Boosting with Multiple Sources

Corinna Cortes
Google Research
New York, NY 10011
corinna@google.com

Mehryar Mohri
Google & Courant Institute
New York, NY 10012
mohri@google.com

Dmitry Storcheus
Courant Institute & Google
New York, NY 10012
dstorcheus@google.com

Ananda Theertha Suresh
Google Research
New York, NY 10011
theertha@google.com

Abstract

We study the problem of learning accurate ensemble predictors, in particular boosting, in the presence of multiple source domains. We show that the standard convex combination ensembles in general cannot succeed in this scenario and adopt instead a domain-weighted combination. We introduce and analyze a new boosting algorithm, MULTIBOOST, for this scenario and show that it benefits from favorable theoretical guarantees. We also report the results of several experiments with our algorithm demonstrating that it outperforms natural baselines on multi-source text-based, image-based and tabular data. We further present an extension of our algorithm to the federated learning scenario and report favorable experimental results for that setting as well. Additionally, we describe in detail an extension of our algorithm to the multi-class setting, MCMULTIBOOST, for which we also report experimental results.

1 Motivation

Ensemble methods such as Bagging, AdaBoost, Stacking, error-correction techniques, Bayesian averaging, AdaNet or other adaptive methods for learning neural networks are general machine learning techniques used to combine several predictors to devise a more accurate one (Breiman, 1996; Freund and Schapire, 1997; Smyth and Wolpert, 1999; MacKay, 1991; Freund et al., 2004; Cortes et al., 2014; Kuznetsov et al., 2014; Cortes et al., 2017). These techniques are often very effective in practice and benefit from favorable margin-based learning guarantees (Schapire et al., 1997). These algorithms assume access to a training sample drawn from the target distribution. But, in many applications, the learner receives labeled data from multiple source domains that it seeks to use to find a good predictor for target domains, which are typically assumed to be mixtures of source distributions (Mansour et al., 2008, 2009a; Hoffman et al., 2018, 2021; Cortes et al., 2021; Muandet et al., 2013; Xu et al., 2014; Hoffman et al., 2012; Saito et al., 2019; Wang et al., 2019a). How can we generalize ensemble methods such as boosting to this scenario?

Several related problems have been tackled in previous work. In the special case of a single target domain, boosting solutions have been derived (Dai et al., 2007; Yao and Doretto, 2010). These algorithms have been further improved by boosting base predictors jointly with source features (or feature *views*) that are predictive in the target (Yuan et al., 2017; Cheng et al., 2013; Zhang et al., 2014; Xu and Sun, 2012, 2011). Such methods have been widely adopted in various domain adaptation scenarios for different types of data. For example, Huang et al. (2010, 2012) showed that by selecting a base learner jointly with a feature that is predictive across multiple domains at every boosting step, one can achieve higher accuracy than standard transfer learning methods. Moreover, the margin

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

provided by boosting-style algorithms can aid in transfer learning where target domain is unlabelled. Habrard et al. (2013) have developed an algorithm that jointly minimizes the source domain error and margin violation proportion on the target domain. Wang et al. (2019a) have demonstrated that boosting classifiers from different domains can be carried out online. Taherkhani et al. (2020) and Becker et al. (2013) have shown that multi-source boosting can be combined with Deep Neural Networks for multi-task learning on large scale datasets.

We give a more detailed discussion of this related prior work in Appendix A. However, as we demonstrate in this paper, several critical questions related to the presence of multiple domains have not been fully addressed by existing boosting methods from both the algorithmic and the theoretical point of view. First, what should be the form of the ensemble solutions? We show that, in general, the standard convex combinations used in much of previous work may not succeed in this problem (Section 2). Instead, we put forward Q-ensembles, which are convex combinations weighted by a domain classifier Q, that is, $\mathsf{Q}(k|x)$ is the conditional probability of domain k given input point x. This is inspired by similar Q-ensembles (Cortes, Mohri, Suresh, and Zhang, 2021) or distribution-weighted combinations (Mansour, Mohri, and Rostamizadeh, 2008, 2009a; Hoffman, Mohri, and Zhang, 2018, 2021) used in the context of multiple-source adaptation and crucially differentiates our work from that of past studies of ensemble methods. Our learning scenario strictly generalizes previous algorithms. In the special case of a single domain, the form of our ensemble solutions coincides with that of the familiar ensembles used in AdaBoost. In Appendix J, we further compare our results and guarantees with the multiple-source adaptation algorithms of (Hoffman, Mohri, and Zhang, 2018, 2021; Cortes, Mohri, Suresh, and Zhang, 2021), when using AdaBoost to derive a predictor for each domain.

Second, what should be the form of the objective? Unlike existing boosting methods which optimize for a specific target distribution or domain, we seek a solution that is accurate for any mixture of the source distributions, where the mixture weights may be constrained to be in a subset of the simplex. This conforms with the learning goal in this scenario where the target distribution is typically assumed to be a mixture of the source ones and is further inspired by the formulation of the *agnostic loss* in the related context of federated learning (Mohri, Sivek, and Suresh, 2019). Additionally, this formulation can be viewed as more risk-averse and robust, and it also ensures more favorable fairness properties, since it does not favor any distribution possibly biased towards some subset of the source domains. In Section 3, we introduce and analyze an algorithm, MULTIBOOST, that is based on such an objective. This further distinguishes our boosting algorithm from related prior work. In Appendix L, we further give a detailed description of a multi-class extension of our algorithm, MCMULTIBOOST, including its pseudocode, the discussion of some of its properties, and the results of several experiments.

Third, in Section 4, we present a theoretical analysis of our MULTIBOOST algorithm, including margin-based generalization bounds that hold for any target mixture of the source distributions. In Appendix F, we derive finer and more flexible learning bounds that can provide more favorable guarantees, in particular when the family of target mixtures is more constrained.

Fourth, in Section 5, we describe FEDMULTIBOOST, an extension of our MULTIBOOST algorithm adapted to the distributed learning scenario of *federated learning*, which is increasingly important in a wide array of applications (Kairouz et al., 2021), including healthcare (Brisimi et al., 2018). FEDMULTIBOOST allows the server to train ensemble models that benefit from clients' data, while the data remains with clients. We provide a detailed description of our algorithm, as well as a brief analysis of the communication costs, which are critical in this scenario. Appendix G contains experimental results comparing FEDMULTIBOOST with several baselines.

Finally, in Section 6, we report the results of extensive experiments with our MULTIBOOST algorithm. We start with a detailed description of the learning scenario we consider.

2 Learning scenario

Let $\mathcal{X}$ denote the input space and $\mathcal{Y} = \{-1, +1\}$ the output space associated to binary classification. We consider a scenario where the learner receives labeled samples from p source domains, each defined by a distribution $\mathcal{D}_k$ over $\mathcal{X} \times \mathcal{Y}$, $k \in [p]$. We denote by $S_k = ((x_{k,1}, y_{k,1}), \ldots, (x_{k,m_k}, y_{k,m_k})) \in (\mathcal{X} \times \mathcal{Y})^{m_k}$ the labeled sample of size m_k received from source k, which is drawn i.i.d. from $\mathcal{D}_k$. For any function $f \colon \mathcal{X} \to \mathbb{R}$ and distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, let

$\mathcal{L}(\mathcal{D}, f)$ be the expected loss of f, that is $\mathcal{L}(\mathcal{D}, f) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(f(x), y)]$, where ℓ is the binary loss $\ell(f(x), y) = \mathbb{I}(yf(x) \leq 0)$.

For any $k \in [p]$, let $\mathcal{H}_k$ denote a hypothesis set of functions mapping from $\mathcal{X}$ to $[-1, +1]$, $|\mathcal{H}_k| = N_k$. The objective of the learner is to find a predictor f that is accurate for *any* target distribution $\mathcal{D}_\lambda$ that is a mixture of the source distributions, where λ may be in a subset Λ of the simplex. Thus, $\mathcal{D}_\lambda$ can written as $\mathcal{D}_\lambda = \sum_{k=1}^p \lambda_k \mathcal{D}_k$, with $\lambda = (\lambda_1, \ldots, \lambda_p) \geq 0$ and $\sum_{k=1}^p \lambda_k = 1$. This leads to the following *agnostic loss* of a predictor f (Mohri et al., 2019):

$$\mathcal{L}(\mathcal{D}_\Lambda, f) = \max_{\lambda\in\Lambda} \mathcal{L}(\mathcal{D}_\lambda, f). \tag{1}$$

To come up with a predictor f, the learner seeks an ensemble of functions from the base classes $\mathcal{H}_k$. What should be the form of such ensembles? A natural solution is a convex combinations of the form

$$f = \sum_{l=1}^{p} \sum_{j=1}^{N_l} \alpha_{l,j} h_{l,j}, \tag{2}$$

where $h_{l,j} \in \mathcal{H}_l$ and $\alpha_{l,j} \geq 0$, $\sum_{l=1}^p \sum_{j=1}^{N_l} \alpha_{l,j} = 1$, as in prominent algorithms such as BAGGING (Breiman, 1999) or ADABOOST (Freund and Schapire, 1997). It is not hard to show, however, that for some distributions, any such convex combination of base predictors would lead to a poor solution.

Proposition 1. *There exist distributions $\mathcal{D}_1$ and $\mathcal{D}_2$ and hypotheses h_1 and h_2 with $\mathcal{L}(\mathcal{D}_1, h_1) = 0$ and $\mathcal{L}(\mathcal{D}_2, h_2) = 0$ such that $\mathcal{L}\big(\frac{1}{2}(\mathcal{D}_1 + \mathcal{D}_2), \alpha h_1 + (1-\alpha)h_2\big) \geq \frac{1}{2}$ for any $\alpha \in [0, 1]$,*

This result is similar to (Mansour et al., 2008, Theorem 1) or similar statements in (Mansour et al., 2008, 2009a; Hoffman et al., 2018, 2021; Cortes et al., 2021), however, there are some technical differences. In particular, we are considering here a binary classification loss and not the absolute loss. The proof is given in Appendix B.

Note that in the example of the proposition, the base predictors h_1, h_2 are both perfect for their respective domains. Nevertheless, unlike the common case of a single source domain, standard convex combinations in general may perform poorly. Thus, we will consider instead the following form for the ensembles of base predictors:

$$\forall x \in \mathcal{X}, \quad f(x) = \sum_{l=1}^{p} \sum_{j=1}^{N_l} \alpha_{l,j} \mathsf{Q}(l|x) h_{l,j}(x), \tag{3}$$

where $\mathsf{Q}(l|x)$ denotes the conditional probability of domain l given x. This helps us account for the fact that the learner combines base predictors from different domains. For a given point x, experts from the domains where x is more likely to appear are allocated more weight in the voted combination. The following result further substantiates the hope for this method to succeed.

Proposition 2. *For the same distributions $\mathcal{D}_1$ and $\mathcal{D}_2$ and hypotheses h_1 and h_2 as in Proposition 1, the equality $\mathcal{L}(\mathcal{D}_\lambda, (\alpha\mathsf{Q}(1|\cdot)h_1 + (1-\alpha)\mathsf{Q}(2|\cdot)h_2)) = 0$ holds for any $\lambda \in \Delta$ and any $\alpha \in (0, 1)$.*

The proof is also given in Appendix B. Thus, our Q-ensembles can be substantially more effective in the multiple-source adaptation problem considered. Note that, in the special case of a single domain, the Q-combinations coincide with standard convex combinations, since $\mathsf{Q}(k|x) = 1$ for all x.

As in the standard boosting scenario, we will adopt a weak-learning assumption. However, our assumption here must hold for each source domain: for each domain $k \in [p]$ and any distribution D over S_k, there exists a base classifier $h \in \mathcal{H}_k$ such that the weighted loss of h is γ-better than random: $\frac{1}{2}[1 - \mathbb{E}_{i\sim\mathsf{D}}[y_{k,i}h(x_{k,i})] \leq \frac{1}{2} - \gamma$, for some edge value $\gamma > 0$. This is equivalent to the existence of a weak-learning algorithm for each domain, which is a mild assumption. As in the standard boosting scenario, this corresponds to the existence of a a good *rule of thumb* for each domain. Unlike the standard scenario, however, here, we seek a Q-ensemble and further require that ensemble to be accurate for any target mixture $\mathcal{D}_\lambda$, $\lambda \in \Lambda$. Note that the definition of the hypothesis set $\mathcal{H}_k$ and the weak-learning assumption for domain k are intimately related: the elements of $\mathcal{H}_k$ are typically simple predictors sufficient to guarantee the weak-learning assumption for that domain.

In the next section, we present an algorithm, MULTIBOOST, for deriving an accurate Q-ensemble for any target mixture domain that belongs to the convex combination of the source domains. In Appendix L, we further present a multi-class extension of our algorithm, MCMULTIBOOST.

3 Algorithm

Let Φ be a convex, increasing and differentiable function such that $u \mapsto \Phi(-u)$ upper-bounds the binary loss $u \mapsto 1_{u \leq 0}$. Φ could be the exponential function as in AdaBoost or the logistic function, as in logistic regression. Using Φ to upper-bound the agnostic loss leads to the following objective function for an ensemble f defined by (3) for any $\alpha = (\alpha_{l,j})_{(l,j) \in [p] \times [N_l]} \geq 0$:

$$F(\alpha) = \max_{\lambda \in \Lambda} \sum_{k=1}^{p} \frac{\lambda_k}{m_k} \sum_{i=1}^{m_k} \Phi\left(-y_{k,i} \sum_{l=1}^{p} \sum_{j=1}^{N_l} \alpha_{l,j} \mathsf{Q}(l|x_{k,i}) h_{l,j}(x_{k,i}) \right). \tag{4}$$

Since a convex function composed with an affine function is convex and a sum of convex functions is convex, F is convex as the maximum of a set of convex functions. In this section, we will consider the case where the set Λ coincides with the full simplex, that is $\Lambda = \Delta$. F can then be expressed more simply as $F = \max_{k=1}^{p} F_k$, with $F_k(\alpha) = \frac{1}{m_k} \sum_{i=1}^{m_k} \Phi\big(-y_{k,i} \sum_{l=1}^{p} \sum_{j=1}^{N_l} \alpha_{l,j} \mathsf{Q}(l|x_{k,i}) h_{l,j}(x_{k,i})\big)$.

It is known that AdaBoost coincides with coordinate descent applied to an exponential loss objective function (Duffy and Helmbold, 1999; Schapire and Freund, 2012; Mohri et al., 2018). Our algorithm similarly applies coordinate descent to the objective just described, as with other boosting-type algorithms (Mason et al., 1999; Cortes et al., 2014; Kuznetsov et al., 2014). While convex, F is not differentiable and, in general, coordinate descent may not succeed in such cases (Tseng, 2001; Luo and Tseng, 1992). This is because the algorithm may be *stuck* at a point where no progress is possible along any of the axis directions, while there exists a favorable descent along some other direction. However, we will show that, under the weak-learning assumption we adopted, at any point $\boldsymbol{\alpha}$ and for each active function F_k, that is F_k such that $F_k(\boldsymbol{\alpha}) = F(\boldsymbol{\alpha})$, there exists a coordinate direction along which a descent progress is possible. We will assume that these directions are also descent directions for F. More generally, it suffices in fact that one such direction admits this guarantee. Alternatively, one can replace the maximum with a *soft-max*, that is the $(x_1, \ldots, x_k) \mapsto \frac{1}{\mu} \log(\sum_{k=1}^{p} e^{\mu x_k})$ for $\mu > 0$, which leads to a continuously differentiable function with a close approximation for μ sufficiently large.

Description. Let α_{t-1} denote the value of the parameter vector $\alpha = (\alpha_{l,j})$ at the end of the $(t-1)$th iteration and f_{t-1} the corresponding function: $f_{t-1} = \sum_{l=1}^{p} \sum_{j=1}^{N_l} \alpha_{t-1,l,j} \mathsf{Q}(l|\cdot)\, h_{l,j}$. Coordinate descent at iteration t consists of choosing a direction $\mathbf{e}_{q,r}$ corresponding to base classifier $h_{q,r}$ and a step value $\eta > 0$ to minimize $F(\alpha_{t-1} + \eta\, \mathbf{e}_{q,r})$.

To select a direction, we consider the subdifferential of F along any $\mathbf{e}_{q,r}$. Since functions F_k are differentiable, by the subdifferential calculus for the maximum of functions, the subdifferential of F at α_{t-1} along the direction $\mathbf{e}_{q,r}$ is given by:

$$\partial F(\alpha_{t-1}, \mathbf{e}_{q,r}) = \operatorname{conv}\{F'_k(\alpha_{t-1}, \mathbf{e}_{q,r}) \colon k \in \mathcal{K}_t\},$$

where $F'_k(\alpha_{t-1}, \mathbf{e}_{q,r})$ is the directional derivative of F_k at α_{t-1} along the direction $\mathbf{e}_{q,r}$ and where $\mathcal{K}_t = \{k \in [p] \colon F_k(\alpha_{t-1}) = F(\alpha_{t-1})\}$. We will therefore consider the direction $\mathbf{e}_{q,r}$ with the largest absolute directional derivative value $|F'_k(\alpha_{t-1}, \mathbf{e}_{q,r})|$, $k \in \mathcal{K}_t$, but will restrict ourselves to $q = k$ since, as we shall see, that will suffice to guarantee a non-zero directional gradient. To do so, we will express $F'_k(\alpha_{t-1}, \mathbf{e}_{k,r})$ in terms of the distribution $\mathsf{D}_t(k, \cdot)$ over S_k defined by $\mathsf{D}_t(k,i) = \frac{\Phi'(-y_{k,i} f_{t-1}(x_{k,i}))}{Z_{t,k}}$, with $Z_{t,k} = \sum_{i=1}^{m_k} \Phi'(-y_{k,i} f_{t-1}(x_{k,i}))$, for all $i \in [m_k]$:

$$F'_k(\alpha_{t-1}, h_{k,r}) = \frac{1}{m_k} \sum_{i=1}^{m_k} -y_{k,i} \mathsf{Q}(k|x_{k,i}) h_{k,r}(x_{k,i}) \Phi'\big(-y_{k,i} f_{t-1}(x_{k,i})\big) = \frac{Z_{t,k}}{m_k} [2\epsilon_{t,k,r} - 1],$$

where $\epsilon_{t,k,r} = \frac{1}{2}\big[1 - \mathbb{E}_{i \sim \mathsf{D}_t(k,\cdot)}[y_{k,i} \mathsf{Q}(k|x_{k,i}) h_{k,r}(x_{k,i})]\big]$ denotes the *weighted error* of $\mathsf{Q}(k|\cdot) h_{k,r}$. For any $s \in [m_k]$, since $x_{k,s}$ is a sample drawn from $\mathcal{D}_k$, we have $\mathsf{Q}(k|x_{k,s}) > 0$ and therefore we have: $\overline{\mathsf{Q}}_{t,k} = \sum_{s=1}^{m_k} \mathsf{D}_t(k,s) \mathsf{Q}(k|x_{k,s}) > 0$. Thus, we can write $\mathbb{E}_{i \sim \mathsf{D}_t(k,\cdot)}[y_{k,i} \mathsf{Q}(k|x_{k,i}) h_{k,r}(x_{k,i})]$ as

$$\sum_{i=1}^{m_k} \frac{\mathsf{D}_t(k,i) \mathsf{Q}(k|x_{k,i}) y_{k,i} h_{k,r}(x_{k,i})}{\overline{\mathsf{Q}}_{t,k}} \overline{\mathsf{Q}}_{t,k}, = \mathop{\mathbb{E}}_{i \sim \mathsf{D}'_t(k,\cdot)}[y_{k,i} h_{k,r}(x_{k,i})]\, \overline{\mathsf{Q}}_{t,k},$$

where $\mathsf{D}'_t(k,i) = \frac{\mathsf{D}_t(k,i) \mathsf{Q}(k|x_{k,i})}{\overline{\mathsf{Q}}_{t,k}}$. By our weak-learning assumption, there exists $r \in N_k$ such that $\mathbb{E}_{i \sim \mathsf{D}'_t(k,\cdot)}[y_{k,i} h_{k,r}(x_{k,i})] \geq \gamma > 0$. For that choice of r, we have $\epsilon_{t,k,r} < \frac{1}{2} - \overline{\gamma}$, with $\overline{\gamma} = \gamma \overline{\mathsf{Q}}_{t,k} > 0$.

