Multi-task Supervised Learning via Cross-learning

Juan Cerviño[1], Juan Andrés Bazerque[2], Miguel Calvo-Fullana[3] and Alejandro Ribeiro[1]

***Abstract*—In this paper we consider a problem known as multi-task learning, consisting of fitting a set of classifier or regression functions intended for solving different tasks. In our novel formulation, we couple the parameters of these functions, so that they learn in their task specific domains while staying close to each other. This facilitates cross-fertilization in which data collected across different domains help improving the learning performance at each other task. First, we present a simplified case in which the goal is to estimate the means of two Gaussian variables, for the purpose of gaining some insights on the advantage of the proposed cross-learning strategy. Then we provide a stochastic projected gradient algorithm to perform cross-learning over a generic loss function. If the number of parameters is large, then the projection step becomes computationally expensive. To avoid this situation, we derive a primal-dual algorithm that exploits the structure of the dual problem, achieving a formulation whose complexity only depends on the number of tasks. Preliminary numerical experiments for image classification by neural networks trained on a dataset divided in different domains corroborate that the cross-learned function outperforms both the task-specific and the consensus approaches.**

***Index Terms*—Supervised learning, multi-task learning, optimization.**

I. Introduction

Supervised learning is one of the traditional problems studied in statistical learning. At its core, it consists of learning a function mapping inputs to outputs based on provided input-output pairs [1]. In practical situations, it is often the case that in order to learn this function (which we also refer as a *task*), we have limited input-output pairs available (called *training data*). It is also common for different tasks to be related in some sense, and thus, one could attempt to exploit this relationship to improve the individual tasks' performance. The study of this problem, in general is known as *multi-task learning* [2]. While expected in the limited sample regime, it is also applicable in the asymptotic case, as exemplified by Stein's paradox [3], which shows that when estimating more than three parameters from Gaussian random variables, a combined estimator exists that has lower Mean Square Error (MSE) than any separate estimator, even if the random variables are independent.

Multi-task learning has been widely used in practice, specially finding success in computer vision [4]–[6], and language applications [7], [8]. Applications aside, approaches to multi-task learning can be grouped in two categories, based on whether they explicitly model or not the relationship between tasks [9]. Out of those assuming a priori knowledge of the tasks, prevalent approaches are based on a commonly shared underlying task representation [10]. This underlying task representation often takes the form of a sparse, low rank representation across tasks [11], [12], or a manifold representation [13]. Some other approaches do not assume previous knowledge, and learn the relationship between tasks directly from data, generally by performing clustering [14]. In the context of support vector machines, some works have bounded the pair-wise difference between the elements of the classifiers' weights [15], [16]. Some other works focus on finding Pareto optimal solutions between tasks [17].

In this paper, we take a constrained approach to the multi-task learning problem. Our formulation is based on the cross-learning framework [18], [19]. Originally used for learning policies in a reinforcement learning scenario, its principles can be also applied to the multi-task supervised learning problem. The proposed cross-learning approach consists on bounding the distance between the parameters of the learned functions (in our case, classifiers or regression functions). We show, via an illustrative Gaussian sample mean example, that the use of this centrality measure can guarantee to outperform both the separate and consensus approaches. Furthermore, since the cross-learning method is constrained, it requires a projection at each step which can be computationally expensive (e.g., for neural networks with millions of parameters). To overcome this hurdle, we propose to solve the projection step on the dual domain, which reduces the dimension of the problem to the number of tasks. Finally, we present numerical results in a complex classification task, which show that our cross-learning strategy outperforms the consensus and agnostic estimators.

