Learn to Expect the Unexpected: Probably Approximately Correct Domain Generalization

Vikas K. Garg
MIT
vgarg@csail.mit.edu

Adam Tauman Kalai
Microsoft Research
noreply@microsoft.com

Katrina Ligett
Hebrew University
katrina@cs.huji.ac.il

Zhiwei Steven Wu
Carnegie Mellon University
zstevenwu@cmu.edu

Abstract

Domain generalization is the problem of machine learning when the training data and the test data come from different "domains" (data distributions). We propose an elementary theoretical model of the domain generalization problem, introducing the concept of a meta-distribution over domains. In our model, the training data available to a learning algorithm consist of multiple datasets, each from a single domain, drawn in turn from the meta-distribution. We show that our model can capture a rich range of learning phenomena specific to domain generalization for three different settings: learning with Massart noise, learning decision trees, and feature selection. We demonstrate approaches that leverage domain generalization to reduce computational or data requirements in each of these settings. Experiments demonstrate that our feature selection algorithm indeed ignores spurious correlations and improves generalization.

1 Introduction

Machine learning algorithms often perform poorly due to the fact that training data is distributed differently from the data on which the algorithm will eventually be used. This often happens when training data is collected from some domains (e.g., certain websites, breeds of dogs, or countries) but good performance is desired on a broader distribution over domains (e.g., the entire web, all dogs, or the whole world). The challenge is to generalize from data collected in these training domains to the full distribution. For some problems, it is of course impossible to generalize from data collected in some domains to others, even when the training data is annotated by domain. In other problems, however, it is not only possible but can in fact be easier to learn with domain-split data than if data were iid from the entire distribution of interest. We propose a novel theoretical framework to study this setting and give algorithms which illustrate such learnability in the framework.

As a concrete example, consider classifying personal web pages at a university as either student or faculty. Think of each university, such as the Toyota Technological Institute, as a domain, often corresponding to a URL domain name such as `ttic.edu`. Data may be scraped from a handful of domains with the goal to generalize to future domains. Moreover, predictors that work on one domain may not generalize to others. For example, the presence of the single word *mission* on a web page is an excellent predictor of faculty web pages because official `ttic.edu` faculty pages all contain a "mission statement," while student pages do not. Given data from multiple domains, one can use the domain splits of the training data to *learn classifiers that generalize across domains* and hence are likely to work on future data. This natural example is motivated by the classic WebKB project, where Craveny et al. (1998) collected and hand-labeled training data from four universities with test data from 100 universities, and in fact we test one of our algorithms on this data.

We suppose there is a distribution of interest ρ over examples (x, y, z) where $z \in \mathcal{Z}$ represents the latent domain of example $x \in \mathcal{X}$ with label $y \in \mathcal{Y}$. In standard iid learning, e.g., PAC learning ρ, training examples are iid from ρ and the z's are unobserved. We consider a setting where training data is divided into d domains, where data from each domain share a common z, and these domains are chosen iid from ρ's marginal distribution on z. Learning is with respect to a family of classifiers $\mathcal{C}$ and an assumption set $\rho \in \mathcal{P}$ on the relationship between the domains—with no such assumption, the problem reduces to agnostic learning.

Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130. Copyright 2021 by the author(s).

Interestingly, for some problems in this framework, having training datasets split by domain can actually make learning easier. We give algorithms for three problems that illustrate three different ways in which this can happen. We first consider a multi-domain variant of the Massart noise model (Massart et al., 2006), where there is a common target concept $c \in \mathcal{C}$ but each domain has a different noise rate in the labels. We provide a general reduction from computationally efficient learning in this model to PAC learning under random classification noise (Angluin and Laird, 1987). This results in Bayes-optimal learning; it is not known how to achieve this for iid learning even for simple classes such as halfspaces with Massart noise (though progress was made in this direction by the recent celebrated result of Diakonikolas et al. (2019)).

Second, we provide a multi-domain feature selection algorithm that identifies features that are robust across multiple domains. Our algorithm augments a black-box PAC learner with an additional correlation-based selection based on data across different domains. We empirically demonstrate its effectiveness on the aforementioned WebKB dataset of university webpages. We show that our approach provides stronger cross-domain generalization than the standard baseline. As hypothesized, we find that features that are highly predictive in one university but not in another are in fact *spurious correlations*; removing them improves prediction on data from further universities not in the training set.

Finally, we turn to another notoriously difficult computational problem—PAC learning decision trees. We make the assumption that there is a target decision tree that labels the examples across all domains, but examples in each domain all belong to a single leaf in this tree. Under this assumption, we provide an efficient algorithm with runtime $O(n+s)$, where n denotes the dimension of the data and s denotes the number of nodes in the target tree. (Without any assumption, the fastest known algorithm runs in time $n^{O(\log s)}$).

The three algorithms we introduce within this model are quite different in nature and serve to illustrate the rich range of algorithm innovations possible. Our model of domain generalization enables two distinct advantages over the traditional PAC learning model. First, PAC-learned models do not come with any guarantee of performance on data drawn from unobserved domains. Second, the additional structure of training on multiple datasets enables in-sample guarantees that are not achievable in the standard PAC model. We view this work as a theoretically-grounded starting point and hope that it leads to further study.

Finally, we note that even in settings where training data are not explicitly partitioned into domains, one can employ *data partitioning* by which we mean artificially partitioning the training data along a feature of interest or by clustering (e.g., one could partition images collected within a city based on time of day, season, weather, or geographic subdivision of the city.) Algorithms may use such partitioned data to learn to generalize to novel types of data. A domain expert would choose such a partition analogously to *data augmentation* where a domain expert anticipates certain transformations that could be applied to data.

2 Related Work

A rich literature sometimes known as domain adaptation (e.g., Blitzer et al. (2006); Ben-David et al. (2007); Blitzer et al. (2008); Mansour et al. (2009a,b); Ben-David et al. (2010); Ganin and Lempitsky (2015); Tzeng et al. (2017); Morerio et al. (2018); Volpi et al. (2018a)) considers settings where the learner has access not only to labeled training data, but also to unlabeled data from the test domain. This is a quite different setting from ours; our learner is given no access to data from the test domain, either labeled or unlabeled.