In view of that, it suffices for us to search along the directions $h_{k,r}$ and we do not need to consider the directional derivative of F_k along directions $h_{q,r}$ with $q \neq k$. The direction chosen by our coordinate descent algorithm is thus defined by: $\operatorname{argmax}_{k \in \mathcal{K}_t, r \in [N_k]} \frac{Z_{t,k}}{m_k}[1 - 2\epsilon_{t,k,r}]$. Given the direction $\mathbf{e}_{k,r}$, the optimal step value η is $\operatorname{argmin}_{\eta > 0} F(\alpha_{t-1} + \eta \mathbf{e}_{k,r})$. The pseudocode of our algorithm, MULTIBOOST, is provided in Figure 1. In the most general case, η can be found via a line search or other numerical methods.

Step size. In some special cases, the line search can be executed using a simpler expression by using an upper bound on $F(\alpha_{t-1} + \eta \mathbf{e}_{k,r})$. Using the convexity of Φ, since $y_{l,i}\mathsf{Q}(k|x_{l,i})h_{k,r}(x_{l,i}) = \frac{1+y_{l,i}\mathsf{Q}(k|x_{l,i})h_{k,r}(x_{l,i})}{2} \cdot (+1) + \frac{1-y_{l,i}\mathsf{Q}(k|x_{l,i})h_{k,r}(x_{l,i})}{2} \cdot (-1)$, the following holds for all $\eta \in \mathbb{R}$:

$$\Phi(-y_{l,i} f_{t-1}(x_{l,i}) - \eta\, y_{l,i}\mathsf{Q}(k|x_{l,i})h_{k,r}(x_{l,i})) \leq \frac{1+y_{l,i}\mathsf{Q}(k|x_{l,i})h_{k,r}(x_{l,i})}{2}\Phi(-y_{l,i} f_{t-1}(x_{l,i}) - \eta) + \frac{1-y_{l,i}\mathsf{Q}(k|x_{l,i})h_{k,r}(x_{l,i})}{2}\Phi(-y_{l,i} f_{t-1}(x_{l,i}) + \eta).$$

In the case of exponential and logistic functions, the following upper bounds can then be derived.

Lemma 1. *For any $k \in [p]$, the following upper bound holds when Φ is the exponential or the logistic function:*

$$F(\alpha_{t-1} + \eta \mathbf{e}_{k,r}) \leq \max_{l \in [p]} \frac{Z_{t,l}}{m_l}\left[(1 - \epsilon_{t,l,k,r})e^{-\eta} + \epsilon_{t,l,k,r}e^{\eta}\right],$$

where $\epsilon_{t,l,k,r} = \frac{1}{2}\left[1 - \mathbb{E}_{i \sim \mathsf{D}_t(l,\cdot)}[y_{l,i}\mathsf{Q}(k|x_{l,i})h_{k,r}(x_{l,i})]\right]$.

For any k, function $\eta \mapsto (1 - \epsilon_{t,l,k,r})e^{-\eta} + \epsilon_{t,l,k,r}e^{\eta}$ reaches its minimum for $\eta = \frac{1}{2}\log\frac{1-\epsilon_{t,l,k,r}}{\epsilon_{t,l,k,r}}$. When the maximum is achieved for $l = k$, the solution coincides with the familiar expression of the step size $\eta_t = \frac{1}{2}\log\frac{1-\epsilon_{t,k,r}}{\epsilon_{t,k,r}}$ used in ADABOOST.

Q-function. The conditional probability functions $\mathsf{Q}(k|\cdot)$ are crucial to the definition of our algorithm. As pointed out earlier, Q-ensembles can help achieve accurate solutions in some cases that cannot be realized using convex combinations. Furthermore, for any $k \in [p]$, since $\mathsf{D}_t(k,\cdot)$ is a distribution over the sample S_k, it is natural to assume that for any $j \neq k$ we have $\mathbb{E}_{s \sim \mathsf{D}_t(k,\cdot)}[\mathsf{Q}(k|x_{k,s})] \geq \mathbb{E}_{s \sim \mathsf{D}_t(k,\cdot)}[\mathsf{Q}(j|x_{k,s})]$. This implies the following lower bound: $\mathbb{E}_{s \sim \mathsf{D}_t(k,\cdot)}[\mathsf{Q}(k|x_{k,s})] \geq \frac{1}{p}$, which in turn implies $\overline{\gamma} \geq \frac{\gamma}{p}$, since for any $x \in \mathcal{X}$, $\sum_{j=1}^{p}\mathsf{Q}(j|x) = 1$. In the special case where all domains coincide, we have $\mathsf{Q}(k|x_{k,s}) = \frac{1}{p}$ for all s and this lower bound is reached. At another extreme, when all domains admit distinct supports, we have $\mathsf{Q}(k|x_{k,s}) = 1$ for all $s \in [m_k]$ and thus $\overline{\gamma} = \gamma$.

MULTIBOOST$(S_1, \ldots, S_p)$
1 $\alpha_0 \leftarrow 0$
2 **for** $t \leftarrow 1$ **to** T **do**
3 $\quad f_{t-1} \leftarrow \sum_{l=1}^{p}\sum_{j=1}^{N_l}\alpha_{t-1,l,j}\mathsf{Q}(l|\cdot)\,h_{l,j}$
4 $\quad \tilde{\Phi}_k \leftarrow \frac{1}{m_k}\sum_{i=1}^{m_k}\Phi(-y_{k,i}f_{t-1}(x_{k,i}))$
5 $\quad \mathcal{K}_t \leftarrow \left\{k\colon k \in \operatorname{argmax}_{k \in [p]}\tilde{\Phi}_k\right\}$
6 $\quad$ **for** $k \in \mathcal{K}_t$ **do**
7 $\quad\quad Z_{t,k} \leftarrow \sum_{i=1}^{m_k}\Phi'\big(-y_{k,i}f_{t-1}(x_{k,i})\big)$
8 $\quad\quad$ **for** $i \leftarrow 1$ **to** m_k **do**
9 $\quad\quad\quad \mathsf{D}_t(k,i) \leftarrow \frac{\Phi'(-y_{k,i}f_{t-1}(x_{k,i}))}{Z_{t,k}}$
10 $\quad (k,r) \leftarrow \operatorname{argmax}_{k \in \mathcal{K}_t, r \in [N_k]}\frac{Z_{t,k}}{m_k}[1 - 2\epsilon_{t,k,r}]$
11 $\quad \eta_t \leftarrow \operatorname{argmin}_{\eta > 0} F(\alpha_{t-1} + \eta \mathbf{e}_{k,r})$
12 $\quad \alpha_t \leftarrow \alpha_{t-1} + \eta_t \mathbf{e}_{k,r}$
13 $f \leftarrow \sum_{l=1}^{p}\sum_{j=1}^{N_l}\alpha_{T,l,j}\mathsf{Q}(l|\cdot)h_{l,j}$
14 **return** f

Figure 1: Pseudocode of the MULTIBOOST algorithm. $\epsilon_{t,k,r} = \frac{1}{2}\left[1 - \mathbb{E}_{i \sim \mathsf{D}_t(k,\cdot)}[y_{k,i}\mathsf{Q}(k|x_{k,i})h_{k,r}(x_{k,i})]\right]$ denotes the weighted error of $\mathsf{Q}(k|\cdot)h_{k,r}$.

In practice, we do not have access to the true conditional probabilities $\mathsf{Q}(k|\cdot)$. Instead, we can derive accurate estimates $\widehat{\mathsf{Q}}(k|\cdot)$ using large unlabeled samples from the source domains, the *label* used for training being simply the domain index. This can be done using algorithms such as conditional maximum entropy models (Berger et al., 1996) (or multinomial logistic regression), which benefit from strong theoretical guarantees (Mohri et al., 2018, Chapter 13), or other rich models based on neural networks.

Other variants of MULTIBOOST. In Appendix D, we present and discuss other variants of our algorithm, including their convergence guarantees.

4 Theoretical analysis

In this section, we present a theoretical analysis of our algorithm, including margin-based learning bounds for multiple-source Q-ensembles.

For any $k \in [p]$, define the family $\mathcal{G}_k$ as follows: $\mathcal{G}_k = \{\mathsf{Q}(k|\cdot)\, h \colon h \in \mathcal{H}_k\}$. Then, the family of ensemble functions $\mathcal{F}$ that we consider can be defined as $\mathcal{F} = \operatorname{conv}(\bigcup_{k=1}^p \mathcal{G}_k)$. For any $\lambda \in \Delta$, let $\overline{\mathcal{D}}_\lambda$ be the distribution defined by $\mathcal{D}_\lambda = \sum_{k=1}^p \lambda_k \widehat{\mathcal{D}}_k$, where $\widehat{\mathcal{D}}_k$ is the empirical distribution associated to an i.i.d. sample S_k drawn from $\mathcal{D}_k$. $\overline{\mathcal{D}}_\lambda$ is distinct from the distribution associated to $\mathcal{D}_\lambda$. We seek to derive margin-based bounds for ensemble functions $f \in \mathcal{F}$ with respect to target mixture distributions $\mathcal{D}_\lambda$, while the empirical data is from multiple source distributions $\mathcal{D}_k$, or from $\overline{\mathcal{D}}_\lambda$. Thus, these guarantees differ from standard learning bounds where the source and target distribution coincide.

For any distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$ and $\rho > 0$, let $\mathcal{L}_\rho(\mathcal{D}, f)$ denote the expected ρ-margin loss of f with respect to $\mathcal{D}$: $\mathcal{L}_\rho(\mathcal{D}, f) = \mathbb{E}_{(x,y)\sim\mathcal{D}_k}[\mathbb{I}(yf(x) \leq \rho)]$. We denote by $\mathfrak{R}_m(\mathcal{F})$ the Rademacher complexity of $\mathcal{F}$ for samples $S = (x_1, \ldots, x_m)$ of size m, defined by $\mathfrak{R}_m(\mathcal{F}) = \mathbb{E}_{S\sim\mathcal{D}^m}[\sup_{f\in\mathcal{F}} \sum_{i=1}^m \sigma_i f(x_i)]$, with σ_is independent random variables taking values in $\{-1, +1\}$. The $\boldsymbol{\sigma}$ following gives a margin-bound for our multiple-source learning with Q-ensembles.

Theorem 1. *Fix $\rho > 0$. Then, for any $\delta > 0$, with probability at least $1 - \delta$ over the draw of a sample $S = (S_1, \ldots, S_p) \sim \mathcal{D}_1^{m_1} \otimes \cdots \otimes \mathcal{D}_p^{m_p}$, the following inequality holds for all ensemble functions $f = \sum_{t=1}^T \alpha_t \mathsf{Q}(k_t|\cdot) h_t \in \mathcal{F}$ and all $\lambda \in \Delta$:*

$$\mathcal{L}(\mathcal{D}_\lambda, f) \leq \mathcal{L}_\rho(\overline{\mathcal{D}}_\lambda, f) + \sum_{k=1}^p \lambda_k \left[\frac{2}{\rho} \max_{l\in[p]} \mathfrak{R}_{m_k}(\mathcal{H}_l) + \sqrt{\frac{\log \frac{p}{\delta}}{2m_k}} \right]. \tag{5}$$

Note that the complexity term depends only on the Rademacher complexities of the families of based predictors $\mathcal{H}_l$ and does not involve the domain classification function Q. Theorem 1 improves upon previous known bounds of Mohri et al. (2019). In particular, it provides a dimension-independent margin guarantee for the zero-one loss, while the bound of Mohri et al. (2019), when applied to the zero-one loss, is dimension-dependent (Mohri et al., 2019, Lemma 3). Additionally, our bound holds for Q-ensembles. Our margin bound can be straightforwardly generalized to hold uniformly for all $\rho \in (0, 1)$, at the cost of an additional mild term depending on $\log \log_2(2/\rho)$ (see Theorem 4 in Appendix E). For the ensemble functions returned by our algorithm, the bound applies to $x \mapsto \frac{f(x)}{\|\alpha\|_1}$. These learning guarantees suggest choosing $\alpha \geq 0$ and ρ to minimize the following:

$$\max_{\lambda\in\Lambda} \sum_{k=1}^p \lambda_k \left\{ \mathop{\mathbb{E}}_{(x,y)\sim S_k} \left[\mathbb{I}\left(y \sum_{t=1}^T \frac{\alpha_t}{\|\alpha\|_1} \mathsf{Q}(k_t|\cdot) h_t(x) \leq \rho \right) \right] + \frac{r_k}{\rho} \right\},$$

where $r_k = 2 \max_{l\in[p]} \mathfrak{R}_{m_k}(\mathcal{H}_l)$. Choosing $\frac{1}{\rho} = \|\alpha\|_1$ and upper bounding the binary loss using a convex function Φ, the problem can then be equivalently cast as the following unconstrained minimization, for example in the case of the exponential function:

$$\min_{\alpha\geq 0} \max_{\lambda\in\Lambda} \sum_{k=1}^p \lambda_k \left\{ \mathop{\mathbb{E}}_{(x,y)\sim S_k} \left[\exp\left(-y \sum_{t=1}^T \alpha_t \mathsf{Q}(k_t|\cdot) h_t(x) \right) \right] + r_k \|\alpha\|_1 \right\}.$$

Thus, the learning guarantees justify a posteriori an ℓ_1-regularized version of our algorithm. It is straightforward to derive that extension of our algorithm and its pseudocode. This is similar to the extension to the ℓ_1-AdaBoost (Rätsch et al., 2001) of the original unregularized AdaBoost (Freund and Schapire, 1997). Let us add that finer learning guarantees such as those for Deep Boosting (Cortes et al., 2014) ones can be derived similarly here, which would lead to a weighted ℓ_1-regularization, where the weights are the Rademacher complexities of the families $\mathcal{H}_k$. For the sake of simplicity, we do not detail that extension. Let us point out, however, that such an extension would also lead to an algorithm generalizing AdaNet (Cortes et al., 2017) to multiple sources.

In some applications, prior knowledge can be used to constrain Λ to a strict subset of the simplex Δ. Finer margin-based learning guarantees can be derived to cover that scenario. Let $\overline{\Lambda}$ be the set of

vertices of a subsimplicial cover of Λ, that is a decomposition of a cover of Λ into subsimplices and let $\overline{\Lambda}_\epsilon$ be the set of λs ϵ-close to $\overline{\Lambda}$ in ℓ_1-distance. Then, the following guarantees hold.

Theorem 2. *Fix $\rho > 0$ and $\epsilon > 0$. Then, for any $\delta > 0$, with probability at least $1 - \delta$ over the draw of a sample $S = (S_1, \ldots, S_p) \sim \mathcal{D}_1^{m_1} \otimes \cdots \otimes \mathcal{D}_p^{m_p}$, each of the following inequalities holds:*
1. for all ensemble functions $f = \sum_{t=1}^T \alpha_t \mathsf{Q}(k_t|\cdot) h_t \in \mathcal{F}$ and all $\lambda \in \overline{\Lambda}_\epsilon$:

$$\mathcal{L}(\mathcal{D}_\lambda, h) \leq \mathcal{L}_\rho(\overline{\mathcal{D}}_\lambda, h) + \frac{2}{\rho} \max_{r \in [p]} \mathfrak{R}_{\mathbf{m}}(\mathcal{H}_r, \lambda) + \frac{3\epsilon}{2} + \sqrt{\sum_{k=1}^p \frac{\lambda_k^2}{2m_k} \log \frac{|\overline{\Lambda}|}{\delta}}. \tag{6}$$

2. for all ensemble functions $f = \sum_{t=1}^T \alpha_t \mathsf{Q}(k_t|\cdot) h_t \in \mathcal{F}$ and all $\lambda = \sum_{k=1}^p \mu_k \beta_k \in \operatorname{conv}(\overline{\Lambda})$:

$$\mathcal{L}(\mathcal{D}_\lambda, h) \leq \mathcal{L}_\rho(\overline{\mathcal{D}}_\lambda, h) + \frac{2}{\rho} \sum_{l=1}^p \mu_k \max_{r \in [p]} \mathfrak{R}_{\mathbf{m}}(\mathcal{H}_r, \beta_l) + \sum_{k=1}^p \mu_k \sqrt{\sum_{l=1}^p \frac{\beta_{l,k}^2}{2m_k} \log \frac{|\overline{\Lambda}|}{\delta}}. \tag{7}$$

The proof of the theorem and further discussion are presented in Appendix F. Here, $\mathfrak{R}_{\mathbf{m}}(\mathcal{H}_r, \lambda)$ is the λ-weighted Rademacher complexity of $\mathcal{H}_r$ (see Appendix F). For Λ a strict subset of the simplex, these learning bounds suggest an algorithm with a regularization term of the form $\sum_{k=1}^p \lambda_k^2 / m_k$. Our analysis and learning guarantees can be similarly extended to cover the multi-class setting.

5 Federated MULTIBOOST algorithm

In this section, we present an extension of our algorithm to the *federated learning* scenario, called FEDMULTIBOOST. The pseudocode is given in Figure 2. Federated learning is a distributed learning scenario where a global model is trained at the server level, based on data remaining at clients (Konečnỳ et al., 2016b,a; McMahan et al., 2017; Yang et al., 2019). Clients are typically mobile phones, network sensors, or other IoT devices. While the data remains at the clients, the trained model significantly benefits from user data, as demonstrated, for example, in next word prediction (Hard et al., 2018; Yang et al., 2018) and healthcare applications (Brisimi et al., 2018). We refer the reader to (Kairouz et al., 2021) for a more detailed survey of federated learning.

