GAN-based Data Mapping for Model Adaptation

Felipe Leno da Silva [1 2] **Ruben Glatt** [2] **Raphael Cobe** [1] **Renato Vicente** [3]

Abstract

Although Machine Learning algorithms are solving tasks of ever-increasing complexity, gathering data and building training sets remains an error prone, costly, and difficult problem. However, reusing knowledge from related previously-solved tasks enables reducing the amount of data required to learn a new task. We here propose a method for learning a mapping model that maps data from a source task with labeled data to a related target task with only unlabeled data. We perform an empirical evaluation showing that our method achieves performance comparable to a model learned directly in the target task.

1. Introduction and Motivation

Machine Learning (ML) applications are progressively becoming pervasive. Due to recent advances in learning and optimization algorithms, dedicated hardware platforms, and open-source implementation frameworks, ML is being employed in challenging domains of ever-increasing complexity nourished by a growing amount of available data, continuously being collected and stored in the internet age. Learning algorithms are designed to train models $P(Y|X)$ that predict a class or a real number Y based on relevant input features X. Most algorithms assume that the data used for training the ML model (training set) is made up of independent samples from the same joint distribution $P(Y, X)$, with $p(Y, X) = p(Y|X)p(X)$. Although ML is becoming increasingly accessible and most companies are able to train and deploy models for several tasks, learning algorithms remain data-hungry. Since crafting new training data is still costly and a time-consuming process additional techniques for reducing the sample-complexity are needed.

Transfer Learning (Pan & Yang, 2010) enables sharing knowledge for reducing sample complexity of learning a task. While transfer can be achieved in various ways, e.g. through communication in multi-agent systems (Dawson et al., 2021), we are here not interested in parallel learning or sample sharing but focus on reusing previously learned knowledge in a sequential way. However, reusing knowledge from one (or more) task(s) for learning a new one is challenging (Glatt et al., 2016) because we can generally expect that the distribution $P(Y|X)$ will change from one (source) task to another (target) task, a phenomenon known as *Concept Drift* (Webb et al., 2016), making it hard to directly reuse samples or models (Glatt et al., 2020). Additionally, bias in the sampling process might cause the distribution $P(X)$ to shift between training and deployment phase, known as *Covariate Drift*. Therefore, both $P(Y|X)$ and $P(X)$ are expected to change from one task to another.

Despite the above-mentioned challenges, reusing knowledge is useful in many practical situations. Consider a company that calculates a creditworthiness score for deciding if a loan should be given to a particular customer. This score is estimated by a model that receives, as input, the customer's previous credit-related relationship with this particular company and its partners, and outputs an estimated probability that the customer defaults the loan. Now imagine that this company expands its operations to a new country. The model cannot be directly reused because the new population has its own particularities (drift in $P(Y|X)$). Moreover, the "average person" in this new country might be different from the previous one (drift in $P(X)$). However, building a new data set in this new country will demand time and investment, because the company will have to start new relationships and wait to see who defaults loans and who does not. For many months, the company will have access to customer profiles but will not have the information of who will be able to pay their loans, which corresponds to having unlabeled data in the target task. Therefore, being able to reuse some knowledge from the source task to accelerate this process could be very beneficial to the company.

2. Background

In this section we explain the underlying theories that form the base for our proposed method. First, we define the Supervised Learning problem and then we describe the GAN framework which we use for our mapping approach. Finally, we present the definition of transfer learning and how

[1]Advanced Institute for Artificial Intelligence (AI2), Brazil [2]Lawrence Livermore National Laboratory [3]Serasa Experian. Correspondence to: Felipe Leno da Silva <leno@llnl.gov>.

Proceedings of the LatinX in AI (LXAI) Research workshop at ICML 2021.

it relates to the challenge we introduce here.

2.1. Supervised Learning

A particular learning task consist of building a predictive model $h : X \rightarrow Y$, where $X = X^1, \ldots, X^f, \forall_{x \in X} x \in \mathbb{R}$ corresponds to a set of features sufficient for fully describing the task and Y is the *target* variable to be predicted. Supervised learning covers mainly two areas and we can define $Y \in \mathbb{R}$ for regression tasks and $Y \in C$ for classification tasks, where C is the set of possible labels. A *Concept* is defined as the joint distribution $P(Y, X)$ (Gama et al., 2014) and can be decomposed as $P(Y, X) = P(Y|X)P(X)$.

