A Survey on Deep Transfer Learning

Chuanqi Tan[1], Fuchun Sun[2], Tao Kong[1],
Wenchang Zhang[1], Chao Yang[1], and Chunfang Liu[2]

State Key Laboratory of Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology (TNList)
Department of Computer Science and Technology, Tsinghua University
[1]{tcq15, kt14, zhangwc14, yang-c15}@mails.tsinghua.edu.cn
[2]{fcsun, cfliu1985}@tsinghua.edu.cn

Zihao Li, 2020.08.21

Why Transfer Learning

- In some domains, like bioinformatics and robotics, it is very difficult to construct a large-scale well-annotated dataset due to the expense of data acquisition and costly annotation, which limits its development;

- Transfer learning relaxes the hypothesis that the training data must be independent and identically distributed (i.i.d.) with the test data, which motivates us to use transfer learning to solve the problem of insufficient training data.

Definitions

- A domain can be represented by $D = \{\chi, P(X)\}$, which contains two parts: the feature space χ and the edge probability distribution $P(X)$ where $\mathrm{X} = \{x_1, \dots, x_n\} \in \chi$;

- A task can be represented by $\mathrm{T} = \{y, f(x)\}$. It consists of two parts: label space y and target prediction function $f(x)$. $f(x)$ can also be regarded as conditional probability function $\mathrm{P(y|x)}$.

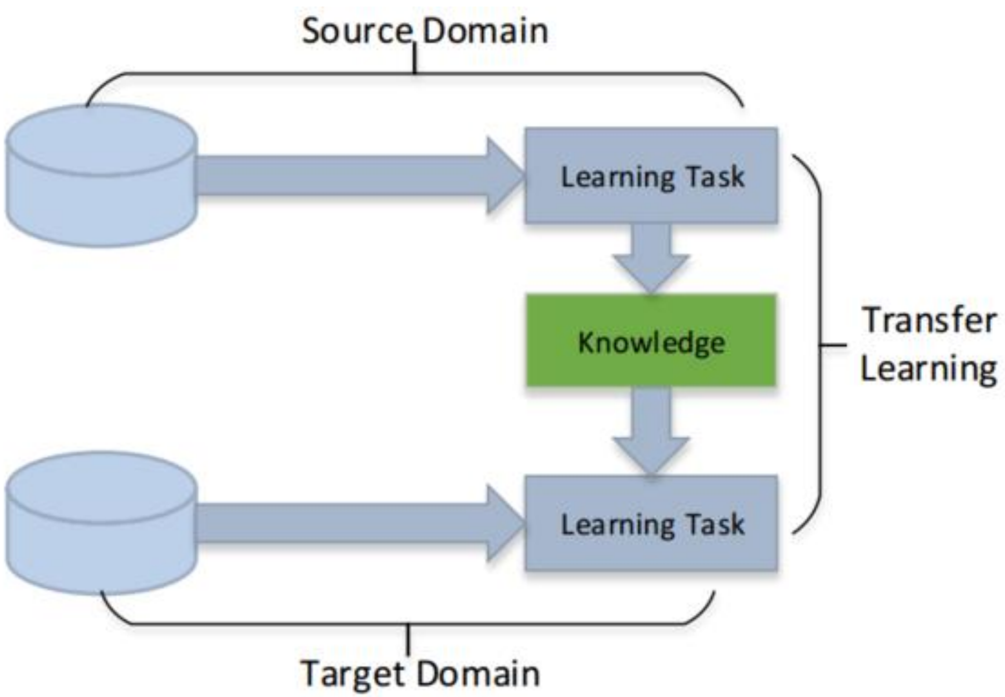

Fig. 1. Learning process of transfer learning.

Definitions

- **Transfer Learning.** Given a learning task T_t based on D_t, and we can get the help from D_s for the learning task T_s. Transfer learning aims to improve the performance of predictive function $f_T(\cdot)$ for learning task T_t by discover and transfer latent knowledge from D_s and T_s, where $D_s \neq D_t$ and/or $T_s \neq T_t$. In addition, in the most case, the size of D_s is much larger than the size of D_t, $N_s \gg N_t$;
- **Deep Transfer Learning.** Given a transfer learning task defined by $< D_s, T_s, D_t, T_t, f_T(\cdot) >$. It is a deep transfer learning task where $f_T(\cdot)$ is a non-linear function that reflected a deep neural network;

Categorizes

Table 1. Categorizing of deep transfer learning.

