Abstract

Learning to align distributions by minimizing an adversarial distance between them has recently achieved impressive results. However, such models are difficult to optimize with gradient descent and they often do not converge without very careful parameter tuning and initialization. We investigate whether turning the adversarial min-max problem into an optimization problem by replacing the maximization part with its dual improves the quality of the resulting alignment. Our empirical results suggest that using the dual formulation for linear and kernelized discriminators results in a more stable convergence to a desirable solution. We test our hypothesis on the problem of aligning two synthetic point clouds on a plane and on a real-image domain adaptation problem using a subset of MNIST and USPS. In both cases, the dual formulation yields an iterative procedure that gives more stable and monotonic improvement over time.

1 Introduction

Adversarial methods have recently become a popular choice for learning distributions of high-dimensional data. The key idea is to learn a parametric representation of a distribution by aligning it with the empirical distribution of interest according to a distance given by a discriminative model. At the same time, the discriminative model is trained to differentiate between true and artificially obtained samples. Generative Adversarial Networks (GANs) that use neural networks to both discriminate samples and parameterize a learned distribution, have achieved particularly impressive results in many applications such as generative modeling of images\cite{goodfellow2014generative, mathieu2016discovering, salimans2016improved}, image super-resolution\cite{ledig2017photo}, and image-to-image translation\cite{isola2016image}. Adversarial matching of empirical distributions has also shown promise for aligning train and test data distributions in scenarios involving domain shift\cite{ganin2014unsupervised, long2015learning, long2016fully, saenko2010adapting}.

However, GANs and related models have proved to be extremely difficult to optimize. It has been widely reported that training GANs is a tricky process that often diverges and requires very careful parameter initialization and tuning. Arjovsky and Bottou\cite{arjovsky2017towards} have recently identified several theoretical problems with loss functions used by GANs, and have analyzed how they contribute to instability and saturation during training.

In this paper, we focus on one of the major barriers to stable optimization of adversarial methods, namely their min-max nature. Adversarial methods seek to match the generated and real distributions by minimizing the discrepancy between the two, which is often defined as a maximum over values of different test functions that could differentiate between distributions. More specifically in the case of GANs, the distance is usually considered to be equal to the likelihood of the best neural network classifier that discriminates between the distributions, assigning “real or generated?” labels to the input points. This way, in order to align these distributions one has to minimize this maximum likelihood w.r.t. the parameters of the learned or aligned distribution.
Unfortunately, solving min-max problems using gradient descent is inherently very difficult. Below we use a simple example to demonstrate that different flavors of gradient descent are very unstable when it comes to solving problems of this kind.

To address this issue, we explore the possibility of replacing the maximization part of the adversarial alignment problem with a dual minimization problem for linear and kernelized linear discriminators. The resulting dual problem turns out to be much easier to solve via gradient descent. Moreover, we find that it is equivalent to the minimization of the iteratively reweighted Maximum Mean Discrepancy (MMD) \[12\], and that support vectors of the resulting classifier form two subsets of points from different distributions that are closest to each other in terms of MMD among all other pairs of subsets of given size.

We first evaluate how well our dual method can handle a point alignment problem on a low-dimensional synthetic dataset. Then, we compare its performance with the analogous primal method on a real-image domain adaptation problem using a subset of MNIST and USPS digit datasets. Here the goal is to align the feature distributions so that a classifier trained on MNIST does not loose accuracy on USPS. In both cases, the scheme that relies on the dual formulation of the adversarial distance monotonically improves over time, whereas using the primal formulation results in a constantly drifting objective and often does not converge to a solution.

Our contributions can be summarized as follows:

- we explore the dual formulation of the adversarial alignment objective for linear and kernelized linear discriminators and how they relate to the Maximum Mean Discrepancy;
- we demonstrate experimentally on both synthetic and real datasets that the resulting objective leads to more stable convergence and better alignment quality;
- we apply this idea to a domain adaptation scenario and show that the stability of reaching high target classification accuracy is also positively impacted by the dual formulation.

2 Related Work

There has been a long line of work on unsupervised generative machine learning, a review of which is beyond the scope of this work. Recently, adversarial methods for learning generative neural networks have achieved popularity due to their ability to effectively model high dimensional data, e.g., generating very realistic looking images. These include the original Generative Adversarial Nets (GANs) \[1\] and follow up work proposing improved formulations such as Wasserstein GANs \[13\] and Conditional GANs \[14\].

