DECIPHERING AND OPTIMIZING MULTI-TASK LEARNING: A RANDOM MATRIX APPROACH

Malik Tiomoko
Laboratoire des Signaux et Systèmes
Université Paris-Sud
Orsay, France
malik.tiomoko@u-psud.fr

Hafiz Tiomoko Ali
Huawei Technologies Research and Development (UK)
London, UK
hafiz.tiomoko.ali@huawei.com

Romain Couillet
Gipsa Lab
Université Grenoble-Alpes
Saint Martin d'Hères, France
romain.couillet@gipsa-lab.grenoble-inp.fr

ABSTRACT

This article provides theoretical insights into the inner workings of multi-task and transfer learning methods, by studying the tractable least-square support vector machine multi-task learning (LS-SVM MTL) method, in the limit of large (p) and numerous (n) data. By a random matrix analysis applied to a Gaussian mixture data model, the performance of MTL LS-SVM is shown to converge, as $n, p \to \infty$, to a deterministic limit involving simple (small-dimensional) statistics of the data.

We prove (i) that the standard MTL LS-SVM algorithm is in general strongly biased and may dramatically fail (to the point that individual single-task LS-SVMs may outperform the MTL approach, even for quite resembling tasks): our analysis provides a simple method to correct these biases, and that we reveal (ii) the sufficient statistics at play in the method, which can be efficiently estimated, even for quite small datasets. The latter result is exploited to automatically optimize the hyperparameters without resorting to any cross-validation procedure.

Experiments on popular datasets demonstrate that our improved MTL LS-SVM method is computationally-efficient and outperforms sometimes much more elaborate state-of-the-art multi-task and transfer learning techniques.

1 INTRODUCTION

The advent of elaborate learning machines capable to surpass human performances on dedicated tasks has reopened past challenges in machine learning. Transfer learning, and multitask learning (MTL) in general, by which known tasks are used to help a machine learn other related tasks, is one of them. The particularly interesting aspects of multi-task learning lie in the possibility (i) to exploit the resemblance between the datasets associated to each task so the tasks "help each other" and (ii) to train a machine on a specific target dataset comprised of few labelled data by exploiting much larger labelled datasets, however composed of different data. Practical applications are numerous, ranging from the prediction of student test results for a collection of schools (Aitkin & Longford, 1986), to survival of patients in different clinics, to the value of many possibly related financial indicators (Allenby & Rossi, 1998), to the preference modelling of individuals in a marketing context, etc.

Since MTL seeks to improve the performance of a task with the help of related tasks, a central issue to (i) understand the functioning of MTL, (ii) adequately adapt its hyperparameters and eventually (iii) improve its performances consists in characterizing how MTL relates tasks to one another and in identifying which features are "transferred". The article aims to decipher these fundamental aspects for sufficiently general data models.

Several data models may be accounted for to enforce relatedness between tasks. A common assumption is that the data lie close to each other in a geometrical sense (Evgeniou & Pontil, 2004), live in a low dimensional manifold (Agarwal et al., 2010), or share a common prior (Daumé III, 2009). We follow here the latter assumption in assuming that, for each task, the data arise from a 2-class Gaussian mixture.[1]

Methodologically, in its simplest approach, MTL algorithms can be obtained from a mere extension of support vector machines (SVM), accounting for more than one task. That is, instead of finding the hyperplane (through its normal vector ω) best separating the two classes of a unique dataset, (Evgeniou & Pontil, 2004) proposes to produce best separating hyperplanes (or normal vectors) $\omega_1, \dots, \omega_k$ for each pair of data classes of k tasks, with the additional constraint that the normal vectors take the form $\omega_i = \omega_0 + v_i$ for some common vector w_0 and dedicated vectors v_i. The amplitude of the vectors v_i is controlled (through an additional hyperparameter) to enforce or relax task relatedness. We study this approach here. Yet, to obtain explicit and thus more insightful results, we specifically resort to a least-square SVM (as proposed e.g., in (Xu et al., 2013)) rather than a margin-based SVM. This only marginally alters the overall behavior of the MTL algorithm and has no impact on the main insights drawn in the article. Moreover, by a now well-established universality argument in large dimensional statistics, (Mai et al., 2019) show that quadratic (least-square) cost functions are asymptotically optimal and uniformly outperform alternative costs (such as margin-based methods or logistic approaches), even in a classification setting. This argument further motivates the choice of considering first and foremost the LS-SVM version of MTL-SVM.

Technically, the article exploits the powerful random matrix theory to study the performance of the MTL least-square SVM algorithm (MTL LS-SVM) for data arising from a Gaussian mixture model, assuming the total number n and dimension p of the data are both large, i.e., as $n, p \to \infty$ with $p/n \to c \in (0, \infty)$. As such, our work follows after the recent wave of interest into the asymptotics of machine learning algorithms, such as studied lately in e.g., (Liao & Couillet, 2019; Deng et al., 2019; Mai & Couillet, 2018; El Karoui et al., 2010). Our analysis reveals the following major conclusions:

- we exhibit the sufficient statistics, which concretely enable task comparison in the MTL LS-SVM algorithm; we show that, even when data are of large dimensions ($p \gg 1$), these statistics remain small dimensional (they only scale with the number k of tasks);
- while it is conventional to manually set labels associated to each dataset within $\{-1, 1\}$, we prove that this choice is largely suboptimal and may even cause MTL to severely fail (causing "negative transfer"); we instead provide the optimal values for the labels of each dataset, which depend on the sought-for objective: these optimal values are furthermore easily estimated from very few training data (i.e., no cross-validation is needed);
- for unknown new data $\mathbf{x}$, the MTL LS-SVM algorithm allocates a class based on the comparison of a score $g(\mathbf{x})$ to a threshold ζ, usually set to zero. We prove that, depending on the statistics and number of elements of the training dataset, a bias is naturally induced that makes $\zeta = 0$ a largely suboptimal choice in general. We provide a correction for this bias, which again can be estimated from the training data alone;
- we demonstrate on popular real datasets that our proposed optimized MTL LS-SVM is both resilient to real data and also manages, despite its not being a best-in-class MTL algorithm, to rival and sometimes largely outperform competing state-of-the-art algorithms.

These conclusions thus allow for an optimal use of MTL LS-SVM with performance-maximizing hyperparameters and strong theoretical guarantees. As such, the present article offers through MTL LS-SVM a viable fully-controlled (even better performing) alternative to state-of-the-art MTL.

Reproducibility. Matlab and Julia codes for reproducing the results of the article are available in the supplementary materials.