Approach category	Brief description	Some related works
Instances-based	Utilize instances in source domain by appropriate weight.	[4], [27], [20], [24], [10], [26], [11]
Mapping-based	Mapping instances from two domains into a new data space with better similarity.	[23], [12], [8], [14], [2]
Network-based	Reuse the partial of network pre-trained in the source domain.	[9], [17], [15], [30], [3], [6], [28]
Adversarial-based	Use adversarial technology to find transferable features that both suitable for two domains.	[1], [5], [21], [22], [13], [16]

Instances-based deep transfer learning

Although there are different between two domains, partial instances in the source domain can be utilized by the target domain with appropriate weights.

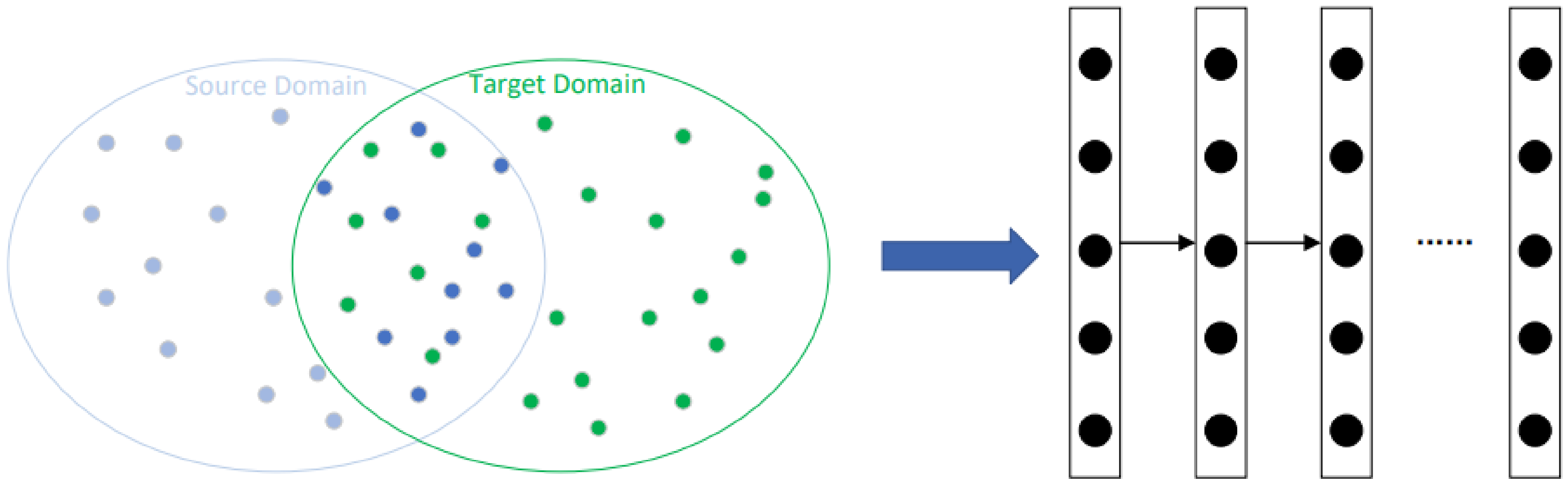

Fig. 2. Sketch map of instances-based deep transfer learning. Instances with light blue color in source domain meanings dissimilar with target domain are exclude from training dataset; Instances with dark blue color in source domain meanings similar with target domain are include in training dataset with appropriate weight.

TrAdaBoost

Algorithm 1 TrAdaBoost

Input the two labeled data sets T_d and T_s, the unlabeled data set S, a base learning algorithm **Learner**, and the maximum number of iterations N.

Initialize the initial weight vector, that $\mathbf{w}^1 = (w_1^1, \ldots, w_{n+m}^1)$. We allow the users to specify the initial values for $\mathbf{w}^1$.