A widely used algorithm in this scenario is *federated averaging*, where the server trains the global model via stochastic updates from the clients (McMahan et al., 2017). It has been argued by Mohri et al. (2019), however, that minimizing the expected loss with respect to a particular training distribution, as done by standard federated learning algorithms, may be risk-prone and benefit only a specific subset of clients. This compromises fairness, which is one of the critical questions in federated learning, where clients are often heterogeneous (Bickel et al., 1975; Hardt et al., 2016; Abay et al., 2020; Li et al., 2020). This issue can be further

FEDMULTIBOOST$(S_1, S_2, \ldots, S_n)$
1 $\alpha_0 \leftarrow 0$
2 **for** $t \leftarrow 1$ **to** T **do**
3 $\quad f_{t-1} \leftarrow \sum_{l=1}^p \sum_{j=1}^{N_l} \alpha_{t-1,l,j} \mathsf{Q}(l|\cdot)\, h_{l,j}$
4 $\quad C_{t,k} \leftarrow$ SELECTCLIENT(k, r)
5 $\quad$ **for** $k \leftarrow 1$ **to** p **do**
6 $\quad\quad$ **for** $c \in C_{t,k}$ **do**
7 $\quad\quad\quad \tilde{\Phi}_{k,c} \leftarrow \frac{1}{m_c} \sum_{i=1}^{m_c} \Phi(-y_{c,i} f_{t-1}(x_{c,i}))$
8 $\quad\quad\quad Z_{t,c} \leftarrow \sum_{i=1}^{m_c} \Phi'\big(-y_{c,i} f_{t-1}(x_{c,i})\big)$
9 $\quad\quad \mathcal{K}_t \leftarrow \Big\{k \in [p] \colon k \in \operatorname{argmax}_{k \in [p]} \sum_{c \in C_{t,k}} \tilde{\Phi}_{k,c}\Big\}$
10 $\quad\quad Z_{t,k} \leftarrow \sum_{c \in C_{t,k}} Z_{t,c}, \forall k \in [p]$
11 $\quad\quad \mathcal{H}_{t,k} \leftarrow$ SELECTBASECLASSIFIER(k, s)
12 $\quad\quad$ **for** $k \in \mathcal{K}_t$ **do**
13 $\quad\quad\quad$ **for** $c \in C_{t,k}$ **do**
14 $\quad\quad\quad\quad$ **for** $i \leftarrow 1$ **to** m_c **do**
15 $\quad\quad\quad\quad\quad \mathsf{D}_t(c,i) \leftarrow \frac{\Phi'(-y_{c,i} f_{t-1}(x_{c,i}))}{Z_{t,k}}$
16 $\quad\quad\quad\quad \beta_{t,c,r} \leftarrow [1 - 2\epsilon_{t,c,r}]$
17 $\quad\quad (k, r) \leftarrow \operatorname*{argmax}_{k \in \mathcal{K}_t, r \in \mathcal{H}_{t,k}} \frac{Z_{t,k}}{\sum_{c \in C_{t,k}} m_c} \sum_{c \in C_{t,k}} \beta_{t,c,r}$
18 $\quad\quad \eta_t \leftarrow \eta_0 / \sqrt{t}$
19 $\quad\quad \alpha_t \leftarrow \alpha_{t-1} + \eta_t \mathbf{e}_{k,r}$
20 $f \leftarrow \sum_{l=1}^p \sum_{j=1}^{N_l} \alpha_{T,l,j} \mathsf{Q}(l|\cdot) h_{l,j}$
21 **return** f

Figure 2: Pseudocode of the FEDMULTIBOOST algorithm. $\epsilon_{t,c,r}$ denotes the weighted error of $\mathsf{Q}(k|\cdot)h_{k,r}$, $\epsilon_{t,c,r} = \frac{1}{2}\big[1 - \mathbb{E}_{i \sim \mathsf{D}_t(c,\cdot)}[y_{c,i} \mathsf{Q}(k|x_{c,i}) h_{k,r}(x_{c,i})]\big]$, where client c belongs to domain k. cobalt steps are carried out by the server, red on the clients.

accentuated when the set of participating clients during inference may significantly differ from those used in training. Such cases may occur, for example, when clients loose access to network connection during training, which results in distinct training and test distributions.

To overcome these problems, we adopt the *agnostic federated learning* approach suggested by Mohri et al. (2019) and Ro et al. (2021). Following the client-partition approach of Ro et al. (2021), let each client belong to one of p domains. The domains can be, for example, based on the type of device or their geographical location. In this setting, the global model is optimized for all convex combinations of domain distributions as in the objective of MULTIBOOST (Equation 4), by taking the maximum over the mixtures weights λ. This ensures that the solution obtained is risk-averse and reduces the worst-case mismatch between the inference and training distributions. We propose FEDMULTIBOOST, a communication-efficient modification of MULTIBOOST, that can overcome the communication bottleneck in federated learning (Konečnỳ et al., 2016b).

Before presenting our algorithm, we first describe the learning scenario and the notation we adopt. Let n be the number of clients, which in the cross-device setting can be in the order of millions. Let m_c denote the number of samples in client c and $(x_{c,i}, y_{c,i})$ denote the i^{th} sample of client c. The server has access to a set of base-predictor classes $\mathcal{H}_k$ for $k \in [p]$ and a domain classifier Q that assigns for each sample x, the likelihood of it belonging to each of the p domains. The base classifiers can be obtained by training on public data or training on client data with federated learning. The domain classifier Q can be trained using unlabeled client data.

Additional discussion of the algorithm as well as experimental results can be found in Appendix G. At each round, the algorithm randomly selects r clients and s base classifiers from each domain. Next, it chooses the best classifier out of the previously selected subset, based on the data from the subsampled clients. Since FEDMULTIBOOST operates only on a small set of base classifiers at each round, the algorithm is communication-efficient. At round t, the server needs to send Q, f_{t-1}, and s base classifiers to the selected clients. Hence, the server-to-client communication cost at round t is $\mathcal{O}((t+s)K + K')$, where K is the number of parameters in a single base classifier and K' is the number of parameters required to specify Q. Furthermore, the clients communicate only the aggregate statistics $\tilde{\Phi}_c, Z_{t,c}, \beta_{t,c,r}$ to the server and hence the client-to-server communication is $\mathcal{O}(p \cdot s)$ real numbers, which is independent of the number of parameters of the base classifiers. We note that FEDMULTIBOOST can be extended to the multi-class setting using techniques presented in Appendix L.

6 Experiments

In this section, we present experimental results for the MULTIBOOST algorithm on several multiple-source datasets. Our study is restricted to learning an ensemble of decision stumps $\mathcal{H}^{stumps}$ using the exponential surrogate loss $\Phi(u) = e^{-u}$. To estimate the probabilities $\mathsf{Q}(k|\cdot)$ for $k \in [p]$, we assigned the label k to each sample from domain k and used multinomial logistic regression. This step can be facilitated with unlabeled examples, as only their domain membership information is required.

We compare MULTIBOOST with a set of boosting-style algorithms that operate on the same hypotheses class $\mathcal{H}^{stumps}$. Those include ADABOOST-k for $k \in [p]$ and ADABOOST-all. The former is a standard ADABOOST algorithm trained only on a single source S_k and the latter is ADABOOST trained on the union of all sources $\cup_{k=1}^{p} S_k$. It is natural to compare MULTIBOOST against ADABOOST-all, since both of them have access to all sources during training. The difference is that while ADABOOST-all treats $\cup_{k=1}^{p} S_k$ as a single dataset, MULTIBOOST trains base learners separately for each source and weights examples by domain probabilities $\mathsf{Q}(k|\cdot)$. We used $T = 100$ boosting steps for all benchmarks. T up to 1,000 were explored, but did not change any results significantly.

We also compare our results with the a multiple-source adaptation algorithm, DMSA, designed for a scenario where no labeled data is available, and where only an accurate predictor per domain and unlabeled data are supplied. DMSA was shown to outperform other algorithms designed for that scenario. To apply DMSA, we use the domain predictors ADABOOST-k, $k \in [p]$.

We report classification errors on the test data for various mixtures of the source distributions $\lambda_1, \ldots, \lambda_p$, including: errors for λ on the simplex Δ edges (errors on each domain separately); errors on the uniform mixture $\forall k : \lambda_k = \frac{1}{p}$; agnostic error, which is the maximum error across all sources. The errors and their standard deviations are reported based on 10-fold cross validation. Each source

S_k is independently split into 10 folds $S_k^1, \ldots, S_k^{10}$. For the i-th cross-validation step, the test set is $\{S_1^i, \ldots, S_p^i\}$, while the rest is used for training.

Datasets and preprocessing steps used are described below with additional dataset details provided in Appendix H. Note, that all datasets are public and do not contain any personal identifiers or offensive information. The experiments were performed on Linux and Mac workstations with Quad-Core Intel Core i7 2.9 GHz and Intel Xeon 2.20 GHz respectively. The time taken to perform 10-fold cross-validation varies per dataset: from roughly 1 hour on CPU for all benchmarks on tabular data to 8 hours on CPU for all benchmarks on digits data. The run-times of a single MULTIBOOST iteration is almost identical to that of ADABOOST-all when a closed form solution is used for step size η_t (line 12 in Figure 1) for the exponential surrogate loss function. Alternatively, for some experiments we used line search with 100 steps. If training time is critical, one can use $\eta_t = C/\sqrt{t}$ instead, at the cost of slightly worse convergence guarantee. Convergence plots are illustrated in Appendix I.

Sentiment analysis. The *Sentiment* dataset (Blitzer et al., 2007) consists of text reviews and rating labels for products sold on Amazon in various categories. We selected $p = 3$ sources from this dataset, where $k = 1$ corresponds to `books`, $k = 2$ means `dvd`, and $k = 3$ is `electronics`. We converted the 5-star rating to binary labels, assigning $y = +1$ for a positive review and $y = -1$ for a negative one. Neutral reviews with 3-star labels are removed from the dataset. The benchmarks are trained using bigram features generated from the raw text reviews. Since the *Sentiment* dataset contains only 2,000 instances for each domain, we used random train/test splits with 10 different seeds instead of the 10-fold cross-validation. To illustrate the importance of the conditional domain probabilities $\mathsf{Q}(K|x)$, for each domain, we illustrate its values along the projections of x onto the first principal component of the joint *Sentiment* dataset in Figure 3. The figure shows that our domain classifiers are able to provide coherent separation between the three domains and narrow the applicability of the base classifiers. In Appendix K, we further discuss the importance of obtaining a high-quality Q function and illustrate the performance degradation resulting from a lower-quality Q function.

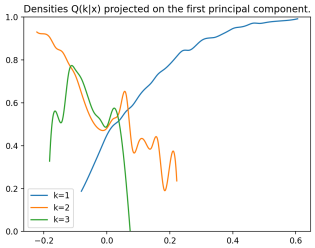

Figure 3: Mean values of $\mathsf{Q}(k|\cdot_k)$ for the three domains of the Sentiment data projected on the first principal component of the joint data.

Digits recognition. For this problem, we aggregated 32x32 pixel handwritten digits images from $p = 3$ sources: *MNIST* ($k = 1$), *SVHN* ($k = 2$), *MNIST-M* ($k = 3$). We compared algorithms on two binary classification problems: digits 4 vs. 9 and digits 1 vs. 7.

Object recognition. We divided images of clothes items from *Fashion-MNIST* (Xiao et al., 2017) dataset into two classes: tops and bottoms from $p = 3$ sources. $k = 1$ consists of `t-shirts, pullovers, trousers`; $k = 2$ consists of `dresses, coats, sandals`; $k = 3$ consists of `shirts, bags, sneakers, ankle-boots`.

Tabular data. In addition to text and images, we tested MULTIBOOST on tabular data. We used the *Adult* dataset (Kohavi, 1996), which consists of numerical and categorical features describing an individual's socioeconomic characteristics, given the task to predict if an person's income exceeds USD $50,000$. We divided this dataset into $p = 3$ sources based on individual's educational background. Source $k = 1$ consisted of individuals with a university degree (BSc, MS or PhD), source $k = 1$ those with only a High School diploma, and source $k = 2$, none of the above.

Discussion As can be seen from Table 4, MULTIBOOST provides agnostic and uniform errors that are significantly better than the baselines on all datasets. While ADABOOST-k predictors on digits recognition and object recognition tasks each show the lowest error on their specific domains $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3$, they fail to generalize to other target distribution in $\mathrm{conv}\{\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3\}$. Moreover, as suggested by the discussion in Section 2, the standard convex combination of domain-specific predictors (ADABOOST-all) also performs poorly on several target distributions in $\mathrm{conv}\{\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3\}$, such as the uniform and agnostic targets. As the experimental results confirm, MULTIBOOST, by leveraging domain predictors $\mathsf{Q}(k|\cdot)$ and the agnostic loss, is able to produce an ensemble of domain-specific predictors that generalizes to different targets. Appendix I provides additional illustrations of $\mathsf{Q}(k|\cdot)$ by projecting each domain on the first principal component of the joint data.

Moreover, for tabular data and the digits (4 vs. 9) classification problem, MULTIBOOST benefits from positive knowledge transfer and obtains an error on individual domains that is even smaller than that

Table 1: Test errors for multiple benchmarks.

Algorithm	Error-1	Error-2	Error-3	Error-Uniform	Error-Agnostic
			Sentiment Analysis		
AdaBoost-1	**0.326 ± 0.019**	0.300 ± 0.008	0.360 ± 0.017	0.329 ± 0.017	0.360 ± 0.019
AdaBoost-2	0.354 ± 0.020	**0.266 ± 0.009**	0.336 ± 0.023	0.318 ± 0.019	0.357 ± 0.019
AdaBoost-3	0.402 ± 0.015	0.334 ± 0.008	**0.258 ± 0.015**	0.331 ± 0.016	0.402 ± 0.018
AdaBoost-all	0.354 ± 0.020	0.325 ± 0.011	0.313 ± 0.022	0.324 ± 0.021	0.354 ± 0.016
DMSA	0.332 ± 0.021	0.308 ± 0.017	0.314 ± 0.015	0.318 ± 0.019	0.332 ± 0.021
MultiBoost	0.332 ± 0.027	0.288 ± 0.018	0.284 ± 0.027	**0.301 ± 0.027**	**0.332 ± 0.024**
			Digits Recognition (4 vs. 9)		
AdaBoost-1	**0.044 ± 0.007**	0.615 ± 0.012	0.476 ± 0.022	0.379 ± 0.008	0.615 ± 0.012
AdaBoost-2	0.455 ± 0.014	0.299 ± 0.011	0.504 ± 0.015	0.420 ± 0.011	0.504 ± 0.015
AdaBoost-3	0.549 ± 0.034	0.488 ± 0.015	0.300 ± 0.013	0.446 ± 0.013	0.549 ± 0.034
AdaBoost-all	0.060 ± 0.009	0.374 ± 0.015	0.353 ± 0.012	0.262 ± 0.009	0.374 ± 0.015
DMSA	0.069 ± 0.005	0.351 ± 0.012	0.310 ± 0.011	0.243 ± 0.009	0.351 ± 0.015
MultiBoost	0.096 ± 0.008	**0.283 ± 0.028**	**0.246 ± 0.014**	**0.209 ± 0.013**	**0.284 ± 0.027**
			Digits Recognition (1 vs. 7)		
AdaBoost-1	**0.005 ± 0.002**	0.613 ± 0.007	0.519 ± 0.012	0.379 ± 0.004	0.613 ± 0.007
AdaBoost-2	0.431 ± 0.022	**0.252 ± 0.009**	0.479 ± 0.012	0.387 ± 0.010	0.479 ± 0.012
AdaBoost-3	0.680 ± 0.031	0.490 ± 0.014	**0.244 ± 0.012**	0.474 ± 0.013	0.680 ± 0.031
AdaBoost-all	0.014 ± 0.003	0.286 ± 0.010	0.306 ± 0.012	0.202 ± 0.005	0.306 ± 0.011
DMSA	0.012 ± 0.003	0.288 ± 0.017	0.286 ± 0.015	0.195 ± 0.013	0.288 ± 0.017
MultiBoost	0.026 ± 0.004	0.261 ± 0.013	0.257 ± 0.015	**0.181 ± 0.005**	**0.261 ± 0.011**
			Objects Recognition (Fashion-MNIST)		
AdaBoost-1	**0.015 ± 0.003**	0.251 ± 0.026	0.602 ± 0.028	0.288 ± 0.017	0.602 ± 0.028
AdaBoost-2	0.435 ± 0.007	**0.015 ± 0.002**	0.169 ± 0.012	0.173 ± 0.003	0.435 ± 0.007
AdaBoost-3	0.311 ± 0.018	0.097 ± 0.005	**0.014 ± 0.002**	0.140 ± 0.006	0.311 ± 0.018
AdaBoost-all	0.036 ± 0.004	0.020 ± 0.002	0.025 ± 0.003	0.027 ± 0.002	0.036 ± 0.004
DMSA	0.033 ± 0.008	0.015 ± 0.002	0.022 ± 0.003	0.023 ± 0.007	0.033 ± 0.009
MultiBoost	0.028 ± 0.003	0.015 ± 0.003	0.022 ± 0.002	**0.021 ± 0.001**	**0.028 ± 0.003**
			Tabular Data (Adult Data)		
AdaBoost-1	0.201 ± 0.012	0.249 ± 0.021	0.215 ± 0.014	0.222 ± 0.012	0.249 ± 0.021
AdaBoost-2	0.298 ± 0.011	**0.131 ± 0.009**	0.142 ± 0.006	0.190 ± 0.007	0.298 ± 0.011
AdaBoost-3	0.225 ± 0.015	0.138 ± 0.010	0.133 ± 0.011	0.165 ± 0.008	0.225 ± 0.015
AdaBoost-all	0.221 ± 0.013	0.134 ± 0.007	0.131 ± 0.010	0.155 ± 0.004	0.221 ± 0.013
DMSA	0.195 ± 0.014	0.137 ± 0.005	0.131 ± 0.008	0.154 ± 0.005	0.195 ± 0.014
MultiBoost	**0.190 ± 0.014**	0.132 ± 0.008	**0.130 ± 0.009**	**0.150 ± 0.005**	**0.190 ± 0.014**

of the domain-specific predictors. From Figure 6 in Appendix I it can be seen that the α-weight for the MultiBoost classifier in the 4 vs. 9 problem is heavily centered on $\mathcal{D}_2$, but contributions from $\mathcal{D}_1$ and $\mathcal{D}_3$ adds to the stronger performance. The same figure also illustrates the α-mass distribution for the other classification tasks. The standard convex ensemble AdaBoost-all, however, does not seem to exhibit the positive transfer property. Similar conclusions carry over for other target distributions in $\mathrm{conv}\{\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3\}$, for the sake of simplicity we do not present them here.

