Transfer Learning for Aided Target Recognition: Comparing Deep Learning to other Machine Learning Approaches

Samuel Rivera[a], Olga Mendoza-Schrock[b], and Ashley Diehl[b]

[a]Matrix Research, Dayton, USA;
[b]Air Force Research Laboratory/Sensors Directorate, Wright-Patterson Air Force Base, OH, USA

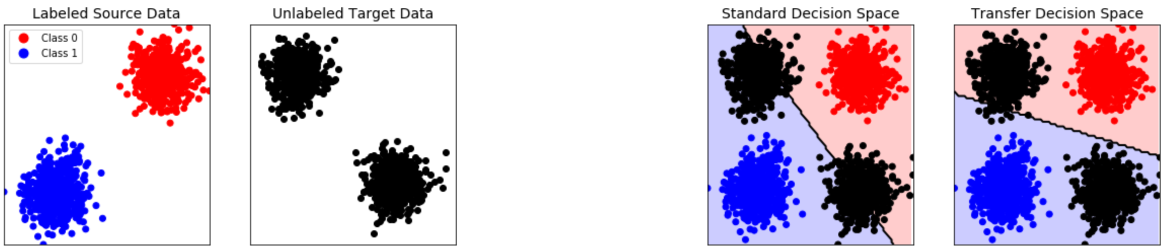
ABSTRACT

Aided target recognition (AiTR), the problem of classifying objects from sensor data, is an important problem with applications across industry and defense. While classification algorithms continue to improve, they often require more training data than is available or they do not transfer well to settings not represented in the training set. These problems are mitigated by transfer learning (TL), where knowledge gained in a well-understood source domain is transferred to a target domain of interest. In this context, the target domain could represents a poorly-labeled dataset, a different sensor, or an altogether new set of classes to identify.

While TL for classification has been an active area of machine learning (ML) research for decades, transfer learning within a deep learning framework remains a relatively new area of research. Although deep learning (DL) provides exceptional modeling flexibility and accuracy on recent real world problems, open questions remain regarding how much transfer benefit is gained by using DL versus other ML architectures. Our goal is to address this shortcoming by comparing transfer learning within a DL framework to other ML approaches across transfer tasks and datasets. Our main contributions are: 1) an empirical analysis of DL and ML algorithms on several transfer tasks and domains including gene expressions and satellite imagery, and 2) a discussion of the limitations and assumptions of TL for aided target recognition - both for DL and ML in general. We close with a discussion of future directions for DL transfer.

Keywords: Deep learning, Machine learning, transfer learning, adversarial networks

1. INTRODUCTION



(a) Source and target data scatter plots. We have some well characterized source data and some unlabeled target data that is related to the source data.

(b) Typical source data based decision space versus a transfer decision space that leverages source data and the unlabeled target data.

Figure 1: We show the overall idea of transfer subspace learning. We wish to learn a subspace where both the source and target data can best be classified.

Further author information: (Send correspondence to S.R.)
S.R.: E-mail: samuel.rivera@matrixresearch.com
O.M-S.: E-mail: olga.mendoza-schrock@us.af.mil

The abundance of data has created a paradox in recent years: while there exists more data than we could probably ever process, we never seem to have enough labeled data or the right type for our application. If training data does not match the target application data distributions, then poor generalization typically results for aided target recognition (AiTR) algorithms and machine learning (ML) models in general. The problem illustrated in Fig. 1 has spawned an entire subfield of ML, called transfer learning (TL) (See Pan[1] for a fantastic survey) where the goal is to *transfer* knowledge from well characterized *source* data to less understood or available *target* data. Source and target are different from the tradition *train* and *test* labels because it is possible for the target data, unlike test data, to bolster the training process. The distinction is that the target set is from a different domain or of different classes than the source.

TL problems for AiTR and classification manifest in several common scenarios. The most common case involves needing to classify data on an unlabeled target set that closely resembles the source set but with different data distributions. This occurs for AiTR if the weather changes, for example, or if we use a different type of sensor to collect new data. This scenario also occurs when attempting to learn from synthetically generated data. In both cases the source and target classes and their domains match, but with slightly different distributions over the data samples. Another case of transfer learning requires transfer to related but not identical classes. We may know how to classify Honda sedans versus SUVs, for example, but would like to classify Toyota sedans versus SUVs. In this case the target set is similar to the source set in class hierarchy, but with slightly different specific target classes. Another type of transfer involves transfer to a different database or different domain, such as from electro-optical (EO) to synthetic aperture radar (SAR) imagery. In that case it is clear that the source and target distributions will not match. The problem becomes even more challenging when the source and target reside in different feature spaces altogether, as we show for lung and breast gene sequences in Sec. 3.3. The challenge comes from needing to learn a decision rule across two separate feature spaces, but with only label information for the source data.