Related ideas have been proposed for unsupervised domain adaptation. Neural domain adaptation methods seek to improve the performance of a classifier network on a target distribution that is different from the original training distribution. This is achieved by adding an objective that minimizes the difference between representations learned for source and target data. Some models align feature representations across domains by minimizing the distance between first or second order feature space statistics \[15\]-\[17\]. When adversarial objectives are used for domain adaptation, a domain classifier is trained to distinguish between the generated source and target representations, both using the standard minimax objective \[7\], as well as alternative losses \[8\]-\[10\]. Instead of aligning distributions in feature space, several models perform generative alignment in pixel-space, e.g., using Coupled GANs \[13\] or conditional GANs \[19\]. These models adapt by “hallucinating” what the raw training images might look like in the target test domain. A related line of work uses adversarial training to translate images in one domain to the “style” of a different domain to create artistic effects \[20\]. Several recent works perform the alignment in an unsupervised way without aligned image pairs \[21\]-\[24\]. Other objectives for distribution matching that have been proposed in the literature, including Maximum Mean Discrepancy \[12\], f-discrepancy \[25\] and others, have also been used for generative modeling \[26\]-\[27\].

Most distribution alignment objectives fall into one of two categories: they are either f-divergences that use a classifier likelihood to measure the distance between distributions, or integral probability metrics (IPMs) that optimize linear combinations of values regressed at sample points. The main difference between these two is in the way they treat correctly classified samples as will be discussed in Section 4.1.
Figure 1: Gradient descent fails to solve the saddle point problem $\min_x \max_y xy$. The plot on the right presents a trajectory of the gradient descent in $(x, y)$ axes over time indicated with color.

In this work we specifically consider the logistic adversarial objective, show that it is useful to optimize its dual, and present a relationship between this adversarial objective and MMD.

3 A Motivating Example

We start with a well-known [28] motivating example of a simple min-max problem to show that, even in this basic case, gradient descent might fail dramatically. Let us consider the simplest min-max problem with a unique solution: finding a saddle point of a hyperbolic surface. Given the function $f(x, y) = xy$, our problem is to solve $\min_x \max_y f(x, y)$, which has a unique solution at $(0, 0)$.

Suppose that we want to apply gradient descent to solve this problem. The intuitive analog of the gradient vector that we might consider using in the gradient descent update rule is defined by the following vector field: $g(x, y) = (x, -y)$. However, at any given point $(x, y)$ the vector $g(x, y)$ will lie in the tangent space of a circular trajectory, thus following this trajectory would never lead to the true solution $(0, 0)$.

One can observe the trajectory produced by the update rule $x_{t+1} = x_t + \alpha g(x, y)$ applied to the problem above in [Fig. 1]. The figure on the left illustrates the saddle point, and the plot on the right shows the $x, y$ parameter as a function of iteration $t$, indicating the number of iterations with changing colors. Neither block coordinate descent nor various learning rate schedules can change the performance of the gradient descent on this problem significantly.

4 Approach

We first propose a new formulation of the adversarial objective for distribution alignment problems. Then we apply this approach to the domain adaptation scenario in Section 5.

4.1 Formulation of the Dual Adversarial Objective

Suppose that we are given a finite set of points $A$ sampled from the distribution $p$, and a finite set of points $B$ sampled from the distribution $q$, and our goal is to match $q$ with $p$ by aligning $B$ with $A$.

More specifically, we aim to learn a matching function $M_\theta(B)$ that maps $B$ to be as close as possible to $A$ by minimizing some empirical estimate of a statistical distance $d(\cdot, \cdot)$ between them where $\theta$ are parameters of the matching function,

$$\theta^* = \arg\min_\theta d(A, M_\theta(B))$$

Let us denote $B'_\theta = M_\theta(B)$. The adversarial approach obtains the distance function by finding the best classifier $D_\theta(x)$ with parameters $\theta$ that discriminates points $x \in A$ from points $x \in B'$ and considers the distance between $A$ and $B'$ to be equal to the likelihood of this classifier. A higher likelihood of separating $A$ from $B'$ means that $A$ is far from $B'$. This can be any form of hypothesis in general, and is often chosen to be a linear classifier [15] or a multi-layer neural network [1]. In this work, we use the class of linear classifiers, specifically, logistic regression in its primal and dual formulations. The solution can also be kernelized to obtain nonlinear discriminators.
We will define the distance between distributions to be minimized as the maximum likelihood of the logistic distribution classifier parametrized by \( w \): 
\[
d(A, B') = \max_w \sum_{x_i \in A} \log(\sigma(w^T x_i)) + \sum_{x_j \in B'} \log(1 - \sigma(w^T x_j)) - \frac{\lambda}{2} w^T w
\]