II. Problem Formulation

We address the problem of learning N functions from data coming from different tasks. Let function $f : \mathcal{X} \times \Theta \to \mathcal{Y}$ be the map between input space $\mathcal{X} \subset \mathbb{R}^P$ and output space $\mathcal{Y} \subset \mathbb{R}^Q$ parameterized by $\theta \in \Theta \subset \mathbb{R}^S$. We seek to minimize the loss function ℓ given N datasets containing pairs $(x_j, y_j) \in (\mathcal{X}, \mathcal{Y})$ drawn according to a joint probability $p_i(x, y)$, for $j = 1, \ldots, M_i$, where M_i and $p_i(x, y)$ are the number of samples and generating probability corresponding to each of the $i = 1, \ldots, N$ tasks. Our objective is to obtain functions $f(x, \theta_i)$ that stay close together in the optimization space via the introduction of a measure of centrality ϵ, and central parameters θ_g. Then, we pose the cross-learning problem

$$\{\theta_i^*\}, \theta_g^* = \underset{\{\theta_i\}, \theta_g \in \Theta}{\operatorname{argmin}} \quad \sum_{i=1}^{N} \mathbb{E}_{p_i(x,y)}[\ell\left(y, f(x, \theta_i)\right)] \qquad \text{(PCL)}$$
$$\text{subject to} \quad \|\theta_i - \theta_g\| \leq \epsilon, \quad i = 1, \ldots, N.$$

1: University of Pennsylvania, United States, e-mails: {jcervino, aribeiro}@seas.upenn.edu. 2: Universidad de la República, Uruguay, e-mail: jbazerque@fing.edu.uy. 3: Massachusetts Institute of Technology, e-mail: cfullana@mit.edu. This work is supported by NSF-Simons MoDLTheorinet and Uruguay's ANII FSE 1-2019-1-157459.

The selection of the centrality parameter ϵ entails a trade-off between learning the specific tasks individually or sharing the data corresponding to all tasks. Indeed, if ϵ is large enough, the constraint is always inactive and the functions are learned separately, being agnostic to the information provided by the data corresponding to the other tasks. On the other extreme, if $\epsilon = 0$, all policies are required to be equal and thus the cross-learning setup reduces to consensus. In practice, the best parameter ϵ to use is not known, and to estimate it one can resort to cross-validation techniques [1].

Before presenting an algorithm to solve the cross-learning problem, and to gain insights on the advantages proposed formulation (PCL), we consider the more tractable problem of estimating the means of two Gaussian variables. This example will allow us to show that by selecting a nonzero ϵ judiciously, the cross-learning formulation shown in (PCL) can outperform both the agnostic and consensus estimators.

A. An illustrative case of sample means

Consider the problem of estimating the means μ_x and μ_y of two independent real-valued Gaussian random variables, from their samples $X_j \sim \mathcal{N}(\mu_x, \sigma^2),\ j = 1, \dots, M$ and $Y_j \sim \mathcal{N}(\mu_y, \sigma^2),\ j = 1, \dots, M$. If μ_x and μ_y are estimated separately, being agnostic of the information provided by the other sample, their maximum likelihood estimators are the sample means $\bar{X}_M$ and $\bar{Y}_M$, and the corresponding mean squared errors equal $\sigma_M^2 := \sigma^2/M$.

Now, if we know $|\mu_x - \mu_y| = \epsilon_0$ a priori, then we can use the cross-learning algorithm to trade off bias for variance and improve the estimation accuracy. Indeed, we will prove

Claim 1. *The Mean Squared Error (MSE) of the of the estimates obtained with the cross-learning algorithm for $\epsilon = \epsilon_0$ satisfy $mse_{CL}(\epsilon_0) \leq (3/4)\sigma_M^2$.*

In words, the agnostic estimator can be outperformed by a factor of $3/4$, at least, regardless of the value of ϵ_0 and σ_M. This is true even when $M \to \infty$, where the agnostic estimator becomes consistent. Although this result assumes exact knowledge of the ground truth $\epsilon = \epsilon_0$, we will characterize the MSE $\text{mse}_{\text{CL}}(\epsilon)$ as a continuous function of ϵ, and demonstrate that there is a range of values around ϵ_0 for which the cross-learning estimator outperforms the agnostic one. Furthermore, we also argue that cross-learning outperforms consensus, corresponding to $\epsilon = 0$. Specifically,

Claim 2. *There exists $\epsilon > 0$ such that $mse_{CL}(\epsilon) < mse_{CL}(0)$, with strict inequality for all values of ϵ_0 and σ_M.*