There is also a rich literature (e.g., Li and Zong (2008); Luo et al. (2008); Crammer et al. (2008); Mansour et al. (2009c); Guo et al. (2018)) that does not always rely on unlabeled data from the test distribution, but rather leverages information about similarity between domains to produce labels for new points. Zhang et al. (2012), relatedly, study the distance between domains in order to draw conclusions about generalization.

Adversarial approaches have recently gained attention (e.g. Zhao et al., 2018), and in particular, Volpi et al. (2018b), like us, generalize to unseen domains, but they attack the problem of domain generalization by augmenting the training data with fictitious, “hard” points. There are also many other empirical approaches to the problem of domain generalization (e.g., Muandet et al., 2013; Khosla et al., 2012; Ghifary et al., 2015; Li et al., 2017; Finn et al., 2017; Li et al., 2018; Mancini et al., 2018; Balaji et al., 2018; Wang et al., 2019; Carlucci et al., 2019; Dou et al., 2019; Li et al., 2019).

There are of course many other related fields of study, including covariate shift (wherein the source and target data generally have different distributions of unlabeled points but the same labeling rule), concept drift and model decay (wherein the distribution over unlabeled points generally remains static, but the labeling rule drifts over time), robust optimization (which aims for worst-case rather than PAC-style guarantees), and multi-task learning (wherein the goal is generally to leverage access to multiple domains to improve performance on each of them, rather than generalizing to new domains).

3 Definitions

For mathematical notation, we let $[n]$ denote $\{1, 2, \ldots, n\}$ and 1_Q denote the indicator function that is 1 if predicate Q holds and 0 otherwise. For vector $x \in \mathbb{R}^n$, let $x[k]$ denote the kth coordinate of x. Finally, let $\Delta(S)$ denote the set of probability distributions over set S. We now define our model of learning from independent datasets.

3.1 Generalizing from multiple domains

We study classification with multiple datasets from independent domains where training data $T = \langle T^1, \ldots, T^d \rangle \sim \rho_m^{\times d}$ consists of *datasets* $T^i = \langle (x_1^i, y_1^i), \ldots, (x_m^i, y_m^i) \rangle$ each of m examples. These d datasets are chosen iid from *dataset distribution* ρ_m over $(\mathcal{X} \times \mathcal{Y})^m$.

In particular, there is a distribution $\rho \in \Delta(\mathcal{X} \times \mathcal{Y} \times \mathcal{Z})$ where $\mathcal{X}$ is a set of examples, $\mathcal{Y}$ is a set of labels, and $\mathcal{Z}$ is a set of *domains*. Based on this ρ_m selects m labeled examples from a common latent domain as follows: (x_1, y_1, z_1) is picked from ρ, and (x_j, y_j) is picked from ρ conditional on its domain being $z_j = z_1$ for $j \geq 2$. It is not difficult to see that this model is equivalent to a meta-distribution over domains z paired with domain-specific distributions over labeled examples, where the domain-specific distributions would simply be the distribution ρ conditioned on the given domain z. For simplicity, in this paper we focus on classification with equal-sized datasets and latent domains but the model can be generalized to other models of learning, unequal dataset sizes, and observed domains.

A *domain-generalization learner* L takes training data T divided into multiple datasets of examples as input and outputs classifier $L_T : \mathcal{X} \to \mathcal{Y}$. L is said to be computationally efficient if it runs in time polynomial in its input length.

The *error* of classifier $c : \mathcal{X} \to \mathcal{Y}$ is denoted by $\text{err}_\rho(c) = \Pr_{x,y,z\sim\rho}[c(x) \neq y]$ and ρ may be omitted when clear from context. This can be thought of in two ways: $\text{err}_\rho(c)$ is the expected error on d' test datasets of m' examples or it is also the average performance across domains, i.e., error rate on a random example (from a random domain) from ρ.

We first define a model of sample-efficient learning, for large d, with respect to a family $\mathcal{C}$ of classifiers. Following the agnostic-learning definition of Kearns et al. (1992), we also consider an *assumption* $\rho \in \mathcal{P}$ where $\mathcal{P}$ is a set of distributions over $\mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$.

Definition 1 (Efficient Domain Generalization). *A learner L is an efficient domain-generalization learner for classifiers $\mathcal{C}$ over assumption $\mathcal{P}$ if there exist polynomials q_d and q_m such that, for all $\rho \in \mathcal{P}$, all $\epsilon, \delta > 0$, and all $d \geq q_d(1/\delta, 1/\epsilon), m \geq q_m(1/\delta, 1/\epsilon)$,*

$$\Pr_{T\sim\rho_m^{\times d}}[\text{err}_\rho(L_T) \leq \min_{c\in\mathcal{C}} \text{err}_\rho(c) + \epsilon] \geq 1 - \delta.$$

Standard models of learning can be fit into this model using iid and noiseless assumptions:

$$\begin{aligned}\mathcal{P}_{iid} = \{\rho \in \Delta(\mathcal{X} \times \mathcal{Y} \times \mathcal{Z}) \mid & \; z \text{ is independent of} \\ & (x, y) \text{ for } x, y, z \sim \rho\} \\ \mathcal{P}_{shh}(\mathcal{C}) = \{\rho \in \Delta(\mathcal{X} \times \mathcal{Y} \times \mathcal{Z}) \mid & \; \min_{c\in\mathcal{C}} \text{err}_\rho(c) = 0\}\end{aligned}$$

In particular, *agnostic learning* can be defined as efficient domain-generalization learning subject to $\mathcal{P}_{iid}$ while *PAC learning* (Valiant, 1984) can be defined as efficient domain-generalization learning with $\mathcal{P}_{PAC} = \mathcal{P}_{iid} \cap \mathcal{P}_{shh}(\mathcal{C})$.

