

---

**1            Data Representation Framework Addressing  
the Training/Test Distributions Gap**

*Shai Ben-David*

We discuss some dataset shift learning problems from a formal, statistical, point of view. We offer definitions for “Multi-Task Learning”, “inductive transfer” and “Domain Adaptation” and discuss the parameters along which such learning scenarios may be taxonomized. We then focus on one concrete setting of domain adaptation and demonstrate how error bounds can be derived for that setting. Our bounds can be reliably estimated from finite samples of training data, and do not rely on any assumptions concerning similarity between the domain from which the labeled training data is sampled and the target (or test) data. However, these bounds are *relative* to the performance of some optimal classifier, rather than providing any *absolute* performance guarantee.

---

## 1.1 Introduction

We consider a setting where a learner has access to labeled training data generated according to some *training data* distribution (or several different data distributions), and wishes to learn a classifier which performs well with respect to a different, ‘target’ (or ‘test’), data distribution.

Such learning scenarios are usually discussed under the titles “domain adaptation”, “inductive transfer” and “multi-task learning”. We propose formal definitions for these learning problems. These definitions may further partition along different input availability settings. Namely, what kind of target distribution data is available to the learner? labeled? unlabeled? distribution description? constraints on possible such distributions? etc.

Clearly, the success of any such learning depends on the knowledge the learner has about the test data or on its relationship to the training data. Common approaches to this problem rely on postulating some prior assumptions in that respect. We propose a different approach. Rather than aiming for *absolute* error bounds, that are (inevitably) conditioned upon such prior assumptions, we make no prior assumptions about the test domain, and derive error bounds which are *relative* to the best possible performance in the relevant setting.

We focus on scenarios where the learner can access samples generated by the test-data distribution, alas, without the labels of these data points (this is on top of having access to labeled examples generated by the training-data distribution).

In such scenarios, datashift learning paradigms can be divided along another aspect – the distinction between, what we call, *conservative* and *adaptive* learners. A conservative algorithm is one that makes its choice of predictor function based only on the labeled sample from the training domain. Such a learner uses the unlabeled target sample only as a tool for evaluating the quality of the chosen predictor. In contrast, adaptive learners incorporate the unlabeled target sample as an integral part of the learning process. We focus our analysis on conservative learners, and argue, in Section 1.5 that, as long as the learner has access to only unlabeled samples

from the target distribution, no reliable adaptive learning can take place (unless further assumptions, concerning that distribution, are made). This claim remains valid even in the *covariate shift* setting, where one assumes that the conditional distribution of label values, given the unlabeled data, is unchanged between the training and target distributions.

We introduce some parameters, depending on the learning algorithm, the distribution of labeled training data and the distribution of unlabeled test data, that determine the generalization error of conservative learning in our framework. A key component in our discussion is the introduction of a special measure for the similarity between probability distributions. We show that, for the purpose of domain adaptation, it is useful to define that similarity as the error that the relevant learning algorithm will make when applied to distinguish between the training and target unlabeled data distributions.

We prove convergence rates for the (quality of) approximation of these relevant parameters from finite samples of labeled training data and unlabeled test data, and derive some basic theoretical performance guarantees for classifiers in terms of these parameters.

---

## 1.2 Formal Framework and Notation

In this paper, a learning task is modeled by a probability distribution  $P$  over labeled examples. Namely, for some domain set,  $\mathcal{X}$ , and a set of labels  $\mathcal{Y}$  (which, for concreteness, we take to be  $\{0, 1\}$ ),  $P$  is a distribution over  $\mathcal{X} \times \mathcal{Y}$ . We shall also consider probability distributions over the space,  $\mathcal{X}$ , of unlabeled examples. Given a task,  $P$ , as above, we use  $D_P$  to denote the probability distribution over the domains set (the data marginal distribution),  $\mathcal{X}$ , obtained by projecting  $P$  to  $\mathcal{X}$  (by erasing the labels). Namely, for a subset  $A \subseteq \mathcal{X}$ ,  $D_P(A) = P(\{(x, y) : x \in A\})$  - the probability of drawing  $x \in A$  regardless of the label it is paired with.

By a *sample* for a task,  $P$ , we mean a multiset of labeled points,  $S = ((x_1, \ell_1), \dots, (x_m, \ell_m))$ , picked i.i.d. according to the distribution  $P$ . An *unlabeled* sample of  $P$  is a multiset of points from  $\mathcal{X}$  picked i.i.d. according to the unlabeled distribution  $D_P$ .

For some parts of our discussion, we shall need to get into some further detail and consider also the measure space over which our distributions (or ‘tasks’) are defined. A measure space is a pair,  $(\mathcal{X}, \mathcal{B})$ , where  $\mathcal{X}$  is any domain space, as above, and  $\mathcal{B}$  is a  $\sigma$ -algebra of subsets of  $\mathcal{X}$ . We shall assume that all the probability distributions considered are defined over some measure space (that will remain implicit for most of our discussion). Given a domain space,  $(\mathcal{X}, \mathcal{B})$ , and a finite label space,  $\mathcal{Y}$ , a task  $P$  over it is assumed to be defined over the space  $(\mathcal{X} \times \mathcal{Y}, \{b \times l : b \in \mathcal{B} \text{ and } l \subseteq \mathcal{Y}\})$ . As mentioned above, we shall focus on the case that  $\mathcal{Y} = \{0, 1\}$  and shall make no explicit reference to  $\mathcal{Y}$  in our notation.