These problems are not new and many approaches try to address at least some components of the problem using different strategies. The three main strategies are 1) traditional TL, 2) learning robust decision rules (traditional ML), and 3) learning robust feature representations. The focus of traditional TL is to transfer the decision model from the source to the target after accounting for the change between source and target distributions. A large subset of traditional TL is transfer subspace learning (TSL), which aims to learn feature representations where source and target distributions match. The transfer diffusion map (TrDM) algorithm,[2] for example, combines the benefits of diffusion maps (DMs)[3] for learning manifold preserving data embeddings, and transfer subspace approaches that minimize the divergence between source and target distributions.[4]

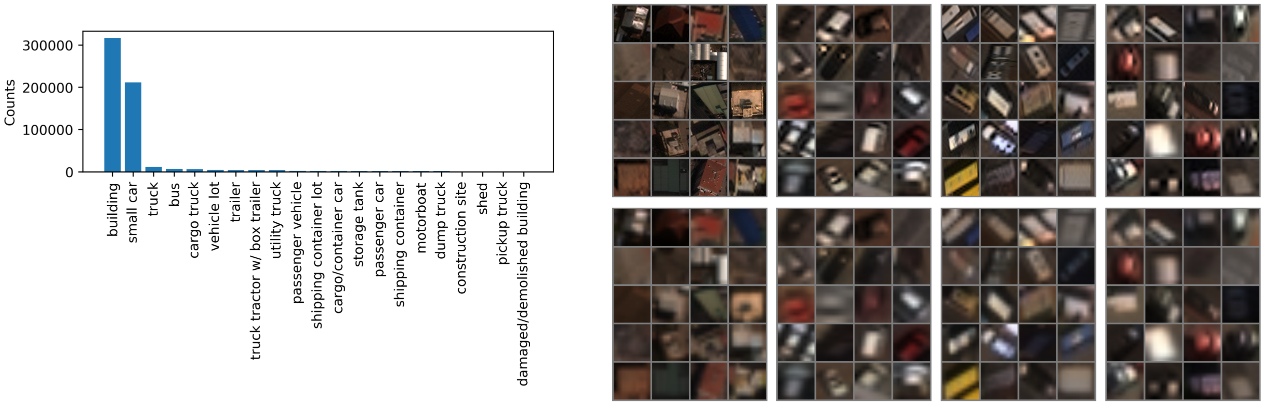
Deep learning (DL) or neural networks (NNs) are other promising areas where transfer learning has been applied. This is an especially interesting area since DL methods have quickly become one of the most popular and accurate approaches for recognition in recent years. Transfer is done by learning robust representations that work across the source and target. This is achieved using a variety of techniques including fine-tuning, regularization, or by explicitly trying to match the source and target similarity in the feature extraction layer during training by using an adversarial network framework.[5–8] A very relevant study applies the adversarial and robust feature learning approaches for AiTR.[9] Our goal is to compare these different approaches across different transfer problems to understand the benefits and shortcomings of the different approaches.

Our contributions are as follows: We lay out some of the important current transfer problems, then establish empirical benchmark comparisons on measured, challenging, and practical datasets for some of the main transfer problems and relevant ML approaches. Finally, we discuss some practical insights into the benefits and limitations of the competing approaches as well as recommendations for the practitioners. Throughout this article, we will not draw a distinction between DL and ML, since DL is one component of the broader field of ML just as TL is one component of ML. When we refer to ML, it is understood that network-based models are included in that set of algorithms unless we explicitly state otherwise.

2. DATASETS AND ALGORITHMS

2.1 Datasets

xView Dataset The xView dataset[10] is a collection of satellite imagery of about a million images over 60 classes of objects at 0.3m ground resolution. In this work we focus on the fully annotated training set. As is the



(a) xView sample counts

(b) xView cropped examples

Figure 2: Here we show a histogram of sample counts for the most frequent classes in the xView dataset in Fig. 2a, and some example images in Fig. 2b. The top row of image blocks shows the high-resolution images while the lower row of blocks shows the corresponding low-resolution images. From left to right, the classes are: building, small car, bus, and passenger vehicle. This subset of images illustrates some of the challenges associated with the xView dataset. Namely, some classes look very similar such as small car versus passenger vehicle. Furthermore, even some high-resolution images are quite difficult for a human observer to classify without the image context.

case with most real datasets, the class counts follow a heavy tail distribution where a small number of classes make up most of the dataset while the remaining classes have relatively few examples. For the xView dataset, the two most frequent classes make up 90% of the samples. Furthermore, of the 60 classes, only 21 have over 1000 samples represented in the set. See Fig. 2a for a histogram of the sample counts for those 21 classes.