Notation. $e_m^{[n]} \in \mathbb{R}^n$ is the canonical vector of $\mathbb{R}^n$ with $[e_m^{[n]}]_i = \delta_{mi}$. Moreover, $e_{ij}^{[2k]} = e_{2(i-1)+j}^{[2k]}$. Similarly, $E_{ij}^{[n]} \in \mathbb{R}^{n \times n}$ is the matrix with $[E_{ij}^{[n]}]_{ab} = \delta_{ia}\delta_{jb}$. The notations $A \otimes B$ and $A \odot B$

[1] In the supplementary material, we extend this setting to a much broader and more realistic scope, and justify in passing the relevance of a Gaussian mixture modelling to address multi-task learning with real data.

for matrices or vectors A, B are respectively the Kronecker and Hadamard products. $\mathcal{D}_x$ is the diagonal matrix containing on its diagonal the elements of the vector x and $A_{i\cdot}$ is the i-th row of A. The notation $\mathring{A}$ is used when a centering operation is performed on the matrix or vector A. Uppercase calligraphic letters ($\mathcal{A}, \mathcal{K}, \Gamma, \mathcal{M}$, $\mathcal{V}$,...) are used for deterministic small dimensional matrices. Finally, $\mathbb{1}_m$ and I_m are respectively the vector of all one's of dimension m and the identity matrix of dimension $m \times m$. The index pair i, j generally refers to Class j in Task i.

2 Related works

Let us first point out the difference between MTL and transfer learning: while MTL makes no distinction between tasks and aims to improve the performance of all tasks, transfer learning aims to maximize the performance of a *target* task with the help of all source tasks. Yet, both methods mostly sharing the same learning process, in this section, we mainly focus on the MTL literature, which is divided into parameter-based versus feature-based MTL.

In the parameter-based MTL approach, the tasks are assumed to share some parameters (e.g., the hyperplanes best separating each class) or their hyperparameters have a common prior distribution. Existing learning methods (SVM, logistic regression, etc.) can then be appropriately modified to incorporate these relatedness assumptions. In this context, (Evgeniou & Pontil, 2004; Xu et al., 2013; Parameswaran & Weinberger, 2010) respectively adapt the SVM, LS-SVM, and Large Margin Nearest Neighbor (LMNN) algorithms to the MTL paradigm. The present article borrows ideas from Evgeniou & Pontil (2004); Xu et al. (2013).

In the feature-based MTL approach, the tasks data are instead assumed to share a low-dimensional common representation. In this context, most of the works aim to determine a mapping of the ambient data space into a low-dimensional subspace (through sparse coding, deep neural networks, principal component analysis, etc.) in which the tasks have high similarity (Argyriou et al., 2007; Maurer et al., 2013; Zhang et al., 2016; Pan et al., 2010); other works simply use a feature selection method by merely extracting a subset of the original feature space (Obozinski et al., 2006; Wang & Ye, 2015; Gong et al., 2012). We must insist that, in the present work, our ultimate objective is to study and improve "data-generic" MTL mechanisms under no structural assumption on the data; this approach is quite unlike recent works exploiting convolutive techniques in deep neural nets or low dimensional feature-based methods to perform transfer or multi-task learning mostly for computer vision.

From a theoretical standpoint though, few works have provided a proper understanding of the various MTL algorithms. To our knowledge, the only such results arise from elementary learning theory (Rademacher complexity, VC dimension, covering number, stability) and only provide loose performance bounds (Baxter, 2000; Ben-David & Schuller, 2003; Baxter, 1997). As such, the present work fills a long-standing gap in the MTL research.

3 The multi-task learning setting

Let $X \in \mathbb{R}^{p\times n}$ be a collection of n independent data vectors of dimension p. The data are divided into k subsets attached to individual "tasks". Specifically, letting $X = [X_1, \ldots, X_k]$, Task i is a binary classification problem from the training samples $X_i = [X_i^{(1)}, X_i^{(2)}] \in \mathbb{R}^{p\times n_i}$ with $X_i^{(j)} = [x_{i1}^{(j)}, \ldots, x_{in_{ij}}^{(j)}] \in \mathbb{R}^{p\times n_{ij}}$ the n_{ij} vectors of class $j \in \{1, 2\}$ for Task i. In particular, $n = \sum_{i=1}^k n_i$ and $n_i = n_{i1} + n_{i2}$ for each $i \in \{1, \ldots, k\}$.

To each $x_{il} \in \mathbb{R}^p$ of the training set is attached a corresponding "label" (or score) $y_{il} \in \mathbb{R}$. We denote $y_i = [y_{i1}, \ldots, y_{in_i}]^\mathsf{T} \in \mathbb{R}^{n_i}$ the vector of all labels for Task i, and $y = [y_1^\mathsf{T}, \ldots, y_k^\mathsf{T}]^\mathsf{T} \in \mathbb{R}^n$ the vector of all labels. These labels are generally chosen to be ±1 but, for reasons that will become clear in the course of the article, we voluntarily do not enforce binary labels here.

Before detailing the multitask classification scheme, a preliminary task-wise centering operation is performed on the data, i.e., we consider in the following the datasets

$$\mathring{X}_i = X_i \left(I_{n_i} - \frac{1}{n_i} \mathbb{1}_{n_i} \mathbb{1}_{n_i}^\mathsf{T} \right), \quad \forall i \in \{1, \ldots, k\}.$$

As such, we systematically work on the labeled datasets $(\mathring{X}_1, y_1), \dots, (\mathring{X}_k, y_k)$. Remark 1 in the supplementary material motivates this choice, which avoids extra biases produced by the algorithm.

3.1 The optimization framework

The multitask learning least square support vector machine (MTL LS-SVM) aims to predict, for input vectors $\mathbf{x} \in \mathbb{R}^p$ not belonging to the training set, their associated score y upon which a decision on the class allocation of $\mathbf{x}$ is taken, *for a given target task*. To this end, based on the labeled sets $(\mathring{X}_1, y_1), \dots, (\mathring{X}_k, y_k)$, MTL LS-SVM determines the normal vectors $W = [\omega_1, \omega_2, \dots, \omega_k] \in \mathbb{R}^{p\times k}$ and intercepts $b = [b_1, b_2, \dots, b_k]^\mathsf{T} \in \mathbb{R}^k$ defining k separating hyperplanes for the corresponding k binary classification tasks. In order to account for task relatedness, each ω_i assumes the form $\omega_i = \omega_0 + v_i$ for some common $\omega_0 \in \mathbb{R}^p$ and task-dedicated $v_i \in \mathbb{R}^p$.

Formally, writing $V = [v_1, \dots, v_k] \in \mathbb{R}^{p\times k}$ (so that $W = \omega_0 \mathbb{1}_k^\mathsf{T} + V$) and following the work of (Evgeniou & Pontil, 2004; Xu et al., 2013), the optimization function is given by

$$\min_{\substack{(\omega_0, V, b) \in \\ \mathbb{R}^p \times \mathbb{R}^{p\times k} \times \mathbb{R}^k}} \frac{1}{2\lambda}\|\omega_0\|^2 + \frac{1}{2}\sum_{i=1}^k \frac{\|v_i\|^2}{\gamma_i} + \frac{1}{2}\sum_{i=1}^k \|\xi_i\|^2, \quad \xi_i = y_i - (\mathring{X}_i^\mathsf{T}\omega_i + b_i \mathbb{1}_{n_i}), \quad 1 \le i \le k.$$

In this expression, the parameter λ enforces more task relatedness while the parameters $\gamma_1, \dots, \gamma_k$ enforce better classification of the data in their respective classes.