For $t = 1, \ldots, N$

1. Set $\mathbf{p}^t = \mathbf{w}^t / (\sum_{i=1}^{n+m} w_i^t)$.
2. Call **Learner**, providing it the combined training set T with the distribution $\mathbf{p}^t$ over T and the unlabeled data set S. Then, get back a hypothesis $h_t : X \to Y$ (or $[0,1]$ by confidence).
3. Calculate the error of h_t on T_s:

$$\epsilon_t = \sum_{i=n+1}^{n+m} \frac{w_i^t \cdot |h_t(x_i) - c(x_i)|}{\sum_{i=n+1}^{n+m} w_i^t}.$$

4. Set $\beta_t = \epsilon_t/(1-\epsilon_t)$ and $\beta = 1/(1+\sqrt{2\ln n/N})$. Note that, ϵ_t is required to be less than $1/2$.
5. Update the new weight vector:

$$w_i^{t+1} = \begin{cases} w_i^t \beta^{|h_t(x_i)-c(x_i)|}, & 1 \le i \le n \\ w_i^t \beta_t^{-|h_t(x_i)-c(x_i)|}, & n+1 \le i \le n+m \end{cases}$$

Output the hypothesis

$$h_f(x) = \begin{cases} 1, & \prod_{t=\lceil N/2 \rceil}^{N} \beta_t^{-h_t(x)} \ge \prod_{t=\lceil N/2 \rceil}^{N} \beta_t^{-\frac{1}{2}} \\ 0, & \text{otherwise} \end{cases}$$

Mapping-based deep transfer learning

Although there are different between two origin domains, they can be more similarly in an elaborate new data space.

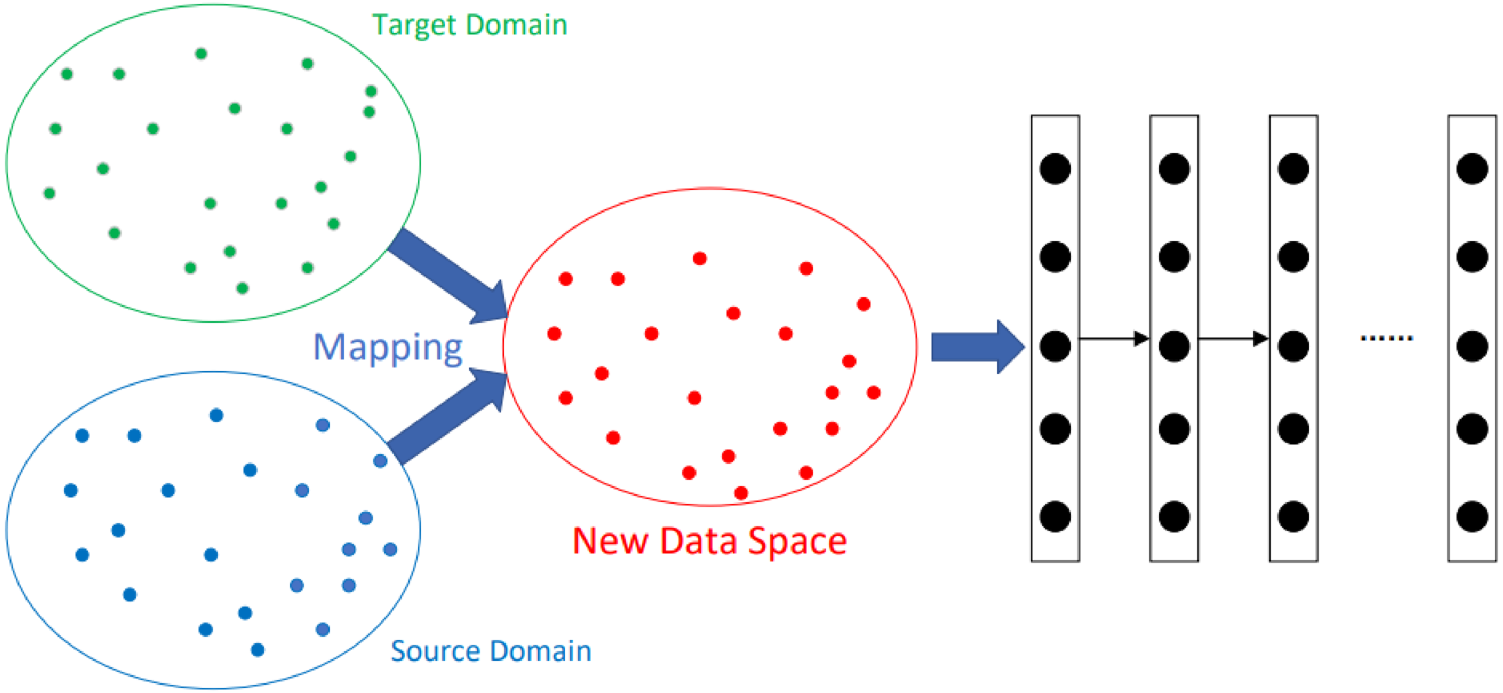

Fig. 3. Sketch map of mapping-based deep transfer learning. Simultaneously, instances from source domain and target domain are mapping to a new data space with more similarly. Consider all instances in the new data space as the training set of the neural network.