Class imbalance causes problems for ML algorithms where typically the most prevalent classes are often favored at the expense of the less represented classes. Several approaches to alleviating this exist including weighting schemes and resampling during training.[11–13] To address the imbalance problem, in our experiments we first balance the classes by randomly selecting 1000 samples for each class of interest. This allows us to evaluate and compare each algorithm in an established setting while avoiding the issues related to imbalanced data that are not within the scope here.

Another challenge of real world data of visual categories is that the object sizes vary dramatically. Images in the xView dataset are collected at a resolution of 0.3m on the ground which means that a *small car*, for example, will have a smaller bounding box than a *vehicle lot* or a *trailer*. Moreover, the aspect ratios of those bounding boxes will also vary widely since *trailers* are typically much longer than *small cars*. To make matters worse, the bounding boxes define the upper and lower corners of horizontal box which means that rectangular objects in the scene that are off-axis with respect to the camera roll angle will have large segments of background included in their bounding boxes compared to their axis-aligned counterparts. The typical approaches to make object images uniform are to pad the image bounding boxes along one or both axes, or to apply some kind of image rescaling. In our implementations we chose to crop the object using the bounding boxes identified in the data annotations and scale all classes to the same square size. Some examples are shown in Fig. 2b.

Gene Sequence Dataset The gene sequence dataset used here was taken from the lung and breast mRNA microarray gene expression datasets from the Biomedical Knowledge Repository (BKR) developed at the National Library of Medicine (NLM).[14] Specifically we used Lung Adenocarcinoma (LUAD) and the Breast Invasive Carcinoma (BRCA) data. In the original dataset, each sample in both domains contain 60,484 different genes (features). The breast cancer data contains 112 positive samples and 117 negative samples. For the lung cancer data, there are 49 positive samples and 49 negative samples. For our purposes, we view this as a 60,484 dimensional vector per subject (sample).

The 60,484 dimensions are not the same for breast and lung cancer. For the transfer learning techniques discussed so far, the attributes (feature spaces) need to be the same for the source and target data. It is often not a trivial task to construct these transfer problems and this is an open area of research. Mendoza-Schrock constructed the transfer learning problem and aligned the feature spaces by taking the intersection of both the breast and lung cancer data while discarding the remaining features.[15] For EO this is analogous to first ensuring that the images are of equivalent size. This gives samples of dimensionality 22,622.

2.2 Learning Models and Algorithms

The goal of this article is to compare broad classes of transfer learning approaches. Therefore, we selected well established methods in subspace transfer learning, nonlinear hyperplane classifiers, and DL.

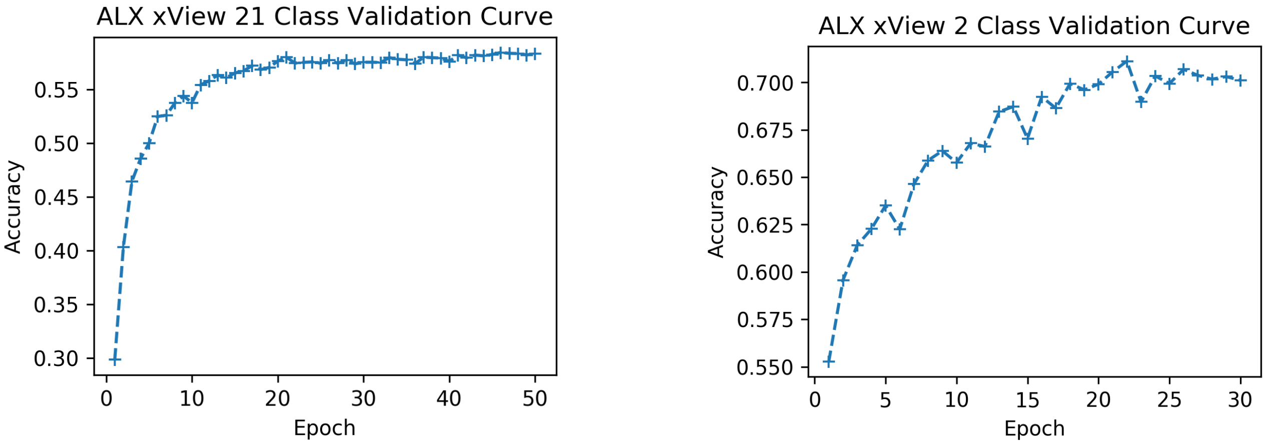
Traditional ML Classifier Support vector machines (SVMs) have become one of the fundamental approaches to classification in ML. The simple and intuitive idea behind the standard linear SVM is to learn the hyperplane that best separates two classes of data in the feature space.[16] A penalty parameter is used to allow for samples to be misclassified, or exist on the wrong side of the hyperplane. The approach has been extended to nonlinear decision boundaries through the kernel trick.[17] By using the kernel, samples are implicitly mapped to a new feature space where a linear hyperplane can better separate the classes. We use the SVM implementations in python's scikit-learn library[18] with the RBF kernel. An important aspect of the traditional ML approaches is that they do not use the target data during model training.