Being a quadratic optimization problem under linear equality constraints, ω_0, V, b are obtained explicitly (see details in Section 1 of the supplementary material). The solution is best described through the expression of the hyperplanes $\omega_1, \dots, \omega_k \in \mathbb{R}^p$ which take the form:

$$\omega_i = \left(e_i^{[k]\mathsf{T}} \otimes I_p\right) AZ\alpha,$$

and $b = (P^\mathsf{T}QP)^{-1}P^\mathsf{T}Qy$, where $\alpha = Q(y - Pb) = Q^{\frac{1}{2}}(I_n - Q^{\frac{1}{2}}P(P^\mathsf{T}QP)^{-1}P^\mathsf{T}Q^{\frac{1}{2}})Q^{\frac{1}{2}}y \in \mathbb{R}^n$ is the Lagrangian dual and

$$Q = \left(\frac{1}{kp}Z^\mathsf{T}AZ + I_n\right)^{-1} \in \mathbb{R}^{n\times n}, \quad Z = \sum_{i=1}^k E_{ii}^{[k]} \otimes \mathring{X}_i \in \mathbb{R}^{pk\times n}$$

$$A = \left(\mathcal{D}_\gamma + \lambda \mathbb{1}\mathbb{1}^\mathsf{T}\right) \otimes I_p \in \mathbb{R}^{kp\times kp}, \quad P = \sum_{i=1}^k E_{ii}^{[k]} \otimes \mathbb{1}_{n_i} \in \mathbb{R}^{n\times k}$$

with $\gamma = [\gamma_1, \dots, \gamma_k]^\mathsf{T}$ and $\mathcal{D}_\gamma = \operatorname{diag}(\gamma)$.

MTL LS-SVM differs from a single-task joint LS-SVM for all data in that the data $\mathring{X}_1, \dots, \mathring{X}_k$ are not treated simultaneously but through k distinct filters: this explains why $Z \in \mathbb{R}^{kp\times n}$ is not the mere concatenation $[\mathring{X}_1, \dots, \mathring{X}_k]$ but a block-diagonal structure isolating each $\mathring{X}_i$. As such, the $\mathring{X}_i$'s-relating matrix A plays an important role in the MTL learning process.

With this formulation for the solution (W, b), the prediction of the class of any new data point $\mathbf{x} \in \mathbb{R}^p$ for the target Task i is then obtained from the classification score

$$g_i(\mathbf{x}) = \frac{1}{kp}\left(e_i^{[k]} \otimes \mathring{\mathbf{x}}\right)^\mathsf{T} AZ\alpha + b_i \tag{1}$$

where $\mathring{\mathbf{x}} = \mathbf{x} - \frac{1}{n_i}X_i\mathbb{1}_{n_i}$ is a centered version of $\mathbf{x}$ with respect to the training dataset for Task i.

3.2 Large dimensional statistical modelling

The first objective of the article is to quantify the MTL performance, and thus of the (a priori intricate) statistics of $g_i(\mathbf{x})$, under a sufficiently simple but telling Gaussian mixture model for training and test data.

Assumption 1 (Distribution of X and $\mathbf{x}$)**.** The columns of $[X, \mathbf{x}]$ are independent Gaussian random variables. Specifically, the n_{ij} samples $x_{i1}^{(j)}, \dots, x_{in_{ij}}^{(j)}$ of class j for Task i are independent $\mathcal{N}(\mu_{ij}, I_p)$ vectors, and we let $\Delta\mu_i \equiv \mu_{i1} - \mu_{i2}$. As for $\mathbf{x}$, it follows an independent $\mathcal{N}(\mu_\mathbf{x}, I_p)$ vector.

In the supplementary material, Assumption 1 is relaxed to $[X, \mathbf{x}]$ arising from a generative model of the type $x_{il}^{(j)} = h_{ij}(z_{il}^{(j)})$ for $z_{il}^{(j)} \sim \mathcal{N}(0, I_p)$ and $h_{ij} : \mathbb{R}^p \to \mathbb{R}^p$ a 1-Lipschitz function. This model encompasses extremely realistic data models, including data arising from generative networks (e.g., GANs (Goodfellow et al., 2014)) and is shown in the supplementary material to be universal in the sense that, as $n, p \to \infty$, the asymptotic performances of MTL LS-SVM only depend on the statistical means and covariances of the $x_{il}^{(j)}$: the performances under complex mixtures are thus proved to *coincide* with those under an elementary Gaussian mixture. This generalized study however comes at the expense of more complex definitions and formulas, which impedes readability; hence the simpler isotropic Gaussian mixture model here.

Our central technical approach for the performance evaluation of the MTL LS-SVM algorithm consists in placing ourselves under the large p, n regime of random matrix theory.

Assumption 2 (Growth Rate)**.** As $n \to \infty$, $n/p \to c_0 > 0$ and, for $1 \leq i \leq k$, $1 \leq j \leq 2$, $\frac{n_{ij}}{n} \to c_{ij} > 0$. We let $c_i = c_{i1} + c_{i2}$, $c = [c_1, \ldots, c_k]^\mathsf{T} \in \mathbb{R}^k$.

With these notations and assumptions, we are in position to present our main theoretical results.

4 The multi-task Learning analysis

4.1 Technical strategy and notations

To evaluate the statistics of $g_i(\mathbf{x})$ (equation 1), we resort to finding so-called *deterministic equivalents* for the matrices Q, AZQ, etc., which appear at the core of the formulation of $g_i(\mathbf{x})$. Those are provided in Lemma 1 of the supplementary material. Our strategy then consists in "decoupling" the effect of the data statistics from those of the MTL hyperparameters $\lambda, \gamma_1, \ldots, \gamma_k$. Specifically, we extract two fundamental quantities for our analysis: a data-related matrix $\mathcal{M} \in \mathbb{R}^{2k \times 2k}$ and a hyperparameter matrix $\mathcal{A} \in \mathbb{R}^{k \times k}$:

$$\mathcal{M} = \sum_{i,j=1}^{k} \Delta\mu_i^\mathsf{T} \Delta\mu_j \left(E_{ij}^{[k]} \otimes \mathbb{c}_i \mathbb{c}_j^\mathsf{T} \right), \; \mathbb{c}_i = \begin{bmatrix} \frac{c_{i2}}{c_i} \sqrt{\frac{c_{i1}}{c_i}} \\ -\frac{c_{i1}}{c_i} \sqrt{\frac{c_{i2}}{c_i}} \end{bmatrix}, \; \mathcal{A} = \left(I_k + \mathcal{D}_{\tilde{\Delta}}^{-\frac{1}{2}} \left(\mathcal{D}_\gamma + \lambda \mathbb{1}_k \mathbb{1}_k^\mathsf{T} \right)^{-1} \mathcal{D}_{\tilde{\Delta}}^{-\frac{1}{2}} \right)^{-1}$$

where $\tilde{\Delta} = [\tilde{\Delta}_1, \ldots, \tilde{\Delta}_k]^\mathsf{T}$ are the unique positive solutions to the implicit system $\tilde{\Delta}_i = \frac{c_i}{c_0} - \mathcal{A}_{ii}$ (this is implicit because $\mathcal{A}$ is a function of the $\tilde{\Delta}_j$'s). In passing, it will appear convenient to use the shortcut notation $\bar{\Delta} = [\bar{\Delta}_{11}, \ldots, \bar{\Delta}_{k2}]^\mathsf{T} \in \mathbb{R}^{2k}$ where $\bar{\Delta}_{ij} = \frac{c_{ij}}{c_i} c_0 \tilde{\Delta}_i$.