It is not difficult to see that Definition 1 is not substantially different from PAC and agnostic learning, with a large number of datasets:

Observation 2. *If $\mathcal{C}$ is PAC learnable, then $\mathcal{C}$ is efficiently domain-generalization learnable with noiseless assumption $\mathcal{P}_{shh}(\mathcal{C})$. If $\mathcal{C}$ is agnostically learnable, then $\mathcal{C}$ is efficiently domain-generalization learnable without assumption, i.e., $\mathcal{P} = \Delta(\mathcal{X} \times \mathcal{Y} \times \mathcal{Z})$.*

Proof. Simply take a PAC (or agnostic) learning algorithm for $\mathcal{C}$ and run it on the first example in each dataset. Since these first examples are in fact iid from ρ, the guarantees of PAC (or agnostic) learning apply to the error for future examples drawn from ρ. □

This is somewhat dissatisfying, as one might hope that error rates would decrease as the number of data points per domain increases. This motivates the following definition, which considers the rate at which the error decreases separately in terms of the number of datasets d and the number of examples per dataset m.

Definition 3 (Dataset-efficient learning). *A learner L is a* dataset-efficient learner *for classifiers $\mathcal{C}$ over assumption $\mathcal{P}$ if there exists polynomials q_d and q_m such that, for all $\rho \in \mathcal{P}$, all $\epsilon, \delta > 0$, and all $d \geq q_d(1/\delta), m \geq q_m(1/\delta, 1/\epsilon)$,*

$$\Pr_{T\sim\rho_m^{\times d}}[\text{err}_\rho(L_T) \leq \min_{c\in\mathcal{C}} \text{err}_\rho(c) + \epsilon] \geq 1 - \delta.$$

This definition requires fewer datasets than the previous definition, requiring a number of datasets that depends only on $1/\delta$ regardless of ϵ.

In PAC and agnostic learning, many problems have a natural *complexity parameter* n where $\mathcal{X} = \bigcup_{n\geq1} \mathcal{X}_n$, $\mathcal{C} = \bigcup_{n\geq1} \mathcal{C}_n$, $\mathcal{Z} = \bigcup_{n\geq1} \mathcal{Z}_n$, $\mathcal{P} = \bigcup_{n\geq1} \mathcal{P}_n$, such as $\mathcal{X}_n = \mathbb{R}^n$. In those cases, we allow the number of examples and datasets, q_d, q_m in Definitions 1 and 3 to also grow polynomially with n. Also note that the set $\mathcal{P}$ can capture a host of other assumptions, such as a margin between positive and negative examples. Finally, while we assume that the chosen domains z^i are not given to the learner—this is without loss of generality as the domains could be redundantly encoded in the examples x.

4 Multi-Domain Massart Noise Model

In the traditional Massart noise model (Massart et al., 2006), each individual example x has its own label noise rate that is, $\Pr[c(x) \neq y] = \eta(x) \leq \eta_b$, for some given upper-bound $\eta_b < 1/2$. Learning under this model is computationally challenging and no efficient Bayes-optimal algorithms are known even for simple concept classes (Diakonikolas et al., 2019), despite the fact that the statistical complexity of learning in this model is no worse than learning with a uniform noise rate η_b. We study a multi-domain variant of the Massart model, in which the learner receives examples with noisy labels from multiple domains such that each domain has its own fixed noise rate. We demonstrate that by leveraging the cross-domain structure of the problem we can obtain a broad class of computationally efficient algorithms. In particular, we provide a reduction from efficient learning a multi-domain variant of the Massart noise model to efficient PAC learning under random classification noise (Angluin and Laird, 1987). Practically every concept class known to be efficiently PAC learnable (Valiant, 1984) can also be learned efficiently with classification noise (either directly or through the statistical query framework (Kearns, 1998), with parity-based constructions being the only known exceptions).

We first state our multi-domain Massart model formally as an assumption over the distributions $\Delta(\mathcal{X} \times \mathcal{Y} \times \mathcal{Z})$.

Assumption $\mathcal{P}_{\mathrm{MDM}}$. There exists an unknown classifier $c \in \mathcal{C}$ and an unknown noise rate function $\eta \colon \mathcal{Z} \to \mathbb{R}$ such that the distribution ρ over $\mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$ satisfies $\Pr_\rho[y \neq c(x) \mid z] = \eta(z) \leq \eta_b < 1/2$. We assume quantity η_b is known to the learner. While this may seem "easier" than learning with a fixed noise rate of η_b because there is less noise, this is not the case just as learning with Massart noise seems to be hard than learning with a fixed noise rate η_b. (If the noise rates for different examples or domains were known, it would be easy to artificially add noise and reduce it to the easier problem of learning with a constant η_b noise.) However, with unknown noise rates, patterns in the examples may correlate with differences in rates of noise to make it harder to learn.

Note that the minimal error rate $\Pr_\rho[y \neq c(x)] \in [0, \eta]$, achieved by the "true" classifier c, can be much smaller than η. Our multi-domain variant is a generalization in that the marginal distribution over labeled examples, ignoring domains, fits the Massart noise model. We will leverage the domain structure to provide a reduction from the learning problem in this model to PAC learning under classification noise due to Angluin and Laird (1987), defined below.

Classification Noise (CN) Learnable. Let $\rho_{\mathcal{X}}$ be a distribution over $\mathcal{X}$. For any *noise rate* $0 \leq \eta_0 < 1/2$, the example oracle $\mathrm{EX}^{\eta}_{\mathrm{CN}}(c, \rho_{\mathcal{X}})$ on each call returns an example (x, y) by first drawing an example x from $\rho_{\mathcal{X}}$ and then drawing a random noisy label y such that $\Pr[y \neq c(x)] = \eta_0 < \overline{\eta}$, where $\overline{\eta}$ is an known upper bound. The concept class $\mathcal{C}$ is *CN learnable* if there exists a learner $\mathcal{L}$ and a polynomial f such that for any distribution $\rho_{\mathcal{X}}$ over $\mathcal{X}$, any noise rate $0 \leq \eta_0 < \overline{\eta} < 1/2$, and for any $0 < \epsilon \leq 1$ and $0 < \delta \leq 1$, the following holds: $\mathcal{L}$ will run in time bounded by $f(1/(1-2\overline{\eta}), 1/\epsilon, 1/\delta)$ and output a hypothesis h that with probability at least $1-\delta$ satisfies $\Pr_{x \sim \rho_{\mathcal{X}}}[h(x) \neq c(x)] \leq \epsilon$.