We can equivalently re-write this expression as:
\[
C_\theta = \{(x_i, y_i) : x_i \in A \cup B'_\theta, \ y_i = 1 \text{ if } x_i \in A \text{ else } -1 \}
\min_\theta d(A, B'_\theta) = \min_\theta \max_w \sum_{x_i, y_i \in C_\theta} \log(\sigma(y_i w^T x_i)) - \frac{\lambda}{2} w^T w
\]

The duality derivation \cite{29} is based on the fact that the sigmoid is tightly upper-bounded by the following expression
\[
\log(\sigma(u)) \leq \alpha^T u + H(\alpha)
\]
\[
H(\alpha) = \alpha^T \log \alpha + (1 - \alpha)^T \log(1 - \alpha)
\]

thus we can upper-bound the distance as
\[
d(A, B'_\theta) \leq \min_\alpha \max_w \sum_{x_i, y_i \in C_\theta} \alpha^T (y_i w^T x_i) + H(\alpha) - \frac{\lambda}{2} w^T w
\]

where the dual variable \( \alpha_i \) corresponds to the weight of the point. Higher weight means that the point is contributing more to the decision hyperplane.

The optimal \( w \) that maximizes the inner expression can be computed in a closed form, 
\[
w^* = \frac{1}{\lambda} \alpha^T (\sum x_i y_i).
\]

By substituting this into the expression above we obtain a min problem
\[
d(A, B'_\theta) \leq g(\alpha, A, B'_\theta) := \min_{0 \leq \alpha \leq 1} \frac{1}{2\lambda} \sum_{ij} \alpha_i^T \alpha_j (y_i x_i)^T (y_j x_j) + H(\alpha)
\]
\[
= \min_{0 \leq \alpha \leq 1} \frac{1}{2} \alpha^T Q_A \alpha_A + \frac{1}{2} \alpha^T Q_B \alpha_B - \alpha^T A_{AB} \alpha_B + H(\alpha_A) + H(\alpha_B)
\]
\[
s.t. \sum \alpha_i^A = \sum \alpha_j^B
\]

The Eq. (3) is obtained by splitting the summation into terms that include samples only from \( A \), only from \( B \), and from both. \( Q_{AB} \) is the kernel matrix between points in \( A \) and points in \( B \), and is equal to the dot product between corresponding data points in the linear case.

The above expression gives us an upper bound on the likelihood of the discriminator. Thus, by minimizing this upper bound, we can minimize the likelihood itself, as in the original loss, and therefore minimize the distance between the distributions:
\[
\theta^*, \alpha^* = \arg\min_{\theta, \alpha} g(\alpha, A, M_\theta(B))
\]

Note that the overall problem has changed from a saddle point problem to a smooth minimization problem, which ultimately converges when gradient descent has a properly chosen learning rate, whereas the descent iterations for the saddle point problem might not converge at all. The resulting smooth optimization problem consists of minimization over \( \alpha \) to improve classification scores and over \( \theta \) to move points towards the decision boundary.

4.2 Relationship to MMD

Maximum Mean Discrepancy is defined \cite{12} as
\[
d(p, q) = \sup_{f \in \mathcal{H}} \mathbb{E}_p f(x) - \mathbb{E}_q f(x)
\]
and was shown to have a closed form solution and a corresponding closed form empirical estimator if $\mathcal{H}$ is a unit ball in RKHS with the reproducing kernel $k(x, y)$:

$$d(p, q) = \frac{1}{2} E_{p \times p} k(x, x') + \frac{1}{2} E_{q \times q} k(y, y') - E_{p \times q} k(x, y)$$

$$d(A, B) = \frac{1}{2 |A|} \sum_{i,j \in A} k(x, x') + \frac{1}{2 |B|} \sum_{i,j \in B} k(y, y') - \frac{1}{|A||B|} \sum_{A \times B} k(x, y).$$