7 Conclusion

We presented a new boosting algorithm, MultiBoost, for a multiple-source scenario – a common setting in domain adaptation problems – where the target distribution can be any mixture of the source distributions. We showed that our algorithm benefits from strong theoretical guarantees and exhibits favorable empirical performance. We also highlighted the extension of our work to the federated learning scenario, which is a critical distributed learning setting in modern applications.

Acknowledgments

This work was partly supported by NSF CCF-1535987, NSF IIS-1618662, and a Google Research Award.

References

A. Abay, Y. Zhou, N. Baracaldo, S. Rajamoni, E. Chuba, and H. Ludwig. Mitigating bias in federated learning, 2020.

C. J. Becker, C. M. Christoudias, and P. Fua. Non-linear domain adaptation with boosting. In *Proceedings of Advances in Neural Information Processing Systems 26: 27th Annual Conference on NIPS*, pages 485–493, 2013.

S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira. Analysis of representations for domain adaptation. In *Advances in neural information processing systems*, pages 137–144, 2007.

S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010.

A. L. Berger, S. D. Pietra, and V. J. D. Pietra. A maximum entropy approach to natural language processing. *Comp. Linguistics*, 22(1), 1996.

P. J. Bickel, E. A. Hammel, and J. W. O'Connell. Sex bias in graduate admissions: Data from Berkeley. *Science*, 187(4175):398–404, 1975. ISSN 0036-8075.

J. Blitzer, M. Dredze, and F. Pereira. Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. In *Proceedings of ACL 2007*, Prague, Czech Republic, 2007.

J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Wortman. Learning bounds for domain adaptation. In *Advances in neural information processing systems*, pages 129–136, 2008.

K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečnỳ, S. Mazzocchi, H. B. McMahan, et al. Towards federated learning at scale: System design. *arXiv preprint arXiv:1902.01046*, 2019.

L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

L. Breiman. Prediction games and arcing algorithms. *Neural Computation*, 11:1493–1517, October 1999.

T. S. Brisimi, R. Chen, T. Mela, A. Olshevsky, I. C. Paschalidis, and W. Shi. Federated learning of predictive models from federated electronic health records. *International journal of medical informatics*, 112:59–67, 2018.

S. Caldas, P. Wu, T. Li, J. Konečnỳ, H. B. McMahan, V. Smith, and A. Talwalkar. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

Y. Cheng, G. Cao, X. Wang, and J. Pan. Weighted multi-source TrAdaBoost. *Chinese Journal of Electronics*, 22(3):505–510, 2013.

C. Cortes and M. Mohri. Domain adaptation and sample bias correction theory and algorithm for regression. *Theoretical Computer Science*, 519:103–126, 2014.

C. Cortes, M. Mohri, and U. Syed. Deep boosting. In *Proceedings of ICML*, pages 1179–1187, 2014.

C. Cortes, M. Mohri, and A. Muñoz Medina. Adaptation algorithm and theory based on generalized discrepancy. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 169–178, 2015.

C. Cortes, X. Gonzalvo, V. Kuznetsov, M. Mohri, and S. Yang. Adanet: Adaptive structural learning of artificial neural networks. In *Proceedings of ICML*, pages 874–883, 2017.

C. Cortes, M. Mohri, A. T. Suresh, and N. Zhang. A discriminative technique for multiple-source adaptation. In *Proceedings of ICML*, volume 139, pages 2132–2143. PMLR, 2021.

W. Dai, Q. Yang, G.-R. Xue, and Y. Yu. Boosting for transfer learning. In *Proceedings of ICML*, pages 874–883, 2007.

N. Duffy and D. P. Helmbold. Potential boosters? In *Proceedings of NIPS*, pages 258–264, 1999.

Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer System Sciences*, 55(1):119–139, 1997.

Y. Freund, Y. Mansour, and R. E. Schapire. Generalization bounds for averaged classifiers. *The Annals of Statistics*, 32:1698–1722, 2004.

Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015.

R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE conference on computer vision and pattern recognition*, pages 2066–2073. IEEE, 2012.

A. Habrard, J.-P. Peyrache, and M. Sebban. Boosting for unsupervised domain adaptation. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 433–448. Springer, 2013.

J. Hamer, M. Mohri, and A. T. Suresh. Fedboost: A communication-efficient algorithm for federated learning. In *International Conference on Machine Learning*, pages 3973–3983. PMLR, 2020.

J. Hampshire and A. Waibel. The meta-pi network: building distributed knowledge representations for robust multisource pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(7):751–769, 1992. doi: 10.1109/34.142911.

A. Hard, K. Rao, R. Mathews, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.

M. Hardt, E. Price, E. Price, and N. Srebro. Equality of opportunity in supervised learning. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Proceedings of NIPS*, volume 29, 2016.

J. Hoffman, B. Kulis, T. Darrell, and K. Saenko. Discovering latent domains for multisource domain adaptation. In *ECCV*, volume 7573, pages 702–715, 2012.

J. Hoffman, E. Rodner, J. Donahue, T. Darrell, and K. Saenko. Efficient learning of domain-invariant image representations. *arXiv preprint arXiv:1301.3224*, 2013.

J. Hoffman, M. Mohri, and N. Zhang. Algorithms and theory for multiple-source adaptation. In *Proceedings of NeurIPS*, pages 8256–8266, 2018.

J. Hoffman, M. Mohri, and N. Zhang. Multiple-source adaptation theory and algorithms. *Annals of Mathematics and Artificial Intelligence*, 89(3-4):237–270, 2021.

P. Huang, G. Wang, and S. Qin. A novel learning approach to multiple tasks based on boosting methodology. *Pattern recognition letters*, 31(12):1693–1700, 2010.

P. Huang, G. Wang, and S. Qin. Boosting for transfer learning from multiple data sources. *Pattern Recognition Letters*, 33(5):568–579, 2012.

R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, R. G. L. D'Oliveira, H. Eichner, S. E. Rouayheb, D. Evans, J. Gardner, Z. Garrett, A. Gascón, B. Ghazi, P. B. Gibbons, M. Gruteser, Z. Harchaoui, C. He, L. He, Z. Huo, B. Hutchinson, J. Hsu, M. Jaggi, T. Javidi, G. Joshi, M. Khodak, J. Konečný, A. Korolova, F. Koushanfar, S. Koyejo, T. Lepoint, Y. Liu, P. Mittal, M. Mohri, R. Nock, A. Özgür, R. Pagh, M. Raykova, H. Qi, D. Ramage, R. Raskar, D. Song, W. Song, S. U. Stich, Z. Sun, A. T. Suresh, F. Tramèr, P. Vepakomma, J. Wang, L. Xiong, Z. Xu, Q. Yang, F. X. Yu, H. Yu, and S. Zhao. Advances and open problems in federated learning, 2021.

R. Kohavi. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *KDD*, volume 96, pages 202–207, 1996.

J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016a.

J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016b.

N. Konstantinov and C. Lampert. Robust learning from untrusted sources. In *International Conference on Machine Learning*, pages 3488–3498, 2019.

V. Kuznetsov, M. Mohri, and U. Syed. Multi-class deep boosting. In *Proceedings of NIPS*, pages 2501–2509, 2014.

P. Lahoti, A. Beutel, J. Chen, K. Lee, F. Prost, N. Thain, X. Wang, and E. H. Chi. Fairness without demographics through adversarially reweighted learning. *arXiv e-prints*, pages arXiv–2006, 2020.

M. Ledoux and M. Talagrand. *Probability in Banach Spaces: Isoperimetry and Processes*. Springer, New York, 1991.

D. Levy, Y. Carmon, J. C. Duchi, and A. Sidford. Large-scale methods for distributionally robust optimization. *Advances in Neural Information Processing Systems*, 33, 2020.

T. Li, M. Sanjabi, A. Beirami, and V. Smith. Fair resource allocation in federated learning. In *Proceedings of ICLR*, 2020.

H. Liao. Speaker adaptation of context dependent deep neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7947–7951. IEEE, 2013.

M. Long, Y. Cao, J. Wang, and M. Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015.

Z.-Q. Luo and P. Tseng. On the convergence of the coordinate descent method for convex differentiable minimization. *Journal of Optimization Theory and Applications*, 72(1):7–35, 1992.

D. J. C. MacKay. *Bayesian methods for adaptive models*. PhD thesis, California Institute of Technology, 1991.

Y. Mansour, M. Mohri, and A. Rostamizadeh. Domain adaptation with multiple sources. *Advances in neural information processing systems*, 21:1041–1048, 2008.

Y. Mansour, M. Mohri, and A. Rostamizadeh. Multiple source adaptation and the Rényi divergence. In *UAI*, pages 367–374, 2009a.

Y. Mansour, M. Mohri, and A. Rostamizadeh. Domain adaptation: Learning bounds and algorithms. In *COLT*, 2009b.

L. Mason, J. Baxter, P. L. Bartlett, and M. R. Frean. Boosting algorithms as gradient descent. In *Proceedings of NIPS*, pages 512–518, 1999.

B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of AISTATS*, pages 1273–1282, 2017.

M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of Machine Learning*. Adaptive computation and machine learning. MIT Press, 2018. Second edition.

M. Mohri, G. Sivek, and A. T. Suresh. Agnostic federated learning. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 4615–4625. PMLR, 2019.

S. Motiian, Q. Jones, S. Iranmanesh, and G. Doretto. Few-shot adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, pages 6670–6680, 2017a.

S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto. Unified deep supervised domain adaptation and generalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5715–5725, 2017b.

K. Muandet, D. Balduzzi, and B. Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pages 10–18. PMLR, 2013.

H. Namkoong and J. C. Duchi. Stochastic gradient methods for distributionally robust optimization with f-divergences. In *Advances in Neural Information Processing Systems*, pages 2208–2216, 2016.

S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

Z. Pei, Z. Cao, M. Long, and J. Wang. Multi-adversarial domain adaptation. In *AAAI*, pages 3934–3941, 2018.

G. Rätsch, S. Mika, and M. K. Warmuth. On the convergence of leveraging. In *NIPS*, pages 487–494, 2001.

J. Ro, M. Chen, R. Mathews, M. Mohri, and A. T. Suresh. Communication-efficient agnostic federated averaging. *arXiv preprint arXiv:2104.02748*, 2021.

K. Saito, D. Kim, S. Sclaroff, T. Darrell, and K. Saenko. Semi-supervised domain adaptation via minimax entropy. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8050–8058, 2019.

R. E. Schapire and Y. Freund. *Boosting: Foundations and Algorithms*. The MIT Press, 2012.

R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. In *ICML*, pages 322–330, 1997.

Z. Shen, H. Hassani, S. Kale, and A. Karbasi. Federated functional gradient boosting. *arXiv preprint arXiv:2103.06972*, 2021.

P. Smyth and D. Wolpert. Linearly combining density estimators via stacking. *Machine Learning*, 36: 59–83, July 1999.

A. Taherkhani, G. Cosma, and T. M. McGinnity. AdaBoost-cnn: An adaptive boosting algorithm for convolutional neural networks to classify multi-class imbalanced datasets using transfer learning. *Neurocomputing*, 404:351–366, 2020.

A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528. IEEE, 2011.

P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal Optimization Theory and Applications*, pages 475–494, 2001.

E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. Simultaneous deep transfer across domains and tasks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4068–4076, 2015.

B. Wang, J. Mendez, M. Cai, and E. Eaton. Transfer learning via minimizing the performance gap between domains. In *Advances in Neural Information Processing Systems*, pages 10645–10655, 2019a.

T. Wang, X. Zhang, L. Yuan, and J. Feng. Few-shot adaptive faster R-CNN. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7173–7182, 2019b.

H. Xiao, K. Rasul, and R. Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017.

R. Xu, Z. Chen, W. Zuo, J. Yan, and L. Lin. Deep cocktail network: Multi-source unsupervised domain adaptation with category shift. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3964–3973, 2018.

Z. Xu and S. Sun. Multi-view transfer learning with adaboost. In *2011 IEEE 23rd International Conference on Tools with Artificial Intelligence*, pages 399–402. IEEE, 2011.

Z. Xu and S. Sun. Multi-source transfer learning with multi-view AdaBoost. In *International conference on neural information processing*, pages 332–339. Springer, 2012.

Z. Xu, W. Li, L. Niu, and D. Xu. Exploiting low-rank structure from latent domains for domain generalization. In *European Conference on Computer Vision*, pages 628–643. Springer, 2014.

J. Yang, R. Yan, and A. G. Hauptmann. Cross-domain video concept detection using adaptive SVMs. In *Proceedings of the 15th ACM international conference on Multimedia*, pages 188–197, 2007.

Q. Yang, Y. Liu, T. Chen, and Y. Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

T. Yang, G. Andrew, H. Eichner, H. Sun, W. Li, N. Kong, D. Ramage, and F. Beaufays. Applied federated learning: Improving google keyboard query suggestions. *arXiv preprint arXiv:1812.02903*, 2018.

Y. Yao and G. Doretto. Boosting for transfer learning with multiple sources. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 1855–1862. IEEE, 2010.

Z. Yuan, D. Bao, Z. Chen, and M. Liu. Integrated transfer learning algorithm using multi-source tradaboost for unbalanced samples classification. In *2017 international conference on computing intelligence and information system (CIIS)*, pages 188–195. IEEE, 2017.

Q. Zhang, H. Li, Y. Zhang, and M. Li. Instance transfer learning with multisource dynamic tradaboost. *The scientific world journal*, 2014, 2014.

H. Zhao, S. Zhang, G. Wu, J. M. Moura, J. P. Costeira, and G. J. Gordon. Adversarial multiple source domain adaptation. In *Advances in neural information processing systems*, pages 8559–8570, 2018.

Contents of Appendix

A Previous work: more detailed discussion

Domain adaptation with a single source domain and a target domain has been widely studied (Ben-David et al., 2007; Blitzer et al., 2008; Mansour et al., 2009b; Ben-David et al., 2010; Cortes and Mohri, 2014; Cortes et al., 2015; Wang et al., 2019b) and has applications to several fields ranging from acoustic modelling (Liao, 2013) to object recognition (Torralba and Efros, 2011). It has been studied in unsupervised settings with unlabeled target domain examples (Gong et al., 2012; Long et al., 2015; Ganin and Lempitsky, 2015), in supervised settings with the aid of labeled target domain examples (Yang et al., 2007; Hoffman et al., 2013; Girshick et al., 2014; Motiian et al., 2017b), and in semi-supervised settings where both labeled and unlabeled target examples are available (Tzeng et al., 2015; Saito et al., 2019).

In a wide variety of applications, the learner has access to information from multiple sources. Such problems are often referred to as *multiple-source adaptation*. Multiple-source adaptation problems, where the learner has access to unlabeled source data together with predictors that are trained for each particular domain has been formalized in Mansour et al. (2008, 2009a); Hoffman et al. (2018). There are other multiple-source adaptation scenarios, where labeled examples are available from multiple sources and unlabeled or labeled examples are available from the target domain. This includes adversarial training, which has been studied by Motiian et al. (2017a); Pei et al. (2018); Zhao et al. (2018); Xu et al. (2018). Algorithms for learning from multiple untrusted sources have been proposed by Konstantinov and Lampert (2019). Another related problem is *domain generalization* (Pan and Yang, 2009; Muandet et al., 2013; Xu et al., 2014), where information from multiple sources is used to obtain a predictor that generalizes to a previously unseen domain.

There are various algorithms, successfully applying boosting with multiple sources to domain adaptation and transfer learning problems, that have inspired our analysis. The TrAdaBoost (Dai et al., 2007) algorithm, having a set of weak learners trained on the source domain, at every boosting round selects those that minimize the error on the target domain. In case of multiple sources and a single target, Yao and Doretto (2010) developed MultiSourceTrAdaBoost algorithm that trains weak learners on the union of each of the sources and the target, thus reducing the negative knowledge transfer effect. These algorithms have been further improved and widely adopted in practice (Yuan et al., 2017; Cheng et al., 2013; Zhang et al., 2014). Another approach that uses multi-view AdaBoost for single and multi-source domain adaptation was proposed by Xu and Sun (2012, 2011). They divide the feature space into two *views* based on the source and target; at each boosting step, two weak learners are trained on these views and the sample distribution is updated according to the errors on the target domain.

A number of experimental studies have shown the benefits of having an ensemble of weak learners for multi-task learning and domain adaptation problems. Moreover, in certain cases the boosting approach can outperform traditional methods. For example, Huang et al. (2010, 2012) showed that by selecting a weak learner jointly with a feature that is predictive across multiple domains at every boosting step, one can achieve higher accuracy than standard transfer learning methods. Moreover, the margin provided by boosting-style algorithms can aid in transfer learning where target domain is unlabelled. Habrard et al. (2013) have developed an algorithm that jointly minimizes the the source domain error and margin violation proportion on the target domain.

Wang et al. (2019a) have demonstrated that boosting classifiers from different domains can be done online and showed efficient algorithm for the AdaBoost-style sample distribution updates. For certain types of high-dimensional data, such as images and text, boosting may be not as efficient as other multi-task learning methods. However, a number of works such as Taherkhani et al. (2020) and Becker et al. (2013) have shown that multi-source boosting can be combined with Deep Neural Networks for multi-task learning on large scale datasets.

In the context of neural networks, the idea of using domain probabilities when combining experts, also termed *gating networks*, goes back to Hampshire and Waibel (1992) and Jacobs et al. (1991).

Agnostic loss has been used in several machine learning problems. Namkoong and Duchi (2016); Levy et al. (2020) proposed efficient algorithms to minimize agnostic loss in the context of distributionally robust optimization. Agnostic loss in federated learning has been studied by Mohri et al. (2019); Hamer et al. (2020); Ro et al. (2021), who provided theoretical guarantees and algorithms. Lahoti et al. (2020) used agnostic loss to achieve fairness in machine learning models.

B Proof of the Propositions 1 and 2

This section contains the proofs for Proposition 1 and Proposition 2 discussed in Section 2.