DL The transfer approaches evaluated for the DL framework were *fine-tuning* and *adversarial learning* for domain adaptation. Recently, research has been done to generate realistic synthetic data for training NNs using the generative adversarial network (GAN) framework.[8] We do not evaluate these methods since the problem of generating quality training data is out of the scope of this paper. However, a recent approach that relies on GANs to match source and target distributions[7] will be considered in future research.

For our image experiments we use the AlexNet[19] implementation of python's PyTorch library that was pre-trained with the ImageNet database.[20] That means that the feature extraction layers were already organized to represent a variety of visual object classes. Images were scaled to 226 × 226 pixels then cropped to 224 × 224 pixel color images as required for AlexNet. We replaced the network classification layers and only updated the classification layer parameters using our training data. We also implemented a domain adversarial (DA) variant of AlexNet by adding a domain classification layer after the feature extraction layers.[5,6] We used the pre-trained feature extraction layers while fine-tuning the classification layers as in the standard AlexNet case. An important distinction between the standard AlexNet and its DA variant is that the DA variant has access to the unlabeled target data during training.

For the genetic sequence data a convolutional neural network (CNN) architecture was inappropriate so we implemented a three layer fully connected network. The input linear transformation layer was followed by a rectified linear units (ReLU) activation layer. The output was connected to another linear transformation layer before the class output. We optimized using the cross entropy loss.

DM and TrDM The TrDM algorithm[2] combines the benefits of DMs[3] that learn a data embedding that preserves the underlying data manifold structure, and transfer subspace approaches that minimize the divergence between source and target distributions.[4] We ported our original MATLAB implementation to Python for this work. In our experiments we evaluated both the standard DM and the TrDM in order to verify if the subspace transfer provided benefit beyond just learning a data embedding. We also considered two types of classification rules. For the standard classification rule we used the k-nearest neighbors (k-NNs) of the source data. For the *1Known* classification rule, we labeled a single random sample from each target class and determined the first nearest neighbor. This models the case where we learn a data embedding using all source and target data, but only know a single sample's label for each target class. The DM algorithm variants implemented make use of both the source and target data when learning the embeddings.



(a) AlexNet validation accuracy for 21 class xView dataset (Experiment 1).

(b) AlexNet validation accuracy for two class xView dataset (Experiment 2).

Figure 3: We show the learning curves for AlexNet on the xView dataset with the 21 most prevalent classes in Fig. 3a and for the two class (*bus* and *cargo truck*) set in Fig. 3b. In both cases the validation set accuracy saturates at about 20 epochs.

3. EXPERIMENTS AND RESULTS

In this section we present our experiments and the associated results. Our goal in the experiments section is to provide an empirical evaluation of different approaches to transfer learning across a variety of situations described above. We also plan to provide some practical guidelines into training the learning models as well as comments on the challenges and assumptions of the different algorithms.

3.1 Experiment 1: High Quality to Degraded Data Transfer

For this experiment we were interested in transfer to degraded versions of the same classes. We setup this transfer task with the xView dataset, where we transferred from high-resolution to low-resolution versions of the same classes. We used the 21 classes having more than 1000 samples in the training set that are shown in Fig. 2a. Low-resolution versions of the high resolution data were synthetically generated by first resizing from the original image chip size to 50×50 pixels to have a common starting size then rescaled to 10×10 pixels before scaling to the size required for the learning model. The high-resolution samples were scaled from their original chip size to the size required for the algorithm without any intermediate steps. All rescaling applied bilinear interpolation using the Pillow image processing library in Python.

