Domain2Vec: Deep Domain Generalization

Aniket Anand Deshmukh, Ankit Bansal, Akash Rastogi∗
Department of EECS
University of Michigan
Ann Arbor, MI 48105
anikede, arbansal, arastog @umich.edu

Abstract

We address the problem of domain generalization where a decision function is learned from the data of several related domains, and the goal is to apply it on an unseen domain successfully. It is assumed that there is plenty of labeled data available in source domains (also called as training domain), but no labeled data is available for the unseen domain (also called a target domain or test domain). We propose a novel neural network architecture, Domain2Vec (D2V) that learns domain-specific embedding and then uses this embedding to generalize the learning across related domains. Proposed algorithm, D2V extends the idea of distribution regression [17, 20] and kernelized domain generalization [5] to the neural networks setting. We propose a neural network architecture to learn domain-specific embedding and then use this embedding along with the data point specific features to label it. We show the effectiveness of the architecture by accurately estimating domain to domain similarity. We evaluate our algorithm against standard domain generalization datasets for image classification and outperform other state of the art algorithms.

1 Introduction

There has been significant interest in the last decade on the idea of generalization of learning, on whether it is possible to use similarities in underlying spaces and distributions [5], [16], [2], [10] or tasks [3] to build general learning algorithms to work with new settings. When working with datasets which have similarities in the underlying data distribution, it is possible to use the variation in the distributions to generalize learning tasks [4].

In our scenario, we are given $N$ datasets with each dataset drawn from different probability distributions such that each point in each dataset has a label or class associated with it. The goal is to learn a classifier such that, given a new dataset (with no training data/labels) drawn from a different but similar probability distribution, it is possible to provide labels to its points. The goal is to predict these labels corresponding to samples drawn from the marginal test distribution for the test dataset.

Classification datasets can be associated with a common structure which allows us to generalize behavior. Consider the example of image classification. We have multiple images of same objects from different cameras, and we train our image classification model for these cameras (e.g., Apple’s iPhone, Sony’s DSLR, and Google’s Pixel). During the test time, the goal is to classify images from Samsung’s Note. Different cameras have different optical structure and images from those cameras may look slightly different or may have different optical properties. What makes this more difficult is that there are no labeled images from Samsung’s Note and the classification model should be able to adapt without any labeled data. Instead of cameras, one could also think of different datasets VOC2007, LabelMe, Caltech-101, and SUN09 (VLCS).

∗All three authors contributed equally.

Preprint. Work in progress.
Our main contributions in this paper are: i) proposing the architecture for domain generalization and domain adaptation, ii) showing the effectiveness of the architecture by learning domain-specific embedding and thereby domain to domain similarity and iii) experimental evaluation of the architecture in different scenarios.

2 Formal Problem Statement

Let $\mathcal{X}$ be the feature space and $\mathcal{Y}$ be the label space. Let $\mathcal{P}_{\mathcal{X} \times \mathcal{Y}}$ be the set of probability distributions on $\mathcal{X} \times \mathcal{Y}$ and $\mathcal{P}_{\mathcal{X}}$ the set of probability distributions on $\mathcal{X}$. Furthermore, it is assumed that there exists a probability measure $\mu$ defined over $\mathcal{P}_{\mathcal{X} \times \mathcal{Y}}$. We are given sample points from $N$ distributions: $S_1 = ((X_{ij}, Y_{ij}))_{1 \leq j \leq n}$ such that $(X_{ij}, Y_{ij}) \sim P_{XY}^i$ and $P_{XY}^i \sim \mu$, $\forall i = \{1, ..., N\}$. For a new test dataset $S^T = (X^T_{ij})_{1 \leq j \leq n_T}$ and $P^T_X \sim \mu_X$, the goal is to accurately predict $(Y^T_{ij})_{1 \leq j \leq n_T}$. Let $\mathcal{X} = \mathcal{X} \times \mathcal{P}_X$ be the extended feature space. In the domain generalization setting, a decision function is a function $f : \mathcal{X} \times \mathcal{P}_X \rightarrow \mathcal{Y}$ that predicts $Y_j = f(X_j, \hat{P}_X)$, where $\hat{P}_X$ is the associated empirical distribution. We aim to learn a decision function $f$ that minimizes the expected loss $\mathbb{E}[\ell(f(\hat{X}), Y)]$, where $\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$ is a loss function.