We will see that $\mathcal{M}$ plays the role, in the limit of large p, n, of a sufficient statistic for the performance of the MTL LS-SVM algorithm only involving (i) the data statistics $\mu_{11}, \ldots, \mu_{k2}$ and (ii) the (limiting) relative number $c_{11}/c_1, \ldots, c_{k2}/c_k$ of elements per class in each task. As for $\mathcal{A}$, it captures the information about the impact of the hyperparameters $\lambda, \gamma_1, \ldots, \gamma_k$ and of the dimension ratios $c_1, \ldots, c_k$ and c_0. These two matrices will be combined in the core matrix $\Gamma \in \mathbb{R}^{2k \times 2k}$ of the upcoming MTL LS-SVM performance analysis, defined as $\Gamma = \left(I_{2k} + \left(\mathcal{A} \otimes \mathbb{1}_2 \mathbb{1}_2^\mathsf{T} \right) \odot \mathcal{M} \right)^{-1}$ where we recall that '$\odot$' is the Hadamard (element-wise) matrix product.

We raised in the introduction of Section 3 that we purposely relax the binary "labels" y_{ij} associated to each datum x_{ij} in each task to become "scores" $y_{ij} \in \mathbb{R}$. This will have fundamental consequences to the MTL performance. Yet, since $x_{i1}, \ldots, x_{in_i}$ are i.i.d. data vectors, we impose equal scores $y_{i1} = \ldots = y_{in_i}$ within each class. As such, we may reduce the complete score vector $y \in \mathbb{R}^n$ under the form $y = \left[\tilde{y}_{11} \mathbb{1}_{n_{11}}^\mathsf{T}, \ldots, \tilde{y}_{k2} \mathbb{1}_{n_{k2}}^\mathsf{T} \right]^\mathsf{T}$ for $\tilde{y} = [\tilde{y}_{11}, \ldots, \tilde{y}_{k2}]^\mathsf{T} \in \mathbb{R}^{2k}$. From Remark 1 of the supplementary material, the performances of MTL are insensitive to a constant shift in the scores y_{i1} and y_{i2} of every given Task i: as such, the recentered version $\mathring{\tilde{y}} = [\mathring{\tilde{y}}_{11}, \ldots, \mathring{\tilde{y}}_{k2}]^\mathsf{T}$ of $\tilde{y}$, where $\mathring{\tilde{y}}_{ij} = \tilde{y}_{ij} - (\frac{n_{i1}}{n_i} \tilde{y}_{i1} + \frac{n_{i2}}{n_i} \tilde{y}_{i2})$ will be central in the upcoming results.

4.2 Main results

Theorem 1 (Asymptotics of $g_i(\mathbf{x})$)**.** *Under Assumptions 1–2, for $\mathbf{x} \sim \mathcal{N}(\mu_\mathbf{x}, I_p)$ with $\mu_\mathbf{x} = \mu_{ij}$,*

$$g_i(\mathbf{x}) - G_{ij} \to 0, \quad G_{ij} \sim \mathcal{N}(m_{ij}, \sigma_i^2)$$

in distribution where, letting $m = [m_{11}, \ldots, m_{k2}]^{\mathsf{T}}$,

$$m = \tilde{y} - \mathcal{D}_{\tilde{\Delta}}^{-\frac{1}{2}} \Gamma \mathcal{D}_{\tilde{\Delta}}^{\frac{1}{2}} \mathring{\tilde{y}}, \quad \sigma_i^2 = \frac{1}{\tilde{\Delta}_i} \mathring{\tilde{y}}^{\mathsf{T}} \mathcal{D}_{\tilde{\Delta}}^{\frac{1}{2}} \Gamma \left(\mathcal{D}_{\frac{\mathcal{K}_{i.}}{c} \otimes \mathbb{1}_2} + \mathcal{V}_i \right) \Gamma \mathcal{D}_{\tilde{\Delta}}^{\frac{1}{2}} \mathring{\tilde{y}}$$

with $\mathcal{V}_i = \frac{1}{c_0} (\mathcal{A} \mathcal{D}_{\frac{c_0 \mathcal{K}_{i.}}{c} + e_i^{[k]}} \mathcal{A} \otimes \mathbb{1}_2 \mathbb{1}_2^{\mathsf{T}}) \odot \mathcal{M}$ *and* $\mathcal{K} = [\mathcal{A} \odot \mathcal{A}](I_k - \mathcal{D}_{\frac{c_0}{kc}} [\mathcal{A} \odot \mathcal{A}])^{-1}$.

As anticipated, the statistics of the classification scores $g_i(\mathbf{x})$ mainly depend on the data statistics $\mu_{i'j'}$ and on the hyperparameters λ and $\gamma_1, \ldots, \gamma_k$ through the matrix Γ (and more marginally through $\mathcal{V}_i$ and $\mathcal{K}$ for the variances).

Since $g_i(\mathbf{x})$ has a Gaussian limit centered about m_{ij}, the (asymptotic) standard decision for $\mathbf{x}$ to be allocated to Class 1 ($\mathbf{x} \to \mathcal{C}_1$) or Class 2 ($\mathbf{x} \to \mathcal{C}_2$) for Task i is obtained by the “averaged-mean” test

$$g_i(\mathbf{x}) \underset{\mathcal{C}_2}{\overset{\mathcal{C}_1}{\gtrless}} \frac{1}{2} (m_{i1} + m_{i2}) \tag{2}$$

the classification error rate $\epsilon_{i1} \equiv P(\mathbf{x} \to \mathcal{C}_2 | \mathbf{x} \in \mathcal{C}_1)$ of which is then

$$\epsilon_{i1} \equiv P\left(g_i(\mathbf{x}) \geq \frac{m_{i1} + m_{i2}}{2} \middle| \mathbf{x} \in \mathcal{C}_1 \right) = \mathcal{Q}\left(\frac{m_{i1} - m_{i2}}{2\sigma_i} \right) + o(1) \tag{3}$$

with m_{ij}, σ_i as in Theorem 1 and $\mathcal{Q}(t) = \int_t^\infty e^{-\frac{u^2}{2}} du$.

