

---

# Learning to Transfer with Triply Adversarial Nets

---

Gilles Louppe  
New York University  
g.louppe@nyu.edu

## Abstract

In classification, transfer learning (or its variants known as covariate shift or domain adaptation) arises whenever target instances are governed by a distribution that may be arbitrarily different from the distribution of the source instances. This problem has traditionally been solved by reweighting approaches or by learning robust representations over domains. In this work, we propose a new paradigm based on the assumption that the covariate shift is only due to a different representation of the same underlying objects. Accordingly, we propose to learn how to transform source instances into target instances, possibly across input spaces of distinct dimensions, structures or supports. For this purpose, we extend the generative adversarial networks framework of [1] to a triply adversarial process: a transformer network  $T$  for generating target instances from source instances, a discriminative network  $D$  for separating transformed source instances from actual target instances, and a classifier network  $C \circ T$  for classifying source instances in the projected space. This 3-player game results in a network  $T$  capable of transforming source into target instances, while preserving separation between classes as enabled by  $C$  in the adversarial setup. Experiments demonstrate the potential of this novative approach, with promising results when the construction of  $C$  can be bootstrapped in a semi-supervised way from a few labeled instances from the target space.

## 1 Introduction

## 2 Problem statement

Let assume a probability space  $(\Omega, \mathcal{F}, P)$ , where  $\Omega$  is a sample space,  $\mathcal{F}$  is a set of events and  $P$  is a probability measure. Let consider a (multivariate) random variable  $X : \Omega \mapsto \mathbb{R}^p$  inducing the source distribution, along with a finite set  $\{x_i\}_{i=1}^N$  of its realizations  $X(\omega_i)$ , for  $\omega_i \in \Omega$ . In classification, let us further assume that realizations from the source distribution are extended with realizations  $\{y_i\}_{i=1}^N$  of a label random variable  $Y : \Omega \mapsto \mathcal{Y}$ . Similarly, let assume a (multivariate) random variable  $U : \Omega \mapsto \mathbb{R}^q$  inducing the target distribution, along with a finite set  $\{u_j\}_{j=1}^M$  of its realizations  $U(\omega_j)$ , for  $\omega_j \in \Omega$ . Assuming it exists, our goal is to find a transfer function  $T : \mathbb{R}^p \times \mathcal{Y} \mapsto \mathbb{R}^q$  such that

$$T(X(\omega), Y(\omega)) = U(\omega) \text{ for all } \omega \in \Omega. \quad (1)$$

Since we do not have training tuples  $((X(\omega), Y(\omega)), U(\omega))$  (for the same unknown  $\omega$ ) from which  $T$  could be learned using standard regression algorithms, we propose instead to solve the closely related problem of finding a transfer function  $\hat{T}$  such that

$$P(\{\omega | \hat{T}(X(\omega), Y(\omega)) = u\}) = P(\{\omega' | U(\omega') = u\}) \text{ for all } u \in \mathbb{R}^q. \quad (2)$$

In words, we are looking for a transfer function  $\hat{T}$  such that realizations of  $\hat{T}(X, Y)$  are undistinguishable from realizations of  $U$ . A function  $T$  for which Eqn. 1 is true necessarily satisfies

Eqn. 2. The converse is however in general not true, since the sets of events  $\{\omega|\hat{T}(X(\omega)) = u\}$  and  $\{\omega'|U(\omega') = u\}$  do not need to be the same for the equality to hold. Accordingly, the contribution of this work is to propose a procedure for constructing transfer functions satisfying Eqn. 2, and for which Eqn. 1 is *plausibly* satisfied.

### **3 Method**

### **4 Experiments**

### **5 Related work**

### **6 Conclusions**

#### **Acknowledgments**

[GL: todo]

#### **References**

- [1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.