A *multi-task* is an array of tasks,  $P_1, \dots, P_n$ , all defined over the same space. For most of our discussion, we shall focus on the case of an array of size 2, where

we have just two tasks, a *training task*, that we shall denote by  $P_{tr}$ , and a *target task* that we shall denote by  $P_{te}$ .

The learner wishes to construct a predictor  $h : \mathcal{X} \rightarrow \{0, 1\}$  for the target task. Namely, the learner wishes to minimize the error of its predictor on the target task distribution. That error is defined as the expectation, w.r.t. the distribution  $P_{te}$ , of the 0/1 loss of  $h$ . That is,  $Er^{P_{te}}(h) = E_{x(x,\ell) \sim P_{te}} L(h, (x, \ell))$ , where  $L(h, (x, \ell)) = 0$  if  $h(x) = \ell$  and  $L(h, (x, \ell)) = 1$  if  $h(x) \neq \ell$ .

### 1.3 A Basic Taxonomy of Tasks and Paradigms

Learning for the case of multiple tasks have been considered in various settings. We attempt to provide a basic taxonomy for such problems by considering several determining aspects. The first is the distinction between *symmetric* and *asymmetric* settings. In the symmetric case, the learner has the same type of information for all the relevant learning tasks, and wishes to use the existence of multiple tasks to improve the learning in all of them. This is the scenario that is considered by Baxter's seminal work ?. In such settings (e.g. in ?), the learner is interested in improving the *average* quality of prediction, over the full array of tasks (compared to learning *each* of them separately), or in improving the learning quality over each of the tasks ?. We call this scenario *multi-task learning*. In the asymmetric setting, there is some designated *target task*, and the learner aims to improve the quality of learning on that particular task (again, compared to learning it without access to the additional tasks in the array). We call this type of tasks *inductive transfer*. Such settings may further partition according to the the type of target-task information available to the learner. We propose using the term *domain adaptation* for situations in which the learner has access to only *unlabeled* target samples.

**Symmetric Setting – Multi-task learning** We say that  $P_1, \dots, P_n$  allows *multi-task learning* if, for any sequence of learning algorithms,  $A_1, \dots, A_n$ , one for each of the tasks, there exist a learning algorithm,  $\hat{A}$  that takes an array of samples, one from each task, as input, and outputs an array of predictors, one for each task, such that for any large enough  $m$ , if, for all  $i$ ,  $S_i$  is an  $m$ -size sample of the task  $P_i$ , then, with high probability (over the choice of the  $S_i$ 's), for any  $k \leq n$ ,  $E_k(\hat{A}(S_1, \dots, S_n)) < E(A_k(S_k))$  (where  $E_k$  is the expectation of the error of  $\hat{A}$  when predicting on the  $k$ 'th task, over the input random samples). That is,  $\hat{A}$  utilizes samples from the different tasks to improve the individual task predictions based on single-task training data.

*Possible variant- weak MTL* - rather than having  $\hat{A}$  beat the single-task algorithms on every task, require only that the *average* of these errors, over the array of tasks, is better than the average one would get by learning each task separately.

**Asymmetric Setting – Domain Adaptation** We say that  $P_1, \dots, P_{n-1}$  allows *domain adaptation* to a task  $T_n$  if, there exist a learning algorithm,  $\hat{A}$  that takes an array of samples  $S_1, \dots, S_{n-1}$ , one from each task, as input, and outputs a

predictor for  $T_n$  such that for every large enough  $m$ , if, for all  $i < n$ ,  $S_i$  is an  $m$ -size sample of the task  $P_i$ , then, with high probability (over the choice of the  $S_i$ 's),  $Er_{(\hat{A}(S_1, \dots, S_{n-1}))}(P_n) < \min\{Er_1(P_n), Er_0(P_n)\}$ , where  $Er_i(P_n)$  is the error of the constant label- $i$  predictor on the task  $P_n$  ( $Er_{(\hat{A}(S_1, \dots, S_{n-1}))}(P_n)$  is the error on  $T_n$  of  $\hat{A}$ , trained on  $S_1, \dots, S_{n-1}$ ). That is, using training samples for the tasks  $P_i$ ,  $i < n$ , one can come up with a non-trivial predictor for  $P_n$ , without having access to training data from that task ('non-trivial' in the sense that it has lower error than any of the two constant-value predictors).

*Variations:* One can readily define several variations of these types of task, e.g., allowing for the use of un-labeled target data as part of the input to  $\hat{A}$  in any of the above, or demanding that, having access to large enough samples, the error of  $\hat{A}$  can be made arbitrarily small.

**Conservative versus Adaptive prediction** An orthogonal dimension along which relevant approaches may be classified is the algorithmic paradigms they employ. Do they address the data shift explicitly or do they ignore the data shift in the learning process and address it only for the evaluation of the resulting predictor? We distinguish between two possible learning paradigms. In the first, the learner chooses the predictor that works best with respect to the training task(s), and wishes to evaluate how good will that predictor be when employed on the target task. We call this "conservative prediction". In the second setting, we consider learning strategies that allow the predictor they pick for the target task to differ from the predictor they would use for the training task. We call it "adaptive prediction".