With the assumption that the marginal distribution $P_X$ of domain represents that particular domain, we propose a general neural network architecture to learn some statistical estimates (eg. moments of distribution) from the data of that domain. These estimates (domain specific embeddings) are then combined with actual data from the domain to learn a final classifier. This procedure is exactly same as learning a decision function $f : \mathcal{X} \times \mathcal{P}_X \rightarrow \mathcal{Y}$ over extended feature space $\mathcal{X} \times \mathcal{P}_X$. We aim to learn a decision function $f$ to minimize the expected loss $\mathbb{E}[\ell(f(\hat{X}), Y)]$. We define empirical training error, empirical test error and generalization error as,

- **Empirical training error:**
  \[
  \hat{\ell}(f) = \frac{1}{N} \sum_{i=1}^N \frac{1}{n_i} \sum_{i=1}^{n_i} \ell(f(P_X^i, X_{ij}), Y_{ij}).
  \]

- **Empirical test error:**
  \[
  \hat{\ell}(f)^T = \frac{1}{n_T} \sum_{i=1}^{n_T} \ell(f(P_X^T, X^T_{ij}), Y^T_{ij}).
  \]

- **Generalization error:**
  \[
  \ell(f) = \mathbb{E}_{P_{XY} \sim \mu} \mathbb{E}_{(X^T, Y^T) \sim P_{XY}^T} \ell(f(P_X^T, X^T), Y^T).
  \]

In the next section, we describe the proposed architecture to solve the problem of domain generalization.

3 Proposed Architecture

The proposed architecture is motivated by the idea that one needs to find a classifier or a decision boundary from $\mathcal{X} = \mathcal{X} \times \mathcal{P}_X$ to $\mathcal{Y}$. In a typical neural network for supervised learning, we just find a classifier from $\mathcal{X}$ to $\mathcal{Y}$. The idea here is then to use a neural network to learn some statistic $T_X$ or the empirical characteristic function of the marginal probability distribution $P_X$ and concatenate the learned vector $T_X$ with a sample $X$ from $\mathcal{X}$. Let $\mathcal{X} \in \mathbb{R}^d$ be the $d$ dimensional feature space and for the $i^{th}$ distribution let $S^i_X = \{X_{ij}^i, ..., X_{in}^i\} \in \mathbb{R}^{n \times d}$. The typical neural network just operates on the dimension $d$ and transforms the original features $S^i_X$ to some $D$ dimensional features $T^i_X$. But this doesn’t learn any statistic from $n$ samples. Our proposed idea is to operate on dimension $n$ and get the domain-specific statistic $T^i_X$ of the data and use it along with the features $S^i_X$. The idea is that for data point $X_{ij}$ to have same label as $X_{pq}$, not only does $X_{ij}$ need to be close to $X_{pq}$ but so should $T^i_X$ to $T^p_X$. 

2
3.1 Related Work

3.1.1 Distribution Regression

Our proposed method is closely related to distribution regression. The goal of distribution regression is to label a set of points (cloud of points) assuming that all points belonging to one set come from one distribution. Some of the applications of distribution regression are to estimate statistic of the population and to draw some conclusions based on the entire data (rather than seeing each point in the set individually). [17] and [19] define the kernel on probability distribution and use kernel methods for classification or regression. Similarly [1] uses Gaussian process for distribution regression. [6] uses variational encoders to estimate statistics of the data which can then be used for classification or regression. [20] and [12] learn the function from a set of points (cloud of points from features space) to the label space. This function is called permutation invariant because any estimated statistic of the set does not depend on the ordering of elements in the set. They also propose permutation equivariant method which is related to our proposed architecture but is very restricted. Our architecture is very general and can be used for any domain generalization and multi-task learning application. Part of our proposed network is motivated by distribution regression, which we will use to get domain specific information.

3.1.2 Domain Generalization

One of the methods to solve the transfer learning problem when there is no labeled data available for the test domain is described in [5] and [2]. The approach is a distribution-free and kernel-based. We shall call this approach as kernelized domain generalization (KDG). It involves identifying an appropriate reproducing kernel Hilbert space (RKHS) and optimizing a regularized empirical risk over the space. The approach results in support vector machine defined over an extended feature space. This approach gives the generalization error bound in terms of number of datasets, number of examples per dataset and number of classes. Though the method is theoretically proven to be powerful, it is limited by the SVM. One needs to define the kernel that’s useful for the dataset in hand and it isn’t as powerful as neural networks when working with images and texts.