Theorem 4. *Let $\mathcal{C}$ be a concept class that is CN learnable. Then there exists an efficient domain generalization learner for $\mathcal{C}$ under the multi-domain Massart assumption $\mathcal{P}_{\mathrm{MDM}}$.*

The basic idea behind the proof is to "denoise" data from each dataset by training a classifier within each dataset and then using that classifier to label another held-out example from that domain. If that classifier had high accuracy, then with high probability the predicted labels will be correct. A noiseless classification algorithm can then be applied to the denoised data.

Proof. Let $\mathcal{L}$ be a CN learner for $\mathcal{C}$ with runtime polynomial f. To leverage this learner to learn under the multi-dataset Massart model, we will aim to create an example oracle $\mathrm{EX}^{\eta}_{\mathrm{CN}}$. Let $c \in \mathcal{C}$ be the target concept, and let $\epsilon, \delta \in (0,1)$ be the target accuracy parameters. We will first draw a collection of $d = f(1, 1/\epsilon, 2/\delta)$ datasets $T = \langle T^1, \ldots T^d \rangle$ from $\rho_m^{\times d}$, where $m > f(1/(1-2\eta_b), \delta/(4d), \delta/(4d))$. We will run the CN learner with a random subset of T_i of size $(m-1)$ as input and obtain an hypothesis h_i such that with probability $1 - \delta/(4d)$,

$$\Pr_{\rho_i}[h_i(x) \neq c(x)] \leq \delta/(4d), \tag{1}$$

where ρ_i denotes the conditional distribution over $\mathcal{X}$ conditioned on the domain being z_i. By a union bound, we know that except with probability $\delta/4$, equation (1) holds for all datasets i. We will condition on this level of accuracy (event E_1). Let (x^i, y^i) denote an example in

T_i that was not used for learning h_i. This provides another dataset $\hat{T} = \langle (x^1, h_i(x^1)), \ldots, (x^d, h_i(x^d)) \rangle$. Note that the x^i's i.i.d. draws from the $\rho_{\mathcal{X}}$, the marginal distribution of ρ over $\mathcal{X}$. Furthermore, by the accuracy guarantee of each h_i, $\Pr[h_i(x^i) \neq c(x^i)] \leq \delta/(4d)$. By a union bound, we know that except with probability $\delta/4$, $h_i(x^i) = c(x^i)$ for all $i \in [d]$. We will condition on this event of correct labeling (event E_2). This means the examples in $\hat{T}$ can simulate random draws from $\mathrm{EX}^0_{\mathrm{CN}}(c, \rho_{\mathcal{X}})$. Finally, we will run $\mathcal{L}$ over the set $\hat{T}$, and by our choice of d, $\mathcal{L}$ will output a hypothesis h such that $\Pr_\rho[h(x) \neq c(x)] \leq \epsilon$ with probability at least $1 - \delta/2$ (event E_3). Finally, our learning guarantee follows by combining the failure probability of the three events E_1, E_2, E_3 with a union bound. □

Open problem in the (multi-domain) Massart model. An open question in the multi-domain Massart noise model is whether there exists an efficient algorithm that only relies on a constant number of examples from each domain. If we can decrease the number of examples in each domain down to 1, we recover the standard Massart noise model. Thus, we view this as an intermediate step towards an efficient algorithm for the standard Massart model (Diakonikolas et al., 2019).

5 Feature Selection Using Domains

Next, we use access to training data from multiple domains to aid in performing feature selection.

We fix $\mathcal{X} = \{0,1\}^n$. For set $R \subseteq [p]$, let $x[R] = \langle x[k] \rangle_{k \in R} \in \{0,1\}^{|R|}$ denote the selected features R of example $x \in \mathcal{X}$. Let z^i denote the domain corresponding to training dataset T^i, for each $i \in [d]$. Define ρ_k to be the correlation of $x[k]$ and y over ρ and let ρ_k^i denote the usual (Pearson) correlation coefficient of feature $x[k]$ with y conditioned on the example having domain $z = z^i$. Let $\hat{\rho}_k^i$ denote the empirical correlation of $x[k]$ and y on T^i.

The following algorithm (**FUD**) performs feature selection using domains.

1. Input: class $\mathcal{C}$, parameters $\beta, \epsilon \geq 0$, training data T consisting of d splits of m examples each.
2. If the overall fraction of positive or negative examples is less than $\epsilon/2$ (massive class imbalance), stop and output the constant classifier $c(x) = 0$ or $c(x) = 1$, respectively.
3. For each variable $i \in [n]$, compute empirical correlation $\hat{\rho}_k^i$ of $x[k]$ and y over each dataset $i \in [d]$.
4. Let $R = \left\{ k \mid \min_i |\hat{\rho}_k^i| \geq \beta \right\}$.
5. Find any $c \in \mathcal{C}$ such that $c(x[R]) = y$ for all $s, x, y \in T$, and output classifier $f(x) = c(x[R])$. If no such c exists, output *FAIL*.