---

## 1.4 Error Bounds for Conservative Domain Adaptation Prediction

In the conservative prediction setting, one wishes to upper bound the error of a predictor on the target (or 'goal') task by its error on the training task. Clearly, such a bound depends on the similarity between the training distribution and the goal distribution. Any performance guarantee of this type should therefore involve a measure of that task similarity. A common measure for the dissimilarity between two probability distributions is the  $L_1$  distance (also called the *total variance* or *statistical distance*). It is defined, as

$$d_{L_1}(P, Q) = 2 \sup_{A \in \mathcal{B}} |P(A) - Q(A)|$$

Recall that  $\mathcal{B}$  denotes the set of all  $P$ -measurable domain subsets.

Based on this  $L_1$  distance between the two (unlabeled) data distributions, one can readily get a rather straightforward bound on the error on the Goal distribution, in terms of the error of a hypothesis on the training distribution (in the covariate shift setting):

**Lemma 1.1** *Under the covariate shift assumption, for every predictor,  $h : x \rightarrow$*

$\{0, 1\}$ ,

$$Er^G(h) \leq Er^T(h) + \frac{d_{L_1}(D_T, D_G)}{2}$$

**Proof:** Note that, due to the covariate shift assumption, for every point  $x \in \mathcal{X}$ ,  $G(1|x) = T(1|x)$ . Let us denote this quantity by  $\ell(x, 1)$  (the probability, under either  $G$  or  $T$ , that the label of  $x$  is 1) and let  $\ell(x, 0) = 1 - \ell(x, 1)$  (namely, the probability that the label of  $x$  is 0). The only possible source of the error of a predictor  $h$  to be greater for  $G$  than for  $T$  is that  $G$  puts more weight on points on which  $h$  has a relatively large error. Namely,

$$Er^G(h) - Er^T(h) \leq D_G\{x : \ell(x, h(x)) > Er^T(h)\} - D_T\{x : \ell(x, h(x)) > Er^T(h)\}$$

Now, note that by the definition of  $d_{L_1}$ , the right hand side of this inequality is at most  $1/2d_{L_1}(D_T, D_G)$ .

The above bound has two major weaknesses. First, it may be overly pessimistic, a second concern is that, based on the data available to the learner in the setting we discuss, there is no way to estimate the crucial similarity parameter  $d_{L_1}(D_T, D_G)$  reliably.

We address these issues by developing an alternative error bound. The new bound is based on a different measure of similarity between distributions.

#### 1.4.1 A Special Measure of Between-Distributions Distance

To measure the similarity between two probability distributions, we shall use the  $d_{\mathcal{A}}$  measure, introduced in ? and ?. The  $d_{\mathcal{A}}$  measure is parameterized by a collection  $\mathcal{A}$  of subsets of the domain over which the probability distributions are defined. Intuitively speaking,  $\mathcal{A}$  is the collection of ‘subsets of interest’ with respect to the properties of the distributions that one wishes to analyze. In our case, when we analyze a domain adaptation learning algorithm, that collection is determined by the learning algorithm that is used to generate the classification predictors (more precisely, by the set of potential predictors that that algorithm may output). The motivation behind the introduction of that measure come from real life scenarios in which one cares only about certain distribution changes. For example, in ? we discuss detecting changes in the generating distribution of streaming real valued data, in that context one cares only about changes that effect the probabilities of real intervals.

**Definition 1.2** Let  $\mathcal{X}$  be some domain set and let  $\mathcal{B}$  be a  $\sigma$ -algebra of subsets of  $\mathcal{X}$ . Let  $\mathcal{A} \subseteq \mathcal{B}$ . For probability distributions  $P, Q$  over  $(\mathcal{X}, \mathcal{B})$ ,

$$d_{\mathcal{A}}(P, Q) = 2 \sup_{A \in \mathcal{A}} |P(A) - Q(A)|$$

Note that the only difference between this measure and the  $L_1$  distance is that in the  $d_{\mathcal{A}}$  distance, we restrict our attention to some fixed collection  $\mathcal{A}$  of subsets, rather than considering the full  $\sigma$ -algebra of measurable sets. This difference becomes meaningful when  $\mathcal{A}$  is ‘small’ compared to the full collection of measurable sets. Below, we demonstrate the benefits gained when that smallness is reflected by the VC-dimension.