Further comment on Γ is due before moving to practical consequences of Theorem 1. From the expression $(\mathcal{A} \otimes \mathbb{1}_2 \mathbb{1}_2^{\mathsf{T}})^{-1} \odot \mathcal{M}$, we observe that: (i) if $\lambda \ll 1$, then $\mathcal{A}$ is diagonal dominant and thus “filters out” in the Hadamard product all off-diagonal entries of $\mathcal{M}$, i.e., all cross-terms $\Delta\mu_i^{\mathsf{T}} \Delta\mu_j$ for $i \neq j$, therefore refusing to exploit the correlation between tasks; (ii) if instead λ is not small, $\mathcal{A}$ may be developed (using the Sherman-Morrison matrix inverse formulas) as the sum of a diagonal matrix, which again filters out the $\Delta\mu_i^{\mathsf{T}} \Delta\mu_j$ for $i \neq j$, and of a rank-one matrix which instead performs a weighted sum (through the γ_i’s and the $\tilde{\Delta}_i$’s) of the entries of $\mathcal{M}$. Specifically, letting $\gamma^{-1} = (\gamma_1^{-1}, \ldots, \gamma_k^{-1})^{\mathsf{T}}$ in the expression of $\mathcal{A}$, we have $\left(D_\gamma + \lambda \mathbb{1}_k \mathbb{1}_k^{\mathsf{T}} \right)^{-1} = D_\gamma^{-1} - \frac{\lambda \gamma^{-1} \gamma^{-1\mathsf{T}}}{1 + \lambda \frac{1}{k} \sum_{i=1}^k \gamma_i^{-1}}$. As such, disregarding the regularization effect of the $\tilde{\Delta}_i$’s, the off-diagonal $\Delta\mu_i^{\mathsf{T}} \Delta\mu_j$ entry of $\mathcal{M}$ is weighted with coefficient $(\gamma_i \gamma_j)^{-1}$: the impact of the γ_i’s is thus strongly associated to the relevance of the correlation between tasks.

A fundamental aspect of Theorem 1 is that it concludes that the performances of the *large dimensional* ($n, p \gg 1$) classification problem at hand merely boils down to $2k$-dimensional statistics, as all objects defined in the theorem statement are at most of size $2k$. More importantly from a practical perspective, these “sufficient statistics” are easily amenable to fast and efficient estimation: it only requires a few training samples to estimate all quantities involved in the theorem. This, as a corollary, lets one envision the possibility of efficient transfer learning methods based on very scarce data samples as discussed in Remark 1.

Estimating m_{ij} and σ_i not only allows one to anticipate theoretical performances but also enables the actual estimation of the decision threshold $\frac{1}{2}(m_{i1} + m_{i2})$ in equation 2 and opens the possibility to largely optimize MTL LS-SVM through an (asymptotically) optimal choice of the training scores $\tilde{y}$. Indeed, the asymptotics in Theorem 1 depend in an elegant manner on the training data labels (scores) $\tilde{y}$. Since the variance σ_i^2 is independent of the classes, we easily determine the vector $\tilde{y} = \tilde{y}^\star$ minimizing the misclassification probability for Task i as

$$\tilde{y}^\star = \underset{\tilde{y} \in \mathbb{R}^{2k}}{\arg\max} \frac{(m_{i1} - m_{i2})^2}{\sigma_i^2} = \underset{\tilde{y} \in \mathbb{R}^{2k}}{\arg\max} \frac{\| \tilde{y}^{\mathsf{T}} (I_{2k} - \mathcal{D}_{\tilde{\Delta}}^{\frac{1}{2}} \Gamma \mathcal{D}_{\tilde{\Delta}}^{-\frac{1}{2}}) (e_{i1}^{[2k]} - e_{i2}^{[2k]}) \|^2}{\tilde{y}^{\mathsf{T}} \mathcal{D}_{\tilde{\Delta}}^{\frac{1}{2}} \Gamma (\mathcal{D}_{\frac{\mathcal{K}_{i.}}{c} \otimes \mathbb{1}_2} + \mathcal{V}_i) \Gamma \mathcal{D}_{\tilde{\Delta}}^{\frac{1}{2}} \tilde{y}}$$

the solution of which is explicit

$$\tilde{y}^\star = \mathcal{D}_{\tilde{\Delta}}^{-\frac{1}{2}} \Gamma^{-1} \mathcal{H} [(\mathcal{A} \otimes \mathbb{1}_2 \mathbb{1}_2^{\mathsf{T}}) \odot \mathcal{M}] \mathcal{D}_{\tilde{\Delta}}^{-\frac{1}{2}} (e_{i1}^{[2k]} - e_{i2}^{[2k]}), \quad \mathcal{H} \equiv (\mathcal{D}_{\frac{\mathcal{K}_{i.}}{c} \otimes \mathbb{1}_2} + \mathcal{V}_i)^{-1} \tag{4}$$

with corresponding (asymptotically) optimal classification error ϵ_{i1} (equation 3) given by

$$\epsilon_{i1}^\star = \mathcal{Q}\left(\frac{1}{2} \sqrt{(e_{i1}^{[2k]} - e_{i2}^{[2k]})^{\mathsf{T}} \mathcal{G} (e_{i1}^{[2k]} - e_{i2}^{[2k]})} \right), \ \mathcal{G} = \mathcal{D}_{\tilde{\Delta}}^{\frac{1}{2}} [(\mathcal{A} \otimes \mathbb{1}_2 \mathbb{1}_2^{\mathsf{T}}) \odot M] \mathcal{V}_i^{-1} [(\mathcal{A} \otimes \mathbb{1}_2 \mathbb{1}_2^{\mathsf{T}}) \odot M] \mathcal{D}_{\tilde{\Delta}}^{\frac{1}{2}}.$$

The only non-diagonal matrices in equation 4 are Γ and $\mathcal{V}_i$ in which $\mathcal{M}$ plays the role of a "variance profile" matrix. In particular, assuming $\Delta\mu_i^\mathsf{T}\Delta\mu_\ell = 0$ for all $\ell \neq i$ (i.e., the statistical means of all tasks are orthogonal to those of Task i), then the two rows and columns of $\mathcal{M}$ associated to Task i are all zero but on the 2×2 diagonal block. Therefore, $\tilde{y}^\star$ must be filled with zero entries but on its Task i two elements. All other values at the zero entry locations of $\tilde{y}^\star$ (such as the usual $\tilde{y} = [1, -1, \ldots, 1, -1]^\mathsf{T}$) would be suboptimal and possibly severely detrimental to the classification performance of Task i (not by altering the means m_{i1}, m_{i2} but by increasing the variance σ_i^2). This extreme example strongly suggests that, in order to maximize the MTL performance on Task i, one must impose low scores $\tilde{y}_{jl}$ to all Tasks j strongly different from Task i.