Proposition 1. *There exist distributions $\mathcal{D}_1$ and $\mathcal{D}_2$ and hypotheses h_1 and h_2 with $\mathcal{L}(\mathcal{D}_1, h_1) = 0$ and $\mathcal{L}(\mathcal{D}_2, h_2) = 0$ such that $\mathcal{L}\big(\frac{1}{2}(\mathcal{D}_1 + \mathcal{D}_2), \alpha h_1 + (1-\alpha)h_2\big) \geq \frac{1}{2}$ for any $\alpha \in [0, 1]$,*

Proof. Consider the case where $\mathcal{X}$ is reduced to two elements, $\mathcal{X} = \{a_1, a_2\}$, where $\mathcal{D}_1$ is the point mass on a_1, $\mathcal{D}_2$ the point mass on a_2, and where the target labeling function is f defined by $f(a_1) = +1$, $f(a_2) = -1$. Let h_1 be defined by $h_1(a_1) = h_1(a_2) = +1$ and and h_2 by $h_2(a_1) = h_2(a_2) = -1$.

Then, h_1 is a perfect predictor for the first domain since $\mathcal{L}(\mathcal{D}_1, h_1) = \mathbb{I}(h_1(a_1)f(a_1) \leq 0) = 0$, and h_2 is a perfect predictor for the second domain since $\mathcal{L}(\mathcal{D}_2, h_2) = \mathbb{I}(h_2(a_2)f(a_2) \leq 0) = 0$. However, for any $\alpha \in [0, 1]$, we have

$$\mathcal{L}\left(\frac{1}{2}(\mathcal{D}_1 + \mathcal{D}_2), \alpha h_1 + (1-\alpha)h_2\right) = \frac{1}{2}\,\mathbb{I}(2\alpha - 1 \leq 0) + \frac{1}{2}\,\mathbb{I}(1 - 2\alpha \leq 0) \geq \frac{1}{2}.$$

This concludes the proof of Proposition 1. □

Proposition 2. *For the same distributions $\mathcal{D}_1$ and $\mathcal{D}_2$ and hypotheses h_1 and h_2 as in Proposition 1, the equality $\mathcal{L}(\mathcal{D}_\lambda, (\alpha \mathsf{Q}(1|\cdot)h_1 + (1-\alpha)\mathsf{Q}(2|\cdot)h_2)) = 0$ holds for any $\lambda \in \Delta$ and any $\alpha \in (0, 1)$.*

Proof. For the counterexample of Proposition 1, for any $\alpha \in (0, 1)$, the Q-ensemble $f(x) = (\alpha \mathsf{Q}(1|x)h_1(x) + (1-\alpha)\mathsf{Q}(2|x)h_2(x))$ admits no loss with respect to any target distribution $\mathcal{D}_\lambda$:

$$\begin{aligned}\mathcal{L}(\mathcal{D}_\lambda, f) &= (\lambda\,\mathbb{I}(f(a_1) \leq 0) + (1-\lambda)\,\mathbb{I}(-f(a_2) \leq 0)) \\ &= \lambda(\mathbb{I}(\alpha \leq 0) + (1-\lambda)\,\mathbb{I}((1-\alpha) \leq 0)) = 0,\end{aligned}$$

since $\mathsf{Q}(1|a_1) = \mathsf{Q}(2|a_2) = 1$ and $\mathsf{Q}(2|a_1) = \mathsf{Q}(1|a_2) = 0$.

This concludes the proof of Proposition 2. □

C Proof of Lemma 1

Lemma 1. *For any $k \in [p]$, the following upper bound holds when Φ is the exponential or the logistic function:*

$$F(\alpha_{t-1} + \eta \mathbf{e}_{k,r}) \leq \max_{l \in [p]} \frac{Z_{t,l}}{m_l}\big[(1 - \epsilon_{t,l,k,r})e^{-\eta} + \epsilon_{t,l,k,r}e^{\eta}\big],$$

where $\epsilon_{t,l,k,r} = \frac{1}{2}\big[1 - \mathbb{E}_{i \sim \mathsf{D}_t(l,\cdot)}[y_{l,i}\mathsf{Q}(k|x_{l,i})h_{k,r}(x_{l,i})]\big]$.

Proof. In the special case where $\Phi = \exp$, we have:

$$\begin{aligned}&\Phi(-y_{l,i}f_{t-1}(x_{l,i}) - \eta\, y_{l,i}\mathsf{Q}(k|x_{l,i})h_{k,r}(x_{l,i})) \\ &\leq \frac{1 + y_{l,i}\mathsf{Q}(k|x_{l,i})h_{k,r}(x_{l,i})}{2}e^{-y_{l,i}f_{t-1}(x_{l,i})}e^{-\eta} + \frac{1 - y_{l,i}\mathsf{Q}(k|x_{l,i})h_{k,r}(x_{l,i})}{2}e^{-y_{l,i}f_{t-1}(x_{l,i})}e^{\eta} \\ &= \frac{1 + y_{l,i}\mathsf{Q}(k|x_{l,i})h_{k,r}(x_{l,i})}{2}\mathsf{D}_t(l,i)Z_{t,l}e^{-\eta} + \frac{1 - y_{l,i}\mathsf{Q}(k|x_{l,i})h_{k,r}(x_{l,i})}{2}\mathsf{D}_t(l,i)Z_{t,l}e^{\eta}.\end{aligned}$$

Thus, we have:

$$\begin{aligned}F(\alpha_{t-1} + \eta \mathbf{e}_{k,r}) &= \max_{l \in [p]} \frac{1}{m_l}\sum_{i=1}^{m_l} \Phi(-y_{l,i}f_{t-1}(x_{l,i}) - \eta\, y_{l,i}\mathsf{Q}(k|x_{l,i})h_{k,r}(x_{l,i})) \\ &\leq \max_{l \in [p]} \frac{Z_{t,l}}{m_l}\sum_{i=1}^{m_l}\frac{1}{m_l}\sum_{i=1}^{m_l}\frac{1 + y_{l,i}\mathsf{Q}(k|x_{l,i})h_{k,r}(x_{l,i})}{2}\mathsf{D}_t(l,i)e^{-\eta} \\ &\quad + \frac{1 - y_{l,i}\mathsf{Q}(k|x_{l,i})h_{k,r}(x_{l,i})}{2}\mathsf{D}_t(l,i)e^{\eta} \\ &= \max_{l \in [p]} \frac{Z_{t,l}}{m_l}\big[(1 - \epsilon_{t,l,k,r})e^{-\eta} + \epsilon_{t,l,k,r}e^{\eta}\big].\end{aligned}$$

The proof is similar in the case of the logistic function. □

D Other variants of MULTIBOOST

As already mentioned, instead of the maximum, the *softmax* function $(x_1, \ldots, x_k) \mapsto \frac{1}{\mu} \log(\sum_{k=1}^{p} e^{\mu x_k})$ can be used in the definition of the algorithm, modulo an approximation that can be controlled via the parameter $\mu > 0$. Using the *softmax* not only leads to a differentiable objective, but also makes the algorithm focus on several top most difficult domains instead of the single most difficult one, thereby offering a useful trade-off in some applications.

Another variant of the algorithm with also a differentiable objective function consists of simply upper-bounding the maximum by a sum:

$$F_{\text{sum}}(\alpha) = \sum_{k=1}^{p} \frac{1}{m_k} \sum_{i=1}^{m_k} \Phi\left(-y_{k,i} \sum_{l=1}^{p} \sum_{j=1}^{N_l} \alpha_{l,j} \mathsf{Q}(l|x_{k,i}) h_{l,j}(x_{k,i})\right). \tag{8}$$

It is straightforward to show that, as with the maximum-based objective, our weak learning assumption implies that, at each round, there exists a coordinate direction along which each active function F_k decreases. Furthermore, our comments and analysis in the maximum case regarding the Q-function and lower bounds on the edge similarly hold here.

In the following, we present convergence guarantees for the F_{sum} objective. A similar guarantee with the same proof holds for the softmax variant of MULTIBOOST.

Theorem 3. *Assume that Φ is twice differentiable and $\Phi''(u) \geq 0$ for all $u \in \mathbb{R}$. Let $F = F_{sum}$, then, projected coordinate descent applied to $F(\boldsymbol{\alpha})$ converges to the optimal solution $\boldsymbol{\alpha}^*$ of $\min_{\alpha \geq 0} F(\boldsymbol{\alpha})$. If further Φ is strongly convex on the path of the iterates α_t, then there exist $\tau > 0$ and $\gamma > 0$ such that for all $t > \tau$:*

$$F(\boldsymbol{\alpha}_{t+1}) - F(\boldsymbol{\alpha}^*) \leq \left(1 - \tfrac{1}{\gamma}\right)(F(\boldsymbol{\alpha}_t) - F(\boldsymbol{\alpha}^*)). \tag{9}$$

Proof. We show that F_{sum} can be represented as $F_{\text{sum}}(\boldsymbol{\alpha}) = G(\mathbf{H}\boldsymbol{\alpha})$, such that $\nabla^2 G(\mathbf{H}\boldsymbol{\alpha})$ is positive definite for all $\boldsymbol{\alpha}$ and apply Theorem 2.1 in Luo and Tseng (1992) to obtain the convergence guarantees. Let $\mathbf{H}$ be the matrix whose row indexes are $\{(k,i)\colon k \in [p], i \in [m_k]\}$ and whose column indexes are $\{(l,j)\colon l \in [p], j \in [N_k]\}$. Define matrix $\mathbf{H}$ by $\mathbf{H}_{(k,i),(l,j)} = -y_{k,i} \mathsf{Q}(l|x_{k,i}) h_{l,j}(x_{k,i})$. Let $\mathbf{e}_{(k,i)}$ be the (k,i)-th unit vector, then for any $\boldsymbol{\alpha}$:

$$\mathbf{e}_{(k,i)}^\top \mathbf{H}\boldsymbol{\alpha} = -y_{k,i} \sum_{l=1}^{p} \sum_{j=1}^{N_l} \alpha_{l,j} \mathsf{Q}(l|x_{k,i}) h_{l,j}(x_{k,i}). \tag{10}$$

Define the function G as follows for all $\mathbf{u}$:

$$G(\mathbf{u}) = \sum_{k=1}^{p} \frac{1}{m_k} \sum_{i=1}^{m_k} \Phi\left(-\mathbf{e}_{(k,i)}^\top \mathbf{u}\right). \tag{11}$$

By definition, we have $F_{\text{sum}}(\boldsymbol{\alpha}) = G(\mathbf{H}\boldsymbol{\alpha})$. Moreover, G is twice differentiable and $\nabla^2 G(\mathbf{u})$ is a diagonal matrix with diagonal entries $\frac{1}{m_k} \Phi''(-\mathbf{e}_{(k,i)}^\top \mathbf{u}) \geq 0$. Thus, $\nabla^2 G(\mathbf{u})$ is positive definite for all $\alpha \geq 0$. Thus, Theorem 2.1 in Luo and Tseng (1992) holds for the optimization problem

$$\min_{\boldsymbol{\alpha} \geq 0} G(\mathbf{H}\boldsymbol{\alpha}), \tag{12}$$

which guarantees the convergence rate of the coordinate descent for F_{sum}. If further F is is strongly convex over the sequence of α_ts, then, by Luo and Tseng (1992)[page 26], the inequality:

$$F(\boldsymbol{\alpha}_{t+1}) - F(\boldsymbol{\alpha}^*) \leq \left(1 - \tfrac{1}{\gamma}\right)(F(\boldsymbol{\alpha}_t) - F(\boldsymbol{\alpha}^*))$$

holds for the projected coordinate method based on the best direction at each round, as with the Gauss-Southwell method. □

Note, that the proof can be extended straightforwardly to a regularized F_{sum} objective, simply by considering $F_{\text{sum}}(\boldsymbol{\alpha}) = G(\mathbf{H}\boldsymbol{\alpha}) + \beta^\top \boldsymbol{\alpha}$ in the proof for some $\beta \geq 0$.

E Proofs of Theorem 1 and Theorem 4

Theorem 1. *Fix $\rho > 0$. Then, for any $\delta > 0$, with probability at least $1 - \delta$ over the draw of a sample $S = (S_1, \ldots, S_p) \sim \mathcal{D}_1^{m_1} \otimes \cdots \otimes \mathcal{D}_p^{m_p}$, the following inequality holds for all ensemble functions $f = \sum_{t=1}^{T} \alpha_t \mathsf{Q}(k_t|\cdot) h_t \in \mathcal{F}$ and all $\lambda \in \Delta$:*

$$\mathcal{L}(\mathcal{D}_\lambda, f) \leq \mathcal{L}_\rho(\overline{\mathcal{D}}_\lambda, f) + \sum_{k=1}^{p} \lambda_k \left[\frac{2}{\rho} \max_{l \in [p]} \mathfrak{R}_{m_k}(\mathcal{H}_l) + \sqrt{\frac{\log \frac{p}{\delta}}{2m_k}} \right]. \tag{5}$$

Proof. Fix $\lambda \in \Delta$ and $\delta > 0$. For any $k \in [p]$, by the standard Rademacher complexity margin bound for $\mathcal{F}$ (Mohri et al., 2018)[Theorem 5.8], with probability at least $1 - \delta$, the following inequality holds for all $f \in \mathcal{F}$:

$$\mathcal{L}(\mathcal{D}_k, f) \leq \mathcal{L}_\rho(\widehat{\mathcal{D}}_k, f) + \frac{2}{\rho} \mathfrak{R}_{m_k}(\mathcal{F}) + \sqrt{\frac{\log \frac{1}{\delta}}{2m_k}}.$$

By the union bound, the following inequalities hold simultaneously for all $k \in [p]$:

$$\mathcal{L}(\mathcal{D}_k, f) \leq \mathcal{L}_\rho(\widehat{\mathcal{D}}_k, f) + \frac{2}{\rho} \mathfrak{R}_{m_k}(\mathcal{F}) + \sqrt{\frac{\log \frac{p}{\delta}}{2m_k}}.$$

Multiplying each by λ_k and summing them up yields:

$$\mathcal{L}(\mathcal{D}_\lambda, f) \leq \mathcal{L}_\rho(\overline{\mathcal{D}}_\lambda, f) + \sum_{k=1}^{p} \lambda_k \left[\frac{2}{\rho} \mathfrak{R}_{m_k}(\mathcal{F}) + \sqrt{\frac{\log \frac{p}{\delta}}{2m_k}} \right].$$

Since the Rademacher of a family coincides with that of its convex hull (Mohri et al., 2018), we have $\mathfrak{R}_{m_k}(\mathcal{F}) = \mathfrak{R}_{m_k}(\bigcup_{k=1}^{p} \mathcal{G}_k) \leq \max_{l \in [p]} \mathfrak{R}_{m_k}(\mathcal{G}_l)$. We will show that the following inequality holds for any $k \in p$: $\mathfrak{R}_{m_k}(\mathcal{G}_l) \leq 2\mathfrak{R}_{m_k}(\mathcal{H}_l)$. Note that we can write for any $h \in \mathcal{H}_k$: $\mathsf{Q}(l|\cdot)h = \frac{1}{4}\left[(\mathsf{Q}(l|\cdot) + h)^2 - (\mathsf{Q}(l|\cdot) - h)^2\right]$. Thus, since the Rademacher complexity of a sum can be bounded by the sum of the Rademacher complexities, we have:

$$\mathfrak{R}_{m_k}(\mathcal{G}_l) \leq \frac{1}{4} \mathfrak{R}_{m_k}\left(\left\{(\mathsf{Q}(l|\cdot) + h)^2 \colon h \in \mathcal{H}_l\right\}\right) + \frac{1}{4} \mathfrak{R}_{m_k}\left(\left\{(\mathsf{Q}(l|\cdot) - h)^2 \colon h \in \mathcal{H}_l\right\}\right).$$

Now, functions $\mathsf{Q}(l|\cdot) + h$ and $\mathsf{Q}(l|\cdot) - h$ both take values in $[-1, 2]$ and the function $x \mapsto \frac{1}{4}x^2$ is 1-Lipschitz on $[-1, 2]$ since the absolute value of its derivative $|x|/2$ reaches it maximum at $x = 2$. Thus, by Talagrand's contraction lemma (Ledoux and Talagrand, 1991), we have

$$\mathfrak{R}_{m_k}(\mathcal{G}_l) \leq \mathfrak{R}_{m_k}(\{(\mathsf{Q}(l|\cdot) + h) \colon h \in \mathcal{H}_l\}) + \mathfrak{R}_{m_k}(\{(\mathsf{Q}(l|\cdot) - h) \colon h \in \mathcal{H}_l\}).$$

Now, these Rademacher complexities can be straightforwardly analyzed as follows:

$$\begin{aligned}
\mathfrak{R}_{m_k}(\{(\mathsf{Q}(l|\cdot) + h) \colon h \in \mathcal{H}_l\}) &= \mathop{\mathbb{E}}_{\substack{S \sim \mathcal{D}^m \\ \boldsymbol{\sigma}}} \left[\sup_{h \in \mathcal{H}_l} \sum_{i=1}^{m_k} \sigma_i [h(x_i) + \mathsf{Q}(l|x_i)] \right] \\
&= \mathop{\mathbb{E}}_{\substack{S \sim \mathcal{D}^m \\ \boldsymbol{\sigma}}} \left[\sup_{h \in \mathcal{H}_l} \sum_{i=1}^{m_k} \sigma_i h(x_i) \right] + \mathop{\mathbb{E}}_{\substack{S \sim \mathcal{D}^m \\ \boldsymbol{\sigma}}} \left[\sum_{i=1}^{m_k} \sigma_i \mathsf{Q}(l|x_i) \right] \\
&= \mathop{\mathbb{E}}_{\substack{S \sim \mathcal{D}^m \\ \boldsymbol{\sigma}}} \left[\sup_{h \in \mathcal{H}_l} \sum_{i=1}^{m_k} \sigma_i h(x_i) \right] + \mathop{\mathbb{E}}_{S \sim \mathcal{D}^m} \left[\sum_{i=1}^{m_k} \mathop{\mathbb{E}}_{\boldsymbol{\sigma}}[\sigma_i] \mathsf{Q}(l|x_i) \right] \\
&= \mathop{\mathbb{E}}_{\substack{S \sim \mathcal{D}^m \\ \boldsymbol{\sigma}}} \left[\sup_{h \in \mathcal{H}_l} \sum_{i=1}^{m_k} \sigma_i h(x_i) \right] = \mathfrak{R}_{m_k}(\mathcal{H}_l).
\end{aligned}$$

Similarly, we have $\mathfrak{R}_{m_k}(\{(\mathsf{Q}(l|\cdot) - h) \colon h \in \mathcal{H}_l\}) = \mathfrak{R}_{m_k}(\mathcal{H}_l)$. This completes the proof. □

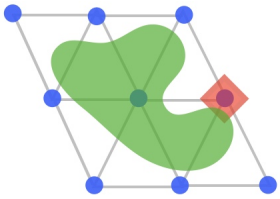

Figure 4: Illustration of the set $\overline{\Lambda}$, vertices of simplices(in blue). The area in green represents the set Λ. The small area in pink shows an ϵ-ball in l_1-distance (and hence a lozenge) around a vertex.