Multi-task Auto Encoder (MTAE) learns domain invariant features using autoencoder setting. It has a single encoder and D decoders, where D are the number of domains in the training data. At the end of autoencoder training, it learns features that are robust to variations across domains. The learned features are then used as inputs to a classifier. The classifier is trained on all the data from domains available during training [9]. Scatter Component Analysis (SCA) proposes a geometrical measure called scatter. Scatter helps to find a representation that minimizes the mismatch between domains and maximizes the separability of the data. Optimization in SCA can be reduced to generalized eigenvalue problem [8]. Classification and contrastive semantic alignment (CCSA) learns the domain invariant embedding by minimizing the sum of classification loss, confusion alignment loss, and
semantic alignment loss. Confusion alignment loss is a distance between distribution and a popular choice for such a loss is maximum mean discrepancy. Semantic alignment loss makes sure that samples from different domains and with different labels are mapped as far apart as possible in the embedding space [15]. [14] this paper is similar to Undo-Bias [11], where domain agnostic model (which benefits all domains) and domain-specific model is learned during training. When one observes a new domain, only domain agnostic model is used. Meta-learning domain generalization approach (MLDG) provides a model agnostic training procedure that improves the domain generality. It minimizes the training loss in such a way that virtual testing loss (which generalizes across different domains) is also minimized during the training phase [13]. The proposed method D2V has several advantages over these existing methods. First, the model is easy to understand, implement and motivated by distribution or dataset similarity. Domain2vec yields domain specific embedding which can be used to represent a particular dataset and quantify the similarity between different datasets.

4 Experimental Results

D2V can be viewed as comprising of two main components: 1) the main model and 2) the task model. In a rather unconventional setting, we operate on the batch size dimension in the tensors of the task network. The main network is fed a smaller batch of data belonging to one task while the task network is fed with a larger batch of data for the same task (typically as large as the entire task). The idea behind this practice is that D2V network learns task/domain specific information and hence ‘sees’ close to entire task at every iteration from the task network while also learning inter-domain nuances from the main network. Together, the two components of D2V enable it to learn robust embeddings across the domains, thereby improving the classification accuracy. For experiments in this paper, we use single hidden layer in both main network and task network. We get best hyper-parameters (learning rate, number of hidden units in the main network and task network, weight decay) using random search.

We implement the architecture in TensorFlow and test the results on a 2-dimensional synthetic dataset and one real dataset. For synthetic dataset, we sample data points uniformly from a 2-D unit box $[-1, 1] \times [0, 1]$. Data points are labeled according to their first dimension, i.e. 1 if it’s non negative and 0 if it’s negative. We then randomly sample $\theta$ from $[0, \pi]$ and rotate all these data points by $\theta$ to get dataset of one domain. Note that when $\theta$ of two different domains are close to each other then their data distribution and their decision functions are also very close to each other. In Fig. 2, we show the result on synthetic dataset by varying number of training domains from 8 to 256 and number of examples per domain from 8 to 1024. Number of test domain are fixed to 44 and number of examples per test domain are fixed to 1024. We compare the results with a pooling Neural Network (pooling NN or 1 HNN), which combines data from all domains and trains one single neural network. Domain2Vec improves the accuracy from 73% to 96% compared to Pooling NN. Also, accuracy of Domain2Vec model increases as it gets more number of training example per domain and more number of training domains.

We also show similarity between embeddings learnt by Domain2Vec for different domains and its comparison to true similarity (based on $\theta$) and random similarity. Let $d_p$ and $d_q$ denote the learnt embeddings for domain $p$ and domain $q$ respectively. Then, estimated similarity between domain $p$ and domain $q$ is $\exp\left(-\frac{\|d_p - d_q\|^2}{\sigma^2}\right)$. Let $\theta_p$ and $\theta_q$ denote the angles with which domain $p$ and domain $q$ were created respectively. In this case, known similarity is $\exp\left(-\frac{|\theta_p - \theta_q|^2}{\sigma^2}\right)$. From the Fig. 4, it can be seen that similarity calculated using embeddings is very close to known similarity.