Our primary metric of interest was the classification accuracy, or the proportion of correct classification decisions on the target set. To verify the reproducibility of results, we ran multiple testing iterations for each experiment and calculated the mean and standard deviation of the accuracy over the test iterations. On each testing iteration we sequestered a random subset of 30% of the dataset for testing while the remaining 70% samples were used for model training and validation. We applied the low-resolution transformation process described above to synthetically degrade the test (target) samples. We did ten test iterations for all models except the DM and TrDM models where we did five iterations due to computational limitations.

For AlexNet, care must be taken when training the model to ensure that the algorithm does not overfit to the training data. Therefore we used a sequestered validation set of 30% of the training data to define a stopping point for fine-tuning the classification layers. We chose a stopping point of 20 epochs based on the validation curve shown in Fig.3a.

Because of the way we synthetically generated low-resolution test samples, it was possible in this case to evaluate the relative performance on a high-resolution versus a low-resolution test set. Therefore, we show

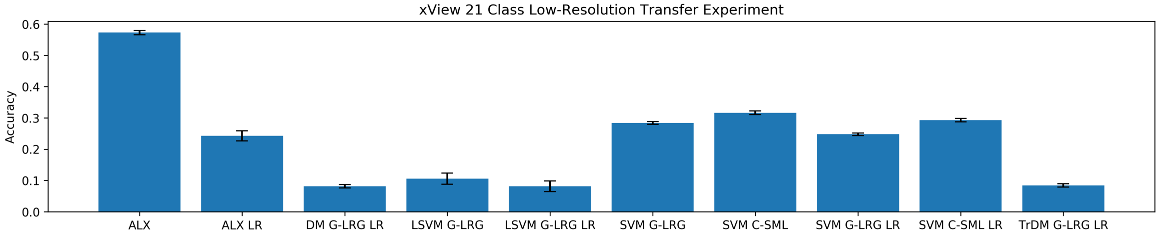


Figure 4: Accuracy for 21-class xView high-resolution and low-resolution transfer experiments. Algorithms are denoted by the first string on the x-axis labels: ALX, DM, SVM, LSVM, and TrDM corresponding to AlexNet, DM, kernel SVM, Linear SVM, and TrDM, respectively. G-LRG and C-SML for the SVM and DM experiment variants denote whether images were 224 × 224 grayscale or 20 × 20 three-channel color images, respectively. The LR at the end denotes if the result is for the low-resolution transfer set. The results without the LR means classification on a high-resolution version of the test set. For example, ALX is the classification result for AlexNet on high resolution data.

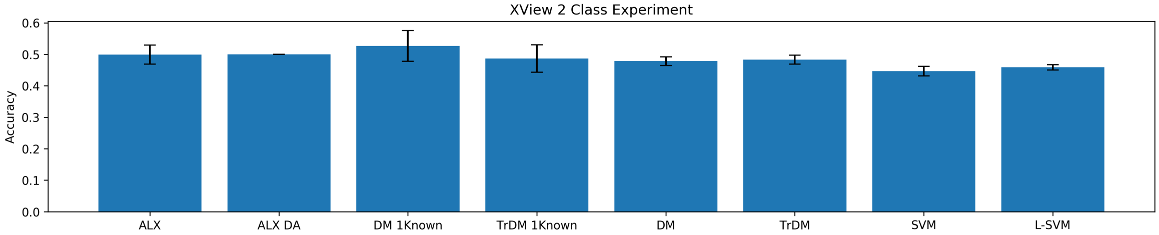


Figure 5: Accuracy for xView cross-class transfer experiments. ALX and ALX DA denote AlexNet and the domain adversarial variant of AlexNet. DM 1Known and TrDM 1Known denote diffusion map and transfer diffusion map algorithms where one labeled sample from each target class was used for a 1-NN classifier. The variants without 1Known used the k-NN classification rule learned from the training source data. SVM and L-SVM denote the kernel and linear SVM.

results where appropriate for both high and low-resolution cases. Mean and standard deviation of accuracy over the test (target) trials is shown in Fig. 4.

The results show that the CNN did best on the source data, but comparably to the traditional ML approaches on the target data. There was also less of a performance drop for the low-resolution versus high-resolution data when training the SVM algorithm on smaller color images versus larger grayscale images.