**Lemma 1.3** *For any pair of probability distributions,  $P, Q$ , over some domain measure space  $(\mathcal{X}, \mathcal{B})$ , and for every family of sets  $\mathcal{A} \subseteq \mathcal{B}$ ,*

$$d_{\mathcal{A}}(P, Q) \leq d_{L_1}(P, Q)$$

**Example 1:** *Let  $\mathcal{X}$  be some Euclidean space,  $\mathbb{R}^d$  and let  $\mathcal{B} = \mathcal{L}(\mathbb{R}^d)$  be the collection of Lebesgue measurable subsets of  $\mathbb{R}^d$ . Let  $P$  be the uniform distribution over  $D_{100}^{odd} = \{(x_1, \dots, x_d) \in [0, 1]^d : \sum_{i=1}^d (\text{the } 100\text{'th decimal digit of } d_i) \text{ is odd}\}$  and let  $Q$  be the uniform distribution over its complement,  $([0, 1]^d \setminus D_{100}^{odd})$ . It is easy to see that  $D_{L_1}(P, Q) = 2$  (note that, for any  $P, Q$ , it is always the case that  $D_{L_1}(P, Q) \leq 2$ ). Just the same, if we let  $\mathcal{A}$  be the set of all linear half-spaces in  $\mathbb{R}^d$ , then  $d_{\mathcal{A}}(P, Q) = 0$ .*

Note that one could easily modify the above example so that the two distributions will be absolutely continuous with respect to each other; let  $0 \leq \lambda \leq 1$ , the distributions  $P' = \lambda U + (1 - \lambda)P$  and  $Q' = \lambda U + (1 - \lambda)Q$  (where  $U$  is the uniform distribution over  $[0, 1]^d$ ), satisfy  $d_{\mathcal{A}}(P', Q') = 0$  and  $D_{L_1}(P', Q') = 2 - 2\lambda$

Kifer et al. [?] use a uniform convergence argument to show that, if  $\mathcal{A}$  has a finite VC-dimension, then it is possible to reliably estimate the  $\mathcal{A}$ -distance from finite samples. We state here the relevant result from that work:

**Lemma 1.4** *Let  $\mathcal{A}$  be a class of subsets of some domain set  $\mathcal{X}$  and let  $d < \infty$  be the VC dimension of  $\mathcal{A}$ . Let  $P, Q$  be any probability distributions over  $\mathcal{X}$  and  $\epsilon \in (0, 1)$ . Then, for any sample size,  $m$ , for i.i.d.  $m$  samples,  $S, S'$ , drawn by  $P, Q$  respectively,*

$$\Pr [|d_{\mathcal{A}}(P, Q) - d_{\mathcal{A}}(S, S')| \geq \epsilon] < (2m)^d 4e^{-m\epsilon^2/4}$$

where  $\Pr[\cdot]$  is over random draws of  $m$ -size independent samples from the distributions  $P$  and  $Q$ , and we identify a finite sample  $S$  with the probability distribution that assigns each point a weight equal to its relative frequency in  $S$ .

Note that the above claim is in sharp contrast to the  $d_{L_1}$  case, as the following claim demonstrates.

**Claim:** *Let  $U$  be the uniform distribution over the unit interval with the Lebesgue algebra of measurable subsets,  $([0, 1]^d, \mathcal{L}([0, 1]^d))$ . For any statistical test  $T$  and any number  $m$ , there exist a probability distribution,  $P$ , over the same domain space, such that  $d_{L_1}(U, P) = 1$  and yet, given a pair of  $m$ -size samples,  $S_1, S_2$  as input,*

the test  $T$  cannot distinguish between the case that both samples were both drawn i.i.d. from  $U$  and the case that  $S_1$  was drawn i.i.d. from  $U$  and  $S_2$  was drawn i.i.d. from  $P$ .

Alternatively, one could define the  $D_{\mathcal{A}}$  measure from a learning perspective. Intuitively speaking, viewing  $\mathcal{A}$  as a set of functions from  $\mathcal{X}$  to  $\{0, 1\}$  (or, label predictors), we ask how well can a function from  $\mathcal{A}$  separate examples generated by the two distributions. It turns out that the prediction error on such a task can be used to define the  $d_{\mathcal{A}}$  distance.

**Definition 1.5 (Alternative, equivalent definition of  $d_{\mathcal{A}}$ )** For  $\mathcal{X}$ ,  $\mathcal{A}$ ,  $P$  and  $Q$  as above, consider the task of finding a predictor  $h : \mathcal{X} \rightarrow \{0, 1\}$  that distinguishes points generated by  $P$  from points generated by  $Q$ . That is, consider the mixture distribution  $(P, Q) = \frac{1}{2}[P \times \{1\} + Q \times \{0\}]$  over  $\mathcal{X} \times \{0, 1\}$  (i.e., with probability  $1/2$ ,  $x$  is drawn from  $P$  and has label  $\ell(x) = 1$ , and with probability  $1/2$ ,  $x$  is drawn from  $Q$  and has label  $0$ ), define the error of such a predictor as  $Err^{(P, Q)}(h) = \Pr_{(P, Q)}[h(x) \neq \ell(x)]$ . Given a set  $A \in \mathcal{A}$ , let  $h_A$  be the characteristic function of  $A$  (that is,  $h(x) = 1$  iff  $x \in A$ ). We can rewrite the definition of  $d_{\mathcal{A}}$  as

$$d_{\mathcal{A}}(P, Q) = 1 - 2 \inf_{h_A | A \in \mathcal{A}} Err^{(P, Q)}(h_A)$$

It is straightforward to see that the two definitions of  $D$  above are equivalent.