Theorem 4. *For any $\delta > 0$, with probability at least $1-\delta$ over the draw of a sample $S = (S_1, \ldots, S_p) \sim \mathcal{D}_1^{m_1} \otimes \cdots \otimes \mathcal{D}_p^{m_p}$, the following inequality holds for all ensemble functions $f = \sum_{t=1}^{T} \alpha_t \mathsf{Q}(k_t|\cdot) h_t \in \mathcal{F}$ and all $\rho \in (0,1)$ and $\lambda \in \Delta$:*

$$\mathcal{L}(\mathcal{D}_\lambda, f) \leq \mathcal{L}_\rho(\overline{\mathcal{D}}_\lambda, f) + \sum_{k=1}^{p} \lambda_k \left[\frac{2}{\rho} \max_{l \in [p]} \mathfrak{R}_{m_k}(\mathcal{H}_l) + \sqrt{\frac{\log \log_2 \frac{2}{\rho}}{m_k}} + \sqrt{\frac{\log \frac{p}{\delta}}{2m_k}} \right]. \tag{13}$$

Proof. By the uniform margin bound (Mohri et al., 2018, Theorem 5.9), for any $k \in [p]$, with probability at least $1-\delta$ the following inequality holds for all $f \in \mathcal{F}$ and $\rho \in (0,1]$:

$$\mathcal{L}(\mathcal{D}_k, f) \leq \mathcal{L}_\rho(\widehat{\mathcal{D}}_k, f) + \frac{2}{\rho} \mathfrak{R}_{m_k}(\mathcal{F}) + \sqrt{\frac{\log \log_2 \frac{2}{\rho}}{m_k}} + \sqrt{\frac{\log \frac{1}{\delta}}{2m_k}}.$$

The rest of the proof is similar to that of Theorem 1. □

F Finer margin-based learning guarantees

In this section, we give finer margin-based generalization bounds for the family of ensemble $\mathcal{F}$. These learning bounds are particularly more relevant in the case where Λ is a strict subset of the simplex Δ. Λ may be in fact a much smaller set, motivated by prior knowledge about the task and thus possible target mixtures. In some instances, it may even be a finite subset, which corresponds to only a finite set of mixtures.

For any family of real-valued functions $\mathcal{G}$, define the *weighted Rademacher complexity of* $\mathcal{G}$ for the vector of samples $S_k = (z_{k,1}, \ldots, z_{k,m_k})$ of sizes $\mathbf{m} = (m_1, \ldots, m_p)$ as follows:

$$\mathfrak{R}_{\mathbf{m}}(\mathcal{G}, \lambda) = \mathop{\mathbb{E}}_{S_k \sim \mathcal{D}_k^{m_k}, \boldsymbol{\sigma}} \left[\sup_{g \in \mathcal{G}} \sum_{k=1}^{p} \frac{\lambda_k}{m_k} \sum_{i=1}^{m_k} \sigma_{k,i}\, g(z_{k,i}) \right]. \tag{14}$$

Fix $\lambda \in \Lambda$ and define $\Psi(S)$ for the vector of training samples $S = (S_1, \ldots, S_p)$ as follows:

$$\Psi(S) = \sup_{h \in \mathcal{G}} \left\{ \mathop{\mathbb{E}}_{z \sim \mathcal{D}_\lambda} [g(z)] - \mathop{\mathbb{E}}_{z \sim \overline{\mathcal{D}}_\lambda} [g(z)] \right\},$$

where $\overline{\mathcal{D}}_\lambda = \sum_{k=1}^{p} \lambda_k \widehat{\mathcal{D}}_k$, with $\widehat{\mathcal{D}}_k$ the empirical distribution associated with the sample S_k. Assume that functions in $\mathcal{G}$ take values in $[0,1]$. For any vector of samples S' differing from S only by point $z'_{k,i}$ in S'_k and $z_{k,i}$ in S_k, we have

$$\begin{aligned}
\Psi(S') - \Psi(S) &\leq \sup_{g \in \mathcal{G}} \left\{ \left\{ \mathop{\mathbb{E}}_{z \sim \mathcal{D}_\lambda} [g(z)] - \mathop{\mathbb{E}}_{z \sim \overline{\mathcal{D}}'_\lambda} [g(z)] \right\} - \left\{ \mathop{\mathbb{E}}_{z \sim \mathcal{D}_\lambda} [g(z)] - \mathop{\mathbb{E}}_{z \sim \overline{\mathcal{D}}_\lambda} [g(z)] \right\} \right\} \\
&= \sup_{g \in \mathcal{G}} \left\{ \mathop{\mathbb{E}}_{z \sim \overline{\mathcal{D}}_\lambda} [g(z)] - \mathop{\mathbb{E}}_{z \sim \overline{\mathcal{D}}'_\lambda} [g(z)] \right\} = \sup_{g \in \mathcal{G}} \frac{\lambda_k}{m_k} \left[g(z_{k,i}) - g(z'_{k,i}) \right] \leq \frac{\lambda_k}{m_k}.
\end{aligned}$$

Furthermore, as with the standard Rademacher complexity (Mohri et al., 2018), the expectation can be upper bounded in terms of the weighted Rademacher complexity:

$\mathbb{E}_{S_k \sim \mathcal{D}_k^{m_k}}\left[\sup_{g \in \mathcal{G}} \mathbb{E}_{z \sim \mathcal{D}_\lambda}[g(z)] - \mathbb{E}_{z \sim \overline{\mathcal{D}}_\lambda}[g(z)]\right] \leq 2\mathfrak{R}_{\mathbf{m}}(\mathcal{G}, \lambda)$. Thus, by McDiarmid's inequality, for any $\delta > 0$, with probability at least $1 - \delta$,

$$\mathop{\mathbb{E}}_{z \sim \mathcal{D}_\lambda}[g(z)] \leq \mathop{\mathbb{E}}_{z \sim \overline{\mathcal{D}}_\lambda}[g(z)] + 2\mathfrak{R}_{\mathbf{m}}(\mathcal{G}, \lambda) + \sqrt{\sum_{k=1}^{p} \frac{\lambda_k^2}{2m_k} \log \frac{1}{\delta}}.$$

Let $\overline{\Lambda}$ be the set of vertices of a subsimplicial cover of Λ, that is a decomposition of a cover of Λ into subsimplices. When the subsimples are formed by vertices that are ϵ-close in ℓ_1-distance, then $\overline{\Lambda}$ is an ϵ-cover of Λ for the ℓ_1-distance. Figure 4 illustrates the sets λ and $\overline{\Lambda}$. By the union bound, with probability at least $1 - \delta$, the following holds for all $\lambda \in \overline{\Lambda}$:

$$\mathop{\mathbb{E}}_{z \sim \mathcal{D}_\lambda}[g(z)] \leq \mathop{\mathbb{E}}_{z \sim \overline{\mathcal{D}}_\lambda}[g(z)] + 2\mathfrak{R}_{\mathbf{m}}(\mathcal{G}, \lambda) + \sqrt{\sum_{k=1}^{p} \frac{\lambda_k^2}{2m_k} \log \frac{|\overline{\Lambda}|}{\delta}}.$$

Now, fix $\rho > 0$. Let $\mathcal{H}$ be a hypothesis set of real-valued functions and let ϕ_ρ denote the ρ-ramp loss. Let $\mathcal{G}$ be the family of ρ-ramp losses of functions in $\mathcal{H}$: $\mathcal{G} = \{z = (x, y) \mapsto \phi_\rho(yh(x)) \colon h \in \mathcal{H}\}$. Then, proceeding as with the derivation of margin-based Rademacher complexity bounds in the standard case and using the $\frac{1}{\rho}$-Lipschitzness of the ρ-ramp loss (Mohri et al., 2018), we obtained that, with probability at least $1 - \delta$, the following holds for all $\lambda \in \overline{\Lambda}$:

$$\mathcal{L}(\mathcal{D}_\lambda, h) \leq \mathcal{L}_\rho(\overline{\mathcal{D}}_\lambda, h) + \frac{2}{\rho}\mathfrak{R}_{\mathbf{m}}(\mathcal{H}, \lambda) + \sqrt{\sum_{k=1}^{p} \frac{\lambda_k^2}{2m_k} \log \frac{|\overline{\Lambda}|}{\delta}}.$$

Now, for any $\lambda, \lambda' \in \Delta$ with $\|\lambda - \lambda'\|_1 \leq \epsilon$, we have:

$$\begin{aligned}
\sum_{k=1}^{p} \frac{\lambda_k'^2}{2m_k} &= \sum_{k=1}^{p} \frac{(\lambda_k' - \lambda_k + \lambda_k)^2}{2m_k} \\
&= \sum_{k=1}^{p} \frac{\lambda_k^2 + 2(\lambda_k' - \lambda_k)\lambda_k + (\lambda_k' - \lambda_k)^2}{2m_k} \\
&= \sum_{k=1}^{p} \frac{\lambda_k^2}{2m_k} + \sum_{k=1}^{p} \frac{2|\lambda_k' - \lambda_k|\lambda_k + (\lambda_k' - \lambda_k)^2}{2m_k} \\
&\leq \sum_{k=1}^{p} \frac{\lambda_k^2}{2m_k} + \epsilon \max_{k \in [p]} \frac{\lambda_k}{m_k} + \frac{\epsilon^2}{2} \max_{k \in [p]} \frac{1}{m_k} && \text{(Hölder's inequality)} \\
&\leq \sum_{k=1}^{p} \frac{\lambda_k^2}{2m_k} + \frac{3\epsilon}{2}.
\end{aligned}$$

Let $\overline{\Lambda}_\epsilon$ denote the family of λs that are ϵ-close to $\overline{\Lambda}$ in ℓ_1-distance, then, for any $\lambda \in \overline{\Lambda}_\epsilon$ we have:

$$\mathcal{L}(\mathcal{D}_\lambda, h) \leq \mathcal{L}_\rho(\overline{\mathcal{D}}_\lambda, h) + \frac{2}{\rho}\mathfrak{R}_{\mathbf{m}}(\mathcal{H}, \lambda) + \frac{3\epsilon}{2} + \sqrt{\sum_{k=1}^{p} \frac{\lambda_k^2}{2m_k} \log \frac{|\overline{\Lambda}|}{\delta}}.$$

Also, for any λ in a subsimplex formed by elements of $\overline{\Lambda}$, there exist $\mu = (\mu_1, \ldots, \mu_p)$ and $\beta_1, \ldots, \beta_p$ in $\overline{\Lambda}$ such that $\lambda = \sum_{k=1}^{p} \mu_k \beta_k$. Thus, for any such λ, we have

$$\mathcal{L}(\mathcal{D}_\lambda, h) \leq \mathcal{L}_\rho(\overline{\mathcal{D}}_\lambda, h) + \frac{2}{\rho} \sum_{l=1}^{p} \mu_k \mathfrak{R}_{\mathbf{m}}(\mathcal{H}, \beta_l) + \sum_{k=1}^{p} \mu_k \sqrt{\sum_{l=1}^{p} \frac{\beta_{l,k}^2}{2m_k} \log \frac{|\overline{\Lambda}|}{\delta}}.$$

Applying these results to the analysis of the Q-ensembles we are interested yields the following margin-based generalization bounds.

Theorem 2. *Fix $\rho > 0$ and $\epsilon > 0$. Then, for any $\delta > 0$, with probability at least $1 - \delta$ over the draw of a sample $S = (S_1, \ldots, S_p) \sim \mathcal{D}_1^{m_1} \otimes \cdots \otimes \mathcal{D}_p^{m_p}$, each of the following inequalities holds:*
1. for all ensemble functions $f = \sum_{t=1}^{T} \alpha_t \mathsf{Q}(k_t|\cdot) h_t \in \mathcal{F}$ and all $\lambda \in \overline{\Lambda}_\epsilon$:

$$\mathcal{L}(\mathcal{D}_\lambda, h) \leq \mathcal{L}_\rho(\overline{\mathcal{D}}_\lambda, h) + \frac{2}{\rho} \max_{r \in [p]} \mathfrak{R}_{\mathbf{m}}(\mathcal{H}_r, \lambda) + \frac{3\epsilon}{2} + \sqrt{\sum_{k=1}^{p} \frac{\lambda_k^2}{2m_k} \log \frac{|\overline{\Lambda}|}{\delta}}. \tag{6}$$

2. for all ensemble functions $f = \sum_{t=1}^{T} \alpha_t \mathsf{Q}(k_t|\cdot) h_t \in \mathcal{F}$ and all $\lambda = \sum_{k=1}^{p} \mu_k \beta_k \in \mathrm{conv}(\overline{\Lambda})$:

$$\mathcal{L}(\mathcal{D}_\lambda, h) \leq \mathcal{L}_\rho(\overline{\mathcal{D}}_\lambda, h) + \frac{2}{\rho} \sum_{l=1}^{p} \mu_k \max_{r \in [p]} \mathfrak{R}_{\mathbf{m}}(\mathcal{H}_r, \beta_l) + \sum_{k=1}^{p} \mu_k \sqrt{\sum_{l=1}^{p} \frac{\beta_{l,k}^2}{2m_k} \log \frac{|\overline{\Lambda}|}{\delta}}. \tag{7}$$

Note that, for a given $\lambda \in \Lambda$, the most favorable of the two statements of the theorem can be used. Observe also that the second learning bound coincides with that of Theorem 1 when $\overline{\Lambda}$ is chosen to be the vertices of the simplex Δ since in that case $|\overline{\Lambda}| = p$, $\mathrm{conv}(\overline{\Lambda}) = \Delta$, and since $\mathfrak{R}_{\mathbf{m}}(\mathcal{H}_r, \beta_l) = \mathfrak{R}_{\mathbf{m}}(\mathcal{H}_r, \beta_l)$ then coincides with $\mathfrak{R}_{m_l}(\mathcal{H}_r)$. Choosing the best statement of the theorem therefore always provides a finer guarantee than that of Theorem 1.

When Λ is a small set, for example the set of λs ϵ-close to a finite set of discrete elements $\overline{\Lambda}$, then the last term of the learning bound of the first statement can be more favorable that of Theorem 1 since $|\overline{\Lambda}|$ can then be in the same order as p or smaller while, by the sub-additivity of the square-root, the following inequality holds: $\sqrt{\sum_{k=1}^{p} \frac{\lambda_k^2}{m_k}} \leq \sum_{k=1}^{p} \sqrt{\frac{\lambda_k^2}{m_k}} = \sum_{k=1}^{p} \lambda_k \sqrt{\frac{1}{m_k}}$.

The theorem suggests a regularization term of the form $\sum_{k=1}^{p} \frac{\lambda_k^2}{m_k}$, especially in the case where Λ is a small subset of the simplex, which can lead to better algorithms in that case.

G FEDMULTIBOOST: related work and experiments

Following MULTIBOOST, we propose a boosting-style approach with the agnostic loss. Boosting in federated learning was first studied by Hamer et al. (2020). The authors proposed a communication-efficient algorithm for minimizing the standard empirical risk and agnostic loss, based on mirror descent. However, their algorithm is optimal only for density estimation (Hamer et al., 2020, Section 3.2) and is sub-optimal for general classification tasks such as in Proposition 1. Furthermore, their mirror descent solution is inadequate for the boosting framework in this paper, where a (block) coordinate descent approach for learning sparser solutions is critical. Recently, Shen et al. (2021) proposed a functional gradient boosting algorithm for federated learning. Their algorithm iteratively determines base classifiers and mixing weights to compute a convex combination in a distributed manner. However, their algorithm minimizes the uniform loss over all samples. In contrast, we propose to minimize the agnostic loss over multiple domains, which is more risk-averse, and seek Q-ensembles which are more adequate than convex combinations in a multiple-source scenario.

Federated learning experiments. We used the *EMNIST* dataset (Caldas et al., 2018; Bonawitz et al., 2019), which contains 32x32 pixel handwritten digits images annotated by users. The images are divided into $p = 2$ sources based on the group of writers that provided the annotation: high school ($k = 1$) and census ($k = 2$). We compared algorithms on two binary classification problems: digits 4 vs. 9 and digits 1 vs. 7. The results are presented in Table 2. The error bars are obtained from breaking the set of clients into 10 random folds.

We compared FEDMULTIBOOST with three benchmarks, FEDADABOOST-k for $k \in [p]$ and FEDADABOOST-all. The former is a federated version of ADABOOST algorithm trained only on a single source k and the latter is federated ADABOOST trained on both the sources. In the federated ADABOOST versions, at each boosting step we randomly select 20 clients and train weak learners on each of those clients, next, the server selects the weak learner with the smallest weighted error and adds it to the ensemble. For the FEDMULTIBOOST algorithm, we randomly select 20 clients per round. Since the number of clients sampled at each round is small, we run each algorithm for 500 boosting steps.

As can be seen from Table 2, FEDMULTIBOOST provides agnostic and uniform errors that are significantly better than the baselines on both the datasets.

Table 2: Test errors for multiple benchmarks in the federated setting.

Algorithm	Error-1	Error-2	Error-Uniform	Error-Agnostic
		EMNIST (4 vs. 9)		
FEDADABOOST-1	0.075 ± 0.008	0.133 ± 0.014	0.104 ± 0.009	0.133 ± 0.014
FEDADABOOST-2	0.095 ± 0.009	0.096 ± 0.014	0.095 ± 0.012	0.096 ± 0.014
FEDADABOOST-all	0.076 ± 0.006	0.125 ± 0.016	0.101 ± 0.011	0.125 ± 0.007
FEDMULTIBOOST	$\mathbf{0.064 \pm 0.013}$	$\mathbf{0.076 \pm 0.008}$	$\mathbf{0.070 \pm 0.009}$	$\mathbf{0.076 \pm 0.016}$
		EMNIST (1 vs. 7)		
FEDADABOOST-1	$\mathbf{0.029 \pm 0.010}$	0.050 ± 0.009	0.039 ± 0.011	0.050 ± 0.009
FEDADABOOST-2	0.062 ± 0.014	$\mathbf{0.030 \pm 0.007}$	0.046 ± 0.014	0.062 ± 0.007
FEDADABOOST-all	0.032 ± 0.007	0.043 ± 0.006	0.037 ± 0.007	0.043 ± 0.014
FEDMULTIBOOST	0.030 ± 0.008	0.035 ± 0.006	$\mathbf{0.032 \pm 0.008}$	$\mathbf{0.035 \pm 0.010}$

H Dataset references and details

- MNIST: `http://yann.lecun.com/exdb/mnist/`
- SVHN: `http://ufldl.stanford.edu/housenumbers/`
- MNIST-M: `http://yaroslav.ganin.net/`
- SENTIMENT: `https://www.cs.jhu.edu/~mdredze/datasets/sentiment`
- FASHION-MNIST: `https://github.com/zalandoresearch/fashion-mnist`
- ADULT: `https://archive.ics.uci.edu/ml/datasets/adult`

Table 3: Number of examples per domain for each classification problem.