Note that if sample weights in Eq. (3) are constant and equal across all samples, so $\alpha_i = c$, then the distance becomes exactly an empirical estimate of the MMD plus the constant from the entropic regularizer. Thus, the adversarial logistic distance introduced in Eq. (2) is a weighted empirical estimator of the MMD distance. Intuitively, what this means is that the optimization procedure consists of two minimization steps: (1) find the best sample weights assignment by changing $\alpha$ so that the regularized weighted MMD is minimized, and then (2) use a fixed $\alpha$ to minimize the resulting weighted MMD distance by moving points with higher weights in $B'$. This makes the resulting procedure similar to Iteratively Reweighted Least Squares [30] for logistic regression. An interesting observation here is that it turns out that high weights in this iterative procedure are given to the most mutually close subsets of $A$ and $B'$, where closeness is measured in terms of Maximum Mean Discrepancy. Moreover, chosen points happen to be exactly support vectors of the corresponding classifier.

4.3 Computational Complexity

One potential drawback of the proposed method is that the computational complexity of a single gradient step grows quadratically with the size of the dataset. However, there exist methods [31] for mini-batch training of classifiers formulated in dual form that can considerably decrease memory and computational requirements for large scale inference.

5 Application to Distribution Alignment in Domain Adaptation

We now show how the above formulation can be applied to distribution alignment for the specific problem of unsupervised domain adaptation. In this scenario, we assume that there exist two distinct distributions on $\mathcal{X} \times \mathcal{Y}$: source distributions $P_S(X, Y)$ and target distribution $P_T(X, Y)$. We denote parts of the original space $\mathcal{X}$ that include support of source and target distributions as $\mathcal{X}_S \supset \text{supp}(P_S(X))$ and $\mathcal{X}_T \supset \text{supp}(P_T(X))$ respectively. We assume that we observe a finite number of training pairs from the source distribution $D_S \subset [\mathcal{X}_S \times \mathcal{Y}]^n$ and unlabeled samples from target distribution $D_T \subset [\mathcal{X}_T]^m$. Our goal then is to find a mapping $h \in \mathcal{H} \subset \mathcal{X}_T \rightarrow \mathcal{Y}$ that minimizes target risk, even though we only have labels for samples from source:

$$\hat{h} = \arg\min_{h} \mathcal{R}_T(h) = \arg\min_{h} \mathbb{E}_{P_T} L(h(x), y).$$

Ben-David et al. [32] showed that, under mild restrictions on probability distributions, the target risk is upper-bounded by the sum of the source risk, the complexity term involving VC-dimensionality of $\mathcal{H}$ and the size of the dataset, and the discrepancy between source and target distributions. Thus, in order to make the target risk closer to the source risk, we need to minimize the discrepancy between these distributions.

Our approach can be applied directly to this scenario if the discrepancy is replaced with an adversarial objective that uses a logistic regression domain classifier. In Section 6.2 we consider an instance of this problem where the main task is classification and the hypothesis space corresponds to multi-layer neural networks. We compare the standard min-max formulation of the adversarial objective in Eq. (1) with our min-min formulation in Eq. (3), and report the accuracy of the resulting classifier on the target domain.

6 Experiments

6.1 Synthetic Distribution Matching

We first tested the performance of our proposed approach on a synthetic point cloud matching problem. The data consists of two clouds of points in a two-dimensional plane, where the goal is to match
points from one cloud to points from the other cloud. There are no restrictions on the transformation between the point clouds, so $M_\theta$ included all possible transformations, and is therefore parameterized by the point coordinates themselves, so the coordinates themselves were updated on each gradient step. We minimized the logistic adversarial distance in primal space by solving the corresponding min-max problem in Eq. (1) and maximized the negative adversarial distance given by the dual of the logistic classifier in Eq. (3) and the corresponding kernelized logistic classifier. More specifically, Gaussian kernel was used everywhere further in the text.

As expected, the optimization of distances given by the dual versions of domain classifiers (linear and nonlinear) worked considerably better than the same distance given by a linear classifier in the primal form. More specifically, the results in the primal case were very sensitive to the choice of learning rate and in general the resulting saddle point problem did not converge to a single solution, whereas both dual versions successfully converged to solutions that matched the two clouds of points both visually and in terms of means and covariances, as presented in Fig. 2.