### 1.4.2 Relative Error Bounds

We wish to bound the error of a predictor w.r.t.  $P_{te}$  in terms of its error w.r.t.  $P_{tr}$ . Since we must rely on finite samples (rather than having access to the actual distributions,  $P_{tr}$  and  $P_{te}$ ), and since we wish to use that bound for selecting the best predictor, we must restrict our attention to some restricted class of predicting functions. Let  $H$  denote such a class. As mentioned above, even under the covariate shift assumption, it may still be the case that predictors that perform well with respect to  $P_{tr}$  fail badly w.r.t.  $P_{te}$ . Can the similarity between  $P_{tr}$  and  $P_{te}$  bound that gap in the quality of predictors over the two tasks? Lemma 1.1 shows that the answer is positive once the two probability distributions are close in the  $L_1$  sense. However, as demonstrated above, such a requirement is severely restrictive and its validity cannot be assessed on the basis of the information available to the learner in the model we consider. Can similarity in terms of  $d_{\mathcal{A}}$ , for some  $\mathcal{A}$  with a finite VC dimension suffice to imply such a bound? The following example shows that it does not.

**Example 2:** Let  $\mathcal{X}$  be the real unit interval  $[0, 1]$ , and let  $\mathcal{A}$ , as well as  $H$ , be the set of all linear half spaces over  $\mathcal{X}$  (that is, the set of all threshold classifiers). Let  $f : [0, 1] \rightarrow \{0, 1\}$  be defined by  $f(x) = 1$  if the 100<sup>th</sup> digit of  $x$  (in its decimal expansion, without a tail of 9's), and  $f(x) = 0$  otherwise. Now let  $P_{tr}$  be the uniform

distribution over  $f^{-1}(1)$  and let  $P_{te}$  be the uniform distribution over  $f^{-1}(0)$ . Now,  $d_H(P_{tr}, P_{te}) = 0$ , and yet, the constant predictor  $h(x) \equiv 1$  has zero error w.r.t.  $P_{tr}$  but has error 1 w.r.t.  $P_{te}$ .

As long as no data about the distribution of labels w.r.t.  $P_{te}$  is available, and no assumption about it is made (apart from the covariate shift assumption), we suggest that, rather than aiming towards *absolute* error bounds, one should settle for *relative* bounds. Namely, bounds with respect to the best possible performance under the given circumstances. More concretely,

**Definition 1.6** Given a domain space  $\mathcal{X}$ , a class of binary predictors,  $H$  over it (a hypothesis class), and probability distributions  $P_{tr}$  and  $P_{te}$  as above, let

$$\lambda_{H, P_{tr}, P_{te}} = \inf_{h \in H} (Er^{P_{te}}(h) + Er^{P_{tr}}(h))$$

**Notation:** We identify a binary function,  $h : \mathcal{X} \rightarrow \{0, 1\}$ , with the subset  $h^{-1}(1)$  of  $\mathcal{X}$ . We denote by  $\Delta(H)$  the class of symmetric differences of functions from  $H$ . Namely,  $\Delta(H) = \{h\Delta h' : h, h' \in H\}$ .

We are now ready to present our main error bound.

**Theorem 1.7** Let  $P_{tr}$  and  $P_{te}$  be probability distributions over  $\mathcal{X} \times \{0, 1\}$ , for some space  $(\mathcal{X}, \mathcal{B})$ , and let  $H$  be a class of measurable binary functions over that space. For any  $h \in H$ ,

$$|Er^{P_{te}}(h) - Er^{P_{tr}}(h)| \leq \lambda_{H, P_{tr}, P_{te}} + \frac{1}{2} d_{\Delta(H)}(D_{P_{tr}}, D_{P_{te}})$$

Where,  $D_{P_{te}}$  and  $D_{P_{tr}}$  are the projections on  $\mathcal{X}$  of  $P_{te}$  and  $P_{tr}$  respectively.

**Proof:** Let  $h^* = \operatorname{argmin}_{h \in H} (Er^{P_{tr}}(h) + Er^{P_{te}}(h))$ , and let  $\lambda_{P_{tr}}$  and  $\lambda_{P_{te}}$  be the errors of  $h^*$  with respect to  $P_{tr}$  and  $P_{te}$  respectively. Notice that  $\lambda_{H, P_{tr}, P_{te}} = \lambda_{P_{tr}} + \lambda_{P_{te}}$ .

$$\begin{aligned} Er^{P_{te}}(h) &\leq \lambda_{P_{te}} + \Pr_{D_{P_{te}}} [h\Delta h^*] \\ &\leq \lambda_{P_{te}} + \Pr_{D_{P_{tr}}} [h\Delta h^*] + (\Pr_{D_{P_{te}}} [h\Delta h^*] - \Pr_{D_{P_{tr}}} [h\Delta h^*]) \\ &\leq \lambda_{P_{te}} + \Pr_{D_{P_{tr}}} [h\Delta h^*] + \frac{1}{2} d_{\Delta(H)}(D_{P_{tr}}, D_{P_{te}}) \\ &\leq \lambda_G + \lambda_{P_{tr}} + Er^{P_{tr}}(h) + \frac{1}{2} d_{\Delta(H)}(D_{P_{te}}, D_{P_{tr}}) \\ &\leq \lambda + Er^{P_{tr}}(h) + \frac{1}{2} d_{\Delta(H)}(D_{P_{te}}, D_{P_{tr}}) \end{aligned}$$