Problem	Source k=1	Source k=2	Source k=3	Total
Sentiment Analysis	2,000	2,000	2,000	6,000
Digit Classification (1 vs 7)	15,170	26,574	14,728	56,472
Digit Classification (4 vs 9)	13,782	16,235	13,406	43,423
Object Recognition	21,000	21,000	28,000	70,000
Tabular Data (Adult Data)	10,628	14,783	19,811	45,222

I Supplementary plots

This section contains additional plots illustrating the performance of the proposed MULTIBOOST algorithm. We illustrate convergence characteristics, Figure 7, $\mathsf{Q}(k|\cdot_k)$ functions, Figure 5, and α-mass distributions over the domains and $\mathsf{Q}(k|\cdot_k)$ functions, Figure 6.

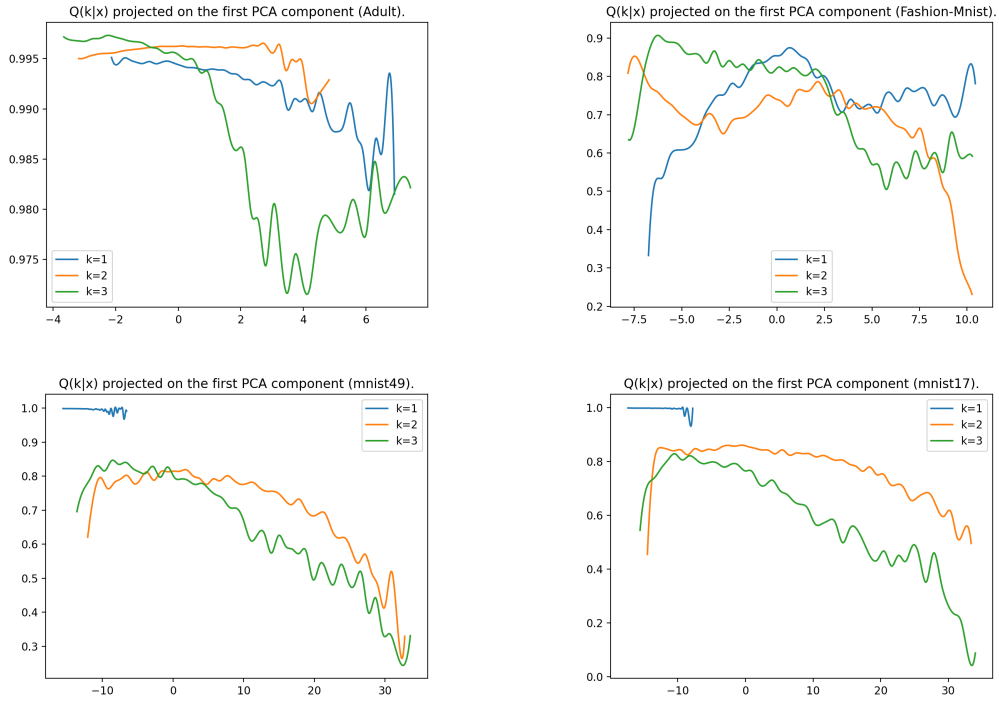

Figure 5: Mean values of $\mathsf{Q}(k|\cdot_k)$ for the three domains of the data projected on the first principal component of the joint data.
Top, left: *Adult* data. Source $k = 1$ consists of individuals with a university degree (BSc, MS or PhD), source $k = 2$ those with only a High School diploma, and source $k = 3$, none of the above.
Top, right: *Fashion-MNIST* data. Source $k = 1$ consists of `t-shirts, pullovers, trousers`; $k = 2$ consists of `dresses, coats, sandals`; $k = 3$ consists of `shirts, bags, sneakers, ankle boots`.
Bottom, left: Digits (4 vs. 9), where pixel handwritten digits images are taken from sources: *MNIST* ($k = 1$), *SVHN* ($k = 2$), *MNIST-M* ($k = 3$).
Bottom, right: Digits (1 vs. 7), where pixel handwritten digits images are taken from sources: *MNIST* ($k = 1$), *SVHN* ($k = 2$), *MNIST-M* ($k = 3$).
Note that for the two bottom plots domain $k = 1$ is significantly further separated from the other two domains, since the pixels for $k = 1$ are [illegible].

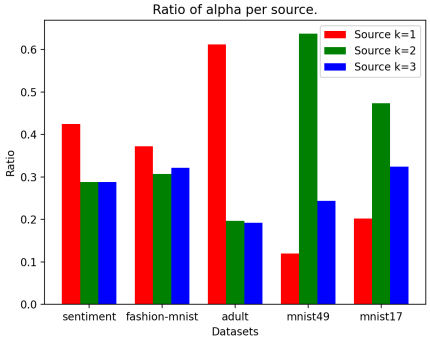

Figure 6: The ratio of ensemble weights $\alpha_{k,j}$ after training that corresponds to each source $k = 1, 2, 3$. For each dataset and fixed k, the bar corresponds to $\sum_{j=1}^{N_k} \alpha_{k,j} / \sum_{k=1}^{p} \sum_{j=1}^{N_k} \alpha_{k,j}$.

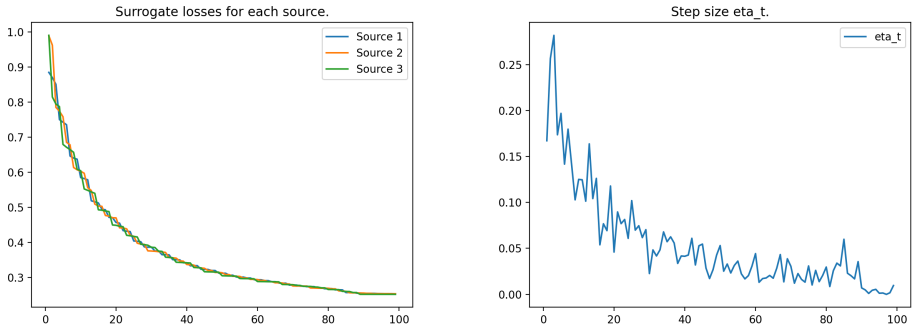

Figure 7: Left: The evolution of surrogate losses Φ for sources $k = 1, 2, 3$ during MULTIBOOST training on object recognition problem (*Fashion-MNIST*). Line k corresponds to $\frac{1}{m_k} \sum_{i=1}^{m_k} \Phi(y_i, f_t(x_i))$ as a function of t, where f_t is the MULTIBOOST ensemble of weak learners obtained at round t. Right: The evolution of the step size η_t during MULTIBOOST training on object recognition problem (*Fashion-MNIST*).

J Comparison with multiple-source adaptation

In this section, we compare our results with an alternative solution based on using a multiple-source adaptation algorithm (Cortes, Mohri, Suresh, and Zhang, 2021; Mansour, Mohri, and Rostamizadeh, 2008, 2009a; Hoffman, Mohri, and Zhang, 2018, 2021). Let us emphasize first that these algorithms are designed for a distinct scenario than the one studied in this paper. They assume no access to labeled data and instead only to a good predictor and unlabeled data for each domain.

The method consists of first training AdaBoost on each domain, which provides an accurate predictor f_k for each domain k. Next, we can use the discriminative technique for multiple-source adaptation (DMSA algorithm) recently presented by Cortes, Mohri, Suresh, and Zhang (2021) to combine these predictors to derive a solution that is robust for any target mixture distribution of the source domains. This algorithm was shown to outperform the GMSA algorithm of Hoffman, Mohri, and Zhang (2018) and Hoffman, Mohri, and Zhang (2021), which is based on density estimation (note that previous work by Mansour, Mohri, and Rostamizadeh (2008, 2009a) did not provide an actual algorithm for this problem), which itself was shown to surpass other existing algorithms for this scenario.

As mentioned earlier, the idea of Q-ensembles in our boosting context is inspired by the distribution-weighted combinations of Mansour, Mohri, and Rostamizadeh (2008, 2009a) or Hoffman, Mohri, and Zhang (2018, 2021) or the domain-classifier based combinations of Cortes, Mohri, Suresh, and Zhang (2021). Our basic motivation via Propositions 1 and 2 are also similar. However, the main technical content of this work and the contributions are all novel. Our formulation of the multiple-source boosting problem, including our weak learning assumption, our algorithmic solutions, including an extension to the Federated Learning setting, and our theoretical analysis of the problem, including finer margin-based learning bounds, our extensive experimental results, are all novel and unrelated to that previous work.

In Section 6, we report experimental results for this AdaBoost and DMSA-based algorithm, and compare them with MULTIBOOST. Note that, as already mentioned, DMSA is designed for a different scenario from the one studied in this paper where no access to labeled data is assumed. The empirical results suggest that MULTIBOOST outperforms DMSA, although DMSA often achieves a competitive performance in the tasks examined.

Let us emphasize, however, that the predictor for the DMSA algorithm (or GMSA) in general does not benefit from informative guarantees in our scenario, for the following reasons.

Target labeling function assumption: the main analysis and results in (Mansour, Mohri, and Rostamizadeh, 2008, 2009a; Hoffman, Mohri, and Zhang, 2018, 2021; Cortes, Mohri, Suresh, and Zhang, 2021) require that the target labeling function (or conditional probability of Y given X for an extension of that analysis) be the same for all domains. This is a strong condition that may not hold in practice and that is not required for our learning guarantees for MULTIBOOST.

Loss function: the guarantees in (Mansour, Mohri, and Rostamizadeh, 2008, 2009a; Hoffman, Mohri, and Zhang, 2018, 2021; Cortes, Mohri, Suresh, and Zhang, 2021) hold only for a continuous loss function, since they rely on Brouwer's fixed-point theorem. In particular, they do not hold for the binary or multi-class mis-classification losses considered here.

One can resort instead to a convex surrogate loss (such a guarantee would be then in terms of the convex loss of the predictors to combine and not their more favorable zero-one loss). But the guarantees in (Mansour, Mohri, and Rostamizadeh, 2008, 2009a; Hoffman, Mohri, and Zhang, 2018, 2021; Cortes, Mohri, Suresh, and Zhang, 2021) also require the loss to be bounded, which would not hold for an unbounded domain. Even for a bounded domain, the bound could be large and the value of the convex loss on the boundary even larger (exponentially larger for AdaBoost) thereby making the desired guarantee essentially vacuous. In contrast, our guarantees hold for the zero-one loss.

Algorithms: the technique of Mansour, Mohri, and Rostamizadeh (2008, 2009a) and Hoffman, Mohri, and Zhang (2018, 2021) requires density estimation for each domain. With estimated densities, the guarantee becomes somewhat looser. Furthermore, the algorithmic solutions of (Hoffman, Mohri, and Zhang, 2018, 2021; Cortes, Mohri, Suresh, and Zhang, 2021) are not based on a convex optimization and thus cannot directly benefit from the theoretical bound, even if it could be applicable (see above).

Hypothesis set: ignoring the normalization, which does not affect the definition of the classifier, the DMSA solution is a specific element of the family of ensembles our algorithm and learning bounds consider. Our algorithm seeks the most favorable ensemble using all the available labeled data, which in general could be quite different and more favorable than the predictor obtained by combining Adaboost and DMSA.

More generally, our algorithm is not restricted to deriving intermediate predictors for each domain and instead directly exploits the labeled training samples from all the sources simultaneously to find a single good predictor. Consider the extreme case where all sources follow the same distribution and all training sets have the same size. The AdaBoost and DMSA-based algorithm consists of first training one AdaBoost model for each source. Assuming perfect density estimation or domain classification, DMSA would then return the uniform average $\frac{1}{p}\sum_{k=1}^{p} f_k$ with each f_k trained on m/p samples. Instead, MULTIBOOST returns a single model trained on all m samples that, in general, can be far superior. To further illustrate this, we ran this experiment on the SVHN dataset with 3 *sources* defined by sub-samples from an original training sample. This led to an error of 26.1% for the predictor obtained via DMSA and only 22.8% for MULTIBOOST.

K The impact of domain classifier Q

We here present results illustrating the importance of selecting a high-accuracy domain classifier Q in the MULTIBOOST algorithm. We experimented with different Q functions by varying the number of steps, $C = 2, 5, 10, 1000$, in training the logistic regression optimizer used to obtain Q. The lower the number of steps in the logistic regression optimizer, the lower the quality of the domain classifier Q, and thus the higher the classification error on the underlying task. In the original MULTIBOOST implementation, the maximum number of L-BFGS steps for the domain classifier Q is set to 1000 by default.

Table 4: Test errors for original MULTIBOOST and MULTIBOOST-C, where C is the maximum number of steps in the L-BFGS optimizer used to fit the multinomial logistic regression as domain classifier Q. In the original MULTIBOOST implementation, the maximum number of L-BFGS steps for the domain classifier Q is set to 1000 by default.

Algorithm-C	Error-1	Error-2	Error-3	Error-Uniform	Error-Agnostic
			Digits Recognition (1 vs. 7)		
MULTIBOOST	**0.026 ± 0.004**	**0.261 ± 0.013**	**0.257 ± 0.015**	**0.181 ± 0.005**	**0.261 ± 0.011**
MULTIBOOST-10	0.029 ± 0.005	0.262 ± 0.013	0.299 ± 0.015	0.197 ± 0.010	0.299 ± 0.011
MULTIBOOST-5	0.032 ± 0.005	0.277 ± 0.012	0.323 ± 0.017	0.211 ± 0.012	0.323 ± 0.015
MULTIBOOST-2	0.127 ± 0.009	0.281 ± 0.015	0.351 ± 0.021	0.253 ± 0.017	0.351 ± 0.019

L Multi-Class extension of MULTIBOOST

Here, we briefly describe the extension of MULTIBOOST to the multi-class setting, MCMULTIBOOST

We denote by $\mathcal{Y}$ the set of output labels or categories. The label associated by a hypothesis $f\colon \mathcal{X}\times\mathcal{Y} \to \mathbb{R}$ to input $x \in \mathcal{X}$ is given by $\operatorname{argmax}_{y\in\mathcal{Y}} f(x,y)$. The margin $\rho_f(x,y)$ of the function f for a labeled example $(x,y) \in \mathcal{X}\times\mathcal{Y}$ is defined by

$$\rho_f(x,y) = f(x,y) - \max_{y'\neq y} f(x,y'). \tag{15}$$

Thus, f misclassifies (x,y) when $\rho_f(x,y) \leq 0$.

As in the binary classification case, we consider a scenario where the learner receives labeled samples from p source domains, each defined by a distribution $\mathcal{D}_k$ over $\mathcal{X}\times\mathcal{Y}$, $k \in [p]$. We denote by $S_k = ((x_{k,1}, y_{k,1}), \ldots, (x_{k,m_k}, y_{k,m_k})) \in \left(\mathcal{X}\times\mathcal{Y}\right)^{m_k}$ the labeled sample of size m_k received from source k, which is drawn i.i.d. from $\mathcal{D}_k$. For any function $f\colon \mathcal{X}\times\mathcal{Y}\to\mathbb{R}$ and distribution $\mathcal{D}$ over $\mathcal{X}\times\mathcal{Y}$, let $\mathcal{L}(\mathcal{D}, f)$ be the expected loss of f, that is $\mathcal{L}(\mathcal{D}, f) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(f,x,y)]$, where ℓ is the multi-class loss $\ell(f(x),y) = \mathbb{I}(\rho_f(x,y)\leq 0)$.

For any $k \in [p]$, let $\mathcal{H}_k$ denote a hypothesis set of functions mapping from $\mathcal{X}\times\mathcal{Y}$ to the interval $[-1,+1]$, $|\mathcal{H}_k| = N_k$. The objective of the learner is to find an accurate predictor f for any target distribution $\mathcal{D}_\lambda$ that is a mixture of the source distributions, where λ may be in a subset Λ of the simplex.

L.1 Form of solution

In the multi-class setting, the general form of our Q-ensemble predictor is the following:

$$\forall (x,y) \in \mathcal{X}\times\mathcal{Y},\ f(x,y) = \sum_{l=1}^{p}\sum_{j=1}^{N_l} \alpha_{l,j}\mathsf{Q}(l|x)h_{l,j}(x,y), \tag{16}$$

where $h_{l,j} \in \mathcal{H}_l$, $\alpha_{l,j} \geq 0$ and $\sum_{j=1}^{N_l}\alpha_{l,j} = 1$ and where $\mathsf{Q}(l|x)$ denotes the conditional probability of domain l given x.

L.2 Weak learning assumption

As in the binary classification scenario, we will adopt a weak-learning assumption. Unlike the standard single source, our assumption here must hold for each source domain: for each domain $k \in [p]$ and any distribution D over $S_k \times \mathcal{Y}$, there exists a base classifier $h \in \mathcal{H}_k$ such that the weighted loss of h is γ-better than random: $\frac{1}{2}[1 - \mathbb{E}_{(i,y)\sim\mathsf{D}}[h(x_{k,i},y_{k,i}) - h(x_{k,i},y)] \leq \frac{1}{2} - \gamma$, for some edge value $\gamma > 0$. This is equivalent to the existence of a weak-learning for a each domain, which is a mild assumption. As in the standard boosting scenario, this suggests that there exists a good *rule of thumb* for each domain. The key difference from the standard learning scenario, however, is that here we seek a Q-ensemble and further require it to be accurate for any target mixture $\mathcal{D}_\lambda$, $\lambda \in \Lambda$. In the next subsection, we present an algorithm, MCMULTIBOOST, for deriving an accurate Q-ensemble for any target mixture domain that belongs to the convex combination of the source domains.

L.3 Algorithm

Let Φ be a convex, increasing and differentiable function such that $u \mapsto \Phi(-u)$ upper-bounds the binary loss $u \mapsto 1_{u\leq 0}$. Φ could be the exponential function as in ADABOOST or the logistic function, as in logistic regression. Using Φ to upper-bound the agnostic loss leads to the following objective function for an ensemble f defined by (16) for any $\alpha = (\alpha_{l,j})_{(l,j)\in[p]\times[N_l]} \geq 0$:

$$F(\alpha) = \max_{\lambda\in\Lambda} \sum_{k=1}^{p} \frac{\lambda_k}{m_k} \sum_{i=1}^{m_k}\sum_{y\in\mathcal{Y}} \Phi\left(-\sum_{l=1}^{p}\sum_{j=1}^{N_l}\alpha_{l,j}\mathsf{Q}(l|x_{k,i})\mathsf{h}_{l,j}(x_{k,i},y_{k,i},y)\right), \tag{17}$$

where $\mathsf{h}_{l,j}(x_{k,i},y_{k,i},y) = h_{l,j}(x_{k,i},y_{k,i}) - h_{l,j}(x_{k,i},y)$.