In proving these claims, we will assume without loss of generality that $\mu_x = \mu_y + \epsilon_0$. Specifying the cross-learning estimator in (PCL) for the problem at hand, we have

$$(\hat{\mu}_x^{\text{CL}}, \hat{\mu}_y^{\text{CL}}) = \underset{\bar{\mu}_x, \bar{\mu}_y}{\text{argmin}} \quad \sum_{i=1}^{M} \left[(X_i - \bar{\mu}_x)^2 + (Y_i - \bar{\mu}_y)^2 \right] \tag{2a}$$

$$\text{subject to} \quad |\bar{\mu}_x - \bar{\mu}_y| \leq \epsilon \tag{2b}$$

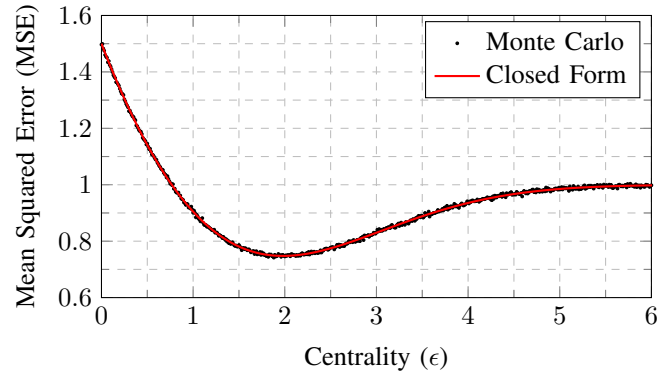


Fig. 1: Mean squared error with respect to the centrality parameter ϵ of the cross-learning problem for one variable. The illustrative case for the estimation of the means of two Gaussian variables is shown. The two variables have same variance $\sigma = 1$ and the distance between their means is given by $\epsilon_0 = 2$.

In this case, the problem admits the following closed-form solution,

$$\hat{\mu}_x^{\text{CL}} = \begin{cases} \bar{X}_M, & \text{if } |\bar{X}_M - \bar{Y}_M| < \epsilon \\ (\bar{X}_M + \bar{Y}_M)/2 + \epsilon/2, & \text{if } \bar{X}_M > \bar{Y}_M + \epsilon \\ (\bar{X}_M + \bar{Y}_M)/2 - \epsilon/2, & \text{if } \bar{X}_M < \bar{Y}_M - \epsilon \end{cases} \tag{3}$$

where the cases correspond to the constraint being activated or not. In order to compute the mean squared error, it is convenient to put $\hat{\mu}_x^{\text{CL}}$ in terms of the variables $\bar{Z}_M := (\bar{X}_M + \bar{Y}_M)/2$ and $\bar{W}_M := (\bar{X}_M - \bar{Y}_M)/2$. Under these definitions, (3) reduces to $\hat{\mu}_x^{\text{CL}} = \bar{Z}_M + T(\bar{W}_M)$, where

$$\begin{aligned} T(\bar{W}_M) := \bar{W}_M \mathbb{1}[|\bar{W}_M| < \epsilon/2] \\ + (\epsilon/2)\mathbb{1}[\bar{W}_M > \epsilon/2] - (\epsilon/2)\mathbb{1}[\bar{W}_M < -\epsilon/2]. \end{aligned} \tag{4}$$

Since $\bar{X}_M$ and $\bar{Y}_M$ are independent, with the same variance, then $\bar{Z}_M$ and $\bar{W}_M$ also are, thus

$$\begin{aligned} \text{mse}_{\text{CL}}(\epsilon) &= \mathbb{E}[(\hat{\mu}_x^{\text{CL}} - \mu_x)^2] && (5) \\ &= \mathbb{E}[(\bar{Z}_M - \mu_z)^2] + \mathbb{E}[(\bar{T}(W_M) - \mu_w)^2] && (6) \\ &= \sigma_M^2/2 + E[(\bar{W}_M - \epsilon_0/2)^2 \mathbb{1}[|\bar{W}_M| < \epsilon/2] \\ &\quad + \mathbb{E}[(1/4)(\epsilon - \epsilon_0)^2 \mathbb{1}[\bar{W}_M > \epsilon/2]] \\ &\quad + \mathbb{E}[(1/4)(\epsilon + \epsilon_0)^2 \mathbb{1}[\bar{W}_M < -\epsilon/2]]. && (7) \end{aligned}$$