Assumption $FS(\mathcal{C}, \beta)$ For $\beta > 0$ we define the Feature Selection assumption $FS(\mathcal{C}, \beta)$ to require that there exists a robust set of features $R \subseteq [p]$ such that:

- Noiselessness $\mathcal{P}_{shh}(\mathcal{C})$: For some $c \in \mathcal{C}$, $\Pr_\rho[c(x[R]) = y] = 1$.
- Independence: $x[R]$ and z are independent over ρ.
- Correlation: For all $k \in R$, $|\rho[k]| > 1.1\beta$
- Idiosyncrasy: For all $k \notin R$, $\Pr_{x,y,z \sim \rho}\big[\, |\rho_k^z| < 0.9\beta \big] > 0.1$.

Note that the constants 1.1, 0.9 and 0.1 in the above assumption can be replaced by parameters (e.g., $1 \pm \epsilon_1$ and ϵ_2) and the dependence of d and m on these parameters in the following theorem would be inverse polynomial.

Theorem 5. *For any $\mathcal{C}$ of finite VC dimension $VC(\mathcal{C})$, with $\mathcal{X} = \{0,1\}^n$, $\mathcal{Y} = \{0,1\}$ and any $\beta > 0$, FUD is a dataset-efficient learner under assumption $FS(\mathcal{C}, \beta)$. In particular, for $d = O\left(\log \frac{n}{\delta}\right)$ and $m = O\left(\frac{VC(\mathcal{C})}{\epsilon} + \frac{\log(n/\delta)}{\beta^4 \epsilon^2}\right)$,*

$$\Pr_T[\mathrm{err}_\rho(FUD_T) \leq \epsilon] \geq 1 - \delta,$$

for any $\epsilon, \delta \in (0, 1/2)$.

Proof. Fix $\rho \in FS(\mathcal{C}, \beta)$. Note that by the noiseless and independent assumptions, the fraction of positives is the same in each domain, i.e., $\mathbb{E}[y|z] = \mathbb{E}[y]$. We first bound the failure probability of outputting the all 0 or all 1 classifier in the second step. However, if $\mathbb{E}[y] \geq \epsilon$, the probability that it outputs the all 0 classifier is at most $\delta/10$ by multiplicative Chernoff bounds over $dm = \Omega(\frac{1}{\epsilon} \log \frac{1}{\delta})$ labeled examples. Similarly, if $\mathbb{E}[y] \leq 1 - \epsilon$, the probability we output the all 1 classifier is at most $\delta/10$. Conversely, if $\mathbb{E}[y] < \epsilon/4$, then multiplicative Chernoff bounds also imply that with probability at least $1 - \delta/10$, we will output the 0 classifier (and hence have error $< \epsilon$), and similarly if $\mathbb{E}[y] > 1 - \epsilon/4$.

Henceforth, let us assume $\mathbb{E}[y] \in [\epsilon/4, 1 - \epsilon/4]$.

Next, note that the set R described in the FS assumption is uniquely determined for ρ. Call this set R^*. It suffices to show that with probability at least $1 - \delta/10$, $R = R^*$ for R defined in the algorithm. This is because if $R = R^*$, by a standard VC bound of Haussler et al. (1991), since $x[R]$ is iid and the total number of examples observed is $dm = \Omega\left(\frac{VC(\mathcal{C})}{\epsilon} \log \frac{1}{\delta}\right)$, with probability at least $1 - \delta/2$ the error is at most ϵ because learning of $(x[R], y)$ is standard PAC learning of $\mathcal{C}$.

Using $\mathbb{E}[y] \in [\epsilon/4, 1-\epsilon/4]$, Lemma 6 below implies that $m = \Omega(\beta^{-4}\epsilon^{-2}\log(dn/\delta))$ examples suffice to estimate all dn correlations accurately to within 0.1β with probability at least $1 - \delta/10$. Assuming this happens, all $k \in R^*$ will necessarily also be in R.

It remains to argue that with probability at least $1 - \delta/10$, $R = R^*$. To see this, note that for each $k \notin R^*$, the Idiosyncrasy assumption means that with probability at most $0.9^d \leq \delta/(10n)$ would there be no k for which $|\rho_i^k| \leq 0.9\beta$. Hence, by a union bound, with probability at least $1 - \delta/10$, there will be simultaneously for each $k \notin R^*$ some dataset $i \in [d]$ such that $|\rho_i^k| \leq 0.9\beta$. Since we are assuming that all correlations are estimated correctly to within 0.1β, it is straightforward to see that $R = R^*$. □

We now bound the number of examples needed to estimate correlations.

Lemma 6. *For any jointly distributed binary random variables* $(R, S) \in \{0,1\}^2$ *with* $\mathbb{E}[S] \in [v, 1-v]$*, and for any* $\epsilon, \delta > 0$*, the probability that the empirical correlation coefficient of* $m \geq 2048\epsilon^{-4}v^{-2}\log(8/\delta)$ *iid samples differs by more than* ϵ *from the true correlation is at most* δ.

The proof of this Lemma is given in Appendix A.

6 Feature Selection Experiments

We conducted simple experiments to evaluate the quality of features selected by our methodology from Section 5. We experimented with the WebKB *Universities* data set,[1] (Craveny et al., 1998) a small dataset that is ideally suited for domain generalization. It conatins webpages from computer science departments of various universities, which can be identified by the url domain, e.g., `cornell.edu`. The data set is classified into categories such as faculty, student, course, etc.; we focused on the faculty and student classes for our experiments. Our training data pertains to 711 faculty and student webpages from four universities: Cornell, Texas, Washington, and Wisconsin. Our test data includes faculty and student pages from 100 universities. None of the four universities in our training set were represented in the test set. We represented each page as a bag-of-words, and preprocessed the data to remove all words that had less than 50 occurrences. As a result, we obtained a vocabulary of 547 unique words. Thus, we represented each page as a 547-dimensional binary vector: each word that occurred at least once in the page had the corresponding coordinate set to 1.

We summarize the statistics of our data in Table 1. Note that we computed the bag density of a domain as the average of the mean vector pertaining to the binary vectors in the domain. The respective densities for student and faculty pages are also shown. Note that the faculty proportion in test data (47%) is about twice the proportion in any domain from the training data (where the fraction of faculty pages hovers around 20%). Thus, investigating this data for domain generalization is a worthwhile exercise.