We are primarily interested in modelling $P(Y|X)$ to predict Y for new samples. However, since $P(Y, X)$, $P(Y|X)$ and $P(X)$ are all initially unknown, we train learning algorithms by gathering a *dataset* of samples with known answers $\boldsymbol{O} = o^1, \ldots, o^n, o^i =< x^i, y^i >$. The learning algorithm then has to learn how to generalize a function to predict Y for new samples based on $\boldsymbol{O}$. Many learning algorithm have been proposed for learning in this scenario, including Linear Discrimination (Lei et al., 2012), Support-Vector Machines (Cortes & Vapnik, 1995), Naive Bayes (John & Langley, 1995), Multilayer Perceptron Networks (Basheer & Hajmeer, 2000), and others.

2.2. GAN and Cycle-Gan

A *Generative Adversarial Network* (GAN) is a popular framework to learn generative models capable of generating realistic samples (Goodfellow et al., 2014).

A basic GAN is illustrated in Fig 1 with two models as main components, a *generator* (red) and a *discriminator* (blue). The objective of the generator is to generate realistic samples while the discriminator learns to distinguish between real samples and artificial ones produced by the generator. As both models co-evolve in an adversarial manner, they eventually stabilize in an equilibrium where the discriminator cannot distinguish between generated and real samples because they are indistinguishable from each other.

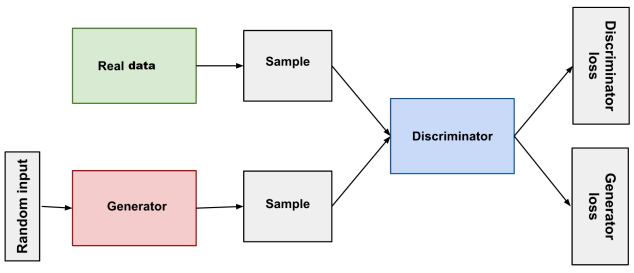

Figure 1. Illustration of a basic GAN model.

The loss function for the discriminator is defined as

$$\mathcal{L}_D = -\frac{1}{2}\mathbb{E}_{\boldsymbol{x}} log D(\boldsymbol{x}) - \frac{1}{2}\mathbb{E}_z log(1 - D(G(z))), \quad (1)$$

where $D(x)$ is the discriminator model, $G(x)$ is the generator model, and z is a random vector. That is, the loss function represents how many of the samples were correctly classified as fake or real. The generator is simply trained with the inverse of this loss,

$$\mathcal{L}_D = -\mathcal{L}_G, \quad (2)$$

which means that the generator will optimize its loss by deceiving the discriminator.

2.3. Transfer Learning

Gathering data and building data sets is a costly and time-consuming process. However, reusing knowledge from a previously solved and related task might reduce the data requirements of learning algorithms. Formally, we define a *domain* $D = \langle X, Y \rangle$ as a feature space X and an output space Y. A *task* $\mathcal{T} = \langle P(X), f(.) \rangle$ consists of a marginal probability distribution $P(X)$ and an objective predictive function $f(.)$, which is not observable but is the *ground truth* function that generates the correct output for any sample. Tasks belonging to a single domain are expected to be similar, but not equal. The most common way of performing transfer is by reusing knowledge from a source task $\mathcal{T}^s$ in a target task $\mathcal{T}^t$. In case $P^s(X) \neq P^t(X)$, we say that a *Covariate Drift* happened from one task to another. In case $f^s(.) \neq f^t(.)$, we say that a *Concept Drift* happened. Knowledge can be reused in several ways, such as reusing $\boldsymbol{O}^s$ as additional samples for learning $\mathcal{T}^t$ (*instance-transfer*). We focus here on reusing and refining $h^s(.)$ in the target task (*model-transfer*). Performing transfer is useful but generally hard because of covariate and concept drift.

3. Problem Statement

This work assumes it is possible to define a mapping function that "translates" a sample from the target task to the source task, that is, if $f : X^t \rightarrow X^t$ is a mapping function,

$$\exists f, P(f(X^t))P(Y|f(X^t)) = P(X^s)P(Y|X^s). \quad (3)$$

The main challenge that we are aiming to overcome is to learn f based on available but insufficient data from a target domain for which we have a good model in a related source domain. Specifically, here, we are interested in the credit risk problem. For example, a potential application would be to define the credit score for people in regions where there is little or no data available but we still need to estimate how likely a person is to default their payments.