With $\bar{Z}_M$ and $\bar{W}_M$ being Gaussian, the expected values can be found in terms of the error function $\text{erf}(x) := (2/\sqrt{\pi}) \int_0^x e^{-t^2} dt$,

$$\begin{aligned} \text{mse}_{\text{CL}}(\epsilon) = \frac{\sigma_M^2}{2} \left(1 + \frac{\alpha e^{-\alpha^2} - \beta e^{-\beta^2}}{\sqrt{\pi}} + \frac{e(\beta) - \text{erf}(\alpha)}{2} \right) \\ + \frac{\sigma_M^2}{2} \left(\alpha^2 (1 + \text{erf}(\alpha)) + \beta^2 (1 - \text{erf}(\beta)) \right), \end{aligned} \tag{8}$$

with $\alpha := -(\epsilon + \epsilon_0)/2\sigma_M$ and $\beta := (\epsilon - \epsilon_0)/2\sigma_M$.

Figure 1 shows the mean squared error in (8) depicted in red as a function of ϵ for a particular choice $\epsilon_0 = 2$ and $\sigma = 1$. Starting from the consensus case at $\epsilon = 0$, the error reduces to a minimum value by ϵ_0, being $3/4$ lower than the limiting error $\lim_{\epsilon \to \infty} \text{mse}_{\text{CL}}(\epsilon) = \sigma_M$. This limit results from the

Algorithm 1 Cross-learning algorithm

Initialize $\theta_g = 0$, $\theta_i^0 = \theta^0$, $i = 1, \dots, N$
repeat for $k = 0, 1, \dots$
  for $i = 1, \dots, N$ **do**
    Obtain $\hat{\nabla}_{\theta_i} \ell(y, f(x, \theta_i^k))$.
  end for
  $(\{\theta_i^{k+1}\}, \theta_g^{k+1}) = \mathcal{P}_{\mathcal{C}} \left[\{\theta_i^k + \eta^k \hat{\nabla}_{\theta_i} \ell(y, f(x, \theta_i^k)), \theta_g^k \right]$
until convergence

constraint being activated with probability zero, which yields $\hat{\mu}_x^{\text{CL}} = \bar{X}_M$, as ϵ grows unbounded. The red curve in is double checked by comparing it to Monte Carlo estimates of the error, over 100,000 realizations of $\hat{\mu}_x^{\text{CL}}$, depicted by black dots in Figure 1.

To see that Claim 1 is true regardless the values of σ_M and ϵ_0, put $\epsilon = \epsilon_0$ in (8), so that $\beta = 0$ and $\alpha = -\epsilon_0/\sigma_M$, thus

$$\text{mse}_{\text{CL}}(\epsilon_0) = \frac{\sigma_M^2}{2} \left(1 + \frac{\alpha e^{-\alpha^2}}{\sqrt{\pi}} - \frac{\text{erf}(\alpha)}{2} + \alpha^2 (1 + \text{erf}(\alpha)) \right).$$

It remains to prove that the term inside brackets in the previous expression does not exceed $3/2$. Indeed, the function

$$f(x) = 1 + \frac{x e^{-x^2}}{\sqrt{\pi}} - \frac{\text{erf}(x)}{2} + x^2 (1 + \text{erf}(x))$$

attains $f(-\infty) = 3/2$, $f(0) = 1$, and $df/dx = 2x(1 + \text{erf}(x)) < 0$ in the interval $(-\infty, 0]$, which yields $f(x) \in (1, 3/2)$, for all $x \in (-\infty, 0)$, holding in particular for $x = -\epsilon_0/\sigma_M = \alpha$, as desired.