3.2 Experiment 2: Cross-Class Transfer

In this experiment we were interested in transfer from well-characterized classes to related but new classes. We again used the xView dataset, where the source classes were *bus* and *cargo truck*, and the target classes were *passenger vehicle* and *utility truck*. The idea behind the transfer is that the two classes in each set represent vehicles that either transport *people* or *cargo*. The target and test classes under consideration, although quite different in appearance, are semantically related. The hope with cross-class transfer learning is that we could use the knowledge learned for identifying the source classes to the identifying completely new, but related classes like these.

As in the previous experiment, we balanced the classes by randomly selecting a subset of 1000 samples from each class randomly for each testing iteration. The full set of 2000 target samples were always used as the testing set. The training set was further split up into 70% for model training and 30% for validation. We did 10 test iterations for all models and show the accuracy in Fig. 5. As with the previous experiment, we identified the stopping point for fine-tuning AlexNet to be 20 epochs based on the validation accuracy curves of Fig. 3b.

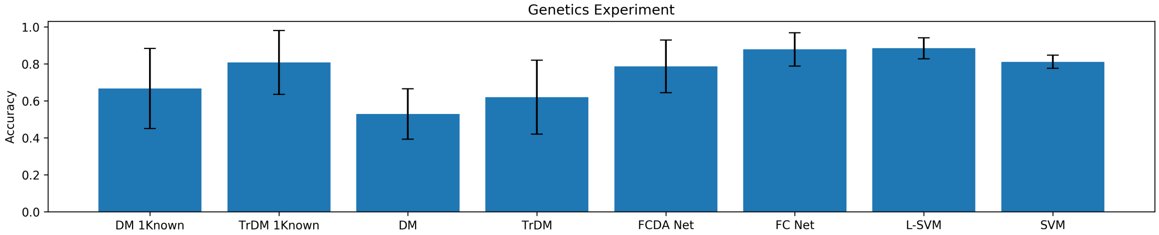


Figure 6: Accuracy for cancer transfer experiments. FC Net and FCDA Net denote the fully connected NN and the domain adversarial variant. The remaining algorithms are defined in Fig. 5.

Overall, no model did particularly well, with all algorithms performing at approximately chance level or 50%. There were some cases where algorithms performed below chance which means that the models may have learned the opposite of the desired mapping. This means that those models saw utility trucks as more similar to busses in appearance, while passenger vehicles looked more like cargo trucks. This class reversal might be avoided if we can use at least a single labeled target exemplar to effectively label the feature space. However, experiments where we used the nearest neighbor classification rule along with a single known target label for each class still gave about chance performance when accounting for the error variance.

3.3 Experiment 3: Cross-Domain Transfer

For this experiment we were interested in transferring a common set of class labels from one domain to another. This represents a general type of problem that comes up for cases such as transfer across modalities like EO to SAR. In this case the two domains corresponded to *lung* versus *breast* gene sequences, and the two common class labels were *positive* or *negative* for cancer.

Unlike the previous experiments that used $2D$ image data, gene sequences are $1D$ vector data. Furthermore, they do not correspond to visual categories of objects in nature, so neither CNNs nor fine-tuning based on visual categories applies. Therefore, we used a 3-layer fully connected network and with the DA complement of that network for the NN models in this experiment. We found that performance saturated very quickly, by about five training epochs, so we used that as the NN training stopping point.

Gene sequences across domains were aligned as described in Sec. 2.1. We did not balance the positive and negative class sample size because they were already the same (lung genes) or very close (breast genes). We evaluated the cases of training on lung data and testing on breast data and vice versa. We always tested with the full target set, while we trained with a random subset of 70% of the training data. To ensure reproducibility of the learned models, we evaluated accuracy over 10 test iterations for each model. This corresponds to 20 test iterations when considering both transfer directions. Results are shown in Fig. 6.

The experiment had two surprising results: that the linear SVM outperformed all other models on the transfer task and that the NN did nearly as well as the linear SVM. This is surprising if we consider the fact that the SVM, unlike the TrDM algorithm, had no access to the unlabeled target data during training. Furthermore, the linear SVM model is the simplest of all the models with the fewest parameters. This suggests two things: i) There exists a good common hyperplane for separating the positive and negative samples across the domains, and ii) the sample size is much too small to learn good nonlinear decision boundaries.