We compare the performance of our algorithm with a standard feature baseline. Specifically, the baseline selects words whose Pearson correlation coefficient with the training labels (i.e., faculty or student) is high. We implemented a regularized version of our feature selection algorithm FUD that penalized those features that have large standard deviation (stdev) of the Pearson coefficient on the train domains. In other words, we computed scores $s_k = |\hat{\rho}_k| - \alpha \;\; \text{stdev}(\hat{\rho}_k^1, \ldots, \hat{\rho}_k^d)$, and selected the features k that were found to have high s_k. We set the value of the regularization parameter α to 2. We call our regularized algorithm FSUS. We trained several classifiers, namely, decision tree, K-nearest neighbor, and logisitic regression, on the features selected by each algorithm (using default values of hyperparameters in the Python *sklearn* library). The performance of the algorithms was measured in terms of the standard *balanced error rate*, i.e., the average of prediction error on each class. Besides the performance on test data, we also show the mean validation error to estimate the generalization performance on domains in the training set. Specifically, we first trained a separate classifier for each domain and measured its prediction error on the data from other domains in the training set, and then averaged these errors to compute the estimate of validation error, denoted by (K=1) in Figure 1. Likewise, for $K = 2$, classifiers were trained on data from two domains at a time, and evaluated for performance on the other domains; similarly for $K \in \{3, 4\}$. As Figure 1 illustrates, our algorithm generally outperformed the baseline method, for different numbers of selected features (horizontal axis) and for different K across classifiers. Note that instead of fixing α beforehand, we could tune it based on the validation error. We found that performance of our algorithm deteriorated only slightly using the tuned α. We omit the details for brevity. These empirical findings substantiate our theoretical foundations, suggesting the benefits of domain generalization.

Figure 2 shows a scatter-plot of the correlations of features, words in this instance, and robustness of this

[1] http://www.cs.cmu.edu/afs/cs/project/theo-20/www/data/

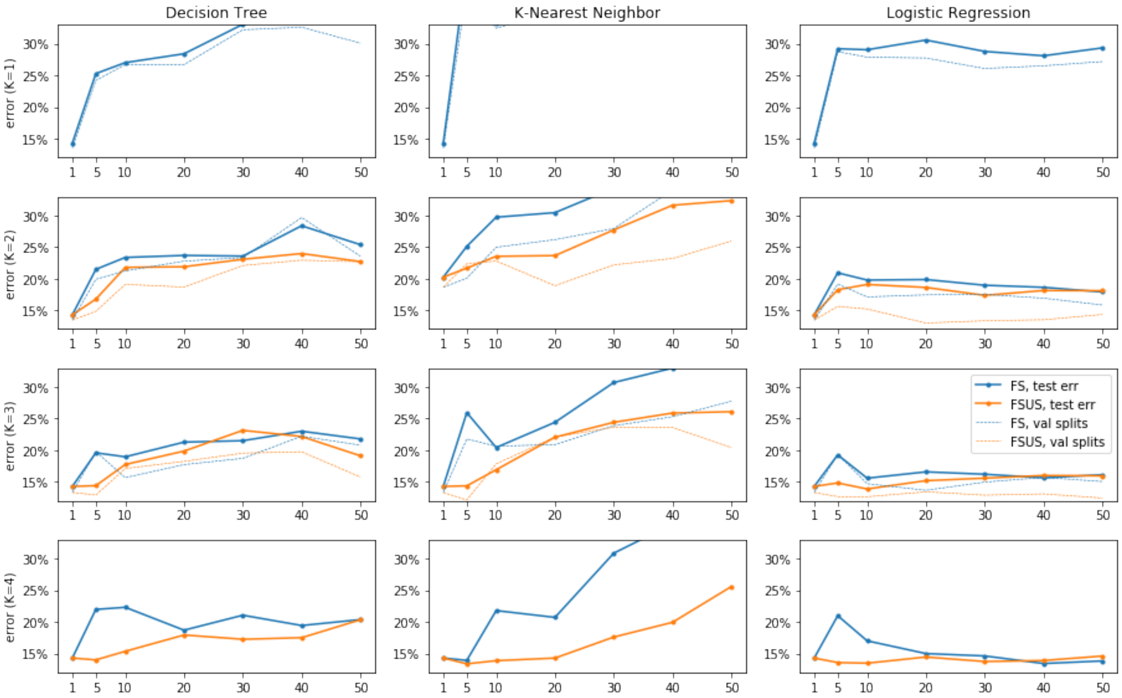

Figure 1: Balanced error rates on University data for varying number of selected features. FSUS is our algorithm, and the baseline is denoted by FS.