The choice $\tilde{y} = [1, -1, \ldots, 1, -1]^\mathsf{T}$ can also be very detrimental when $\Delta\mu_i^\mathsf{T}\Delta\mu_j < 0$ for some i, j: that is, when the mapping of the two classes within each task is reversed (e.g., if Class 1 in Task 1 is closer to Class 2 than Class 1 in Task 2). In this setting, it is easily seen that $\tilde{y} = [1, -1, \ldots, 1, -1]^\mathsf{T}$ works against the classification and performs much worse than a single-task LS-SVM.

Another interesting conclusion arises from the simplified setting of equal number of samples per task and per class, i.e., $n_{11} = \ldots = n_{k2}$. In this case, $\tilde{y}^\star = \Gamma^{-1}\mathcal{H}\left((\mathcal{A} \otimes \mathbb{1}_2\mathbb{1}_2^\mathsf{T}) \odot \mathcal{M}\right)(e_{i1}^{[2k]} - e_{i2}^{[2k]})$ in which all matrices are organized in 2×2 blocks of equal entries. This immediately implies that $\tilde{y}_{j1}^\star = -\tilde{y}_{j2}^\star$ for all j. So in particular the detection threshold $\frac{1}{2}(m_{i1} + m_{i2})$ of the averaged-mean test (equation 2) is zero, as conventionally assumed. In all other settings for the n_{jl}'s, it is very unlikely that $\tilde{y}_{i1}^\star = -\tilde{y}_{i2}^\star$ and the optimal decision threshold *must* also be estimated.

These various conclusions give rise to an improved MTL LS-SVM algorithm. A pseudo-code (Algorithm 1) along with discussions on (i) the estimation of the statistics met in Theorem 1 (Remark 1) and on (ii) its multi-class extension (Remark 2) are covered next. Matlab and Julia implementations of Algorithm 1 and its extensions are proposed in the supplementary material.

5 Experiments

Our theoretical results (the data-driven optimal tuning of MTL LS-SVM, as well as the anticipation of classification performance) find various practical applications and consequences. We exploit them here in the context of transfer learning, first on a binary decision on synthetic data, and then on a multiclass classification on real data.

To this end, and before proceeding to our experimental setup, a few key remarks to practically exploit Theorem 1 are in order.

Remark 1 (On the estimation of m_{ij} and σ_i). *All quantities defined in Theorem 1 are a priori known, apart from the k^2 inner products $\Delta\mu_i^\mathsf{T}\Delta\mu_j$ for $1 \leq i, j \leq k$. For these, define $\mathcal{S}_{il} \subset \{1, \ldots, n_{il}\}$ ($l = 1, 2$) and the corresponding indicator vector $\mathring{\mathbb{j}}_{il} \in \mathbb{R}^{n_i}$ with $[\mathring{\mathbb{j}}_{ij}]_a = \delta_{a \in \mathcal{S}_{ij}}$. For $i = j$, further let $\mathcal{S}'_{il} \subset \{1, \ldots, n_{il}\}$ with $\mathcal{S}'_{il} \cap \mathcal{S}_{il} = \emptyset$ and the corresponding indicator vector $\mathring{\mathbb{j}}'_{il} \in \mathbb{R}^{n_i}$. Then, the following estimates hold:*

$$\Delta\mu_i^\mathsf{T}\Delta\mu_j - \left(\frac{\mathring{\mathbb{j}}_{i1}}{|\mathcal{S}_{i1}|} - \frac{\mathring{\mathbb{j}}_{i2}}{|\mathcal{S}_{i2}|}\right)^\mathsf{T} X_i^\mathsf{T} X_j \left(\frac{\mathring{\mathbb{j}}_{j1}}{|\mathcal{S}_{j1}|} - \frac{\mathring{\mathbb{j}}_{j2}}{|\mathcal{S}_{j2}|}\right) = O\left((p \min_{l \in \{1,2\}} \{|\mathcal{S}_{il}|, |\mathcal{S}_{jl}|\})^{-\frac{1}{2}}\right)$$

$$\Delta\mu_i^\mathsf{T}\Delta\mu_i - \left(\frac{\mathring{\mathbb{j}}_{i1}}{|\mathcal{S}_{i1}|} - \frac{\mathring{\mathbb{j}}_{i2}}{|\mathcal{S}_{i2}|}\right)^\mathsf{T} X_i^\mathsf{T} X_i \left(\frac{\mathring{\mathbb{j}}'_{i1}}{|\mathcal{S}'_{i1}|} - \frac{\mathring{\mathbb{j}}'_{i2}}{|\mathcal{S}'_{i2}|}\right) = O\left((p \min_{l \in \{1,2\}} \{|\mathcal{S}_{il}|, |\mathcal{S}'_{il}|\})^{-\frac{1}{2}}\right).$$

Observe in particular that a single sample *(two when $i = j$) per task and per class ($|\mathcal{S}_{il}| = 1$) is sufficient to obtain a consistent estimate for all quantities, so long that p is large.*

Remark 2 (From binary to multiclass MTL). *Accessing the vector m in Theorem 1 allows for the extension of the MTL framework to a multiclass-per-task MTL by discarding the well known inherent biases of multiclass SVM. In the context of L_i classes for Task i, using a one-versus-all approach (for each $\ell \in \{1, \ldots, L_i\}$, one MTL LS-SVM algorithms with Class "1" being the target class ℓ and Class "2" all other classes at once), one needs to access L_i pairs of values $(m_{i1}(\ell), m_{i2}(\ell))$ and, for a new $\mathbf{x}$, decide on the genuine class of $\mathbf{x}$ based on the largest value among $g_i(\mathbf{x}; 1) - m_{i1}(1), \ldots, g_i(\mathbf{x}; L_i) - m_{i1}(L_i)$ with $g_i(\mathbf{x}; \ell)$ the output score for "Class ℓ versus all".*

For simplicity, from Remark 1 in the supplementary material, one may choose smart shift vectors $\bar{y}(\ell) \in \mathbb{R}^k$ for the scores $\tilde{y}(\ell) \in \mathbb{R}^{2k}$ (i.e., replace $\tilde{y}(\ell)$ by $\tilde{y}(\ell) + \bar{y}(\ell) \otimes \mathbb{1}_2$) so that $m_{i1}(\ell) = 0$ for each ℓ. Under this shift, the selected class is the one for which $g_i(\mathbf{x}; \ell)$ is maximum.