Probably the only step that requires explanation is the inequality  $\Pr_{D_{P_{tr}}} [h\Delta h^*] \leq \lambda_{P_{tr}} + Er^{P_{tr}}(h)$  used for moving from the third line above to the forth. This inequality holds since, for every point  $x \in h\Delta h^*$ , the error probability of  $h$  on  $x$  and the error probability of  $h^*$  on  $x$  sum to 1. The theorem now follows by noting that the above argument is symmetric - the roles of  $P_{tr}$  and  $P_{te}$  are interchangeable

**Corollary 1.8** *Let  $P_{tr}$  and  $P_{te}$  and  $H$ , be as above. If  $H$  has a finite VC-dimension,  $d$ , then for every  $m, m' \in \mathbb{N}$ . If a random labeled sample,  $S_{tr}$ , of size  $m$  is i.i.d by  $P_{tr}$  and random unlabeled samples,  $U_{tr}, U_{te}$ , each of size  $m'$ , are generated i.i.d. by  $D_{P_{tr}}$  and  $D_{P_{te}}$  respectively, then, with probability at least  $1 - \delta$  (over the choice of the samples), for every  $h \in H$ :*

1.

$$Er^{P_{te}}(h) \leq Er^{P_{tr}}(h) + \lambda_{P_{tr}, P_{te}, H} + \frac{1}{2}d_{\Delta(H)}(U_{tr}, U_{te}) + 4\sqrt{\frac{d \log(2m') + \log(\frac{4}{\delta})}{m'}}$$

Where, for finite samples,  $S$ , the same notation is used to denote both the sample itself and the uniform probability distribution over its elements.

2.

$$Er^{P_{tr}}(h) \leq Er^{S_{tr}}(h) + \frac{4}{m} \left( d \log \frac{2em}{d} + \log \frac{4}{\delta} \right) + \lambda_{P_{tr}, P_{te}, H} + \frac{1}{2}d_{\Delta(H)}(U_{tr}, U_{te}) + 4\sqrt{\frac{d \log(2m') + \log(\frac{4}{\delta})}{m'}}$$

**Proof:** The corollary follows from Theorem 1.7 by replacing  $Er^{P_{tr}}$  by its VC based empirical upper bound, and upper bounding  $d_{\Delta(H)}(D_{P_{tr}}, D_{P_{te}})$  in terms of its empirical value on samples, through Lemma ?.

Note, that while part 2 of Corollary 1.8 sounds more practical than part 1 and than Theorem 1.7 (since all the parameters involved in the bound are derived from available training data), the latter formulations are more general. Theorem 1.7 allows flexibility in estimating the true error of hypotheses, and in choosing the preferred predictor. For example, theorem 1.7 and part 1 of the corollary apply also to algorithms employing regularization methods (such as margin or description complexity penalty term) rather than empirical risk minimization. Likewise, Theorem 1.7 is applicable also in cases where the method used to asses the distance between the training and goal task distribution is different than picking random unlabeled samples and computing their empirical distance.

### 1.4.3 Distinguishing Features of Our Bounds

Let us summarize the main aspects in which the bounds presented above may differ from other theoretical analysis of domain adaptation.

- Maybe the most significant merit of this bound is that it can be *reliably estimated* using the data available to the learner. A learner can run any learning algorithm on the training data (the labeled training-task sample), estimate the error of the outcome predictor for the training-task (using any common error estimation technique), then ,using the unlabeled sample from the goal-task, apply the abound of Theorem 1.7 to obtain a (guaranteed) upper bound on the error that predictor

on the goal-task. Such a bound may be used by a learner to determine whether the application of a more elaborate paradigm (like adaptive prediction) is required.

- Another distinctive feature of our bounds is that they do not rely on any assumptions concerning the relationship between the training data domain and the goal (or test) data domain. Obviously, This feature makes our bounds more general. However, the other side of the coin is that this generality has a cost in terms of the tightness of the bounds. Whenever prior knowledge about the relationship between the two learning domains is available, it is conceivable that that prior knowledge can be utilized to get better bounds. Furthermore, such prior domain knowledge may allow the application of *adaptive prediction* - choosing predictors that have better performance on the goal domain than the predictors chosen just based on their training-domain performance (as is the case with conservative prediction that we analyzed above).
- Finally, it should be noted that our bounds are *Relative* bounds, in the sense that rather than providing absolute upper bounds for the learned predictors error, they bound only the difference between that error and the error of some baseline predictor - the sum of the training and test errors of the best predictor in the hypothesis class,  $H$  (this is what  $\lambda_{T,G,H}$  denotes). This relaxation of the guarantee is an inevitable consequence of not making any prior task assumptions (as discussed in the previous point). In a way, this is similar to the distinction between the agnostic and PAC models of learning. The first makes no prior assumptions about the label generating distribution, but settles for generalization bounds that are relative to those of the best predictor in some reference hypotheses class. The latter, the PAC model, assumes that the hypothesis class contains a perfect (i.e., zero error) predictor, and, under that assumptions, can expect absolute numeric bounds on the error of the learner's predictor.