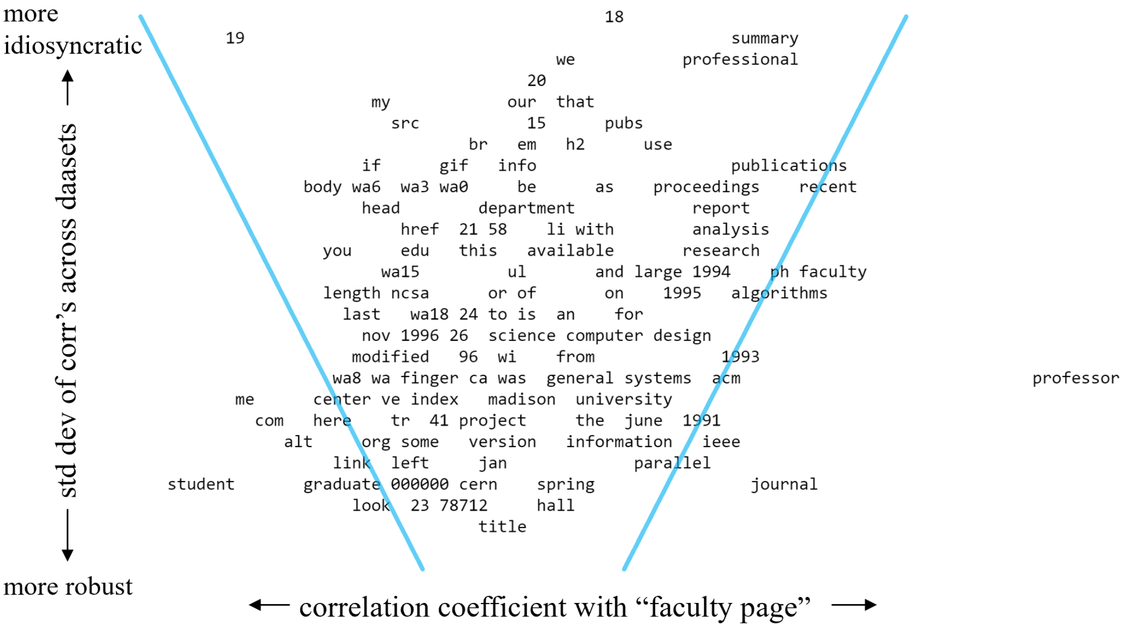

Figure 2: Correlations of words with faculty page (x-axis) vs the std. dev. of correlations over universities. Words to the right such as *professor* correlate most strongly with faculty pages, while words to the left such as *student* correlate most with student pages. Words towards the bottom such as *student* have robust correlations across universities while words towards the top are more idiosyncratic. The words selected are those outside the diagonal lines, where the slope of the line is determined by parameter α, and the horizontal positions of the lines is determined by the number of words to be selected.

Table 1: Data statistics

Domain	Pages	Faculty proportion	Bag density (student pages, faculty pages)
Cornell	162	21%	23% (22%, 28%)
Texas	194	24%	23% (23%, 22%)
Washington	157	20%	24% (24%, 20%)
Wisconsin	198	21%	23% (21%, 29%)
Test	2,054	47%	21% (22%, 21%)

correlation across datasets. Interestingly, one of the most correlated features was the token *19*, which was later discovered to be correlated in certain datasets simply because student webpages at certain universities were downloaded at 7pm, and the datafiles included header information which revealed the download times. It is normally considered the job of a data scientist to decide to ignore features such as data collection time, but this illustrates how our algorithm identified this problem automatically using the idea of robustness across domains.

7 Decision Tree Multi-Domain Model

Finally, we consider learning binary decision trees on $\mathcal{X} = \{0,1\}^n$ in the multi-domain model. Despite years of study, there is no known polynomial-time PAC learner for decision trees, with the fastest known algorithm learning binary decision trees of size $\leq s$ in time $n^{O(\log s)}$ (Hellerstein and Servedio, 2007). Formally, a decision tree is a rooted binary tree where each internal node is annotated with an attribute $1 \leq i \leq n$, and the two child edges are annotated with 0 and 1 corresponding to the restrictions $x[i] = 0$ and $x[i] = 1$. Each leaf is annotated with a label $\{0,1\}$, and on x the classifier computes the function that is the label of the leaf reached by following the path starting at the root of tree and following the corresponding restrictions.

Assumption $\mathcal{P}_{DT}(s,n)$. Let $\mathcal{T}_{s,n}$ be the class of decision trees with at most s leaves. The domains simply correspond to the leaves of the tree in which the (noiseless) example belongs. To make this assumption denoted $\mathcal{P}_{DT}(s,n)$ formal, let the set of domains $\mathcal{Z}$ is simply the set of all 3^n possible *conjunctions* (each $x[j]$ can appear as positive, negative, or not at all) on n variables. We identify each leaf ℓ in a tree with domain $z_\ell \equiv x[j_1] = v_1 \wedge x[j_2] = v_2 \wedge \ldots \wedge x[j_k] = v_k$, where k is the depth of the leaf, $j_1, j_2, \ldots, j_k \leq n$ are the annotations of the internal nodes on the path, and $v_k \in \{0,1\}$ correspond to the edges on the path to that leaf. Using this notation, the assumption $\mathcal{P}_{DT}(s,n)$ is that there is a tree $T \in \mathcal{T}_{s,n}$ for which, with probability 1 over ρ, every example (x,y,z) satisfies $z = z_\ell$ for the leaf ℓ in T which x belongs to, i.e., conjunction z_ℓ holds, and $y = T(x)$, i.e., noiselessness $\mathcal{P}_{DT}(s,n) \subset \mathcal{P}_{shh}(\mathcal{T}_{s,n})$.

Recall that the chosen domains z^i themselves are not observed, otherwise learning would be trivial. Instead, we think of the decision tree simply as the union (OR) of the conjunctions corresponding to leaves labeled positively. It is known to be easy to PAC-learn conjunctions *from positive examples alone* by outputting the *largest consistent conjunction* (Kearns et al., 1994, Section 1.3): the hypothesis given by the conjunction of the subset of possible terms $\{x[j] = b \mid j \in [n], b \in \{0,1\}\}$ that are consistent with every positively labeled example.[2] It is largest in terms of the number of terms, but it is minimal in terms of the positive predictions it makes, and it never has any false positives. The following algorithm learns decision trees in the above multi-domain decision tree model.

1. Input: training data $T^1, T^2, \ldots, T^d$.

2. Let $\textsc{PosDomains} = \{i \mid y_1^i = 1\}$.

3. For each $i \in \textsc{PosDomains}$, find the largest consistent conjunction c_i for T^i.

4. Output the classifier

$$\hat{c}(x) = \begin{cases} 1 & \text{if } c_i(x) = 1 \text{ for any } i \in \textsc{PosDomains} \\ 0 & \text{otherwise.} \end{cases}$$

Theorem 7. *Let $s, n \geq 1$ and $\mathcal{T}_s$ be the family of binary decision trees of size at most s on $\{0,1\}^n$. Then the above algorithm is an efficient domain-generalization learner for $\mathcal{P}_{DT}(s,n)$ for complexity parameter $N = n + s$.*

For decision trees, the complexity of the class depends on both the number of variables and the size of the tree, hence we use $N = n + s$ as a complexity measure. The proof appears in Appendix B.