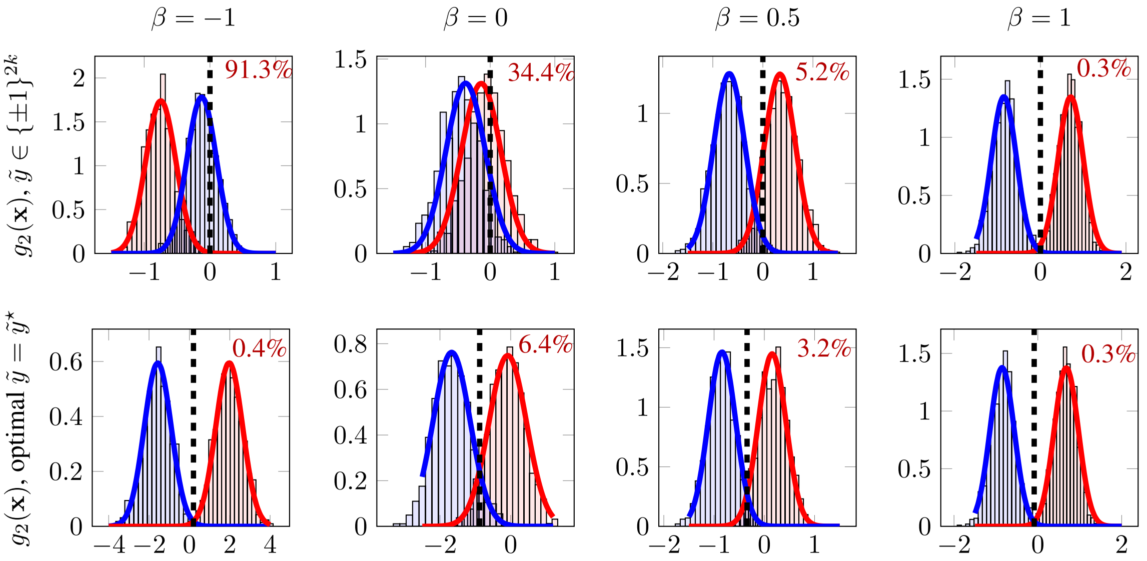

Figure 1: Scores $g_2(\mathbf{x})$ [empirical histogram vs. theory in solid lines] for $\mathbf{x}$ of Class $\mathcal{C}_1$ (red) or Class $\mathcal{C}_2$ (blue) for Task 2 in a 2-task ($k = 2$) setting of isotropic Gaussian mixtures for: **(top)** classical MTL-LSSVM with $y \in \{\pm 1\}$ and threshold $\zeta = 0$; **(bottom)** proposed optimized MTL-LSSVM with $\tilde{y}^\star$ and estimated threshold ζ; decision thresholds ζ represented in dashed vertical lines; red numbers are misclassification rates; task relatedness with $\beta = 0$ for orthogonal tasks, $\beta > 0$ for positively correlated tasks, $\beta < 0$ for negatively correlated tasks; $p = 100$, $[c_{11}, c_{12}, c_{21}, c_{22}] = [0.3, 0.4, 0.1, 0.2]$, $\gamma = \mathbb{1}_2$, $\lambda = 10$. Histograms drawn from 1 000 test samples of each class.

5.1 Effect of input score (label) and threshold decision choices

In order to support the theoretical insights drawn in the article, our first experiment illustrates the effects of the bias in the decision threshold for $g_i(\mathbf{x})$ (in general not centered on zero) and of the input score (label) optimization $\tilde{y}^\star$ on synthetic data.

Specifically, MTL-LSSVM is first applied to the following two-task ($k = 2$) setting: for Task 1, $x_{1l}^{(j)} \sim \mathcal{N}((-1)^j \mu_1, I_p)$ and for Task 2, $x_{2l}^{(j)} \sim \mathcal{N}((-1)^j \mu_2, I_p)$, where $\mu_2 = \beta\mu_1 + \sqrt{1-\beta^2}\mu_1^\perp$ and $\mu_1^\perp$ is any vector orthogonal to μ_1 and $\beta \in [0, 1]$. This setting allows us to tune, through β, the similarity between tasks. For four different values of β, Figure 1 depicts the distribution of the binary output scores $g_i(\mathbf{x})$ both for the classical MTL-LSSVM (top displays) and for our proposed random matrix improved scheme, with optimized input labels (bottom displays).

As a first remark, note that both theoretical prediction and empirical outputs closely fit for all values of β, thereby corroborating our theoretical findings. In practical terms, the figure supports (i) the importance to estimate the threshold decision which is non-trivial (not always close to zero) and (ii) the relevance of an appropriate choice of the input labels to improve the discrimination performance between both classes, especially when the two tasks are not quite related as shown by the classification error presented in red in the figure.

5.2 Multiclass transfer learning

We next turn to the classical Office+Caltech256 (Saenko et al., 2010; Griffin et al., 2007) real data (images) benchmark for transfer learning, consisting of the 10 categories shared by both datasets. For fair comparison with previous works, we compare images using $p = 4096$ VGG features. Half of the samples of the target is randomly selected for the test data and the accuracy is evaluated over 20 trials. We use here Algorithm 1, the results of which (Proposed) are reported in Table 1 against the non-optimized LS-SVM (Xu et al., 2013) and alternative state-of-the-art algorithms: MMDT, CDLS and ILS.[3]

[3] MMDT: Max Margin Domain Transform, proposed in (Hoffman et al., 2013), applies a weighted SVM on a learnt transformation of the source and target. CDLS: Cross-Domain Landmark Selection, proposed in (Hubert Tsai et al., 2016), derives a domain invariant feature space. ILS: Invariant Latent Space, proposed in (Herath et al., 2017), learns an invariant latent space to reduce the discrepancy between source and target.

Algorithm 1 Proposed Multi Task Learning algorithm.

```
Input: Training samples $X = [X_1, \dots, X_k]$ with $X_i = [X_i^{(1)}, \dots, X_i^{(L_i)}]$, $X_i^\ell \in \mathbb{R}^{p \times n_{i\ell}}$ and test data $\mathbf{x}$.
Output: Estimated class $\hat{\ell} \in \{1, \dots, L_t\}$ of $\mathbf{x}$ for target Task $t$.
for $j = 1$ to $L_t$ do
    Center and normalize data per task: for all $i' \in \{1, \dots, k\}$,
        • $\mathring{X}_{i'} \leftarrow X_{i'} \left( I_{n_{i'}} - \frac{1}{n_{i'}} \mathbb{1}_{n_{i'}} \mathbb{1}_{n_{i'}}^\intercal \right)$
        • $\mathring{X}_{i'} \leftarrow \mathring{X}_{i'} / \frac{1}{n_{i'} p} \mathrm{tr}(\mathring{X}_{i'} \mathring{X}_{i'}^\intercal)$
    Estimate: Matrix $\mathcal{M}$ from Remark 1 and $\tilde{\Delta}$, using $X_1^{(j)}, \dots, X_k^{(j)}$ as data of class 1 and $X \setminus \{X_1^{(j)}, \dots, X_k^{(j)}\}$ as data of class 2.
    Create scores
        $$\tilde{y}^\star(j) = \mathcal{D}_{\tilde{\Delta}}^{-\frac{1}{2}} \Gamma^{-1} \mathcal{H}[(\mathcal{A} \otimes \mathbb{1}_2 \mathbb{1}_2^\intercal) \odot \mathcal{M}] \mathcal{D}_{\tilde{\Delta}}^{-\frac{1}{2}} (e_{t1}^{[2k]} - e_{t2}^{[2k]}).$$
    Estimate and center $m = m(j)$ from Theorem 1 as per Remark 1.
    (Optional) Estimate the theoretical classification error $\epsilon_t\,(\lambda, \gamma)$ minimize over $(\lambda, \gamma)^2$.
    Compute classification scores $g_t(\mathbf{x}; j)$ according to equation 1.
end for
Output: $\hat{\ell} = \arg\max_{\ell \in \{1, \dots, L_t\}} g_t(\mathbf{x}; \ell)$.
```

Table 1: Classification accuracy over Office+Caltech256 database. c(Caltech), w(Webcam), a(Amazon), d(dslr), for different "Source to target" task pairs ($S \to T$) based on VGG features. Best score in boldface, second-to-best in italic.