We suggest one more intuitive explanation of why the dual procedure might work better, in addition to the fact that optimization problems are just inherently easier than saddle point problems. The decision boundary of the classifier in the dual space is defined implicitly as a weighted average of observed data points, so that when these data points move, the decision boundary moves with them. If points move too rapidly and the discriminator explicitly parametrizes the decision boundary, weights of the discriminator may change drastically to keep up with moved points, leading to the overall instability of the training procedure. In support of this hypothesis, we observed interesting patterns in the behavior of the linear primal discriminator: when point clouds become sufficiently aligned, the decision boundary starts "spinning" around these clouds, slightly pushing them in corresponding directions. In contrast, both dual classifiers end up gradually converging to solutions that assigned

Figure 2: (Top row) The procedure of min-max adversarial alignment of points from $A$ and $B$ lying in a two-dimensional plane does not converge to a single solution and the objective (first column) oscillates over time. (Second row) In contrast, our proposed minimization-based adversarial objective steadily converges to an optimum. (Bottom row) We observe similar behavior for the kernelized version. Even though the accuracy of the learned discriminators (column two) drifts over time in all cases, the distance between the covariance matrices (column three) and means (not shown in the figure) decreased in all cases; however, the dual cases (rows two and three) converged to a single stable solution that indeed corresponds to the desired point cloud alignment. See Figure 3 for an illustration.
Figure 3: When trained on a point cloud matching task, the primal approach (first row) leads to an unstable solution that makes the decision boundary spin around data points when they are almost aligned, whereas both the linear (second row) and kernel (third row) dual approaches lead to stable solutions that gradually assign 0.5 probability of belonging to either $A$ or $B$ to all points, which is exactly the desired behaviour. Yellow and blue points are the original point clouds, red indicates movement of yellow points. See the supplemental material for an animation.

6.2 Real-Image Unsupervised Domain Adaptation

We also evaluated the performance of proposed model on a visual domain adaptation task. We attempted to adapt a two-layer multi-layered perceptron (MLP) with two hundred hidden units trained to classify a subset of digits from MNIST (source domain) into "fours" and "nines" to perform well in the same task on USPS dataset (target domain). All images are scaled to 16x16 pixel.

Let us take a closer look at a two-layered MLP domain adaptation architecture we are using below. We will denote the first layer of the MLP as the embedding part and the latter one as the classification part.

Figure 4: Samples from MNIST (first row) and USPS (second row) showing diversity and the domain shift between datasets. Training and testing a two-layer fully-connected network on images from the same domain yields an accuracy above 0.95, but the accuracy drops to 0.5-0.6 when the network is tested on samples from the other domain. This is known as the domain shift problem.
Figure 5: Results for the unsupervised domain adaptation of a binary classifier trained to classify digits into nines and fours from MNIST (source domain) and tested on USPS (target domain). Both the negative likelihood objective (first column) and the accuracy of the resulting classifier on target (second column) are not monotonic in the primal case (top row), but show stable improvement over time for the dual case (bottom row). Note that while the discriminator accuracy changes drastically over time in the dual case, the accuracy on target continually improves. The adaptation method based on the dual formulation steadily reaches and retains an accuracy around 0.76, whereas the primal method reached similar accuracy at some point, but also diverged from it over time. These plots were obtained using a basic learning rate scheduling heuristic: divide the learning rate by two if it clearly overshoots on the last step. Without it, results are little worse, but the general trends are still the same.

We train different embedding parts - one for each domain, and use a shared classification part. Our architecture also includes a linear discriminator (adversary) that is trained to distinguish features computed by the corresponding embedding parts for different domains.

After training the source embedding and shared classification part of the network on the source domain, we initialized the target embedding with source embedding weights and optimized the target embedding weights to make the distributions of source and target feature representations less distinguishable. This approach to domain adaptation can be treated as the most basic version of the ADDA architecture [10]. After learning and fixing the source embedding map, we observed that both the negative likelihood objective and the quality of the resulting classifier on the target gradually improved when we were updating target embeddings with the procedure given by the dual linear classifier, but changed quite unexpectedly for a primal linear classifier. Detailed results are shown in Fig. 5.