Mapping-based deep transfer learning

A natural idea is extend the TCA (Transfer component analysis) method to deep neural network extend MMD to comparing distributions in a deep neural network, by introduces an adaptation layer and an additional domain confusion loss to learn a representation that is both semantically meaningful and domain invariant. The MMD distance used in this work is defined as

$$D_{\mathcal{MMD}}(X_S, X_T) = \left\| \frac{1}{|X_S|} \sum_{x_s \in X_S} \phi(x_s) - \frac{1}{|X_T|} \sum_{x_t \in X_T} \phi(x_t) \right\| \quad (1)$$

The loss function is defined as

$$\mathcal{L} = \mathcal{L}_C(X_L, y) + \lambda D^2_{\mathcal{MMD}}(X_S, X_T). \quad (2)$$

Network-based deep transfer learning

Neural network is similar to the processing mechanism of the human brain, and it is an iterative and continuous abstraction process. The front-layers of the network can be treated as a feature extractor, and the extracted features are versatile.

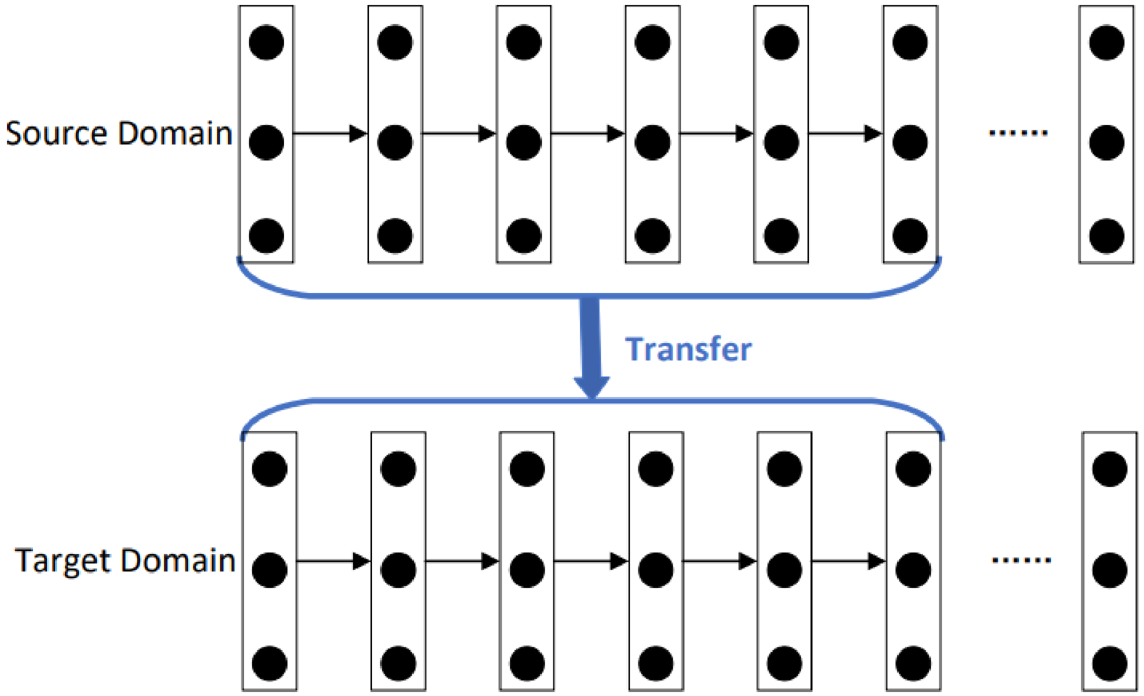

Fig. 4. Sketch map of network-based deep transfer learning. First, network was trained in source domain with large-scale training dataset. Second, partial of network pre-trained for source domain are transfer to be a part of new network designed for target domain. Finally, the transfered sub-network may be updated in fine-tune strategy.

Adversarial-based deep transfer learning

For effective transfer, good representation should be discriminative for the main learning task and indiscriminate between the source domain and target domain.