Given that the simplest linear model did the best and that the small sample size may be a contributing factor, the NN result is surprising because the NN did nearly as well on transfer while being more flexible and having more parameters than the linear SVM. Another interesting finding is that adding the DA penalty to the network and allowing the network to see unlabeled target data caused a slight performance decrease. Issues related the DA regularization will be further discussed in Sec. 4.

4. DISCUSSION

In this section we summarize some the important practical issues related the the models and transfer problems discussed in this article. The purpose is to aid practitioners in applying these methods appropriately, or selecting the right approach for their problem.

Training and Computational Load Training these models is a challenging and time consuming process that requires specialized hardware such as graphical processing units (GPUs) and a few dozens gigabytes (GBs) of random access memory (RAM) depending on the dataset size. The largest dataset used here only contained 21,000 samples which is considered to be a small to medium sized learning problem by today's standards. While the CNN algorithms could be trained on a standard laptop (with or without GPUs), the DM algorithms required a server with a large amount of RAM to process the $N \times N$ Gram matrix. Our off-the-shelf SVM implementation was somewhere in the middle in terms of memory requirements. In terms of time, the DM also took the longest time to train for the largest experiment (on the order of days). For the smaller problems, all algorithms could be trained on a standard laptop in a matter of a few seconds to a few hours depending on the stopping criteria and model parameters. An important point regarding the training time and memory requirements is that the TrDM and DM implementations were based on the original research code and were not optimized.[2] The SVM and NN libraries, however, were quite mature and highly optimized. This means that there could potentially be some big performance gains to be seen for the TrDM algorithm.

Parameter Tuning All algorithms require tuning some parameters as discussed above. SVM was the simplest for our experiments, with a heuristic kernel parameter based on the inverse of the squared mean pairwise distance between samples giving good results across experiments. The TrDM neighborhood size parameter, random walk distance, and regularization parameter were the most challenging to determine which is one of the major drawbacks of that approach. It is quite likely that the TrDM did not achieve their potential for accuracy due to suboptimal parameter choices. We found that setting the kernel parameter similarly to the SVM kernel parameter gave respectable starting point, but computational demands limited our ability to finely tune the TrDM parameters. One important avenue for future research is to determine data-driven ways to set all the TrDM parameters before model training.

DLs parameter tuning is a bit different since the entire model selection process can be argued to be a parameter selection problem. Selecting the model in itself is a parameter. Then that network can be adjusted with all sorts of different regularization schemes and hyper-parameters controlling learning update behavior. This process of iterating over networks typically takes some intuition along with constant trial and error. Although these models are incredibly flexible and achieve state-of-the-art classification results in practice, it is not a trivial task to make DLs work on real problems. Our best advice is to use the simplest model that best characterizes the data. As we saw in the gene sequence experiment, a simple linear model is sometimes best.

Transfer Learning Assumptions And Limitations Across Algorithms The assumption for the transfer learning problems discussed in this article is that the source and target classes share some common underlying structure so that a category in the source set can be mapped to the corresponding category in the target set. We saw from the cross-class transfer algorithm that although there may be a similar semantic relationship between the source and target classes, the relationship might not be directly extracted from the sample features. Even explicitly considering the target data manifold as with the TrDM did not lead to generalization for such a challenging problem. An important area for future research will be to understand the cases where transfer learning should be applied.

The assumption of isomorphic class structure implies that that the source and target categories have same number of categories or classes. However, that is not a requirement for TL. Some algorithms not evaluated here such as zero-shot and one-shot models can generalize to unseen classes. Schemes also exist for extending the DM and DL based algorithms to new numbers of classes. The DM algorithms, for example, learn a feature embedding where any sort of decision rule could be applied. Similarly, the feature extraction output of a NN could be used for another decision rule.

5. CONCLUSION

The big goals of transfer learning are transfer to new categories and new feature spaces. In this paper we evaluated several ML approaches to transfer learning on high quality to degraded data, cross-class, and cross-domain transfer. The degraded data experiment showed that while a CNN surpasses other approaches on source data, the transfer performance is comparable to traditional ML approaches. The cross-class experiment was especially challenging, with all algorithms performing at about chance level. An important avenue of future research will be to understand the cases where cross-class transfer is possible as well the the types of algorithms that will work best. The cross-domain gene sequence experiment showed that a simple linear model can sometimes outperform TL and NN approaches to transfer. This result is likely due to the specific datasets evaluated, and is an important reminder that it is important to understand the datasets under consideration because one algorithm will not be the best across all datasets. That experiment also built on previous research that aligned the source and target feature spaces. Future work should be done to eliminate the barrier that source and target should be in the same feature space. There were some limitations in this paper that would be addressed in future work. We did not perform an exhaustive set of transfer experiments. For example, we did not do a cross-domain experiment for the visual classes or a high quality to degraded experiment for the non visual classes. We would also like to automate the parameter selection for the TrDM approach.