S/T	c →	w →	c →	a →	w →	a →	d →	w →	c →	d →	a →	d →	Mean
	w	c	a	c	a	d	a	d	d	c	w	w	score
LSSVM	96.69	**89.90**	92.90	*90.00*	93.80	78.70	93.50	95.00	85.00	**90.20**	94.70	**100**	91.70
MMDT	93.90	87.05	90.83	84.40	*94.17*	86.25	**94.58**	97.50	86.25	87.23	92.05	*97.35*	90.96
ILS	77.89	73.55	86.85	76.22	86.22	71.34	74.53	82.80	68.15	63.49	78.98	92.88	77.74
CDLS	*97.60*	88.30	*93.54*	88.30	93.54	*92.50*	93.54	93.75	**93.75**	88.30	*97.35*	96.70	*93.10*
Ours	**98.68**	**89.90**	**94.40**	**90.60**	**94.40**	**93.80**	*94.20*	**100**	*92.50*	*89.90*	**98.70**	*99.30*	**94.70**

Table 1 demonstrates that our proposed improved MTL LS-SVM, despite its simplicity and unlike the competing methods used for comparison, has stable performances and is quite competitive. We further recall that, in additions to these high levels of performance, the method comes along with theoretical guarantees, which none of the competing works are able to provide.

6 Conclusion: beyond MTL

Through the example of transfer learning (and more generally multitask learning), we have demonstrated the capability of random matrix theory to (i) predict and improve the performance of machine learning algorithms and, most importantly, to (ii) turn simplistic (and in theory largely suboptimal) methods, such as here LS-SVM, into competitive state-of-the-art algorithms. As Gaussian mixtures are quite "universal" and thus already appropriate to handle real data (as shown in supplementary material), one may surmise the optimality of the least square approach, thereby opening the possibility to prove that MTL LS-SVM is likely close-to-optimal even in real data settings.

This is yet merely a first step into a generalized use of random matrix theory and large dimensional statistics to devise much-needed *low computational cost* and *explainable*, yet highly competitive, machine learning methods from elementary optimization schemes.

Acknowledgments

Couillet's work is partially supported by MIAI at University Grenoble-Alpes (ANR-19-P3IA-0003) and the HUAWEI LarDist project.

REFERENCES

Arvind Agarwal, Samuel Gerber, and Hal Daume. Learning multiple tasks using manifold regularization. In *Advances in neural information processing systems*, pp. 46–54, 2010.

Murray Aitkin and Nicholas Longford. Statistical modelling issues in school effectiveness studies. *Journal of the Royal Statistical Society: Series A (General)*, 149(1):1–26, 1986.

Greg M Allenby and Peter E Rossi. Marketing models of consumer heterogeneity. *Journal of econometrics*, 89(1-2):57–78, 1998.

Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Multi-task feature learning. In *Advances in neural information processing systems*, pp. 41–48, 2007.

Jonathan Baxter. A bayesian/information theoretic model of learning to learn via multiple task sampling. *Machine learning*, 28(1):7–39, 1997.

Jonathan Baxter. A model of inductive bias learning. *Journal of artificial intelligence research*, 12: 149–198, 2000.

Shai Ben-David and Reba Schuller. Exploiting task relatedness for multiple task learning. In *Learning Theory and Kernel Machines*, pp. 567–580. Springer, 2003.

Hal Daumé III. Bayesian multitask learning with latent hierarchies. *arXiv preprint arXiv:0907.0783*, 2009.

Zeyu Deng, Abla Kammoun, and Christos Thrampoulidis. A model of double descent for high-dimensional binary linear classification. *arXiv preprint arXiv:1911.05822*, 2019.

Noureddine El Karoui et al. On information plus noise kernel random matrices. *The Annals of Statistics*, 38(5):3191–3216, 2010.

Theodoros Evgeniou and Massimiliano Pontil. Regularized multi–task learning. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 109–117. ACM, 2004.

Pinghua Gong, Jieping Ye, and Chang-shui Zhang. Multi-stage multi-task feature learning. In *Advances in neural information processing systems*, pp. 1988–1996, 2012.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. *technical report*, 2007.

Samitha Herath, Mehrtash Harandi, and Fatih Porikli. Learning an invariant hilbert space for domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3845–3854, 2017.

Judy Hoffman, Erik Rodner, Jeff Donahue, Trevor Darrell, and Kate Saenko. Efficient learning of domain-invariant image representations. *arXiv preprint arXiv:1301.3224*, 2013.

Yao-Hung Hubert Tsai, Yi-Ren Yeh, and Yu-Chiang Frank Wang. Learning cross-domain landmarks for heterogeneous domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5081–5090, 2016.

Zhenyu Liao and Romain Couillet. A large dimensional analysis of least squares support vector machines. *IEEE Transactions on Signal Processing*, 67(4):1065–1074, 2019.

Xiaoyi Mai and Romain Couillet. A random matrix analysis and improvement of semi-supervised learning for large dimensional data. *The Journal of Machine Learning Research*, 19(1):3074–3100, 2018.

Xiaoyi Mai, Zhenyu Liao, and Romain Couillet. A large scale analysis of logistic regression: Asymptotic performance and new insights. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3357–3361. IEEE, 2019.

Andreas Maurer, Massi Pontil, and Bernardino Romera-Paredes. Sparse coding for multitask and transfer learning. In *International conference on machine learning*, pp. 343–351, 2013.

Guillaume Obozinski, Ben Taskar, and Michael Jordan. Multi-task feature selection. *Statistics Department, UC Berkeley, Tech. Rep*, 2(2.2):2, 2006.

Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2010.

Shibin Parameswaran and Kilian Q Weinberger. Large margin multi-task metric learning. In *Advances in neural information processing systems*, pp. 1867–1875, 2010.

Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pp. 213–226. Springer, 2010.

Jie Wang and Jieping Ye. Safe screening for multi-task feature learning with multiple data matrices. *arXiv preprint arXiv:1505.04073*, 2015.

Shuo Xu, Xin An, Xiaodong Qiao, Lijun Zhu, and Lin Li. Multi-output least-squares support vector regression machines. *Pattern Recognition Letters*, 34:1078–1084, 07 2013. doi: 10.1016/j.patrec.2013.01.015.

Wenlu Zhang, Rongjian Li, Tao Zeng, Qian Sun, Sudhir Kumar, Jieping Ye, and Shuiwang Ji. Deep model based transfer and multi-task learning for biological image analysis. *IEEE transactions on Big Data*, 2016.