283
Entropy_sampling	60.025	56.654	67.169	73.283	79.597	79.371	89.590

representations are not proper, the classifier needs more data to modify and the corresponding curve are much more volatile.

6 Conclusion and Future Work

In this paper, we design an active sampling strategy based on MMD to select valuable data in transfer learning process. We propose a new criterion *characteristics* to select data that can capture target-specific feature well. And the aforementioned experiments have shown our method's efficacy. Through this method, we can find MMD is a powerful tool. In the feature work, We can also filter the data in source domain similar to target domain at the beginning. It can effectively resist the impact of noisy data. What's more, we hope to seek for an efficient and scalable algorithm to extend our method to larger datasets.

Acknowledgements. This research was supported by Natural Science Foundation of China (Grant no. 61572252, 61672284). Meanwhile, this research work was supported by Zayed University Research Cluster Award \# R18038.

References

1. Chattopadhyay, R., Fan, W., Davidson, I., Panchanathan, S., Ye, J.: Joint transfer and batch-mode active learning. In: Proceedings of the 30th International Conference on Machine Learning, ICML 2013, 16–21 June 2013, Atlanta, GA, USA, pp. 253–261 (2013). http://jmlr.org/proceedings/papers/v28/chattopadhyay13.html
2. Chattopadhyay, R., Wang, Z., Fan, W., Davidson, I., Panchanathan, S., Ye, J.: Batch mode active sampling based on marginal probability distribution matching. TKDD **7**(3), 13:1–13:25 (2013). https://doi.org/10.1145/2513092.2513094
3. Gal, Y., Islam, R., Ghahramani, Z.: Deep Bayesian active learning with image data. In: Proceedings of the 34th International Conference on Machine Learning, ICML 2017, 6–11 August 2017, Sydney, NSW, Australia, pp. 1183–1192 (2017). http://proceedings.mlr.press/v70/gal17a.html
4. Ganin, Y., et al.: Domain-adversarial training of neural networks. J. Mach. Learn. Res. **17**(1), 2096–2030 (2016)
5. Ghifary, M., Kleijn, W.B., Zhang, M., Balduzzi, D., Li, W.: Deep reconstruction-classification networks for unsupervised domain adaptation. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9908, pp. 597–613. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_36

6. Gretton, A., Borgwardt, K.M., Rasch, M.J., Schölkopf, B., Smola, A.J.: A kernel two-sample test. J. Mach. Learn. Res. **13**, 723–773 (2012). http://dl.acm.org/citation.cfm?id=2188410
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, 27–30 June 2016, Las Vegas, NV, USA, pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90
8. Hoffman, J., et al.: CyCADA: cycle-consistent adversarial domain adaptation. In: Proceedings of the 35th International Conference on Machine Learning, ICML 2018, 10–15 July 2018, Stockholmsmässan, Stockholm, Sweden, pp. 1994–2003 (2018). http://proceedings.mlr.press/v80/hoffman18a.html
9. Huang, J., Smola, A.J., Gretton, A., Borgwardt, K.M., Schölkopf, B.: Correcting sample selection bias by unlabeled data. In: Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, 4–7 December 2006, Vancouver, British Columbia, Canada, pp. 601–608 (2006). http://papers.nips.cc/paper/3075-correcting-sample-selection-bias-by-unlabeled-data
10. Huang, S., Jin, R., Zhou, Z.: Active learning by querying informative and representative examples. IEEE Trans. Pattern Anal. Mach. Intell. **36**(10), 1936–1949 (2014). https://doi.org/10.1109/TPAMI.2014.2307881
11. Huang, S., Zhao, J., Liu, Z.: Cost-effective training of deep CNNs with active model adaptation. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, 19–23 August 2018, London, UK, pp. 1580–1588 (2018). https://doi.org/10.1145/3219819.3220026
12. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Commun. ACM **60**(6), 84–90 (2017). https://doi.org/10.1145/3065386
13. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
14. Long, M., Cao, Y., Wang, J., Jordan, M.I.: Learning transferable features with deep adaptation networks. In: Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, 6–11 July 2015, Lille, France, pp. 97–105 (2015). http://jmlr.org/proceedings/papers/v37/long15.html
15. Long, M., Zhu, H., Wang, J., Jordan, M.I.: Deep transfer learning with joint adaptation networks. In: Proceedings of the 34th International Conference on Machine Learning, ICML 2017, 6–11 August 2017, Sydney, NSW, Australia, pp. 2208–2217 (2017). http://proceedings.mlr.press/v70/long17a.html
16. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Trans. Knowl. Data Eng. **22**(10), 1345–1359 (2010). https://doi.org/10.1109/TKDE.2009.191
17. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6314, pp. 213–226. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15561-1_16
18. Saito, K., Watanabe, K., Ushiku, Y., Harada, T.: Maximum classifier discrepancy for unsupervised domain adaptation. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, 18–22 June 2018, Salt Lake City, UT, USA, pp. 3723–3732 (2018). https://doi.org/10.1109/CVPR.2018.00392, http://openaccess.thecvf.com/content_cvpr_2018/html/Saito_Maximum_Classifier_Discrepancy_CVPR_2018_paper.html

19. Settles, B.: Active Learning. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers (2012). https://doi.org/10.2200/S00429ED1V01Y201207AIM018
20. Shu, Y., Cao, Z., Long, M., Wang, J.: Transferable curriculum for weakly-supervised domain adaptation (2019)
21. Szegedy, C., et al.: Going deeper with convolutions. CoRR abs/1409.4842 (2014). http://arxiv.org/abs/1409.4842
22. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 21–26 July 2017, Honolulu, HI, USA, pp. 2962–2971 (2017). https://doi.org/10.1109/CVPR.2017.316
23. Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., Darrell, T.: Deep domain confusion: Maximizing for domain invariance. CoRR abs/1412.3474 (2014). http://arxiv.org/abs/1412.3474
24. Wang, K., Zhang, D., Li, Y., Zhang, R., Lin, L.: Cost-effective active learning for deep image classification. IEEE Trans. Circuits Syst. Video Techn. **27**(12), 2591–2600 (2017). https://doi.org/10.1109/TCSVT.2016.2589879