Figure 2 illustrates an idea of how this could be established in practice. In our example, a basic source model is initially

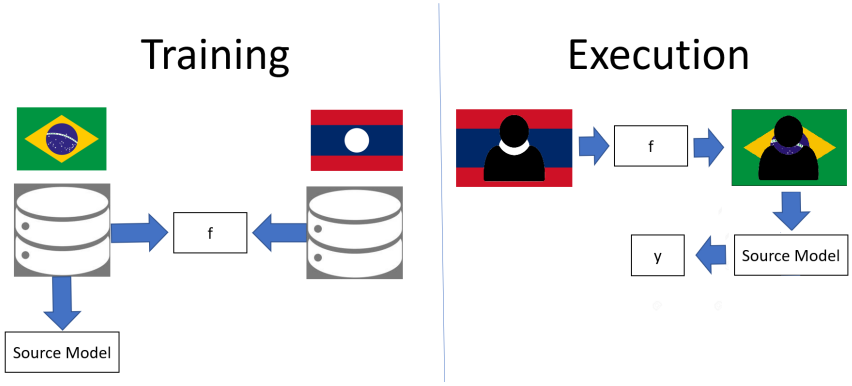

Figure 2. Illustration of a localized example: In a supervised manner, a source model for credit risk assessment is trained for the Brazilian population, where there is abundant labeled data. The Laotian credit data is largely unlabeled and a mapping function f is learned to map Laotians into "virtual Brazilians" using the GAN framework. f is then simply integrated between Laotian data and Brazilian source model, making the model directly usable for Laotians.

trained from data collected in a country with lots of available data. Then, instead of learning a whole new model for a different country to calculate the credit score, we only learn a mapping function that allows to map an average person from the new country to their counterpart in the original country. The effectiveness of the original model trained with massive amounts of data can then be leveraged to rate people from new countries taking local differences into account without the need for extensive data. Over time, after enough data is gathered, a regular specialized model can be trained for the new country.

4. Solution Description

We propose to learn the function f using a GAN framework where the generator represents the mapping function that "translates" between countries. However, its easy to see that the regular GAN loss would not be appropriate for this problem. The generator could simply map all Laotians to a seemingly realistic Brazilian profile, with no relation with their probability of defaulting, still getting optimal cost. Instead, we are following the CycleGAN approach introduced by (Zhu et al., 2017). In addition to the regular generator loss (Eq. 5), it also includes a cost component

$$\mathcal{L}_{Cycle} = ||F(G(\boldsymbol{x}))||, \tag{4}$$

where $F()$ is a function that translates the sample back from source to target task. This cost function enables the model to translate the samples in a way that the relevant information is preserved to recover the original sample.

However, inspecting this cost functions shows that it is not enough to solve the learning task. Not only should the model be able to translate the samples back and forth across the tasks, it should also be able to perform the correct classification. Therefore, we propose to train the GAN model using a novel cost:

$$\mathcal{L} = -\mathcal{L}_D + \mathcal{L}_{Cycle} + |h(G(\boldsymbol{x})) - y| \tag{5}$$

where h is the model trained in the source task and specific purpose components $-\mathcal{L}_D$ and $\mathcal{L}_{Cycle}$. While the negative discriminator loss $-\mathcal{L}_D$ aims at deceiving the discriminator (thus generating realistic "virtual Brazilians"), $\mathcal{L}_{Cycle}$ makes sure the transformation is revertible, allowing a seamless restoration of the original data. Finally, the last component, $|h(G(\boldsymbol{x})) - y|$, encourages the generator to produce samples in a way that the source model is able to estimate the correct credit rate for Laotian samples.

5. Experiment

In this paper we describe an initial experiment we performed in the *Circle* domain. Our artificially-generated target task consists of classifying samples as "inside" or "outside" a circle of arbitrary radius r and origin (x_o, y_o), as illustrated in Fig. 3. With r, x_o, and y_o unknown, the classifier has to learn to label a new sample (x,y) based on a training set. To evaluate our mapping approach, we simulate both covariate and concept drift. Covariate drift is simulated by sampling points in the left side of the circle with a higher probability for $\mathcal{T}^s$. Concept drift is simulated by changing the radius of the circle across tasks.