We provide the results on a popular dataset in domain generalization community called VLCS which consists of 4 different datasets: VOC2007 (V), LabelMe (L), Caltech-101 (C), and SUN09 (S). We evaluate the classification accuracy of our model on the VLCS dataset by following the leave-one-domain-out methodology discussed in [9]. Specifically, we train our model on three out of four datasets (source datasets) and test it on the fourth dataset (target dataset). We follow the same experimental procedure as in [7]. From Table 1, we can see that Domain2Vec improves the accuracy over all baselines.

\[^2\text{Code is available here: https://github.com/aniketde/Domain2Vec}\]
5 Conclusion

We address the problem of domain generalization and propose a neural network architecture to learn domain-specific embedding and then use this embedding along with data point specific features.
to label it. We show the effectiveness of the architecture by learning domain-specific embeddings and then accurately estimating domain to domain similarity. We outperform other state of the art algorithms on a standard domain generalization dataset. Extensive empirical studies are needed to make the domain embedding model more robust. Domain2Vec model can be easily extended to multi-task learning with homogeneous tasks [13]. In future, we plan to devise a architecture similar to Domain2Vec that can learn domain-specific embeddings for other data types like text and speech.

References

[1] F. Bachoc, F. Gamboa, J.-M. Loubes, and N. Venet. A gaussian process regression model for distribution inputs. *IEEE Transactions on Information Theory*, 2017.

[2] G. Blanchard, G. Lee, and C. Scott. Generalizing from several related classification tasks to a new unlabeled sample. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 2178–2186. 2011.

[3] R. Caruana. Multitask learning. In *Learning to learn*, pages 95–133. Springer, 1998.

[4] A. Christmann and I. Steinwart. Universal kernels on non-standard input spaces. In *Advances in Neural Information Processing Systems*, pages 406–414, 2010.

[5] A. A. Deshmukh, S. Sharma, J. W. Cutler, and C. Scott. Multiclass domain generalization. *Limited Labeled Data Workshop at Advances in Neural Information Processing Systems (NIPS)*, 2017.

[6] H. Edwards and A. Storkey. Towards a neural statistician. *arXiv preprint arXiv:1606.02185*, 2016.

[7] C. Gan, T. Yang, and B. Gong. Learning attributes equals multi-source domain generalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 87–97, 2016.

[8] M. Ghifary, D. Balduzzi, W. B. Kleijn, and M. Zhang. Scatter component analysis: A unified framework for domain adaptation and domain generalization. *IEEE transactions on pattern analysis and machine intelligence*, 39(7):1414–1430, 2017.

[9] M. Ghifary, W. B. Kleijn, M. Zhang, and D. Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In *Proceedings of the IEEE international conference on computer vision*, pages 2551–2559, 2015.

[10] A. Gretton, A. Smola, J. Huang, M. Schmittfull, K. Borgwardt, and B. Schölkopf. Covariate shift by kernel mean matching. In *Dataset shift in machine learning*, Neural Information Processing series, chapter 8, pages 131–160. MIT Press, Cambridge, MA, 2009.

[11] A. Khosla, T. Zhou, T. Malisiewicz, A. A. Efros, and A. Torralba. Undoing the damage of dataset bias. In *European Conference on Computer Vision*, pages 158–171. Springer, 2012.

[12] H. C. L. Law, D. Sutherland, D. Sejdinovic, and S. Flaxman. Bayesian approaches to distribution regression. *population*, 2:3, 2018.
[13] D. Li, Y. Yang, Y. Song, and T. M. Hospedales. Learning to generalize: Meta-learning for domain generalization. *arXiv preprint arXiv:1710.03463*, 2017.

[14] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5542–5550, 2017.

[15] S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto. Unified deep supervised domain adaptation and generalization. *arXiv preprint arXiv:1709.10190*, 2017.

[16] K. Muandet, D. Balduzzi, and B. Schölkopf. Domain generalization via invariant feature representation. In *ICML (1)*, pages 10–18, 2013.

[17] B. Póczos, L. Xiong, D. J. Sutherland, and J. Schneider. Support distribution machines. 2012.

[18] S. Ruder, J. Bingel, I. Augenstein, and A. Søgaard. Learning what to share between loosely related tasks. *arXiv preprint arXiv:1705.08142*, 2017.

[19] Z. Szabó, B. Sriperumbudur, B. Póczos, and A. Gretton. Learning theory for distribution regression. *Journal of Machine Learning Research*, 17(152):1–40, 2016.

[20] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola. Deep sets. In *Advances in Neural Information Processing Systems*, pages 3394–3404, 2017.