To see Claim 2 is also true, we compute the derivative of (8) with respect to ϵ and show that it is negative at $\epsilon = 0$,

$$D(\epsilon) := \frac{d}{d\epsilon} \text{mse}_{\text{CL}}(\epsilon) = \sigma_M \left(\beta(1 - \text{erf}(\beta)) - \alpha(1 + \text{erf}(\alpha)) \right).$$

For $\epsilon = 0$ $\alpha = \beta = -\epsilon_0/2\sigma_M < 0$, hence $D(0) = (\epsilon_0/\sigma_M)\text{erf}(-\epsilon_0/2\sigma_M) < 0$, which is strictly negative as desired.

III. Algorithm Construction

In order to solve the cross-learning problem (PCL) in a general setup, we can resort to a stochastic projected gradient descent scheme. To take gradient steps on the objective function, we can take derivatives of the loss function with respect to the parameters,

$$\bar{\theta}_i^k = \theta_i^k - \eta^k \hat{\nabla}_{\theta_i} \ell(y, f(x, \theta_i^k)), \tag{9}$$

where $\hat{\nabla}_{\theta_i} \ell(y, f(x, \theta_i^k))$ is a stochastic version of the gradient associated to the loss function at a data point, and η^k possibly constant, is a learning step. After each gradient step is taken, the restriction may not be satisfied and thus a projection must be enforced. We define the cross-learning projection $\mathcal{P}_{\mathcal{C}}$ as,

$$\mathcal{P}_{\mathcal{C}} \left[\{\bar{\theta}_i\}, \bar{\theta}_g \right] = \underset{\{\theta_i\}, \theta_g}{\text{argmin}} \sum_{i=1}^{N} \left\| \theta_i - \bar{\theta}_i \right\|^2 + \left\| \theta_g - \bar{\theta}_g \right\|^2 \tag{10a}$$

$$\text{s.t.} \quad \left\| \theta_i - \theta_g \right\|^2 \leq \epsilon^2, i = 1, \dots, N \tag{10b}$$

Algorithm 2 Cross-learning projection

Initialize $\mu_i = \mu_i^0$ and prescribe error $\delta > 0$
repeat for $k = 0, 1, \dots$
  for $i = 1, \dots, N$ **do**
    $\mu_i^{k+1} = [\mu_i^k + \alpha^k \partial_{\mu_i} \mathcal{L}(\theta_i, \theta_g, \mu_i^k)]_+$ with eq. (14)
  end for
until $|\langle \hat{\partial}_\mu \mathcal{L}(\theta_i, \theta_g, \mu_i^k), \mu^k \rangle| \leq \delta$

By combining the gradient step given in equation (9) with the projection (10), we obtain the cross-learning algorithm illustrated in Algorithm 1. As a projected gradient descent form, it can be shown to converge to the optimal value of the cross-learning problem (PCL) in the case of a convex problems [20]. In general, the cross-learning problem (PCL) may not be convex due to both the loss function ℓ and the parameterization function $f(x, \theta)$ used. However, recent results have shown that even in those cases, problems akin to (PCL) have tractable duality gaps [21], motivating primal-dual approaches. Nonetheless, for the specific case of the cross-learning Algorithm 1, previous results have shown its converge in high probability to a neighborhood of a first-order stationary point of problem (PCL) in the context of reinforcement learning [19].

A. Projection in the dual domain

On a neural network, the function parameterization vectors θ_i may have millions of parameters rendering projection (10) challenging in practice. However, we can exploit the fact that only one constraint is added per function, resorting to a dual domain algorithm which has only one variable per function. With Lagrange multipliers $\mu_i \geq 0$, we can write the Lagrangian $\mathcal{L}(\theta_i, \theta_g, \mu_i)$ of problem (PCL) as,

$$\mathcal{L}(\theta_i, \theta_g, \mu_i) = \sum_{i=1}^{N} \left\| \theta_i - \bar{\theta}_i \right\|^2 + \left\| \theta_g - \bar{\theta}_g \right\|^2 + \sum_{i=1}^{N} \mu_i (\left\| \theta_i - \theta_g \right\|^2 - \epsilon^2). \tag{11}$$