These results are encouraging because they show that the training procedure of adversarial unsupervised adaptation could be made more stable. The stability of the unsupervised adaptation methods is very important because in real-world scenarios we do not have an access to the labeled target set to estimate how the performance of the method changes over time and whether it is converging. Therefore, we are not able to find peak performance on target similar to the one presented in the second column of the first row in Fig. 5. On the other hand, our proposed dual methods can be trained until convergence without the risk of stopping at a sub-optimal solution.

7 Conclusion

We presented a dual formulation of an adversarial alignment objective in both linear and kernel cases. Our preliminary empirical results indicate that the proposed dual optimization objective is indeed better suited for learning with gradient descent than the saddle point objective that naturally arises from the original primal formulation of adversarial alignment. This suggests that further attempts to reformulate other notions of statistical distances in adversarial settings as computationally feasible minimization problems may be promising.
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 27. 2014.

[2] Emily L Denton, Soumith Chintala, Rob Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In Neural Information Processing Systems (NIPS), pages 1486–1494, 2015.

[3] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434, 2015.

[4] Junbo Zhao, Michael Mathieu, and Yann LeCun. Energy-based generative adversarial network. arXiv preprint arXiv:1609.03126, 2016.

[5] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew P. Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. CoRR, abs/1609.04802, 2016.

[6] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. arXiv preprint arXiv:1611.07004, 2016.

[7] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In David Blei and Francis Bach, editors, Proceedings of the 32nd International Conference on Machine Learning (ICML-15), pages 1180–1189. JMLR Workshop and Conference Proceedings, 2015.

[8] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In International Conference in Computer Vision (ICCV), 2015.

[9] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In European Conference on Computer Vision (ECCV), pages 597–613. Springer, 2016.

[10] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In Computer Vision and Pattern Recognition (CVPR), 2017.

[11] Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. In NIPS 2016 Workshop on Adversarial Training. In review for ICLR, volume 2016, 2017.

[12] Arthur Gretton. A Kernel Two-Sample Test. Journal of Machine Learning Research, 13:723–773, 2012.

[13] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. CoRR, abs/1701.07875, 2017.

[14] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. CoRR, abs/1411.1784, 2014.

[15] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. CoRR, abs/1412.3474, 2014.

[16] Mingsheng Long and Jianmin Wang. Learning transferable features with deep adaptation networks. In International Conference on Machine Learning (ICML), 2015.

[17] Baochen Sun and Kate Saenko. Deep CORAL: Correlation alignment for deep domain adaptation. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 9915 LNCS:443–450, 2016.

[18] Ming-Yu Liu and Oncel Tuzel. Coupled Generative Adversarial Networks. Nips, (Nips), 2016.

[19] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised Pixel-Level Domain Adaptation with Generative Adversarial Networks. 2016.
[20] Michael Mathieul, Junbo Zhao, Pablo Sprechmann, Aditya Ramesh, and Yann Lecun. Disentangling factors of variation in deep representations using adversarial training. (Nips), 2016.

[21] Hao Dong, Paarth Neekhara, Chao Wu, and Yike Guo. Unsupervised Image-to-Image Translation with Generative Adversarial Networks. 2, 2017.

[22] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. mar 2017.

[23] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jungkwon Lee, and Jiwon Kim. Learning to Discover Cross-Domain Relations with Generative Adversarial Networks. 2017.

[24] C Onsiderations, D Iagnostic, C Hallenges, P Ostbiopsy, and C Linical M Anagement. Unsuper vised Cross-Domain Image Generation. pages 1–14, 2017.

[25] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization. arXiv, page 17, 2016.

[26] Gintare Karolina Dziugaite, Daniel M Roy, and Zoubin Ghahramani. Training generative neural networks via Maximum Mean Discrepancy optimization. Uai, 2015.

[27] Yujia Li, Kevin Swersky, and Rich Zemel. Generative Moment Matching Networks. Proceedings of The 32nd International Conference on Machine Learning, 37:1718–1727, 2015.

[28] Ian Goodfellow. Nips 2016 tutorial: Generative adversarial networks. arXiv preprint arXiv:1701.00160, 2016.

[29] Thomas P Minka. A comparison of numerical optimizers for logistic regression.

[30] Peter J Green. Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives. Journal of the Royal Statistical Society. Series B (Methodological), pages 149–192, 1984.

[31] Martin Takác, Avleen Singh Bijral, Peter Richtárik, and Nati Srebro. Mini-batch primal and dual methods for svms.

[32] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. Advances in Neural Information Processing Systems, 19:137–144, 2007.