Here, we will consider the case where the set Λ coincides with the full simplex, that is $\Lambda = \Delta$. F can then be expressed more straightforwardly as $F = \max_{k=1}^{p} F_k$, with $F_k(\alpha) = \frac{1}{m_k} \sum_{i=1}^{m_k} \sum_{y \in \mathcal{Y}} \Phi\left(- \sum_{l=1}^{p} \sum_{j=1}^{N_l} \alpha_{l,j} \mathsf{Q}(l|x_{k,i}) \mathsf{h}_{l,j}(x_{k,i}, y_{k,i}, y) \right)$.

Since a convex function composed with an affine function is convex and a sum of convex functions is convex, F is convex as the maximum of a set of convex functions. While convex, F is not differentiable and, in general, coordinate descent may not succeed in such cases (Tseng, 2001; Luo and Tseng, 1992). This is because the algorithm may be *stuck* at a point where no progress is possible along any of the axis directions, while there exists a favorable descent along some other direction. However, we will show that, under the weak-learning assumption we adopted, at any point $\boldsymbol{\alpha}$ and for each active function F_k, that is F_k such that $F_k(\boldsymbol{\alpha}) = F(\boldsymbol{\alpha})$, there exists a coordinate direction along which a descent progress is possible for each F_k. We will assume that these directions are also descent directions for F. More generally, it suffices in fact that one such direction admits this guarantee. Alternatively, as in the binary classification setting, one can replace the maximum with a *soft-max*, that is the $(x_1, \ldots, x_k) \mapsto \frac{1}{\mu} \log(\sum_{k=1}^{p} e^{\mu x_k})$ for $\mu > 0$, which leads to a continuously differentiable function with a close approximation for μ sufficiently large.

Description. Let α_{t-1} denote the value of the parameter vector $\alpha = (\alpha_{l,j})$ at the end of the $(t-1)$th iteration and let f_{t-1} be defined by $\mathsf{f}_{t-1} = \sum_{l=1}^{p} \sum_{j=1}^{N_l} \alpha_{t-1,l,j} \mathsf{Q}(l|\cdot)\, \mathsf{h}_{l,j}$. Coordinate descent at iteration t consists of choosing a direction $\mathbf{e}_{q,r}$ corresponding to base classifier $h_{q,r}$ and a step value $\eta > 0$ to minimize $F(\alpha_{t-1} + \eta\, \mathbf{e}_{q,r})$. To select a direction, we consider the subdifferential of F along any $\mathbf{e}_{q,r}$. Since functions F_k are differentiable, by the subdifferential calculus for the maximum of functions, the subdifferential of F at α_{t-1} along the direction $\mathbf{e}_{q,r}$ is given by:

$$\partial F(\alpha_{t-1}, \mathbf{e}_{q,r}) = \operatorname{conv}\{F'_k(\alpha_{t-1}, \mathbf{e}_{q,r}) \colon k \in \mathcal{K}_t\},$$

where $F'_k(\alpha_{t-1}, \mathbf{e}_{q,r})$ is the directional derivative of F_k at α_{t-1} along the direction $\mathbf{e}_{q,r}$ and where $\mathcal{K}_t = \{k \in [p] \colon F_k(\alpha_{t-1}) = F(\alpha_{t-1})\}$. We will therefore consider the direction $\mathbf{e}_{q,r}$ with the largest absolute directional derivative value $|F'_k(\alpha_{t-1}, \mathbf{e}_{q,r})|$, $k \in \mathcal{K}_t$, but will restrict ourselves to $q = k$ since, as we shall see, that will be sufficient to guarantee a non-zero directional gradient. To do so, we will express $F'_k(\alpha_{t-1}, \mathbf{e}_{k,r})$ in terms of the distribution $\mathsf{D}_t(k, \cdot, \cdot)$ over $S_k \times \mathcal{Y}$ defined by $\mathsf{D}_t(k, i, y) = \frac{\Phi'(-\mathsf{f}_{t-1}(x_{k,i}, y_{k,i}, y))}{Z_{t,k}}$, with $Z_{t,k} = \sum_{i=1}^{m_k} \sum_{y \in \mathcal{Y}} \Phi'(-\mathsf{f}_{t-1}(x_{k,i}, y_{k,i}, y))$, for all $i \in [m_k]$ and $y \in \mathcal{Y}$:

$$\begin{aligned} F'_k(\alpha_{t-1}, h_{k,r}) &= \frac{1}{m_k} \sum_{i=1}^{m_k} \sum_{y \in \mathcal{Y}} -\mathsf{Q}(k|x_{k,i}) \mathsf{h}_{k,r}(x_{k,i}, y_{k,i}, y) \Phi'\big(-\mathsf{f}_{t-1}(x_{k,i}, y_{k,i}, y)\big) \\ &= \frac{Z_{t,k}}{m_k} [2\epsilon_{t,k,r} - 1], \end{aligned}$$

where $\epsilon_{t,k,r} = \frac{1}{2}\big[1 - \mathbb{E}_{(i,y) \sim \mathsf{D}_t(k,\cdot,\cdot)}[\mathsf{Q}(k|x_{k,i}) \mathsf{h}_{k,r}(x_{k,i}, y_{k,i}, y)]\big]$ denotes the *weighted error* of $\mathsf{Q}(k|\cdot) h_{k,r}$. For any $s \in [m_k]$, since $x_{k,s}$ is a sample drawn from $\mathcal{D}_k$, we have $\mathsf{Q}(k|x_{k,s}) > 0$ and therefore we have: $\overline{\mathsf{Q}}_{t,k} = \sum_{s=1}^{m_k} \sum_{y \in \mathcal{Y}} \mathsf{D}_t(k, s, y) \mathsf{Q}(k|x_{k,s}) > 0$. Thus, we can write $\mathbb{E}_{(i,y) \sim \mathsf{D}_t(k,\cdot,\cdot)}[\mathsf{Q}(k|x_{k,i}) \mathsf{h}_{k,r}(x_{k,i}, y_{k,i}, y)]$ as

$$\sum_{i=1}^{m_k} \sum_{y \in \mathcal{Y}} \frac{\mathsf{D}_t(k, i, y) \mathsf{Q}(k|x_{k,i}) \mathsf{h}_{k,r}(x_{k,i}, y_{k,i}, y)}{\overline{\mathsf{Q}}_{t,k}} \overline{\mathsf{Q}}_{t,k}, = \mathop{\mathbb{E}}_{(i,y) \sim \mathsf{D}'_t(k,\cdot,\cdot)} [\mathsf{h}_{k,r}(x_{k,i}, y_{k,i}, y)]\, \overline{\mathsf{Q}}_{t,k},$$

where $\mathsf{D}'_t(k, i, y) = \frac{\mathsf{D}_t(k,i,y) \mathsf{Q}(k|x_{k,i})}{\overline{\mathsf{Q}}_{t,k}}$. By our weak-learning assumption, there exists $r \in N_k$ such that $\mathbb{E}_{(i,y) \sim \mathsf{D}'_t(k,\cdot,\cdot)}[\mathsf{h}_{k,r}(x_{k,i}, y_{k,i}, y)] \geq \gamma > 0$. For that choice of r, we have $\epsilon_{t,k,r} < \frac{1}{2} - \overline{\gamma}$, with $\overline{\gamma} = \gamma \overline{\mathsf{Q}}_{t,k} > 0$. In view of that, it suffices for us to search along the directions $h_{k,r}$ and we do not need to consider the directional derivative of F_k along directions $h_{q,r}$ with $q \neq k$.

The direction chosen by our coordinate descent algorithm is thus defined by: $\operatorname{argmax}_{k \in \mathcal{K}_t, r \in [N_k]} \frac{Z_{t,k}}{m_k}[1 - 2\epsilon_{t,k,r}]$. Given the direction $\mathbf{e}_{k,r}$, the optimal step value η is $\operatorname{argmin}_{\eta > 0} F(\alpha_{t-1} + \eta \mathbf{e}_{k,r})$. The pseudocode of our algorithm, MCMULTIBOOST, is provided in Figure 8. In the most general case, η can be found via a line search or other numerical methods.

Step size. In some special cases, the line search can be executed using a simpler expression by using an upper bound on $F(\alpha_{t-1} + \eta \mathbf{e}_{k,r})$. Denoting $z_{l,i} = (x_{l,i}, y_{l,i}, y)$ and using the convexity of Φ, since $\mathsf{Q}(k|x_{l,i})\mathsf{h}_{k,r}(z_{l,i}) = \frac{1+\mathsf{Q}(k|x_{l,i})\mathsf{h}_{k,r}(z_{l,i})}{2} \cdot (+1) + \frac{1-\mathsf{Q}(k|x_{l,i})\mathsf{h}_{k,r}(z_{l,i})}{2} \cdot (-1)$, the following holds for all $\eta \in \mathbb{R}$:

$$\Phi(-\mathsf{f}_{t-1}(z_{l,i}) - \eta \mathsf{Q}(k|x_{l,i})\mathsf{h}_{k,r}(z_{l,i})) \leq \frac{1+\mathsf{Q}(k|x_{l,i})\mathsf{h}_{k,r}(z_{l,i})}{2} \Phi(-\mathsf{f}_{t-1}(z_{l,i}) - \eta) + \frac{1-\mathsf{Q}(k|x_{l,i})\mathsf{h}_{k,r}(z_{l,i})}{2} \Phi(-\mathsf{f}_{t-1}(z_{l,i}) + \eta).$$

In the case of exponential and logistic functions, the following upper bounds can then be derived.

Lemma 2. *For any $k \in [p]$, the following upper bound holds when Φ is the exponential or the logistic function:*

$$F(\alpha_{t-1} + \eta \mathbf{e}_{k,r}) \leq \max_{l \in [p]} \frac{Z_{t,l}}{m_l} \left[(1 - \epsilon_{t,l,k,r})e^{-\eta} + \epsilon_{t,l,k,r} e^{\eta}\right],$$

where $\epsilon_{t,l,k,r} = \frac{1}{2}\left[1 - \mathbb{E}_{(i,y)\sim \mathsf{D}_t(l,\cdot,\cdot)}[\mathsf{Q}(k|x_{l,i})\mathsf{h}_{k,r}(x_{l,i}, y_{l,i}, y)]\right]$.

For any k, function $\eta \mapsto (1 - \epsilon_{t,l,k,r})e^{-\eta} + \epsilon_{t,l,k,r}e^{\eta}$ reaches its minimum for $\eta = \frac{1}{2}\log\frac{1-\epsilon_{t,l,k,r}}{\epsilon_{t,l,k,r}}$. When the maximum is achieved for $l = k$, the solution coincides with the familiar expression of the step size $\eta_t = \frac{1}{2}\log\frac{1-\epsilon_{t,k,r}}{\epsilon_{t,k,r}}$ used in AdaBoost.

Q-function. As discussed in the binary classification setting, the conditional probability functions $\mathsf{Q}(k|\cdot)$ are crucial to the definition of our algorithm. As pointed out earlier, Q-ensembles can help achieve accurate solutions in some cases that cannot be realized using convex combinations. Furthermore, for any $k \in [p]$, since $\mathsf{D}_t(k,\cdot,\cdot)$ is a distribution over the $S_k \times \mathcal{Y}$, it is natural to assume that for any $j \neq k$ we have $\mathbb{E}_{(s,y)\sim \mathsf{D}_t(k,\cdot)}[\mathsf{Q}(k|x_{k,s})] \geq \mathbb{E}_{(s,y)\sim \mathsf{D}_t(k,\cdot)}[\mathsf{Q}(j|x_{k,s})]$. This implies the following lower bound: $\mathbb{E}_{(s,y)\sim \mathsf{D}_t(k,\cdot)}[\mathsf{Q}(k|x_{k,s})] \geq \frac{1}{p}$, which in turn implies $\overline{\gamma} \geq \frac{\gamma}{p}$, since for any $x \in \mathcal{X}$, $\sum_{j=1}^{p} \mathsf{Q}(j|x) = 1$. In the special case where all domains coincide, we have $\mathsf{Q}(k|x_{k,s}) = \frac{1}{p}$ for all s and this lower bound is reached. At another extreme, when all domains admit distinct supports, we have $\mathsf{Q}(k|x_{k,s}) = 1$ for all $s \in [m_k]$ and thus $\overline{\gamma} = \gamma$.

MCMultiBoost$(S_1, \ldots, S_p)$
1 $\alpha_0 \leftarrow 0$
2 **for** $t \leftarrow 1$ **to** T **do**
3 $\quad \mathsf{f}_{t-1} \leftarrow \sum_{l=1}^{p}\sum_{j=1}^{N_l} \alpha_{t-1,l,j}\mathsf{Q}(l|\cdot)\,\mathsf{h}_{l,j}$
4 $\quad \tilde{\Phi}_k \leftarrow \frac{1}{m_k}\sum_{i=1}^{m_k}\sum_{y\in\mathcal{Y}} \Phi(-\mathsf{f}_{t-1}(x_{k,i}, y_{k,i}, y))$
5 $\quad \mathcal{K}_t \leftarrow \left\{k\colon k \in \operatorname{argmax}_{k\in[p]} \tilde{\Phi}_k\right\}$
6 $\quad$ **for** $k \in \mathcal{K}_t$ **do**
7 $\quad\quad Z_{t,k} \leftarrow \sum_{i=1}^{m_k}\sum_{y\in\mathcal{Y}} \Phi'\left(-\mathsf{f}_{t-1}(x_{k,i}, y_{k,i}, y)\right)$
8 $\quad\quad$ **for** $i \leftarrow 1$ **to** $m_k, y \in \mathcal{Y}$ **do**
9 $\quad\quad\quad \mathsf{D}_t(k,i,y) \leftarrow \frac{\Phi'(-\mathsf{f}_{t-1}(x_{k,i},y_{k,i},y))}{Z_{t,k}}$
10 $\quad (k,r) \leftarrow \operatorname*{argmax}_{k\in\mathcal{K}_t, r\in[N_k]} \frac{Z_{t,k}}{m_k}[1 - 2\epsilon_{t,k,r}]$
11 $\quad \eta_t \leftarrow \operatorname{argmin}_{\eta>0} F(\alpha_{t-1} + \eta \mathbf{e}_{k,r})$
12 $\quad \alpha_t \leftarrow \alpha_{t-1} + \eta_t \mathbf{e}_{k,r}$
13 $f \leftarrow \sum_{l=1}^{p}\sum_{j=1}^{N_l} \alpha_{T,l,j}\mathsf{Q}(l|\cdot)h_{l,j}$
14 **return** f

Figure 8: Pseudocode of the MCMultiBoost algorithm. $\epsilon_{t,k,r} = \frac{1}{2}\left[1 - \mathbb{E}_{(i,y)\sim \mathsf{D}_t(k,\cdot,\cdot)}[\mathsf{Q}(k|x_{k,i})\mathsf{h}_{k,r}(x_{k,i}, y_{k,i}, y)]\right]$ denotes the weighted error of $\mathsf{Q}(k|\cdot)h_{k,r}$.

In practice, we do not have access to the true conditional probabilities $\mathsf{Q}(k|\cdot)$. Instead, as in the binary classification setting, we can derive accurate estimates $\widehat{\mathsf{Q}}(k|\cdot)$ using large unlabeled samples from the source domains, the *label* used for training being simply the domain index. This can be done using algorithms such as conditional maximum entropy models (Berger et al., 1996) (or multinomial logistic regression), which benefit from strong theoretical guarantees (Mohri et al., 2018, Chapter 13), or other rich models based on neural networks.

Other variants of MCMultiBoost. Other variants of MCMultiBoost, such as the one where, instead of the maximum, the *softmax* function or the sum is used can be defined and analyzed as in the binary setting (Appendix D).

Table 5: Test errors for multi-class problems with RandomForest.

Algorithm	Error-1	Error-2	Error-3	Error-Uniform	Error-Agnostic
			Fashion-MNIST		
AdaBoost.MR-1	**0.131 ± 0.019**	0.185 ± 0.008	0.193 ± 0.017	0.169 ± 0.017	0.193 ± 0.019
AdaBoost.MR-2	0.157 ± 0.020	**0.152 ± 0.009**	0.159 ± 0.023	0.156 ± 0.019	0.159 ± 0.019
AdaBoost.MR-3	0.158 ± 0.015	0.173 ± 0.008	**0.155 ± 0.015**	0.162 ± 0.016	0.173 ± 0.018
AdaBoost.MR-all	0.141 ± 0.020	0.166 ± 0.011	0.163 ± 0.022	0.156 ± 0.021	0.166 ± 0.016
MCMultiBoost	0.132 ± 0.027	0.161 ± 0.018	0.155 ± 0.027	**0.149 ± 0.027**	**0.161 ± 0.024**

Table 6: Test errors for multi-class problems with CNNs.

Algorithm	Error-1	Error-2	Error-3	Error-Uniform	Error-Agnostic
AdaBoost.MR-1	**0.083 ± 0.007**	0.098 ± 0.006	0.104 ± 0.006	0.095 ± 0.006	0.104 ± 0.006
AdaBoost.MR-2	0.104 ± 0.007	**0.090 ± 0.003**	0.099 ± 0.006	0.097 ± 0.003	0.104 ± 0.004
AdaBoost.MR-3	0.098 ± 0.005	0.106 ± 0.003	**0.092 ± 0.003**	0.099 ± 0.005	0.106 ± 0.004
AdaBoost.MR-all	0.093 ± 0.005	0.096 ± 0.007	0.098 ± 0.007	0.095 ± 0.004	0.098 ± 0.007
MCMultiBoost	0.086 ± 0.006	0.090 ± 0.003	0.093 ± 0.005	**0.089 ± 0.003**	**0.091 ± 0.006**

L.4 Experiments

Here, we report the results of several experiments with the MCMultiBoost algorithm on a multiple-source dataset with multiple classes. We present two sets of experiments: one where we used as base predictors random forest classifiers $\mathcal{H}^{RF}$, and another one where we used convolutional neural networks (CNNs).

We use multi-source data based on images of clothes items from the *Fashion-MNIST* (Xiao et al., 2017) dataset. We defined 3 domains: the first domain $(k = 1)$ coincides with the original Fashion-MNIST; the second and third domains $(k = 2, 3)$ are both defined as in Fashion-MNIST but with additive noise, with a different type of noise for $k = 2$ and $k = 3$.

As in Section 6, the probabilities $\mathsf{Q}(k|\cdot)$, $k \in [p]$, are estimated using multinomial logistic regression (or conditional maximum entropy models). We compared MCMultiBoost with a set of multi-class extensions of AdaBoost that operate on the same hypotheses class $\mathcal{H}^{RF}$, which include AdaBoost-k for $k \in [p]$ and AdaBoost-all.

Tables 5 and 6 report our empirical results. The results show that, as in the binary classification setting, the extension of our algorithm to the multi-class setting, MCMultiBoost, outperforms all baselines for both sets of experiments (using Random forests or CNNs).