Upon defining $\lambda_i = \frac{\mu_i}{1+\mu_i}$, $z = [1, \lambda_1, \dots, \lambda_N]$ and $a = |z|_1$, the primal minimizers of the Lagrangian given μ are

$$\theta_g = \frac{1}{a} (\bar{\theta_g} + \sum_{i=1}^{N} \lambda_i \bar{\theta}_i), \tag{12}$$

$$\theta_i = (1 - \lambda_i) \bar{\theta}_i + \lambda_i \theta_g. \tag{13}$$

Hence, we can obtain the *subgradient* of the Lagrangian with respect to the Lagrangian multipliers μ_i, by substituting (12) and (13),

$$\partial_{\mu_i} \mathcal{L}(\theta_i, \theta_g, \mu_i) = \left\| \theta_i - \theta_g \right\|^2 - \epsilon^2 \tag{14}$$

$$= (1 - \lambda_i)^2 \left\| \bar{\theta}_i - \frac{1}{a} (\bar{\theta_g} + \textstyle\sum_{i=1}^{N} \lambda_i \bar{\theta}_i) \right\|^2 - \epsilon^2.$$

By expanding the norm in (14), the gradient of the Lagrangian depends on the parameters $\bar{\theta}_i$ through their inner products

Fig. 2: Example images from 5 of the 65 categories from the 4 domains composing the Office-Home dataset [23]. The 4 domains are Art, Clipart, Product and Real World. In total, the dataset contains 15,500 images of different sizes and shapes.

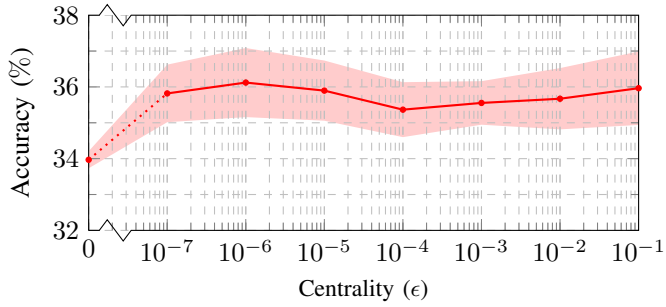


Fig. 3: Accuracy in the test set of a classifier trained with the cross-learning algorithm for different values of the parameter ϵ. Mean values and one standard deviation band are plotted, corresponding to the accuracy over all the 65 categories and 4 different domains, obtained after after 30 epochs. The best overall accuracy is achieved at $\epsilon = 10^{-6}$. The consensus case corresponds to $\epsilon = 0$ and the agnostic case corresponds to $\epsilon = \infty$ (not plotted, achieving a lower accuracy of 24.44%).

$\bar{\theta}_i^T \bar{\theta}_{i'}$, and these products can be computed once when initializing the primal-dual algorithm. Other than that, Algorithm 2 operates on the reduced dimension of the dual variables.

The subgradient (14) is then embedded in the projection described in Algorithm 2. As μ_i must be nonnegative [22], we project them to the nonnegative orthant at every step. The stopping condition of Algorithm 2 is the duality gap of the projection (10), which can be made arbitrarily small. Note that the cross-learning projection (10) is a convex problem, by selecting a non-summable and square summable step-size [20], once the algorithm is halted, the optimal parameters $\{\theta_i\}, \theta_g$ of can be recovered using the primal solutions (12)−(13).