---

## 1.5 Adaptive Predictors

Conservative predictors seem quite limited. Rather than performing a learning process over the target-domain examples (in a way that utilizes training domain data), they just perform learning over the training domain and apply the resulting predictor to the target task. Strategies that allow the predictor they pick for the target task to differ from the predictor they would use for the training task seem to carry greater promise. However, in the context of domain adaptation, when no target domain labeled data is available, the reliability of such paradigms is not guaranteed.

It should be realized that to allow reliable success of such paradigms, rather strong assumptions concerning the learning tasks at and should be made. In particular, assuming *covariate shift* (namely, that the conditional distribution of label values, given the unlabeled data, is unchanged between the training and target distributions) is far from being sufficient to guarantee domain adaptation.

In Example 2 above, we described a pair of tasks for which there exists a perfect label predictor for the training distribution that has error probability 1 on a goal probability, in spite of both tasks being defined on the same domain and sharing the same conditional label distribution ( $\Pr[\text{label}|x]$ ). In that example, availability of unlabeled samples of the target task does not help to overcome the training/test discrepancy.

Some recent work on learning under the covariate shift assumption suggest to overcome this problem by estimating the data density ratio between the training and target distributions by using sample-based empirical values (see, e.g., ? and ?). However, it should be noted that, without restricting the family of possible distributions, no finite sample can yield reliable approximating the actual distribution. This is sharply demonstrated by Example 1, above. Even in settings where the support of the data distribution is finite, in order to obtain reliable empirical estimates of the target distribution one needs sample sizes that approach the cardinality of that support (see ?), which seem way too much for all practical applications.

### 1.5.1 Some solutions

We shall briefly list below some common approaches providing settings that allow reliable adaptive inductive transfer learning algorithms. These solutions are all based on assuming some prior knowledge about the learning tasks.

**Restriction of the family of potential target distributions:** Ben-David and Schuler ? consider a framework in which there is some known family of distribution transformations, such that the target task is obtained by applying one of these transformations to the training task. They show that in cases where that family of transformations has a finite VC-dimension, reliable adaptive learning can be guaranteed (in fact, that paper consider the multi-task setting, and shows that under such conditions the learnability of each of the tasks improves as a result of having access to training samples from other tasks).

**Existence of ‘good’ domain embedding** Ben-David, Blitzer, Crammer and Pereira ? consider the domain adaptation setting where the learner has access to unlabeled target task sample (but no access to labeled samples from the target task). They show that if the learner can come up with data embedding for both the training and target domains, such that the images of the unlabeled distributions (of the training and target tasks) are similar, and such that under that embedding learnability of the training task is possible, then that embedding can be used to achieve reliable adaptive learning of the target task. the key component in their argument can be viewed as an ‘embedding version’ of Theorem 1.7 above.

It is interesting to note that by using such an embedding the learner sacrifices the covariate shift assumption (in cases that that assumption holds for the original task domains) in order to gain similarity between the unlabeled distributions.

---

## Notation and Symbols



### Sets of Numbers

$\mathbb{N}$	the set of natural numbers, $\mathbb{N} = \{1, 2, \dots\}$
$\mathbb{R}$	the set of reals
$[n]$	compact notation for $\{1, \dots, n\}$
$x \in [a, b]$	interval $a \leq x \leq b$
$x \in (a, b]$	interval $a < x \leq b$
$x \in (a, b)$	interval $a < x < b$
$ C $	cardinality of a set $C$ (for finite sets, the number of elements)

### Data

$\mathcal{X}$	the input domain
$d$	(used if $\mathcal{X}$ is a vector space) dimension of $\mathcal{X}$
$M$	number of classes (for classification)
$n$	a number of data examples.
$n_{\text{tr}}$	number of training examples.
$n_{\text{te}}$	number of test examples.
$i, j$	indices, often running over $[n_{\text{te}}]$ or $[n_{\text{tr}}]$ .
$x_i$	input patterns $x_i \in \mathcal{X}$
$x_i^{\text{tr}}$	input training patterns $x_i^{\text{tr}} \in \mathcal{X}$
$x_i^{\text{te}}$	input test patterns $x_i^{\text{te}} \in \mathcal{X}$
$y_i$	classes $y_i \in [M]$ (for regression: target values $y_i \in \mathbb{R}$ )
$y_i^{\text{tr}}$	training data classes $y_i^{\text{tr}} \in [M]$ (for regression: target values $y_i^{\text{tr}} \in \mathbb{R}$ )
$y_i^{\text{te}}$	test data classes $y_i^{\text{te}} \in [M]$ (for regression: target values $y_i^{\text{te}} \in \mathbb{R}$ )
$X$	a sample of input patterns, $X = (x_1, \dots, x_n)$
$X^{\text{tr}}$	a sample of training input patterns, $X^{\text{tr}} = (x_1^{\text{tr}}, \dots, x_n^{\text{tr}})$
$X^{\text{te}}$	a sample of test input patterns, $X^{\text{te}} = (x_1^{\text{te}}, \dots, x_n^{\text{te}})$
$Y$	a sample of output targets, $Y = (y_1, \dots, y_n)$
$Y^{\text{tr}}$	a sample of training output targets, $Y^{\text{tr}} = (y_1^{\text{tr}}, \dots, y_n^{\text{tr}})$
$Y^{\text{te}}$	a sample of training output targets, $Y^{\text{te}} = (y_1^{\text{te}}, \dots, y_n^{\text{te}})$