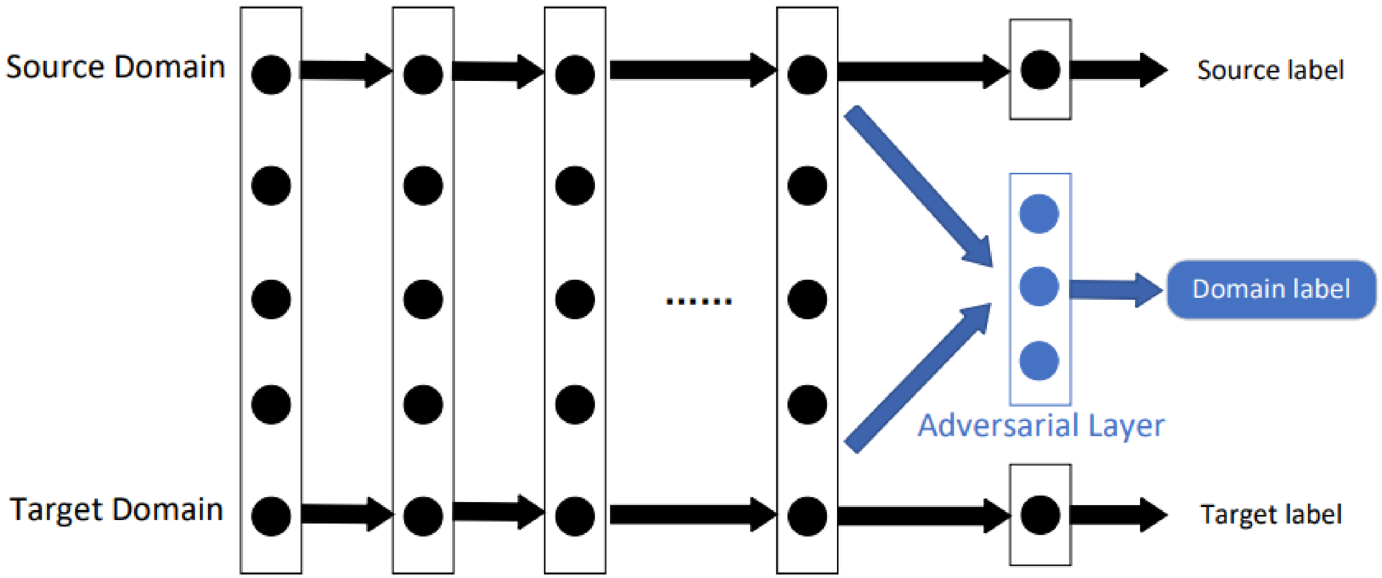

Fig. 5. Sketch map of adversarial-based deep transfer learning. In the training process on large-scale dataset in the source domain, the front-layers of network is regarded as a feature extractor. It extracting features from two domains and sent them to adversarial layer. The adversarial layer try to discriminates the origin of the features. If the adversarial network achieves worse performance, it means a small difference between the two types of feature and better transferability, and vice versa. In the following training process, the performance of the adversarial layer will be considered to force the transfer network discover general features with more transferability.

Summary & Conclusion

- Deep transfer learning is classified into four categories: instances-based deep transfer learning, mapping-based transfer learning, network-based deep transfer learning, and adversarial-based deep transfer learning;

- In most practical applications, the above multiple technologies are often used in combination to achieve better results;

- Network-based deep transfer learning are widely used, especially for pre-training and fine tuning.

References

[1] Tan C, Sun F, Kong T, et al. A survey on deep transfer learning[C]//International conference on artificial neural networks. Springer, Cham, 2018: 270-279.

[2] Dai, W., Yang, Q., Xue, G.R., Yu, Y.: Boosting for transfer learning. In: Proceedings of the 24th international conference on Machine learning. pp. 193{200. ACM (2007).

[3] Pan, S.J., Tsang, I.W., Kwok, J.T., Yang, Q.: Domain adaptation via transfer component analysis. IEEE Transactions on Neural Networks 22(2), 199{210 (2011).

[4] Zhang, J., Li, W., Ogunbona, P.: Joint geometrical and statistical alignment for visual domain adaptation. In: CVPR (2017).

[5] Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., Darrell, T.: Deep domain confusion: Maximizing for domain invariance. arXiv preprint arXiv:1412.3474 (2014).

[6] Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M.: Domainadversarial neural networks. arXiv preprint arXiv:1412.4446 (2014);

Q&A

Thanks