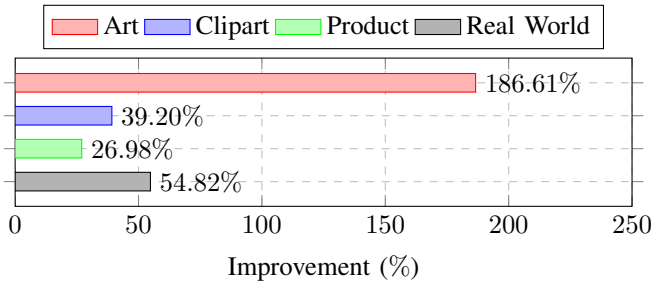
IV. Numerical Results

In this section we test our cross-learning framework on a classification problem with real data. Our goal is to classify images belonging to P different categories, and the problem is divided in N tasks corresponding to images belonging to N different domains. Specifically, we use the Office-Home dataset [23]. It consists of $N = 4$ different domains; Art: an artistic representation of the object, Clipart: a clip art reproduction, Product: an image of a product for sale, and Real World: pictures of the object captured with a camera. The overall dataset contains 15,500 RGB images divided in $P = 65$ categories, with five examples given in Figure 2, including Alarm, Bike, Glasses, Pen, and Speaker. Notice that within each category there are images belonging to each of the domains. The minimum number of images per domain and category is 15 and the image size varies from the smallest image size of 18×18 to the largest being 6500×4900 pixels. We preprocessed the images by normalizing them and fitting their size to 224×224 pixels.

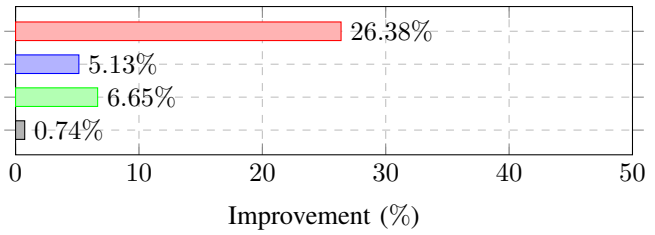
We use our cross-learning strategy for classification with the intuition that seeing an image of one domain could help identifying an image of the same category in other domain. For instance, there is a large amount of speakers pictured as products, which could help identify Clipart versions of the speakers which are available in much smaller numbers. We also want to check how the cross-learning classifier compares to using a single classifier that merges the dataset disregarding the different domains.

We use neural networks as the classifiers $f(x, \theta_i)$, $i = 1, \ldots, N$ for cross-learning, with the architecture being based on AlexNet [24] with a reduction on the size of the last fully connected layers to 256 neurons per layer, corresponding to $\theta \in \mathbb{R}^S$, with $S = 4{,}911{,}745$. In this case, it is crucial to make use of the cross-learning projection (10) in the dual domain, thus reducing the dimensionality from S to $N = 4$ variables. Furthermore, we split the dataset in two parts, using $4/5$ of the images for training and $1/5$ for testing. We train each neural network in a domain according to Algorithm 1. For the gradient step, we use a stepsize $\eta = 0.001$ and we take one image per step. As it is standard for image classification, we use the cross-entropy loss [1]. We train the $N = 4$ neural networks according to the cross-learning algorithm for different values of the centrality measure ϵ. We also consider the case of consensus ($\epsilon = 0$) which is equivalent to merging the images from all domains and training a single neural network. Additionally, we train the neural networks separately one in each domain, which corresponds to $\epsilon = \infty$.

In Figure 3, we present the result of these experiments using the classification accuracy of the trained classifier as figure of merit. We corroborate the intuitive idea drawn by looking at the images (cf. Figure 2). Namely, that domains are correlated and adding samples from different domains improves the overall performance of the agnostic policy, only trained on its own specific samples. This is exemplified by consensus ($\epsilon = 0$) outperforming the agnostic training ($\epsilon = \infty$), with an accuracy of 33.97% against 24.44%. More importantly, the appropriate choice of the ϵ parameter in the cross-learning method outperforms both consensus and agnostic approaches. In particular, Figure 3 shows that the maximum performance is achieved at $\epsilon = 10^{-6}$, indicating that keeping the parameters of the neural networks close, yet not merging them, induces



(a) Percentage of improvement in the accuracy of the classifier between the task-specific agnostic classifier ($\epsilon = \infty$) and the best cross-learning classifier.



(b) Percentage of improvement in the accuracy of the classifier between the task-specific consensus classifier ($\epsilon = 0$) and the best cross-learning classifier.

Fig. 4: Percentage of improvement in the accuracy of the best cross-learning classifier with respect to the agnostic, and consensus classifiers.

an improvement in the overall accuracy. In this sense, these experimental results with real data recover our theoretical findings about the behavior of the cross-learning estimator in the Gaussian example of Section II-A.