We observed promising results showing the feasibility of the approach in a simple domain (Table 1). The benchmark result is represented by a model that we trained directly in the target domain without using any transfer and which achieved an average accuracy of 92.4%. Simply using the source model directly on the target domain data fails drastically and just showed similar performance as a random coin flip averaging just below 50%. Our approach with a learned mapping function was able to achieve very high per-

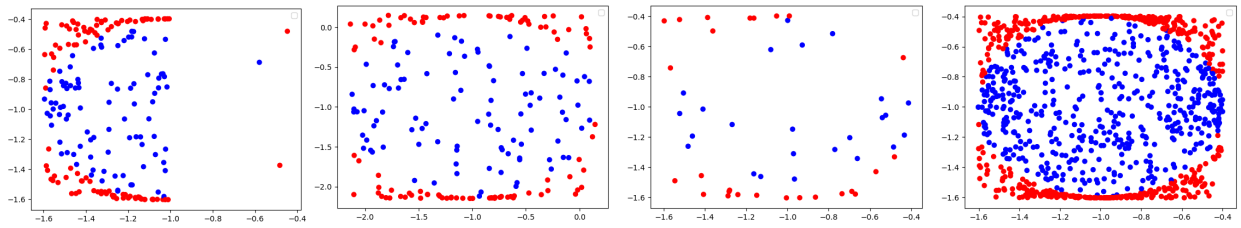

Figure 3. Data sampled from the *circle* environment. (a): $\boldsymbol{O}^s$ with covariate drift; (b): $\boldsymbol{O}^s$ without any drift; (c): $\boldsymbol{O}^t$ available to the algorithm (covariate and concept drift); (d): data for evaluating accuracy.

formance with an average of 83.3%, rivaling the benchmark model despite a source model developed in another task.

Using source model directly	49.3%
Training target model directly	92.4%
Mapped through modified GAN	83.3%

Table 1. Results for Circle domain using modified GAN model.

6. Conclusion and Further Work

While Machine Learning techniques have been successful in providing predictive abilities for many domains, building data sets remains a costly and difficult task. For this reason, being able to reuse data across different tasks is useful. We propose a novel mapping method based on GANs that enables reusing models in similar yet different tasks. Our preliminary experiment in the *circle* domains shows promising results. Even when fed only data without labels, the mapping approach achieved performance similar to the model trained with labeled data. Further work will evaluate our method in the creditworthiness estimation domain, which motivated our approach in the first place.

Acknowledgements

Ruben Glatt's and part of Felipe Leno Da Silva's work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC. LLNL-CONF-820966.

References

Basheer, I. and Hajmeer, M. Artificial neural networks: fundamentals, computing, design, and application. *Journal of Microbiological Methods*, 43(1):3 – 31, 2000.

Cortes, C. and Vapnik, V. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

Dawson, W. A., Glatt, R., Rusu, E., Soper, B. C., and Goldhahn, R. A. Hybrid information-driven multi-agent reinforcement learning. *arXiv preprint arXiv:2102.01004*, 2021.

Gama, J. a., Žliobaite, I., Bifet, A., Pechenizkiy, M., and Bouchachia, A. A survey on concept drift adaptation. *ACM Comput. Surv.*, 46(4), March 2014. ISSN 0360-0300. doi: 10.1145/2523813.

Glatt, R., Silva, F. L. D., and Costa, A. H. R. Towards knowledge transfer in deep reinforcement learning. In *2016 5th Brazilian Conference on Intelligent Systems (BRACIS)*, pp. 91–96. IEEE, 2016.

Glatt, R., Da Silva, F. L., da Costa Bianchi, R. A., and Costa, A. H. R. Decaf: Deep case-based policy inference for knowledge transfer in reinforcement learning. *Expert Systems with Applications*, 156:113420, 2020.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

John, G. H. and Langley, P. Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence (UAI)*, 1995.

Lei, Z., Liao, S., and Li, S. Z. Efficient feature selection for linear discriminant analysis and its application to face recognition. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pp. 1136–1139. IEEE, 2012.

Pan, S. J. and Yang, Q. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22 (10):1345–1359, 2010.

Webb, G. I., Hyde, R., Cao, H., Nguyen, H. L., and Petitjean, F. Characterizing concept drift. *Data Mining and Knowledge Discovery*, 30(4):964–994, 2016.

Zhu, J., Park, T., Isola, P., and Efros, A. A. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 2242–2251, 2017.