ACKNOWLEDGMENTS

We would like to thank Ms. Nicole Robinson for her preliminary work on SVM and other techniques for transfer learning, Mr. Ryan Sperl for his work in porting the Diffusion Transfer algorithms to Python as well as preliminary work with AlexNet and fine-tuning, and Mr. Todd Rovito for supporting this research. This work was supported by Air Force Research Laboratory (AFRL), Air Force Office of Scientific Research (AFOSR), Dynamic Data Driven Application Systems (DDDAS) Program, and Dr. Erik Blasch.

REFERENCES

[1] Pan, S. J. and Yang, Q., "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering* **22**, 1345–1359 (Oct 2010).

[2] Mendoza-Schrock, O., Rizki, M. M., Raymer, M. L., and Velten, V. J., "Manifold transfer subspace learning (mtsl) for high dimensional data — applications to handwritten digits and health informatics," *in Proceedings of the International Conference on IP, Comp. Vision, and Pattern Recognition (IPCV'17)* , 3–9 (2017).

[3] Coifman, R. R. and Lafon, S., "Diffusion maps," *Applied and Computational Harmonic Analysis* **21**(1), 5 – 30 (2006). Special Issue: Diffusion Maps and Wavelets.

[4] Si, S., Tao, D., and Geng, B., "Bregman divergence-based regularization for transfer subspace learning," *IEEE Transactions on Knowledge and Data Engineering* **22**, 929–942 (July 2010).

[5] Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V., "Domain-Adversarial Training of Neural Networks," **17**, 1–35 (2015).

[6] Ganin, Y. and Lempitsky, V., "Unsupervised domain adaptation by backpropagation," in [*Proceedings of the 32nd International Conference on Machine Learning*], Bach, F. and Blei, D., eds., *Proceedings of Machine Learning Research* **37**, 1180–1189, PMLR, Lille, France (07–09 Jul 2015).

[7] Sankaranarayanan, S., Balaji, Y., Castillo, C. D., and Chellappa, R., "Generate To Adapt: Aligning Domains using Generative Adversarial Networks," in [*CVPR*], (2018).

[8] Peng, X. and Saenko, K., "Synthetic to real adaptation with deep generative correlation alignment networks," *CoRR* **abs/1701.05524** (2017).

[9] Elshamli, A., Taylor, G. W., Berg, A., and Areibi, S., "Domain adaptation using representation learning for the classification of remote sensing images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* **10**, 4198–4209 (Sept 2017).

[10] Lam, D., Kuzma, R., McGee, K., Dooley, S., Laielli, M., Klaric, M., Bulatov, Y., and McCord, B., "xview: Objects in context in overhead imagery," *CoRR* **abs/1802.07856** (2018).

[11] He, H. and Garcia, E. A., "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering* **21**, 1263–1284 (Sep. 2009).

[12] Japkowicz, N. and Stephen, S., "The class imbalance problem: A systematic study," *Intell. Data Anal.* **6**, 429–449 (Oct. 2002).

[13] Buda, M., Maki, A., and Mazurowski, M. A., "A systematic study of the class imbalance problem in convolutional neural networks," *Neural networks : the official journal of the International Neural Network Society* **106**, 249–259 (2018).

[14] B. I. of MIT and Harvard, "The biomedical knowledge repository (bkr) data set," (2016).

[15] Mendoza-Schrock, O., *Diffusion Maps and Transfer Subspace Learning*, PhD thesis, Wright State University (2017).

[16] Burges, C. J. C., "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery* **2**, 121–167 (1998).

[17] Duda, R. O., Hart, P. E., and Stork, D. G., [*Pattern Classification, 2nd Ed*], Wiley-Interscience, New York, NY, USA (2000).

[18] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E., "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research* **12**, 2825–2830 (2011).

[19] Krizhevsky, A., Sutskever, I., and Hinton, G. E., "ImageNet Classification with Deep Convolutional Neural Networks," *Advances In Neural Information Processing Systems* , 1–9 (2012).

[20] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L., "ImageNet: A Large-Scale Hierarchical Image Database," in [*CVPR09*], (2009).