The advantage of cross-learning can be better seen in Figure 4, as in all 4 domains there exists a value of ϵ that outperforms both the consensus and agnostic counterparts across all of them. In particular, for the domain that has the minimum number of samples (Art, with 2,427), the accuracy almost triples with cross-learning when compared with the agnostic classifier. This should not come as a surprise, as due to its limited number of samples, it admits a larger margin of improvement by including samples from additional domains. Likewise, considerable improvements are still found when compared with the consensus classifier.

V. Conclusion

In this paper, we introduced a cross-learning framework for multi-task supervised learning. The goal is to learn a set of functions, either for classification or regression, corresponding to different tasks. The proposed strategy entails the fitting of the parameters of each function to their task-specific data, while keeping the parameters of all functions close to each other. We derived intuition from an illustrative case with Gaussian distributions, corroborating the advantage of our formulation compared both to the consensus or the separate approach. For generic set tasks, we have derived a stochastic projected gradient algorithm, together with a dual implementation of the projection for reducing the problem dimension to the number of tasks. Numerical experiments on a dataset with different domains validate that the cross-learning approach is capable of outperforming both the domain specific and the consensus classifiers.

References

[1] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

[2] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[3] C. Stein, "Inadmissibility of the usual estimator for the mean of amultivariate normal distribution," in *Proceeding of the fourth Berkeley symposium on mathematical statistics and probability*, vol. 1. University of California Press, 1956, pp. 197–206.

[4] S. Liu, E. Johns, and A. J. Davison, "End-to-end multi-task learning with attention," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1871–1880.

[5] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert, "Cross-stitch networks for multi-task learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3994–4003.

[6] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7482–7491.

[7] P. Liu, X. Qiu, and X.-J. Huang, "Adversarial multi-task learning for text classification," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 2017, pp. 1–10.

[8] D. Dong, H. Wu, W. He, D. Yu, and H. Wang, "Multi-task learning for multiple language translation," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, 2015, pp. 1723–1732.

[9] Y. Zhang and Q. Yang, "A survey on multi-task learning," *arXiv preprint arXiv:1707.08114*, 2017.

[10] S. Ben-David and R. Schuller, "Exploiting task relatedness for multiple task learning," in *Learning Theory and Kernel Machines*. Springer, 2003, pp. 567–580.

[11] A. Argyriou, T. Evgeniou, and M. Pontil, "Convex multi-task feature learning," *Machine learning*, vol. 73, no. 3, pp. 243–272, 2008.

[12] T. Evgeniou and M. Pontil, "Regularized multi–task learning," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2004, pp. 109–117.

[13] A. Agarwal, S. Gerber, and H. Daume, "Learning multiple tasks using manifold regularization," in *Advances in neural information processing systems*, 2010, pp. 46–54.

[14] L. Jacob, J.-p. Vert, and F. R. Bach, "Clustered multi-task learning: A convex formulation," in *Advances in neural information processing systems*, 2009, pp. 745–752.

[15] T. Kato, H. Kashima, M. Sugiyama, and K. Asai, "Multi-task learning via conic programming," in *Advances in Neural Information Processing Systems*, 2008, pp. 737–744.

[16] ——, "Conic programming for multitask learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 7, pp. 957–968, 2009.

[17] O. Sener and V. Koltun, "Multi-task learning as multi-objective optimization," in *Advances in Neural Information Processing Systems*, vol. 31. Curran Associates, Inc., 2018.

[18] J. Cerviño, J. A. Bazerque, M. Calvo-Fullana, and A. Ribeiro, "Meta-learning through coupled optimization in reproducing kernel hilbert spaces," in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 4840–4846.

[19] ——, "Multi-task reinforcement learning in reproducing kernel hilbert spaces via cross-learning," *arXiv preprint arXiv:2008.11895*, 2020.

[20] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1999.

[21] L. F. Chamon, S. Paternain, M. Calvo-Fullana, and A. Ribeiro, "The empirical duality gap of constrained statistical learning," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 8374–8378.

[22] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2009.

[23] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *(IEEE) Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.