[2] For example, for the two positive examples $(0,0,1)$ and $(0,1,1)$, the largest consistent conjunction is $x[1] = 0 \wedge x[3] = 1$.

Acknowledgments

KL's work was carried out in part while at Microsoft Research New England. This work was supported in part by NSF FAI grant#1939606, J.P. Morgan Faculty Award, Simons Foundation Collaboration 733792, Israel Science Foundation (ISF) grants 1044/16 and 2861/20, the United States Air Force and DARPA under contract FA8750-19-2-0222, and the Federmann Cyber Security Center in conjunction with the Israel national cyber directorate. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force and DARPA.

References

Dana Angluin and Philip D. Laird. Learning from noisy examples. *Machine Learning*, 2(4):343–370, 1987. doi: 10.1007/BF00116829. URL http://dx.doi.org/10.1007/BF00116829.

Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. In *Advances in Neural Information Processing Systems*, pages 998–1008, 2018.

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *Advances in neural information processing systems*, pages 137–144, 2007.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010.

John Blitzer, Ryan McDonald, and Fernando Pereira. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pages 120–128, 2006.

John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman. Learning bounds for domain adaptation. In *Advances in neural information processing systems*, pages 129–136, 2008.

Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2229–2238, 2019.

Koby Crammer, Michael Kearns, and Jennifer Wortman. Learning from multiple sources. *Journal of Machine Learning Research*, 9(Aug):1757–1774, 2008.

Mark Craveny, Dan DiPasquoy, Dayne Freitagy, Andrew McCallumzy, Tom Mitchelly, Kamal Nigamy, and Se an Slatteryy. Learning to extract symbolic knowledge from the world wide web. In *Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence. Menlo Park: American Association for Artificial Intelligence*, pages 509–516. Citeseer, 1998.

Ilias Diakonikolas, Themis Gouleakis, and Christos Tzamos. Distribution-independent pac learning of halfspaces with massart noise. In *Advances in Neural Information Processing Systems 32*, pages 4751–4762. Curran Associates, Inc., 2019.

Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. In *Advances in Neural Information Processing Systems*, pages 6447–6458, 2019.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017.

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–1189, 2015.

Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, and David Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In *Proceedings of the IEEE international conference on computer vision*, pages 2551–2559, 2015.

Jiang Guo, Darsh Shah, and Regina Barzilay. Multi-source domain adaptation with mixture of experts. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4694–4703, 2018.

David Haussler, Michael Kearns, Nick Littlestone, and Manfred K Warmuth. Equivalence of models for polynomial learnability. *Information and Computation*, 95(2):129–161, 1991.

Lisa Hellerstein and Rocco A Servedio. On pac learning algorithms for rich boolean function classes. *Theoretical Computer Science*, 384(1):66–76, 2007.

Michael J. Kearns. Efficient noise-tolerant learning from statistical queries. *J. ACM*, 45(6):983–1006, 1998. doi: 10.1145/293347.293351. URL http://doi.acm.org/10.1145/293347.293351.

Michael J. Kearns, Robert E. Schapire, Linda M. Sellie, and Lisa Hellerstein. Toward efficient agnostic learning. In *In Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, pages 341–352, 1992.

Michael J Kearns, Umesh Virkumar Vazirani, and Umesh Vazirani. *An introduction to computational learning theory*. MIT press, 1994.

Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei A Efros, and Antonio Torralba. Undoing the damage of dataset bias. In *European Conference on Computer Vision*, pages 158–171. Springer, 2012.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Da Li, Jianshu Zhang, Yongxin Yang, Cong Liu, Yi-Zhe Song, and Timothy M Hospedales. Episodic training for domain generalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1446–1455, 2019.

Shoushan Li and Chengqing Zong. Multi-domain sentiment classification. In *Proceedings of ACL-08: HLT, Short Papers*, pages 257–260, 2008.

Ping Luo, Fuzhen Zhuang, Hui Xiong, Yuhong Xiong, and Qing He. Transfer learning from multiple source domains via consensus regularization. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 103–112, 2008.

Massimiliano Mancini, Samuel Rota Bulò, Barbara Caputo, and Elisa Ricci. Best sources forward: domain generalization through source-specific nets. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 1353–1357. IEEE, 2018.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation with multiple sources. In *Advances in neural information processing systems*, pages 1041–1048, 2009a.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. In *22nd Conference on Learning Theory, COLT 2009*, 2009b.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Multiple source adaptation and the rényi divergence. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 367–374, 2009c.

Pascal Massart, Élodie Nédélec, et al. Risk bounds for statistical learning. *The Annals of Statistics*, 34(5): 2326–2366, 2006.

Pietro Morerio, Jacopo Cavazza, and Vittorio Murino. Minimal-entropy correlation alignment for unsupervised deep domain adaptation. In *International Conference on Learning Representations*, 2018.

Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pages 10–18, 2013.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7167–7176, 2017.

Leslie G. Valiant. A theory of the learnable. *Commun. ACM*, 27:1134–1142, 1984.

Riccardo Volpi, Pietro Morerio, Silvio Savarese, and Vittorio Murino. Adversarial feature augmentation for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5495–5504, 2018a.

Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. In *Advances in Neural Information Processing Systems 31*, 2018b.

Haohan Wang, Zexue He, Zachary C Lipton, and Eric P Xing. Learning robust representations by projecting superficial statistics out. In *ICLR*, 2019.

Chao Zhang, Lei Zhang, and Jieping Ye. Generalization bounds for domain adaptation. In *Advances in neural information processing systems*, pages 3320–3328, 2012.

Han Zhao, Shanghang Zhang, Guanhang Wu, José MF Moura, Joao P Costeira, and Geoffrey J Gordon. Adversarial multiple source domain adaptation. In *Advances in neural information processing systems*, pages 8559–8570, 2018.