### Kernels

$\mathcal{H}$	feature space induced by a kernel
$\Phi$	feature map, $\Phi : \mathcal{X} \rightarrow \mathcal{H}$
$k$	(positive definite) kernel
$K$	kernel matrix or Gram matrix, $K_{ij} = k(x_i, x_j)$

### Vectors, Matrices and Norms

$\mathbf{1}$	vector with all entries equal to one
$\mathbf{I}$	identity matrix
$A^{\top}$	transposed matrix (or vector)
$A^{-1}$	inverse matrix (in some cases, pseudo-inverse)
$\text{tr}(A)$	trace of a matrix
$\det(A)$	determinant of a matrix
$\langle \mathbf{x}, \mathbf{x}' \rangle$	dot product between $\mathbf{x}$ and $\mathbf{x}'$
$\ \cdot\ $	2-norm, $\ \mathbf{x}\  := \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$
$\ \cdot\ _p$	$p$ -norm, $\ \mathbf{x}\ _p := \left( \sum_{i=1}^N  x_i ^p \right)^{1/p}$ , $N \in \mathbb{N} \cup \{\infty\}$
$\ \cdot\ _{\infty}$	$\infty$ -norm, $\ \mathbf{x}\ _{\infty} := \sup_{i=1}^N  x_i $ , $N \in \mathbb{N} \cup \{\infty\}$

### Functions

$\ln$	logarithm to base $e$
$\log_2$	logarithm to base 2
$f$	a function, often from $\mathcal{X}$ or $[n]$ to $\mathbb{R}$ , $\mathbb{R}^M$ or $[M]$
$\mathcal{F}$	a family of functions
$L_p(\mathcal{X})$	function spaces, $1 \leq p \leq \infty$

### Probability

$P\{\cdot\}$	probability of a logical formula
$P_{\text{tr}}\{\cdot\}$	probability of a logical formula associated with training data distribution.
$P_{\text{te}}\{\cdot\}$	probability of a logical formula associated with test data distribution.
$P(C)$	probability of a set (event) $C$
$p(x)$	density evaluated at $x \in \mathcal{X}$
$p_{\text{tr}}(x)$	density associated with training data distribution evaluated at $x \in \mathcal{X}$
$p_{\text{te}}(x)$	density associated with test data distribution evaluated at $x \in \mathcal{X}$
$\mathbf{E}[\cdot]$	expectation of a random variable
$\mathbf{Var}[\cdot]$	variance of a random variable
$\mathcal{N}(\mu, \sigma^2)$	normal distribution with mean $\mu$ and variance $\sigma^2$

### Graphs

$\mathbf{g}$	graph $\mathbf{g} = (V, E)$ with nodes $V$ and edges $E$
$\mathcal{G}$	set of graphs
$W$	weighted adjacency matrix of a graph ( $W_{ij} \neq 0 \Leftrightarrow (i, j) \in E$ )
$D$	(diagonal) degree matrix of a graph, $D_{ii} = \sum_j W_{ij}$
$\mathcal{L}$	normalized graph Laplacian, $\mathcal{L} = D^{-1/2} W D^{-1/2}$
$L$	un-normalized graph Laplacian, $L = D - W$

### SVM-related

$\rho_f(x, y)$	margin of function $f$ on the example $(x, y)$ , i.e., $y \cdot f(x)$
$\rho_f$	margin of $f$ on the training set, i.e., $\min_{i=1}^m \rho_f(x_i, y_i)$
$h$	VC dimension
$C$	regularization parameter in front of the empirical risk term
$\lambda$	regularization parameter in front of the regularizer
$\mathbf{w}$	weight vector
$b$	constant offset (or threshold)
$\alpha_i$	Lagrange multiplier or expansion coefficient
$\beta_i$	Lagrange multiplier
$\boldsymbol{\alpha}, \boldsymbol{\beta}$	vectors of Lagrange multipliers
$\xi_i$	slack variables
$\boldsymbol{\xi}$	vector of all slack variables
$Q$	Hessian of a quadratic program

### Miscellaneous

$I_A$	characteristic (or indicator) function on a set $A$ i.e., $I_A(x) = 1$ if $x \in A$ and 0 otherwise
$\delta_{ij}$	Kronecker $\delta$ ( $\delta_{ij} = 1$ if $i = j$ , 0 otherwise)
$\delta_x$	Dirac $\delta$ , satisfying $\int \delta_x(y)f(y)dy = f(x)$
$O(g(n))$	a function $f(n)$ is said to be $O(g(n))$ if there exist constants $C > 0$ and $n_0 \in \mathbb{N}$ such that $ f(n)  \leq Cg(n)$ for all $n \geq n_0$
$o(g(n))$	a function $f(n)$ is said to be $o(g(n))$ if there exist constants $c > 0$ and $n_0 \in \mathbb{N}$ such that $ f(n)  \geq cg(n)$ for all $n \geq n_0$
rhs/lhs	shorthand for “right/left hand side”
■	the end of a proof